

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Luka Grah

**DCTool - Programsko orodje za klasifikacijo
podatkov na diskovnih poljih**

DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor:
prof. dr. Dušan Kodek

Ljubljana, 2008

Namesto te strani vstavite original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

Zahvala

Mentorju prof. dr. Dušanu Kodeku, ki me je pri nastajanju dela spodbudno vodil in usmerjal z nasveti.

Sodelavcu Viliju Tajniču za pomoč s področja diskovnih polj in upravljanja informacij.

Sodelavki Urši Gros, ki je nalogo pregledala s tehničnega in slovničnega vidika.
Sodelavcu Igor Alfireviću, ki je nalogo pregledal s tehničnega stališča.

Punci Maji in celotni družini, ki so mi stali ob strani skozi celoten študij.

Kazalo

Povzetek	1
Abstract	3
1. Uvod	5
1.1 Zgodovina storitvenega računalništva	6
1.2 Storitveno računalništvo na področju diskovnih polj	6
1.3 Ostoječe rešitve na trgu	8
2. DCTool - orodje za upravljanje nestrukturiranih podatkov	8
2.2 DCTool Agent	9
2.3 DCTool Reports	10
2.4 Tok podatkov v sistemu DCTool	10
3. Uporabniški pogled na sistem DCTool	11
3.1 DCTool Agent	11
3.1.1 Predpogoji za pravilno delovanje DCTool Agent	11
3.1.2 Namestitev .NET ogrodja verzije 3.5	11
3.1.3 Namestitev gonilnikov za dostop do podatkovne zbirke Oracle 10gR2	11
3.1.4 Preverjanje komunikacije z domenskim strežnikom	12
3.1.5 Uporabniški vmesnik	13
3.1.6 Zajem podatkov	14
3.2 DCTool Reports	14
3.2.1 Predpogoji za pravilno delovanje DCTool Reports	15
3.2.2 Namestitev DCTool Reports	15
3.2.3 Uporabniški vmesnik	16
3.2.4 DCTool Reports Poročila	17
3.2.4.1 Zavihek Automatic Reports	17
3.2.4.2 Zavihek Manual Reports	19

3.2.4.3	Zavihek Storage Management	20
3.2.4.4	Zavihek Storage Tiers	21
4.	Notranja zgradba	23
4.1	Podatkovni model	23
4.2	SQL stavki poročil	25
4.2.1	Automatic Reports	25
4.2.2	Manual Reports	27
4.2.3	Storage Management	30
4.2.4	Storage Tiers	32
4.3	DCTool Agent	33
4.3.1	Preiskovanje datotečnega sistema	34
4.3.2	Povezava s podatkovno zbirko	34
4.3.3	Vpis podatkov v podatkovno zbirko	35
4.3.4	Podatki o uporabniku in uporabniški skupini datoteke	35
4.3.5	Zajem podatkov	36
5.	Predlogi za nadaljnji razvoj sistema	37
6.	Sklepne ugotovitve	39
	Dodatek A	41
	Slike	43
	Diagrami	43
	Tabele	43
	Viri	45
	Izjava o samostojnosti dela	47

Seznam kratic in simbolov

IT	(<i>angl.</i>) <i>Information technology</i> ; informacijska tehnologija.
UML	(<i>angl.</i>) <i>Unified Modeling Language</i> ; Poenoten jezik modeliranja.
SQL	(<i>angl.</i>) <i>Structured Query Language</i> ; Strukturiran povpraševalni jezik.
LUN	(<i>angl.</i>) <i>Logical Unit Number</i> ; Logični trdi disk.
NTFS	(<i>angl.</i>) <i>NT File System</i> ; NT datotečni sistem.
DLM	(<i>angl.</i>) <i>Data Lifecycle Management</i> ; Upravljanje z življenjskim ciklusom podatkov.
HSM	(<i>angl.</i>) <i>Hierarchical Storage Management</i> ; Tehnika hierarhičnega shranjevanja podatkov .
IP	(<i>angl.</i>) <i>Internet Protocol</i> ; Internetni protokol.
ERD	(<i>angl.</i>) <i>Entity-relationship Diagram</i> ; Entitetni diagram.
LDAP	(<i>angl.</i>) <i>Lightweight Directory Access Protocol</i> ; Protokol za dostop do hierarhične podatkovne zbirke.
CSV	(<i>angl.</i>) <i>Comma separated values</i> ; Z vejicami ločene vrednosti.
NAS	(<i>angl.</i>) <i>Network Attached Storage</i> ; Omrežno diskovje.
SAN	(<i>angl.</i>) <i>Storage Area Network</i> ;
RAID	(<i>angl.</i>) <i>Redundant Array of Inexpensive Disks</i> ; Polje neodvisnih diskov z zagotovljeno stopnjo redundance.
ODBC	(<i>angl.</i>) <i>Open Database Connectivity</i> ; Protokol za povezavo s podatkovnimi bazami.
ERP	(<i>angl.</i>) <i>Enterprise resource planning</i> ; Sistem za podporo poslovanja organizacij.
IBM	(<i>angl.</i>) <i>International Business Machines</i>

Povzetek

Vlaganja v informacijsko tehnologijo predstavljajo podjetjem precejšne breme in so pogosto brez pravih povratnih informacij o uspešnosti naložbe. To postavlja IT oddelke v podrejen položaj v primerjavi z ostalimi oddelki. Organizacije jih vidijo samo kot velike porabnike sredstev. Z uporabo storitvenega računalništva pridobijo organizacije jasnejšo sliko o ustreznosti naložb v IT sektor, hkrati pa omogoča IT oddelkom izboljšati storitve in jih še bolj prilagoditi potrebam organizacij. V pričujočem diplomskem delu je predstavljeno programsko orodje DCTool za klasifikacijo nestrukturiranih podatkov na diskovnih poljih, ki so pomemben del storitvenega računalništva.

V uvodnem poglavju je predstavljeno storitveno računalništvo ter njegova zgodovina. Poudarek je predvsem na področju storitev, ki vključujejo diskovna polja. Podrobno je predstavljen problem klasifikacije nestrukturiranih podatkov in princip upravljanja življenjskega cikla ter hierarhičnega nivojskega shranjevanja podatkov. Predstavitev se zaključuje z opisom obstoječih rešitev za klasifikacijo podatkov.

Osrednja poglavja diplomskega dela so namenjena predstavitvi orodja DCTool z uporabniškega in tehničnega stališča. Nazorno je opisana uporaba orodja, predstavitev njegovih delov in njihova zgradba. Predstavljena je tudi tehnologija, v kateri je orodje razvito ter postopek namestitve.

Delo zaključujejo smernice za nadaljnji razvoj in možne izboljšave. Podani so tudi rezultati testne uporabe orodja.

Ključne besede: diskovna polja, storitveno računalništvo.

Abstract

Investments in information technology represent a substantial burden for firms and are often without a real feedback on investment returns. This puts IT departments in an inferior position relative to the other departments. IT departments are seen only as a major consumer of resources. With the advent of utility computing the organizations can obtain a clearer picture of the adequacy of investments in the IT sector, while enabling the IT departments to improve services and make them more suited to the needs of organizations. It is the purpose of this thesis to present the DCTool software for the classification of unstructured data on disk arrays which are an important part of utility computing.

The introductory chapter describes utility computing and its historical development. The focus is on services which include disk arrays. A detailed presentation of the classification problem of unstructured data, the principle of life-cycle management, and hierarchical storage management are included. The presentation is concluded with a description of existing solutions for data classification.

The main chapters of the thesis are used to describe the DCTool from the user and technical standpoint. A detailed description of the tools usage, its technical details, and structure are included. The technology that was used for its development and the installation process are also given.

The thesis is concluded with the guidelines for further development and possible improvements. A detailed set of tool's test results is also included.

Keywords: disk arrays, utility computing

1. Uvod

Storitveno računalništvo je najnovejši računalniški model, ki se pojavlja v IT (Information Technology) industriji. Z njim želijo IT oddelki in ponudniki IT storitev za svoje stranke zagotoviti storitve, ki bi ustrezale njihovim ekonomskim in funkcionalnim potrebam.

Organizacije brez vlaganja v IT oddelke in informatizacijo poslovanja ne morejo konkurenčno nastopati na svetovnem trgu. Informacijska podpora jim zmanjšuje stroške, povečuje prednost na trgu, omogoča hitro prilagajanje spremembam trga in zagotavlja potrebne informacije za poslovne odločitve. Vlaganje v IT predstavlja organizacijam precejšne breme brez merljivih povratnih informacij o ustreznosti in o dejanski uporabnosti naložb. Povezava med uspešnostjo poslovanja in vlaganjem v IT oddelke je zabrisana, vodstvo pa ima možnost spremljati naložbe samo z ekonomskega stališča, preko amortizacije. Tudi IT oddelki od ostalih poslovnih enot ne dobivajo povratnih informacij o ustreznosti, uporabnosti in dejanski uporabi IT storitev, ki jih nudijo. Nejasnosti se pojavljajo predvsem pri uporabi skupnih virov kot so diskovna polja, elektronska pošta, interni spletni portali in podobno.

V IT industriji se je razvil nov model nujenja storitev – storitveno računalništvo (utility computing). V tem modelu stopajo v ospredje poslovne zahteve, dovolj visok nivo storitev, konsolidacija strojne in programske opreme in zaračunavanje storitev po uporabi. Z modelom storitvenega računalništva podjetje pridobi jasnejšo sliko o uporabi virov, ki jih nudi IT oddelek. S pridobljenimi informacijami lahko IT oddelek izboljša svoje storitve in jih bolje prilagodi potrebam organizacije, prepozna podvojene in odvečne storitve, predvsem pa lažje upraviči opravljene investicije. Storitveno računalništvo omogoča IT oddelku, da porazdeli operativne stroške po poslovnih enotah v podjetju in jim tudi zaračuna uporabo virov. Transparentno prikaže porabo procesorske moči, diskovnih polj, število aplikacij in količino prenesenih podatkov. S tem dobi IT oddelek pravilno sliko o uporabi nujenih virov in storitev, kar mu omogoča še boljšo optimizacijo delovanja in mu da potrebne informacije za lažje sprejemanje odločitev pri planiranju vlaganj v tehnologijo in širjenju nabora storitev na dovolj visokem nivoju.

Z uporabo modela storitvenega računalništva lahko IT oddelek nudi storitve in računsko moč kjer in kadar jo poslovne enote potrebujejo, zagotavlja uporabnikom boljšo uporabniško izkušnjo in omogoča jasn pregled nad izboljšanjem storitev IT oddelka k izboljšanju poslovanja podjetja.

1.1 Zgodovina storitvenega računalništva

Že leta 1961 je profesor John McCarthy v svojem govoru ob stoletnici MIT [5] govoril o prodaji procesorske moči in aplikacij po uporabi, tako kot se prodajajo električna, plin, voda. Ideja o zaračunavanju računalniških storitev glede na uporabo je bila zelo popularna v 60. letih dvajsetega stoletja, ko so se uporabljali predvsem dragi centralni računalniki (IBM – International Business Machines) s terminalskim načinom dela.

Ideja je v 70. letih počasi zbledela s pojavom osebnih računalnikov, ki so bili sorazmerno poceni ter dostopni. Računsko breme se je s centralnih strežnikov preneslo na osebne računalnike in tam tudi ostalo zaradi neprimernih telekomunikacijskih rešitev. Z razvojem interneta in hitrih računalniških omrežij je ideja o storitvenem računalništvu ponovno pridobila na pomenu zaradi možnosti centralizacije in konsolidacije računalniških storitev ter hitrih in zanesljivih omrežnih povezav. Razvoj modela storitvenega računalništva je šel še korak dlje. Ponujanje računalniških storitev ni več omejeno znotraj organizacije, kjer IT oddelek nudi storitve ostalim poslovnim enotam, ampak so se s širšo dostopnostjo interneta pojavila podjetja, ki nudijo storitve tudi drugim podjetjem po modelu storitvenega računalništva, kjer se storitve zaračunavajo po uporabi.

1.2 Storitveno računalništvo na področju diskovnih polj

Eno od področij storitvenega računalništva so tudi diskovna polja. V 90. letih dvajsetega stoletja se je shranjevanje podatkov iz internih strežniških diskov prestavilo na diskovna polja. Diskovna polja so naprave, ki omogočajo shranjevanje velikih količin podatkov na trdih diskih, ki, združeni v RAID (Redundant Array of Inexpensive Disks) polja, zagotavljajo večjo varnost ter dosegljivost podatkov s ščitjenjem pred fizičnimi okvarami podatkovnih nosilcev.

Diskovna polja delimo v dve družini (NAS in SAN), ki se razlikujeta po načinu dostopa do podatkov. Pri NAS (Network Attached Storage) diskovnih poljih dostopamo do podatkov preko mrežnih protokolov (SMB – Server Message Block, NFS – network filesystem). Dostop izvajamo na nivoju datotek. Pri SAN (Storage Area Network) diskovnih poljih pa strežniki vidijo diskovno polje kot lokalne diske, ki so vgrajeni v strežnik. Dostop je na nivoju blokov.

Diskovna polja so lahko sestavljena iz različno hitrih diskov, ki so lahko organizirani v različna RAID polja.

Uporaba diskovnega polja omogoča organizaciji centralizirano upravljanje in ščitjenje podatkov ter enostavno zagotavljanje podatkovnih kapacitet. Podatkovne kapacitete se širijo centralizirano in ne po posameznih strežnikih. Diskovna polja zagotavljajo tudi večjo skupno kapaciteto, kar je zelo pomembno pri trenutni rasti podatkov v podjetjih.

Nakup diskovnega polja predstavlja od 40 do 60% [1] povprečnega letnega proračuna IT oddelka. Pri tako veliki investiciji so povratne informacije o uporabi in izboljšanju poslovanja zelo pomembne. Še bolj pa so pomembne informacije zbrane pred samim nakupom diskovnega polja. To so informacije o strukturi, količini, starosti, porazdeljenosti in

predvideni rasti podatkov. Količina in rast podatkov vplivata na velikost in tip diskovnega polja, struktura in starost podatkov pa na hitrost ter tip diskov v diskovnem polju.

Ena od smernic storitvenega računalništva je tudi zaračunavanje storitev. Nudenje prostora za shranjevanje je storitev, ki jo IT oddelek ponuja ostalim poslovnim enotam in zaračunava po količini zasedenega prostora.

Za diskovno polje si podjetje, oziroma IT oddelek, izračuna ceno enega GB na časovno enoto v celotni predvideni dobi delovanja diskovnega polja. Cena se oblikuje tudi glede na kvaliteto vgrajenih diskov v diskovna polja. Po tej ceni potem zaračunava uporabo ostalim poslovnim enotam. Količina podatkov, ki jih mora podjetje hraniti, eksponentno raste. S tem se povečujejo zahteve po kapacitetah in posledično cena storitve. Pri hranjenju informacij so organizacije podvržene več zakonskim pravilom, ki jim določajo čas hranjenja in tudi postopek uničenja poslovnih podatkov. Eksponentni rasti so podvrženi predvsem nestrukturirani podatki izven poslovnih aplikacij, to so dokumenti, preglednice, slike, predstavitve in podobno. V organizacijah predstavljajo nestrukturirani podatki 60% vseh podatkov.

Za upravljanje nestrukturiranih podatkov organizacije oblikujejo seznam pravil o njihovem shranjevanju za celotno življenjsko obdobje. To so DLM (Data Lifecycle Management) pravila, ki določajo življenjski cikel podatkov; kje se bo datoteka hranila skozi celotno življenjsko obdobje, kdaj in kako bo uničena ali arhivirana.

Z DLM pravili razdelimo podatkovno polje na več različnih nivojev, ki predstavljajo različne faze življenjskega cikla datotek. Produkti, ki implementirajo DLM pravila, avtomatsko prenašajo podatke z enega nivoja na drugega glede na attribute datotek (tip datoteke, starost, velikost) po kriterijih, ki smo jih določili. Diskovna polja hierarhično razdelimo v 4 ali več nivojev [4], ki se razlikujejo po velikosti in po tipu diskov.

Ime nivoja	Namen	Tip medija	Starost datotek
Tier 1	Datoteke, ki jih potrebujemo vsak dan	Hitri in dragi trdi diski	0-6 mesecev
Tier 2	Datoteke, ki jih potrebujemo na mesečni bazi	Hitri in dragi trdi diski	6- 12 mesecev
Tier 3	Datoteke, ki jih potrebujemo na letni bazi	Počasni in ceneni diski	12 > mesecev
Tier 4	Datoteke, ki jih ne potrebujemo več, vendar smo jih po zakonu dolžni hraniti	Trakovi, optični mediji ali posebna diskovna polja	

Tabela 1.1 – Razdelitev diskovnih polj na nivoje

Z razdeljevanjem podatkov na nivoje izboljšamo izrabo diskovnih polj in zmanjšamo ceno GB na časovno enoto

1.3 Ostoječe rešitve na trgu

Na trgu obstaja kar nekaj rešitev za določanje in izvajanje DLM pravil na osnovi hierarhičnega shranjevanja podatkov. V tem poglavju predstavljam nekaj najbolj znanih komercialnih rešitev, ki delujejo neodvisno od diskovnih polj. Večina proizvajalcev ima tudi strojne rešitve, kjer tretji in četrti nivoji niso trakovi ali optični mediji, ampak posebna diskovna polja za hrambo datotek, ki so združljiva z zakonodajo.

- **Symantec Enterprise Vault** (<http://www.symantec.com/business/enterprise-vault>)
Enterprise Vault je vodilni produkt za klasifikacijo in hierarhično shranjevanje podatkov. Omogoča nadzor nad tipi datotek. Podjetja si lahko izdelajo pravila in celo določijo tipe datotek, ki se lahko shranjujejo na diskovnih poljih. Omogoča tudi klasifikacijo datotek po njihovih atributih, arhiviranje po načelu HSM (Hierarchical Storage Management) ter izdelavo poročil o izrabi diskovnih polj.
- **EMC DiskXtender** (<http://www.emc.com/products/family/diskxtender-family.htm>)
DiskXtender je produkt vodilnega proizvajalca diskovnih polj, podjetja EMC. Produkt omogoča klasifikacijo podatkov in prenos podatkov na ustrezne nivoje na diskovnih poljih glede na vnaprej definirana DLM pravila. Obstajata različici za Windows in Linux operacijska sistema. Datoteke klasificira glede na velikost, uporabnika, skupino, datum nastanka in datum zadnjega dostopa. Čeprav je to produkt proizvajalca diskovnih polj, deluje z diskovnimi polji različnih proizvajalcev.
- **CommVault Data Archiver** (<http://www.commvault.com/products/archive.asp>)
To je produkt, ki omogoča izvajanje DLM pravil in hierarhično shranjevanje elektronske pošte na NAS diskovnih poljih. Omembe vreden je zato, ker je specializiran samo za določen tip nestrukturiranih podatkov.
- **IBM System Storage Archive Manager** (<http://www-01.ibm.com/software/tivoli>)
System Storage Archive Manager je produkt podjetja IBM, ki omogoča shranjevanje datotek po principu večnivojskega shranjevanja (HSM) in vpeljavo lastnih DLM pravil. Deluje z diskovnimi polji različnih proizvajalcev in omogoča tudi hranjenje datotek na prenosnih medijih (trak, DVD, CD). Datoteke lahko arhivira tudi z delovnih postaj in strežnikov.

2. DCTool - orodje za upravljanje nestrukturiranih podatkov

Ob nakupu diskovnih polj se pojavijo problemi pri določanju količine, faktorja rasti in pravila za arhiviranje nestrukturiranih podatkov. Izkušnje so pokazale, da imajo stranke probleme pri optimizaciji izrabe diskovnih polj in optimizaciji operativnih stroškov diskovnih polj.

Iz dolgoletnih izkušenj implementacij diskovnih polj in svetovanja strankam se je podjetje S&T Slovenija odločilo za razvoj lastnega sistema za upravljanje z nestrukturiranimi podatki. Sistem je namenjen predvsem pomoči pri svetovanju strankam pri nakupu novih ali optimizaciji že obstoječih diskovnih polj.

Pri načrtovanju sistema so bili v ospredju trije načini uporabe sistema:

- **Svetovanje strankam pred nakupom**
Sistem mora omogočati izdelavo posnetka trenutnega stanja nestrukturiranih podatkov v podjetjih, na osnovi katerega se izvede svetovalno študijo in predstavi možne implementacije diskovnih polj različnih proizvajalcev.
- **Optimizacija izrabe diskovnih polj, ki so že implementirana**
Optimizacija izrabe diskovnih polj temelji na pravilni porazdelitvi datotek po nivojih glede na attribute datotek (tip datoteke, lastnik, skupina, datum zadnjega dostopa, datum kreiranja) in uvedbi DLM pravil, ki vzpostavijo upravljanje življenjskega cikla podatkov.
- **Zaračunavanje uporabe diskovnih polj po oddelkih, uporabnikih**
Zahteva po tej funkcionalnost, ki omogoča izračun stroškov izrabe diskovnih polj pa je prišla od večjih strank, pri katerih je IT oddelek organiziran kot samostojna enota, ki nudi plačljive usluge drugim poslovnim enotam in zato potrebuje sistem zaračunavanja uporabe diskovnih polj.

Sistem DCTool je sestavljen iz dveh ločenih aplikacij:

- **DCTool Agent** (agenti za zbiranje podatkov)
- **DCTool Reports** (orodje za obdelavo podatkov, zbranih preko agentov)

2.2 DCTool Agent

DCTool Agent je manjša aplikacija za zbiranje podatkov o datotekah na diskovnem polju. Agent se namesti na strežnik, kateremu so prikazani LUN-i (Logical Unit Number) diskovnega polja. Nato rekurzivno zbere informacije o datotekah na izbranih LUN-ih in jih zapiše v podatkovno bazo. Vgrajeno ima podporo za več podatkovnih zbirk. Trenutna verzija agenta je pripravljena za operacijski sistem družine Microsoft Windows in podpira NTFS (NT File System) ter FAT32 datotečna sistema. Za shranjevanje podatkov uporablja podatkovno zbirko Oracle 10gR2.

DCTool Agent ima v trenutni verziji implementiran uporabniški vmesnik, vendar je to narejeno zgolj za predstavitvene namene in za izdelavo analiz pri stranki. V okolju, kjer je predvideno zbiranje podatkov v daljšem časovnem obdobju, bo agent brez interakcije avtonomno zbiral podatke in jih zapisoval v podatkovno zbirko.

2.3 DCTool Reports

DCTool Reports je strežniški del aplikacije, ki je sestavljen iz podatkovne baze in spletnega orodja za pregled in obdelavo podatkov. Trenutna verzija je zgrajena na relacijski podatkovni zbirki Oracle verzije 10gR2 in Oracle Application Express ogrodju za izdelavo spletnih aplikacij, ki je vgrajeno v samo podatkovno bazo. Uporabniku nudi sledečo funkcionalnost:

- Pregled zbranih podatkov po različnih kriterijih:
 - po uporabnikih,
 - po časovnih slikah in
 - po tipih datotek..
- Urejanje podatkov o diskovnih poljih.
- Izdelavo avtomatskih, vnaprej definiranih poročil.
- Izdelavo samostojno izdelanih poročil.
- Izdelavo poročil za obračun uporabe diskovnih polj.
- Izdelavo poročil za razdeljevanje datotek po nivojih.
- Izdelavo scenarijev uporabe različnih modelov diskovnih polj na zbranih podatkih.

DCTool Reports je namenjen skrbnikom diskovnih polj, da dobijo boljši pregled nad uporabo z vidika uporabnikov in vodjem IT oddelkov pri pripravi poročil o uporabi in zaračunavanju uporabe. Namenjen je tudi svetovalcem v podjetju S&T Slovenija za lažje in bolj učinkovito svetovanje strankam pri nakupu novih diskovnih polj ali izboljšanju izkoriščenosti obstoječih. V dodatku A.1 je podan UML diagram primerov uporabe za sistem DCTool.

2.4 Tok podatkov v sistemu DCTool

Agenti sistema DCTool so nameščeni na strežnikih, ki shranjujejo podatke na diskovnih poljih. Preko ODBC (Open Database Connectivity) povezave zapišejo podatke o datotekah v podatkovno zbirko. Podatki se zapisujejo v vnaprej pripravljene tabele, ki jih uporablja DCTool Reports za izdelavo poročil. Vse informacije o diskovnih poljih, uporabnikih ali uporabniških skupinah se vnašajo neposredno preko spletnega vmesnika DCTool Reports.

3. Uporabniški pogled na sistem DCTool

Sistem DCTool po funkcionalnosti razdelimo na dva dela. Agenti so namenjeni zbiranju podatkov, Reports pa je namenjen pregledovanju podatkov, izdelavi poročil in urejanju podatkov o diskovnih poljih, uporabnikih in uporabniških skupinah.

3.1 DCTool Agent

Agent je realiziran kot Microsoft Windows aplikacija z uporabniškim vmesnikom, v katerem določimo izvor in ponor podatkov. Omogoča tudi izdelavo dnevniških datotek, v katerih so s časovnimi oznakami dopolnjeni izvedeni koraki. DCTool Agent namestimo na strežnike, ki uporabljajo diskovno polje za shranjevanje podatkov.

Če strežnikov ne želimo obremenjevati z dodatnim delom zbiranja podatkov, lahko agenta namestimo na računalnik, ki ga povežemo neposredno na diskovno polje. Pred zbiranjem podatkov na diskovnem polju naredimo posnetek stanja¹.

3.1.1 Predpogoji za pravilno delovanje DCTool Agenta

Na strežnikih, kjer je nameščen DCTool Agent, je potrebno namestiti .NET ogrodje verzije 3.5, potrebne gonilnike za dostop do podatkovne zbirke Oracle 10g in ustrezno nastaviti vir za podatkovno zbirko. Strežnik mora imeti omogočeno komunikacijo z domenskim strežnikom, oziroma mora biti član domene, iz katere se črpajo informacije o uporabnikih in uporabniških skupinah.

3.1.2 Namestitev .NET ogrodja verzije 3.5

.NET ogrodje verzije 3.5 je prosto dostopno z domače strani podjetja Microsoft. Za namestitev lahko uporabimo storitev WindowsUpdate² podjetja Microsoft (<http://windowsupdate.microsoft.com>), ki nam avtomatsko namesti vse potrebne popravke, ki jih zahteva .NET ogrodje verzije 3.5 [8].

3.1.3 Namestitev gonilnikov za dostop do podatkovne zbirke Oracle 10gR2

Gonilnik za dostop do podatkovne zbirke dobimo z domače strani podjetja Oracle (<http://otn.oracle.com>). Gonilnik se imenuje Oracle 10gR2 Client. Po uspešni namestitvi je potrebno ročno dodati zapis za komunikacijo s podatkovno bazo. Zapis dodamo v datoteko »tnsnames.ora« [7], ki se nahaja v mapi »network/admin« znotraj namestitvenega drevesa gonilnikov za dostop do podatkovne zbirke Oracle.

¹ Funkcionalnost izdelave posnetka stanja diskovnega polja mora omogočati diskovno polje.

² Za uporabo storitve WindowsUpdate morate imeti legalno kopijo operacijskega sistema Windows

Primer zapisa v datoteki »tnsname.ora«:

```

<ime_zapisa> =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <ime_streznika>)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SID = <ime_instance>)
    )
  )

```

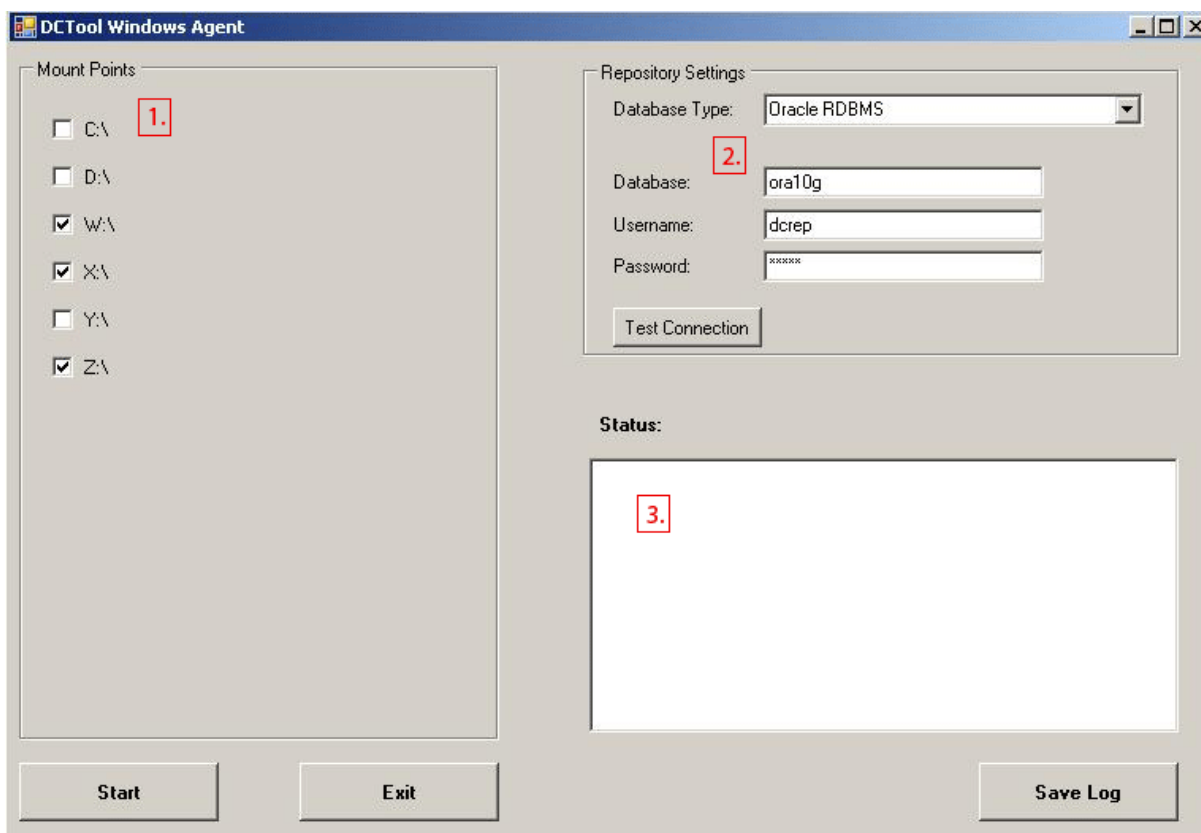
Spremenljivka	Opis spremenljivke
<ime_zapisa>	Ime, na katerega se bomo sklicevali, ko bomo dostopali do podatkovne zbirke.
<ime_streznika>	Ime ali IP številka strežnika, na katerem je nameščena podatkovna zbirka. To je ime strežnika, na katerem je nameščena DCTools Reports programska oprema.
<ime_instance>	Ime podatkovne zbirke, v katero bomo shranjevali podatke o datotekah.

Tabela 3.1 – Spremenljivke v tnsnames zapisu

3.1.4 Preverjanje komunikacije z domenskim strežnikom

Strežnik, na katerem je nameščen DC Tool Agent, mora biti član domene oziroma mora imeti omogočeno komunikacijo z domenskim strežnikom. Od njega pridobi informacije o uporabnikih in uporabniških skupinah. To preverimo tako, da v nadzorni plošči operacijskega sistema Microsoft Windows odpremo ikono »System«. V prikazanem oknu preverimo na zavihku »Computer Name«, če je poleg zapisa »Domain:« napisano tudi ime domene v katero je vključen strežnik.

3.1.5 Uporabniški vmesnik



Slika 3.1 – DCTool Agent uporabniški vmesnik

Uporabniški vmesnik DCTool Agenta je sestavljen iz treh delov. Levi del (1) je namenjen izbiri diskov, ki jih želimo pregledati. Agent nam avtomatsko ponudi vse diske, ki so priključni na strežnik, na katerem je nameščen. V pregled lahko vključimo tudi omrežne diske.

Desno zgoraj (2) imamo možnost izbire podatkovne zbirke, v katero želimo pisati podatke o pregledanih datotekah. Za dostop do podatkovne zbirke potrebujemo ime, kot smo ga definirali v poglavju 4.1.3, uporabniško ime in geslo sheme, v kateri je pripravljena tabela za shranjevanje podatkov o datotekah.

Desno spodaj (3) pa je prostor za beleženje korakov in morebitnih napak, ki se pojavijo pri zbiranju podatkov.

Na uporabniškem vmesniku so tudi štiri gumbi:

Ime gumba	Funkcija gumba
Start	Zagon zbiranja podatkov
Exit	Izhod iz programa
Test Connection	Preverjanje povezave s podatkovno zbirko. Program se z vnešenimi podatki poskuša prijaviti na podatkovno zbirko.
Save Log	Shranjevanje sporočil v dnevniško datoteko.

Tabela 3.2 - Seznam gumbov na zaslonski maski DCTool Agenta

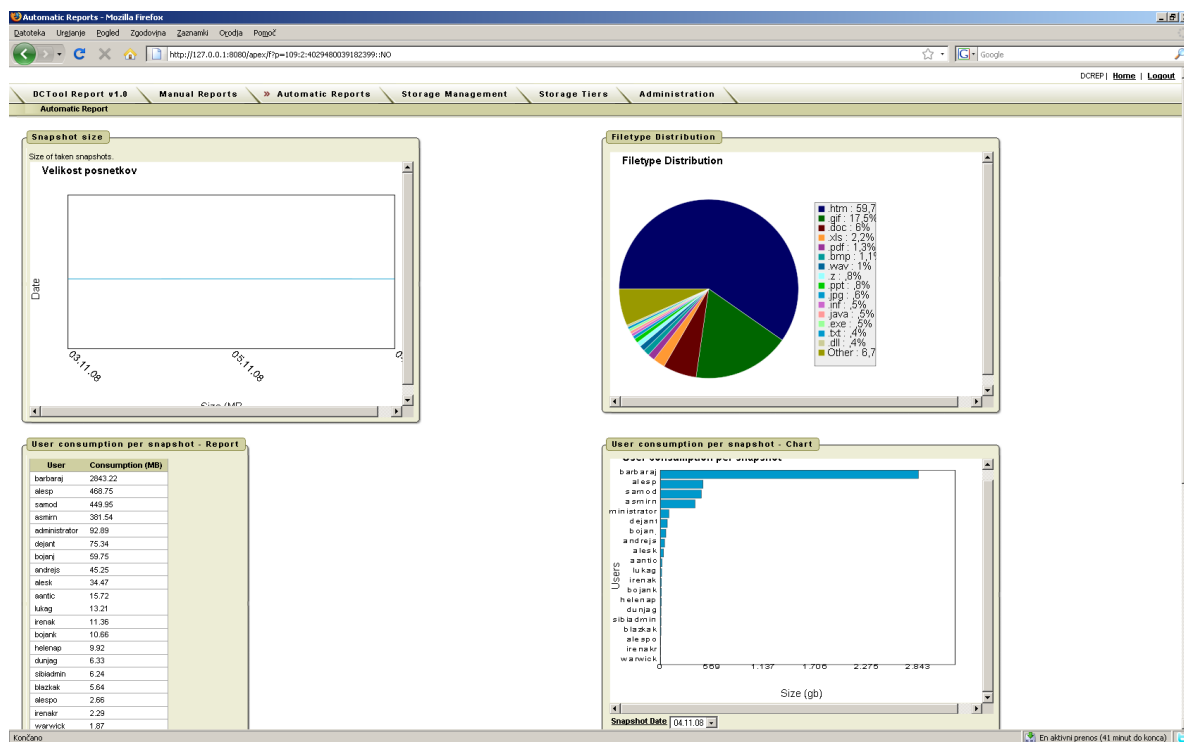
3.1.6 Zajem podatkov

Zajem podatkov začnemo tako, da v uporabniškem vmesniku označimo vse diske, ki jih želimo pregledati in izberemo tip podatkovne zbirke za zapis podatkov. Pri izbiri podatkovne zbirke moramo vpisati ime podatkovne zbirke, uporabniško ime ter geslo za dostop. Priporočeno je, da pred začetkom zbiranja podatkov s klikom na gumb »Test Connection« preverimo pravilnost podatkov za prijavo. Agent bo vrnil pogovorno okno z obvestilom o uspešnosti komunikacije s podatkovno zbirko. Zbiranje podatkov poženemo s klikom na gumb »Start.«

Vse akcije in vse morebitne napake med zbiranjem podatkov nam bo sistem javil v gradniku pod napisom »Status:«. Vsebinsko gradnika lahko preko gumba »Save Log« shranimo v dnevniško datoteko.

3.2 DCTool Reports

DCTool Reports je orodje za pregledovanje in analiziranje podatkov. Realizirano je kot aplikacija v Oracle Application Express ogrodju in je nameščeno direktno na podatkovno zbirko, v katero se zapisujejo podatki zbrani z DCTool Agenti. Namestimo ga lahko na katerokoli podatkovno zbirko verzije 10gR2, ki ima nameščeno Oracle Application Express ogrodje verzije 2.0 ali višje. DCTool Reports deluje na vseh platformah, na katerih deluje podatkovna zbirka Oracle 10gR2. DCTool Reports lahko namestimo tudi v že obstoječo podatkovno zbirko.



Slika 3.2 – DCTool Reports uporabniški vmesnik

3.2.1 Predpogoji za pravilno delovanje DCTool Reports

DCTool Reports potrebuje za delovanje podatkovno zbirko Oracle 10gR2 in Oracle Application Express ogrodje verzije 2.0 ali višje. Namestitvi obeh produktov sta precej kompleksna postopka, ki ju je potrebno opraviti v sodelovanju z administratorjem okolja, v katerega nameščamo DCTool.

3.2.2 Namestitev DCTool Reports

Namestitev DCTool Reports je sestavljena iz dveh delov. Prvi del obsega namestitev elementov v zbirko podatkov, drugi del pa namestitev spletnega vmesnika v ogrodje Oracle Application Express.

V prvem delu je potrebno ustvariti podatkovno shemo in prostor ter namestiti objekte v zbirko podatkov, v katero bo DCTool Agent shranjeval podatke.

- Prostor v podatkovni zbirki kreiramo tako, da se prijavimo kot uporabnik s sistemskimi pravicami in izvedemo ukaz:

```
SQL> Create tablespace dcrep datafile
'<lokacija_datoteke>' size <velikost>M extent management
local;
```

Spremenljivka	Opis spremenljivke
<lokacija_datoteke>	Lokacija, kjer želimo, da nam podatkovna zbirka kreira datoteko.
<velikost>	Velikost prostora v MB.

Tabela 3.3 – Spremenljivke v SQL ukazu za kreiranje prostora v podatkovni zbirki

- Podatkovno shemo ustvarimo tako, da se prijavimo na podatkovno zbirko s sistemskimi pravicami in izvedemo spodnja ukaza:

```
SQL> Create user dcrep identified by dcrep default
tablespace dcrep temporary tablespace temp;
SQL> Grant connect, resource to dcrep;
```

- Objekte v podatkovni zbirki namestimo tako, da se prijavimo kot »dcrep« uporabnik in izvedemo ukaze v datoteki »create_db.sql«. Kot uporabnik izvedemo spodnji ukaz:

```
SQL> @create_db.sql
```

V drugem delu je potrebno namestiti spletni vmesnik v Oracle Application Express ogrodje. To storimo tako, da se preko spletnega vmesnika ogrodja prijavimo kot uporabnik »dcrep« in uporabimo funkcijo za uvoz aplikacije, kot izvorno datoteko pa izberemo »dcrep_gui.sql«. Ko se uvoz konča, je aplikacija pripravljena za delo.

3.2.3 Uporabniški vmesnik

DCTool Reports aplikacija je realizirana kot spletna aplikacija. V zgornjem delu so zavihki, preko katerih dostopamo do posameznih podstrani. Na posameznih podstraneh so pripravljene gradniki za izdelavo poročil in pregledovanje zajetih podatkov.

Ime zavihka	Opis
DCTool Reports v1.0	Opis programa in splošne informacije
Manual Reports	Poročila, pri katerih lahko sami omejujemo podatke in izbiramo pogoje.
Automatic Reports	Avtomatska poročila, ki ne dopuščajo spreminjanja.
Storage Management	Poročila za obračun stroškov uporabe diskovnih polj
Storage Tiers	Poročila za pomoč pri vpeljavi večnivojskega sistema za shranjevanje datotek.
Administracija	Dodajanje diskovnih polj in njihovih karakteristik in upravljanje uporabniških skupin.

Tabela 3.4 – Seznam zavihkov v DCTool Reports

3.2.4 DCTool Reports Poročila

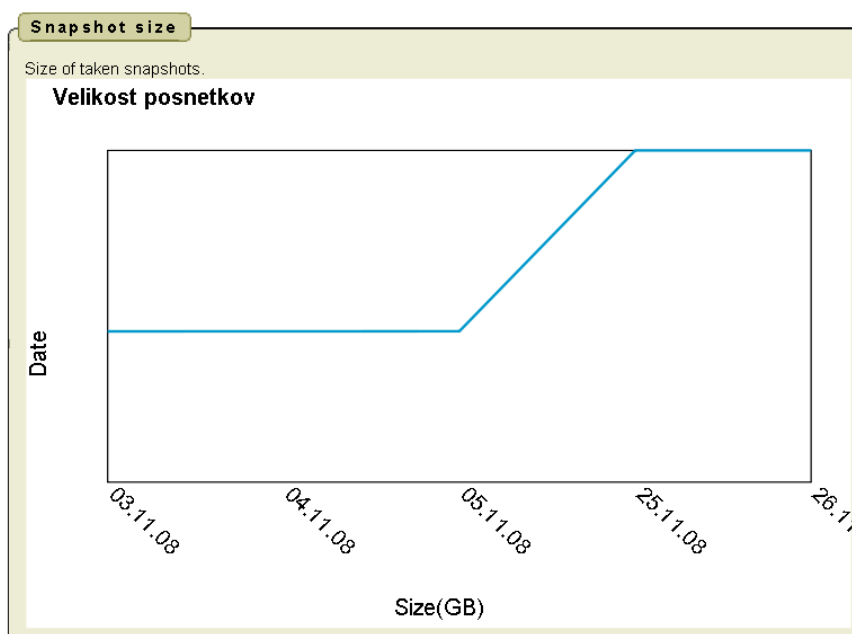
Poročila, ki jih lahko izdelamo z DCTool Reports, so v spletnem vmesniku razdeljena na podstrani, ki so predstavljene v poglavju 3.2.3.

3.2.4.1 Zavihek Automatic Reports

Na tej podstrani so zbrana poročila, ki jih sistem pripravi avtomatsko, poudarek je predvsem na strukturi zajetih podatkov.

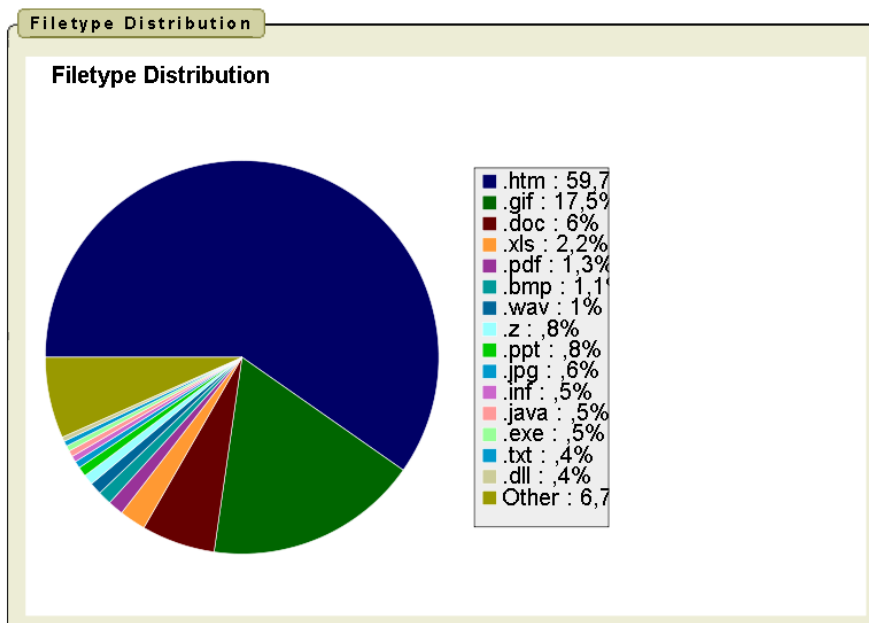
- Poročilo Snapshot Size

Poročilo je predstavljeno v obliki grafa in prikazuje skupno velikost zajetih datotek po datumih zajemanja. Poročilo je namenjeno spremljanju zasedenosti diskov na diskovnih poljih, ki smo jih vključili v zajem podatkov.



Slika 3.3 – Poročilo Snapshot Size

- Poročilo Filetype Distribution

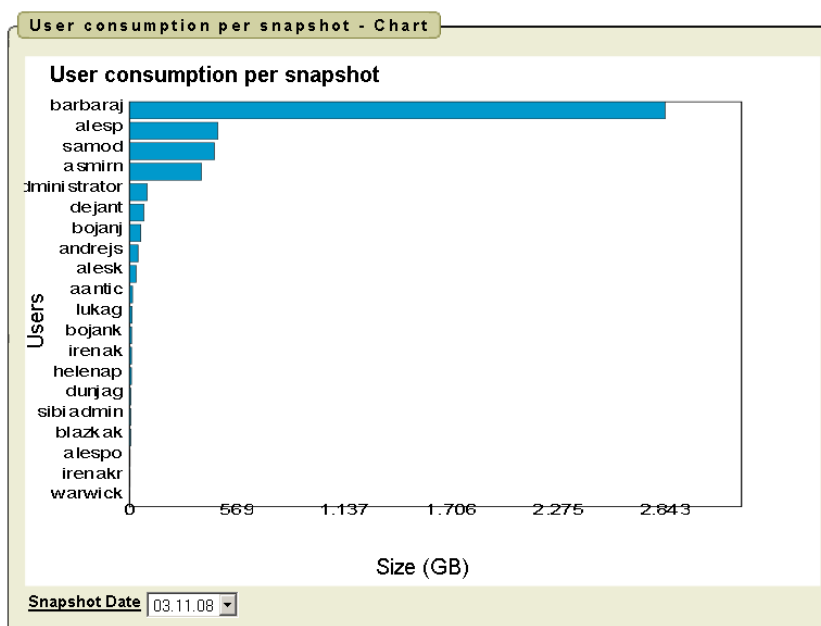


Slika 3.4 – Poročilo Filetype distribution

Poročilo je predstavljeno v obliki grafa in prikazuje porazdeljenost posameznih tipov datotek v posameznem zajemu.

- Poročilo User consumption per snapshot (Report in Chart)

Poročilo prikazuje porabo prostora uporabnikov po posameznih zajemih. Prikazano je z dvema prikaznima elementoma, tabelo in grafom ter z možnostjo izbire uporabnika.



Slika 3.5 – Poročilo User consumption per snapshot (Graf)

V tabeli so predstavljeni podatki:

Ime polja	Opis
User	Izbrani uporabnik
Consumption (GB)	Poraba v GB

Tabela 3.5 – Polja v poročilu User consumption per snapshot (Report in Chart)

- **Poročilo Filetype size per snapshot**

Poročilo prikazuje, kako se spreminja velikost posameznih tipov datotek skozi različne zajeme. Poročilo je pripravljeno v obliki grafa in z možnostjo izbire tipa datotek.

- **Poročilo User consumption per snapshots**

Poročilo prikazuje porabo prostora posameznega uporabnika skozi različne zajeme podatkov. Poročilo je predstavljeno v obliki grafa z možnostjo izbire uporabnika.

3.2.4.2 Zavihek Manual Reports

Na podstrani »Manual Reports« so zbrana poročila, ki nam omogočajo več možnosti spreminjanja atributov. Uporabnik lahko izbira časovno okno, uporabnika ter tip datoteke.

- **Poročilo User consumption per time frame**

Poročilo prikazuje podatke o porabi prostora za posameznega uporabnika v izbranem časovnem obdobju. Podatki so predstavljeni v obliki grafa in tabele.

V tabeli so predstavljeni podatki:

Ime polja	Opis
Snapshot date	Datum zajema
Size (GB)	Velikost v GB

Tabela 3.6 – Polja v poročilu User consumption per time frame

- **Poročilo Filetype consumption per time frame**

Poročilo prikazuje porabo prostora za posamezen tip datoteke v izbranem časovnem obdobju. Podatki so predstavljeni v obliki grafa in tabele.

V tabeli so predstavljeni podatki:

Ime polja	Opis
Snapshot date	Datum zajema
Size (GB)	Velikost v GB

Tabela 3.7 – Polja v poročilu Filetype consumption per time frame

- **Poročilo File Age per time frame**

Poročilo prikazuje starost datotek po posameznih zajemih. Podatki so predstavljeni v obliki grafa in tabele.

3.2.4.3 Zavihek Storage Management

Podstran »Storage Management« je namenjena obračunavanju uporabe diskovnih polj. Poročila na strani omogočajo, da izračunamo porabo za posameznega uporabnika. Omogoča nam tudi, da lahko stranki predstavimo stroške diskovnega polja na njenih podatkih in s preigravanjem možnosti prikažemo vse možne kombinacije. Tako si stranka lahko izbere rešitev, ki ji najbolj ustreza.

- Poročilo Storage Cost per User

Poročilo je namenjeno izračunu stroškov uporabe diskovnega polja za posameznega uporabnika v časovnem oknu, ki ga lahko definiramo. Pri izdelavi poročila lahko izberemo diskovno polje in tip diskov v njem.

Rezultat je prikazan v obliki tabele v kateri so podatki:

Ime polja	Opis
Snapshot date	Datum posnetka
User	Izbrani uporabnik
Disk type	Tip Diskov
Cost	Poraba uporabnika v EUR
Disk cost EUR/GB	Cena 1 GB na izbranih diskih

Tabela 3.8 – Polja v poročilu Storage Cost per User

- Poročilo Storage Cost

Poročilo prikazuje stroške uporabe za vse uporabnike diskovnega polja ter tudi seštevek vseh stroškov diskovnega polja. Tako kot pri prejšnjem poročilu lahko izberemo časovno okno, diskovno polje in tip diskov.

3.2.4.4 Zavihek Storage Tiers

Zavihek »Storage Tiers« je namenjen vpeljavi večnivojskega sistema po principu HSM. Pripravljena poročila nam omogočajo razdelitev datotek na posamezne nivoje po uporabnikih in po tipih datotek. Na koncu strani pa je še skupno poročilo razdelitve po nivojih za vse zajete datoteke.

Razdelitev po nivojih je narejena glede na čas zadnjega dostopa do datotek:

Nivo	Časovno obdobje
Tier 1	0-6 mesecev
Tier 2	6-12 mesecev
Tier 3	12 ali več mesecev

Tabela 3.9 – Časovna opredelitev nivojev na diskovnem polju

- **Poročilo Tier breakdown (per User)**

Poročilo prikazuje, koliko podatkov ima izbrani uporabnik v nekem časovnem obdobju, ki bi bili primerni za prenos na drugi ter tretji nivo. Podatki so prikazani v treh tabelah, vsaka prikazuje en nivo.

- **Poročilo Tier breakdown (per filetype)**

Poročilo prikazuje porazdelitev datotek po nivojih. Izbiramo lahko tip datoteke in časovno okno. Rezultat je prikazan v treh tabelah, vsaka za en nivo.

- **Poročilo Full Tier Breakdown**

Poročilo prikazuje porazdelitev vseh zajetih datotek po posameznih zajemih in po posameznih nivojih.

4. Notranja zgradba

DCTool sistem je sestavljen iz dveh ločenih aplikacij, ki sta med seboj povezani samo s podatkovnim modelom. DCTool Agent vpisuje podatke v podatkovni model, DCTool Reports pa podatke prikazuje in jih združuje v poročila. Zaradi razvoja na področju diskovnih sistemov in izpopolnjevanja modela uporabnega računalništva bomo DCTool orodje razvijali v skladu s spremembami na zgoraj omenjenih področjih. Zato je zagotovljena enostavna razširljivost in vzdrževanje podatkovnega modela.

4.1 Podatkovni model

Osrednja tabela podatkovnega modela je tabela »FILES«, v katero agenti zapisujejo prebrane podatke in je tudi osnova za izdelavo poročil.

- Tabela FILES

Ime stolpca	Podatkovni tip	Opis
ID	NUMBER	Primarni ključ
FILE_NAME	VARCHAR	Ime prebrane datoteke
FILE_SIZE	NUMBER	Velikost prebrane datoteke
EXTENSION	VARCHAR	Končnica prebrane datoteke
PATH	VARCHAR	Lokacija prebrane datoteke
OWNER	VARCHAR	Lastnik prebrane datoteke
GROUP_NAME	VARCHAR	Skupina prebrane datoteke
CREATION_TIME	DATE	Čas in datum nastanka datoteke
LAST_WRITE_TIME	DATE	Čas zadnjega dostopa do prebrane datoteke
ATTRIBUTES	VARCHAR	Atributi prebrane datoteke
SNAPSHOT_DATE	VARCHAR	Datum zajema podatkov

Tabela 4.1 – Polja v tabeli FILES

Poleg tabele »FILES« so v podatkovnem modelu še tabele:

- **Tabela STORAGE**

V tabeli so zapisana diskovna polja.

Podatki se vpisujejo preko aplikacije DCTool Reports.

Ime stolpca	Podatkovni tip	Opis
ID	NUMBER	Primarni ključ
VENDOR	VARCHAR	Izdelovalec diskovnega polja
STORAGE_NAME	VARCHAR	Ime družine diskovnega polja
MODEL	VARCHAR	Model diskovnega polja

Tabela 4.2 – Polja v tabeli STORAGE

- **Tabela DISKS**

V tabeli so zapisani tipi diskov in njihova cena na GB podatkov. Tabela je preko STORAGE_ID polja povezana s tabelo STORAGE.

Podatke v tabelo vpisuje DCTool Reports aplikacija.

Ime stolpca	Podatkovni tip	Opis
ID	NUMBER	Primarni ključ
STORAGE_ID	NUMBER	Tuj ključ (ID v tabeli STORAGE)
DISK_TYPE	VARCHAR2	Tip diska
DISK_COST	NUMBER	Cena GB prostora na disku (v EUR)

Tabela 4.3 – Polja v tabeli DISKS

- **Tabela USERS**

V tabeli so zapisani uporabniki in uporabniške skupine v organizaciji. Tabela je preko DEP_NO polja povezana s tabelo DEP.

Ime stolpca	Podatkovni tip	Opis
USER_ID	NUMBER	Primarni ključ
USER_NAME	VARCHAR2	Ime uporabnika
GROUP_NAME	VARCHAR2	Ime uporabniške skupine
DEP_NO	NUMBER	Tuj ključ (ID v tabeli DEP)

Tabela 4.4 – Polja v tabeli USERS

- Tabela DEP

V tabeli so shranjeni podatki o oddelkih in vodjih oddelkov.

Ime stolpca	Podatkovni tip	Opis
ID	NUMBER	Primarni ključ
DEP_NAME	VARCHAR2	Ime oddelka
DEP_HEAD	VARCHAR2	Vodja oddelka

Tabela 4.5 – Polja v tabeli DEP

V tabeli »DEP« in »USERS« se podatki vpisujejo preko mehanizmov, ki so odvisni od načina upravljanja uporabnikov v okolju, kjer je DCTool sistem nameščen. Če so uporabniki shranjeni v LDAP drevesih, jih lahko preko CSV datotek uvozimo neposredno v zbirko podatkov.

Celoten podatkovni model v obliki ERD diagrama se nahaja v dodatku A.2.

4.2 SQL stavki poročil

Poročila, ki so opisana v poglavjih od 3.2.4.1 do 3.2.4.4 temeljijo na SQL stavkih, ki se izvajajo na podatkovni zbirki preko Oracle Application Express ogrodja.

4.2.1 Automatic Reports

- SQL stavek za Snapshot Size poročilo

a) Izvorna koda SQL stavka:

```
SELECT null link,
       snapshot_date label,
       sum(file_size)/1024/1024 value
FROM   files
GROUP BY snapshot_date
```

- SQL stavek za Filetype Distribution poročilo

a) Izvorna koda stavka:

```
SELECT LINK, label, extension
FROM (SELECT NULL LINK, label, COUNT (*) extension,
            RANK () OVER (ORDER BY COUNT (*)) rnk,
            RANK () OVER (ORDER BY COUNT (*) DESC) rnk2
      FROM (SELECT extension label, extension
            FROM files
            WHERE snapshot_date = :p2_snapshot_date) a
      GROUP BY label)
ORDER BY GREATEST (rnk, rnk2 - .1) DESC
```

b) Spremenljivke v stavku:

Ime spremenljivke	Opis
:p2_snapshot_date	Datum zajema

- **SQL stavek za User consumption per snapshot poročilo**

Tabela SQL stavek:

a) Izvorna koda stavka

```
SELECT owner AS "user",
       TO_CHAR (SUM (file_size) / 1024 / 1024,
               '9999.99'
               ) AS "consumption (MB)"
FROM files
WHERE snapshot_date = :p2_snapshot_date
GROUP BY owner
ORDER BY 2 DESC
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p2_snapshot_date	Datum zajema

Graf SQL stavek:

a) Izvorna koda stavka

```
SELECT NULL LINK, owner label, SUM (file_size) / 1024 / 1024 VALUE
FROM files
WHERE snapshot_date = :p2_snapshot_date
GROUP BY owner
ORDER BY 3 DESC
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p2_snapshot_date	Datum zajema

- **SQL stavek za Filetype size per snapshot poročilo**

a) Izvorna koda stavka

```
SELECT NULL LINK, snapshot_date label, SUM (file_size) / 1024 /
1024 VALUE
FROM files
WHERE extension = :p2_extension
GROUP BY snapshot date
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p2_extension	Končnica datoteke

- **SQL stavek za User consumption per snapshots poročilo**

a) Izvorna koda stavka

```
SELECT NULL LINK, snapshot_date label, SUM (file_size) / 1024 /
1024 VALUE
FROM files
WHERE owner = :p2_owner
GROUP BY snapshot_date
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p2_owner	Lastnik datoteke

4.2.2 Manual Reports

- **SQL stavek za User consumption per time frame poročilo**

Graf SQL stavek:

a) Izvorna koda stavka

```
SELECT NULL LINK, snapshot_date label, SUM (file_size) / 1024 /
1024 VALUE
FROM files
WHERE INSTR (':' || :p5_snapshot_date,
':' || TO_CHAR (snapshot_date, 'DD.MM.YY')
) > 0
AND owner = :p5_owner
GROUP BY snapshot_date
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p5_snapshot_date	Datum zajema
:p5_owner	Lastnik datoteke

Tabela SQL stavek:

a) Izvorna koda stavka

```
SELECT snapshot_date AS "snapshot date",
TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB)"
FROM files
WHERE INSTR (':' || :p5_snapshot_date,
':' || TO_CHAR (snapshot_date, 'DD.MM.YY')
) > 0
AND owner = :p5_owner
GROUP BY snapshot_date
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p5_snapshot_date	Datum zajema
:p5_owner	Lastnik datoteke

- **SQL stavek za Filetype consumption per time frame poročilo**

Graf SQL stavek:

a) Izvorna koda stavka

```
SELECT NULL, snapshot_date label, SUM (file_size) / 1024 / 1024
VALUE
FROM files
WHERE INSTR (':' || :p6_snapshot_date,
            ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')
            ) > 0
AND extension = :p5_extension
GROUP BY snapshot_date
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p5_snapshot_date	Datum zajema
:p5_extension	Končnica datoteke

Tabela SQL stavek:

a) Izvorna koda stavka

```
SELECT snapshot_date AS "Snapshot date",
       TO_CHAR (SUM (file_size) / 1024 / 1024,
               '9999.99'
               ) AS "files size (MB)"
FROM files
WHERE INSTR (':' || :p5_snapshot_date,
            ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')
            ) > 0
AND extension = :p5_extension
GROUP BY snapshot_date
```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p5_snapshot_date	Datum zajema
:p5_extension	Končnica datoteke

- SQL stavek za File Age per time frame poročilo

Graf SQL stavek:

a) Izvorna koda stavka

```

SELECT  NULL LINK, snapshot_date label, SUM (file_size) VALUE
        FROM files
        WHERE last_write_time BETWEEN (SYSDATE - 180) AND SYSDATE
GROUP BY snapshot_date

SELECT  NULL LINK, snapshot_date label, SUM (file_size) VALUE
        FROM files
        WHERE last_write_time BETWEEN (SYSDATE - 360) AND (SYSDATE - 180)
GROUP BY snapshot_date

SELECT  NULL LINK, snapshot_date label, SUM (file_size) VALUE
        FROM files
        WHERE last_write_time BETWEEN (SYSDATE - 540) AND (SYSDATE - 360)
GROUP BY snapshot_date

SELECT  NULL LINK, snapshot_date label, SUM (file_size) VALUE
        FROM files
        WHERE last_write_time BETWEEN (SYSDATE - 720) AND (SYSDATE - 540)
GROUP BY snapshot_date

SELECT  NULL LINK, snapshot_date label, SUM (file_size) VALUE
        FROM files
        WHERE last_write_time < (SYSDATE - 720)
GROUP BY snapshot_date

```

Tabela SQL stavek:

a) Izvorna koda stavka

```

SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024,
                '9999.99') AS "number of files",
        DECODE (owner, :p7_owner, '<6 months') AS "age"
        FROM files
        WHERE INSTR (':' || :p6_snapshot_date,
                ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')) > 0
        AND owner = :p7_owner
        AND last_write_time BETWEEN (SYSDATE - 180) AND SYSDATE
GROUP BY snapshot_date, owner
UNION
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024,
                '9999.99') AS "number of files",
        DECODE (owner, :p7_owner, '6-12 months') AS "age"
        FROM files
        WHERE INSTR (':' || :p6_snapshot_date,
                ':' || TO_CHAR (snapshot_date, 'DD.MM.YY') ) > 0
        AND owner = :p7_owner
        AND last_write_time BETWEEN (SYSDATE - 360) AND (SYSDATE - 180)
GROUP BY snapshot_date, owner
UNION
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024,

```

```

        '9999.99' ) AS "number of files",
        DECODE (owner, :p7_owner, '12-18 months') AS "age"
    FROM files
    WHERE INSTR (':' || :p6_snapshot_date,
                ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')) > 0
        AND owner = :p7_owner
        AND last_write_time BETWEEN (SYSDATE - 540) AND (SYSDATE - 360)
    GROUP BY snapshot_date, owner
UNION
SELECT  snapshot_date AS "snapshot date"
        TO_CHAR (SUM (file_size) / 1024 / 1024,
                '9999.99') AS "number of files",
        DECODE (owner, :p7_owner, '18-24 months') AS "age"
    FROM files
    WHERE INSTR (':' || :p6_snapshot_date,
                ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')) > 0
        AND owner = :p7_owner
        AND last_write_time BETWEEN (SYSDATE - 720) AND (SYSDATE - 540)
    GROUP BY snapshot_date, owner
UNION
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024,
                '9999.99') AS "number of files",
        DECODE (owner, :p7_owner, '>24 months') AS "age"
    FROM files
    WHERE INSTR (':' || :p6_snapshot_date,
                ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')) > 0
        AND owner = :p7_owner
        AND last_write_time < (SYSDATE - 720)
    GROUP BY snapshot_date, owner

```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p6_snapshot_date	Datum zajema
:p7_owner	Lastnik datoteke

4.2.3 Storage Management

- SQL stavek za Storage Cost per User poročilo

a) Izvorna koda stavka

```

SELECT  a.snapshot_date AS "Snapshot date", a.owner AS "User",
        TO_CHAR ((SUM (a.file_size) / 1024 / 1024 / 1024) *
        '9999.99'
                ) AS "Cost",
        b.disk_type AS "Disk type", b.disk_cost AS "Disk Cost
    EUR/GB"
    FROM files a, disks b, STORAGE c
    WHERE INSTR (':' || :p8_snapshot_date,
                ':' || TO_CHAR (a.snapshot_date, 'DD.MM.YY')
                ) > 0

```



```

AND c.storage_name = :p8_storage
AND a.owner = :p8_owner
AND b.disk_type = :p8_disk_type
AND c.ID = b.storage_id
GROUP BY owner, disk_cost, disk_type, snapshot_date

```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p8_snapshot_date	Datum zajema
:p8_owner	Lastnik datoteke
:p8_storage	Diskovno polje
:p8_disk_type	Tip diska v diskovnem polju

- **SQL stavek za Storage Cost poročilo**

a) Izvorna koda stavka

```

SELECT a.snapshot_date AS "Snapshot date", a.owner AS "User",
       TO_CHAR ((SUM (a.file_size) / 1024 / 1024 / 1024) *
b.disk_cost, '9999.99') AS "Cost EUR",
       b.disk_type AS "Disk type", b.disk_cost AS "Disk Cost
EUR/GB"
FROM files a, disks b, STORAGE c
WHERE INSTR (':' || :p9_snapshot_date,
            ':' || TO_CHAR (a.snapshot_date, 'DD.MM.YY') ) > 0
AND c.storage_name = :p9_storage
AND b.disk_type = :p9_disk_type
AND c.ID = b.storage_id
GROUP BY owner, disk_cost, disk_type, snapshot_date

SELECT TO_CHAR (SUM ((SUM (a.file_size) / 1024 / 1024 / 1024) *
b.disk_cost),
              '9999.99'
          ) AS "Cost EUR"
FROM files a, disks b, STORAGE c
WHERE INSTR (':' || :p9_snapshot_date,
            ':' || TO_CHAR (a.snapshot_date, 'DD.MM.YY')
          ) > 0
AND c.storage_name = :p9_storage
AND b.disk_type = :p9_disk_type
AND c.ID = b.storage_id
GROUP BY snapshot date, disk cost

```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p9_snapshot_date	Datum zajema
:p9_owner	Lastnik datoteke
:p9_storage	Diskovno polje
:p9_disk_type	Tip diska v diskovnem polju

4.2.4 Storage Tiers

- SQL stavek za Tier breakdown (per User) poročilo

a) Izvorna koda stavka

```

SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
  FROM files
 WHERE INSTR (':' || :p18_snapshot_date,
             ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')) > 0
        AND owner = :p19_owner
        AND last_write_time BETWEEN (SYSDATE - 180) AND SYSDATE
 GROUP BY snapshot_date, owner
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
  FROM files
 WHERE INSTR (':' || :p18_snapshot_date,
             ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')) > 0
        AND owner = :p19_owner
        AND last_write_time BETWEEN (SYSDATE - 360) AND (SYSDATE - 180)
 GROUP BY snapshot_date, owner
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
  FROM files
 WHERE INSTR (':' || :p18_snapshot_date,
             ':' || TO_CHAR (snapshot_date, 'DD.MM.YY') ) > 0
        AND owner = :p19_owner
        AND last_write_time < (SYSDATE - 720)
 GROUP BY snapshot_date, owner

```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p18_snapshot_date	Datum zajema
:p19_owner	Lastnik datoteke

- SQL stavek za Tier breakdown (per filetype) poročilo

a) Izvorna koda stavka

```

SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
  FROM files
 WHERE INSTR (':' || :p19_snapshot_date,
             ':' || TO_CHAR (snapshot_date, 'DD.MM.YY') ) > 0
        AND extension = :p19_extension
        AND last_write_time BETWEEN (SYSDATE - 180) AND SYSDATE
 GROUP BY snapshot_date, extension
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "

```

```

FROM files
WHERE INSTR (':' || :p19_snapshot_date,
            ':' || TO_CHAR (snapshot_date, 'DD.MM.YY') ) > 0
      AND extension = :p19_extension
      AND last_write_time BETWEEN (SYSDATE - 360) AND (SYSDATE - 180)
GROUP BY snapshot_date, extension
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
FROM files
WHERE INSTR (':' || :p19_snapshot_date,
            ':' || TO_CHAR (snapshot_date, 'DD.MM.YY')) > 0
      AND extension = :p19_extension
      AND last_write_time < (SYSDATE - 720)
GROUP BY snapshot_date, extension

```

b) Spremenljivke v stavku

Ime spremenljivke	Opis
:p19_snapshot_date	Datum zajema
:p19_extension	Končnica datoteke

- **SQL stavek za Full Tier Breakdown poročilo**

a) Izvorna koda stavka

```

-- tier 1
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
FROM files
WHERE last_write_time BETWEEN (SYSDATE - 180) AND SYSDATE
GROUP BY snapshot_date
-- tier 2
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
FROM files
WHERE last_write_time BETWEEN (SYSDATE - 360) AND (SYSDATE - 180)
GROUP BY snapshot_date
-- tier 3
SELECT  snapshot_date AS "snapshot date",
        TO_CHAR (SUM (file_size) / 1024 / 1024, '9999.99') AS "Size
(MB) "
FROM files
WHERE last_write_time < (SYSDATE - 720)
GROUP BY snapshot_date

```

4.3 DCTool Agent

DCTool Agent aplikacijo sestavljajo funkcije in procedure za branje datotek ter zapis podatkov v podatkovno bazo. V nadaljevanju sledi podroben opis funkcij in procedur definiranih v DCTool Agent aplikaciji.

4.3.1 Preiskovanje datotečnega sistema

a) **Oblika funkcije**

```
void DirSearch(string sDir)
```

Vhodni parametri:

Ime	Podatkovni tip	Opis
sDir	String	Ime diska

b) **Klic funkcije**

Funkcija se kliče v zanki, ki izvede rekurzivni klic funkcije čez vse diske, ki smo jih izbrali preko grafičnega vmesnika.

c) **Opis funkcionalnosti**

Ob klicu se preko vhodnega parametra prenese ime diska (C:\, X:\,...), ki ga želimo pregledati. Funkcija se rekurzivno sprehodi čez vse mape in datoteke ter prebere podatke o datotekah, ki se zapišejo v tabelo FILES.

4.3.2 Povezava s podatkovno zbirko

a) **Oblika funkcije**

```
public int connectOracle(String connectString)
```

Vhodni parametri:

Ime	Podatkovni tip	Opis
connectString	String	Podatki za povezavo s podatkovno bazo

b) **Klic funkcije**

Funkcija se pokliče pred pregledovanjem datotek na označenih diskih.

c) **Opis funkcionalnosti**

Funkcija poskuša vzpostaviti povezavo s podatkovno zbirko in kot izhodni parameter vrne rezultat poskusa. Če ima izhodni parameter vrednost 0, potem se je DCTool Agent uspešno povezal s podatkovno zbirko. Napako funkcija javi preko pogovornega okna in z vrednostjo izhodnega parametra 1.

Primer connectString spremenljivke:

```
"provider=MSDAORA;Data Source=" + texDatabase.Text + ";User Id=" + textUsername.Text + ";Password=" + textPassword.Text + ";"
```

4.3.3 Vpis podatkov v podatkovno zbirko

a) Oblika funkcije

```
public void oraInsert(String file_name, String extension, String path, String
owner, String group_name, long size_file, String creation_time String
last_write_time, String attributes)
```

Vhodni parametri:

Ime	Podatkovni tip	Opis
File_name	String	Ime datoteke.
Extension	String	Končnica datoteke.
Path	String	Pot do datoteke.
Owner	String	Lastnik datoteke.
Group_name	String	Uporabniška skupina.
Size_file	Long	Velikost datoteke.
Creation_time	String	Datum nastanka datoteke.
Last_write_time	String	Datum zadnjega dostopa do datoteke.
Attributes	String	Atributi datoteke.

b) Klic funkcije

Funkcija se pokliče za vsako prebrano datoteko.

c) Opis funkcionalnosti

Preko vhodnih parametrov se prenesejo lastnosti datoteke v funkcijo, ki zgradi SQL stavek za vnos podatkov in podatke zapiše v podatkovno zbirko.

4.3.4 Podatki o uporabniku in uporabniški skupini datoteke

a) Oblika funkcije

```
string QueryFileSecurity(string fileName)
```

Vhodni parametri:

Ime	Podatkovni tip	Opis
fileName	String	Ime datoteke

a) Klic funkcije

Funkcija se izvede za vsako prebrano datoteko.

b) Opis funkcionalnosti

Funkcija za vsako prebrano datoteko prebere lastnika datoteke in njemu pripadajočo skupino.

4.3.5 Zajem podatkov**a) Oblika funkcije**

`public void DataCollection()`

b) Klic funkcije

Funkcija se kliče s klikom na gumb »Start« na uporabniškem vmesniku.

c) Opis funkcionalnosti

Funkcija poskrbi za povezavo na podatkovno zbirko, izpisovanje dnevnika na uporabniškem vmesniku in klic funkcije za branje podatkov o datotekah.

5. Predlogi za nadaljnji razvoj sistema

Sistem DCTool je nastal iz dolgoletnih izkušenj, ki smo jih v podjetju S&T Slovenija pridobili skozi prodajo in vzdrževanje diskovnih sistemov različnih proizvajalcev. Menim, da je DCTool sistem pravi pristop k izvajanju storitvenega računalništva na področju diskovnih polj. Ker je sistem še v nastajanju, obstaja še veliko prostora za izboljšave. V tem poglavju je predstavljeno nekaj izboljšav, za katere menim, da bi jih bilo smiselno dodati v prihodnosti.

Izvoz poročil

V trenutni verziji DCTool Reports aplikacije si lahko poročila ogledamo samo preko spletnega vmesnika. Okrnjena funkcionalnost vmesnika nam omogoča samo izdelavo in pregled poročil. Sistemu bi bilo smiselno dodati možnost izpisa v zelen format. DCTool Reports temelji na Oracle APEX (Application Express) tehnologiji, ki omogoča enostavno povezovanje z ostalimi tehnologijami programske hiše Oracle, kot je orodje Oracle BI Publisher za izdelavo poročil nad podatkovnim modelom. Potrebno bi bilo razviti vmesnik za povezavo predlog narejenih v Oracle BI Publisher s spletnim vmesnikom DCTool Reports. Spletni vmesnik bi moral podpirati tudi enostaven način dodajanja novih predlog preko spletnega vmesnika, saj bo vsaka stranka želela svojo obliko poročil, ki bodo skladna z njihovo grafično podobo.

Izvoz podatkov o uporabi diskovnih polj v XML format

V organizacijah, kjer morajo oddelki plačevati storitve IT oddelku, bi moral sistem podpirati izvoz podatkov o uporabi diskovnih polj v XML format. Tega bi lahko kasneje uvozili v organizacijski ERP sistem z namenom zaračunavanja storitev. Namesto izvoza v XML format bi lahko pripravili tudi spletno storitev, preko katere bi ponujali podatke ERP sistemom v organizaciji. Za implementacijo spletnih storitev bi potrebovali še dodatno strežniško aplikacijo. Stranke, ki uporabljajo podatkovno zbirko Oracle 11g, pa dodatne strežniške aplikacije ne bi potrebovale, saj ima podatkovna zbirka že vključeno funkcionalnost za izdelavo spletnih storitev.

Izračun predvidene rasti zasedenosti diskovnih polj

Izračun predvidene rasti zasedenosti diskovnih polj je še ena funkcionalnost, ki bi jo bilo potrebno dodati DCTool Reports aplikaciji. Potrebno bi bilo razviti algoritem, ki bi iz obstoječih posnetkov stanj razbral trend rasti podatkov po uporabnikih, uporabniških skupinah, tipih datotek ali po kvartalnih. Napoved rasti bi stranke uporabljale kot pomoč pri odločitvah o nakupu diskovnih polj.

Izdelava agentov v Javi

V trenutni verziji sistema DCTool so agenti realizirani v programskem jeziku C# in delujejo samo na Microsoft Windows platformi. Z agenti razvitimi v Javi bi pokrili večji spekter operacijskih sistemov pri čemer bi vzdrževali samo eno verzijo agenta. Prvotni načrt je bil, da bodo agenti razviti v Javi, vendar trenutna verzija Jave (1.6) še ne vsebuje vseh metod za delo

z datotekami, ki jih agent protrebuje. Naslednja verzija Jave (1.7) bo prišla na trg predvidoma sredi leta 2009. S prepisom agentov v Javo bi si poenostavili tudi namestitev in vzdrževanje verzij agentov, saj bi lahko uporabili sistem webstart, ki omogoča enostavno namestitev preko spleta in avtomatsko nadgrajevanje. Ko se pojavi nova verzija na lokaciji s katere se razpršujejo agenti, se agent avtomatsko nadgradi.

Poenostavi se tudi komunikacija s podatkovnimi zbirkami. Java agenti ne bi potrebovali nameščene dodatne programske opreme kot je Oracle 10gR2 Client, ker bi lahko komunicirali s podatkovnimi zbirkami preko JDBC vmesnikov, ki so že vsebovani v Javi. Potrebno je dodati samo gonilnike za posamezno podatkovno zbirko, ki pa jih lahko vključimo v samo aplikacijo.

6. Sklepne ugotovitve

Model storitvenega računalništva hitro prodira v svet računalništva. Vsi veliki proizvajalci strojne in programske opreme se tega zavedajo in pripravljajo produkte, pri katerih upoštevajo ta model. Nekaj komercialnih rešitev s področja diskovnih polj sem predstavil v 3. poglavju.

Sistem DCTool je samostojno razvit produkt za vpeljavo modela storitvenega računalništva na področju diskovnih polj. Produkt sem začel razvijati z namenom, da bi poenotili in poenostavili izdelavo študij implementacij diskovnih polj pri naših strankah. Ker predstavlja nakup diskovnega polja 40-60% letnega proračuna IT oddelka, želi stranka pred nakupom pretehtati vse možnosti. Za samostojen razvoj smo se odločili, ker komercialni produkti ne nudijo vseh funkcionalnosti, ki jih potrebujemo.

Prvi rezultati uporabe sistema bodo vidni šele naslednje leto. Vse izboljšave omenjene v 5. poglavju izhajajo iz internega testiranja. Poleg izboljšav se je pojavilo tudi nekaj problemov, ki jih še nisem uspel rešiti. To so predvsem problemi s povezovanjem seznamov uporabnikov ter urejanje uporabnikov v skupine, ki so zrcalna slika organigrama podjetja.

Z vpeljavo sistema DCTool se bo delo svetovalcev na področju diskovnih polj poenostavilo in poenotilo. Menim, da bo z uporabo narejen korak naprej pri kvaliteti izdelanih študij na področju uvajanja diskovnih polj v podjetju S&T Slovenija.

Dodatek A

A.1 Dodatek

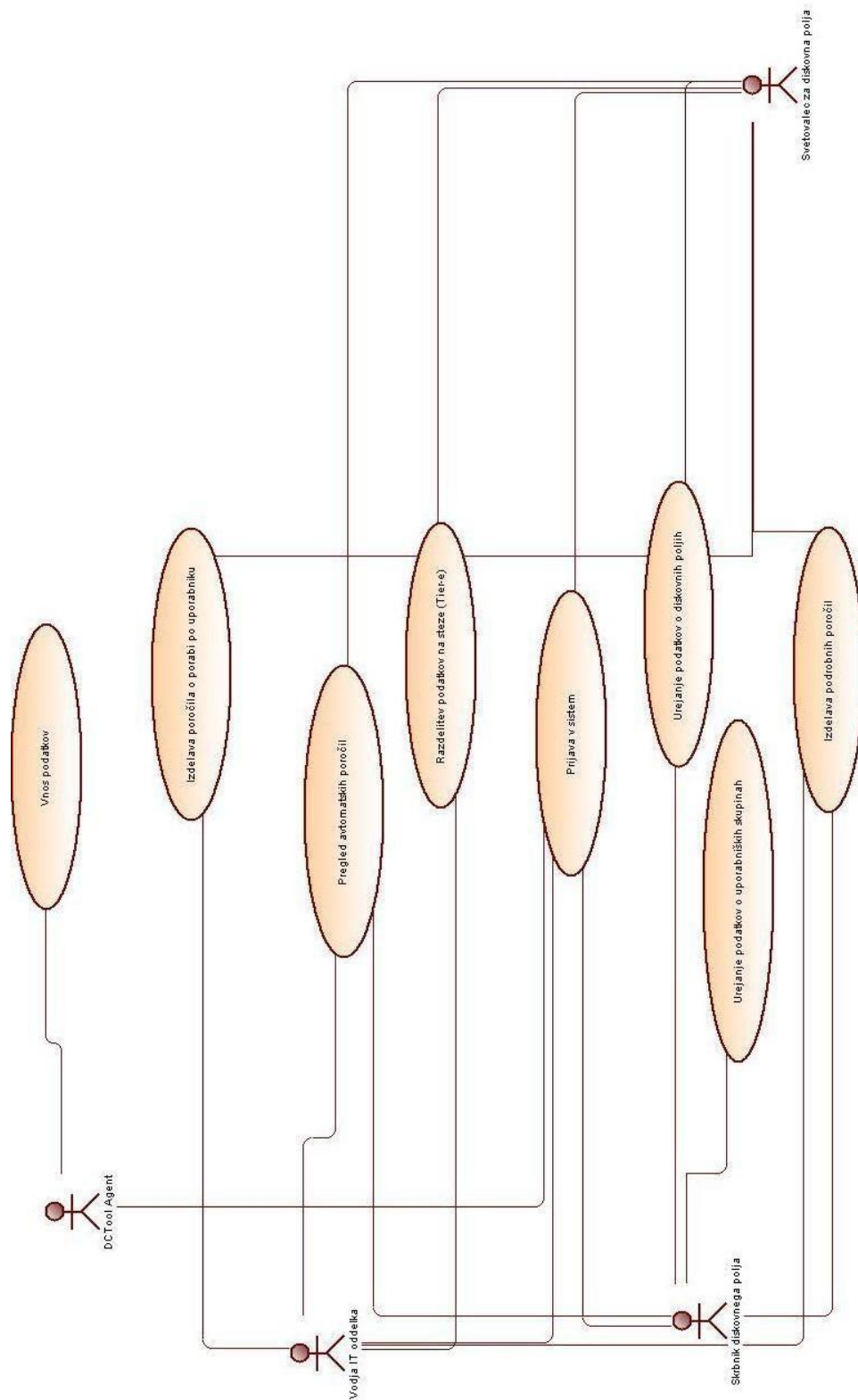


Diagram 7.2 – Use Case diagram aplikacije DCTool

A.2 Dodatek

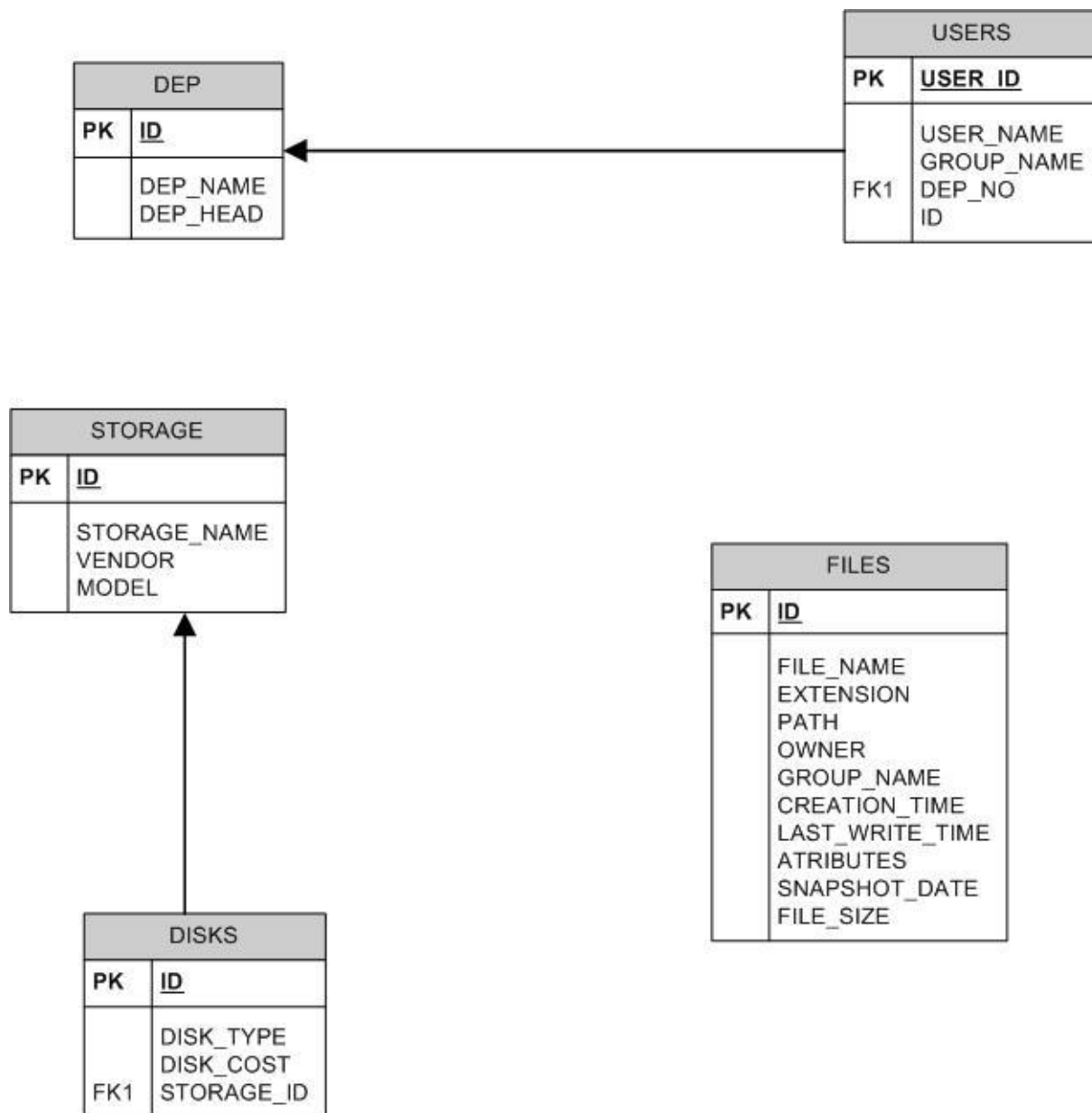


Diagram 7.2 – ERD diagram - podatkovni model DCTool Reports

Slike

Slika 3.2 – DCTool Reports uporabniški vmesnik	15
Slika 3.3 – Poročilo Snapshot Size	17
Slika 3.4 – Poročilo Filetype distribution	18
Slika 3.5 – Poročilo User consumption per snapshot (Graf).....	18

Diagrami

Diagram 7.2 – Use Case diagram aplikacije DCTool	41
Diagram 7.2 – ERD diagram - podatkovni model DCTool Reports	42

Tabele

Tabela 3.1 – Spremenljivke v tnsnames zapisu.....	12
Tabela 3.2 - Seznam gumbov na zaslonski maski DCTool Agenta	14
Tabela 3.3 – Spremenljivke v SQL ukazu za kreiranje prostora v podatkovni zbirki..	16
Tabela 3.4 – Seznam zavihkov v DCTool Reports	16
Tabela 3.5 – Polja v poročilu User consumption per snapshot (Report in Chart).....	19
Tabela 3.6 – Polja v poročilu User consumption per time frame	19
Tabela 3.7 – Polja v poročilu Filetype consumption per time frame	19
Tabela 3.8 – Polja v poročilu Storage Cost per User	20
Tabela 3.9 – Časovna opredelitev nivojev na diskovnem polju.....	21
Tabela 4.1 – Polja v tabeli FILES	23
Tabela 4.2 – Polja v tabeli STORAGE.....	24
Tabela 4.3 – Polja v tabeli DISKS.....	24
Tabela 4.4 – Polja v tabeli USERS.....	24
Tabela 4.5 – Polja v tabeli DEP.....	25

Viri

[1] Roger S. Reid, Gareth Fraser-King, W. David Schwaderer: Data Lifecycles Managing data for strategic advantages, 2007

[2] On-Demand, Grid & Utility Computing Business Resources - Independent industry information for the fifth utility

URL: <http://utilitycomputing.com/>

[3] IBM Systems Journal - Vol. 43, No. 1, 2004 - Utility Computing

URL: <http://www.research.ibm.com/journal/sj43-1.html>

[4] IBM System Storage Archive Manager

URL: <http://www-01.ibm.com/software/tivoli/products/storage-mgr-data-reten/>

[5] Cybernetics, Time-sharing, Human-Computer Symbiosis and On-line Communities

URL: <http://www.columbia.edu/~rh120/ch106.x06>

[6] Oracle Application Express documentation

URL: <http://www.oracle.com/technology/documentation/apex.html>

[7] Oracle 10gR2 RDBMS documentation

URL: <http://tahiti.oracle.com>

[8] C# .NET documentation

URL: <http://msdn.microsoft.com/en-us/library/ms269115.aspx>

[9] Symantec Enterprise Vault

<http://www.symantec.com/business/enterprise-vault>

[10] EMC DiskXtender

<http://www.emc.com/products/family/diskxtender-family.htm>

[11] CommVault Data Archiver

<http://www.commvault.com/products/archive.asp>

Izjava o samostojnosti dela

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja prof. dr. Dušana Kodeka. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.