

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

MAGISTRSKO DELO

BAZA ZNANJA ZA PODPORO UPORABNIKOM

Ljubljana, maj 2002

Mateja Bezjak

IZJAVA

Študentka **Mateja Bezjak** izjavljam, da sem avtorica tega magistrskega dela, ki sem ga napisala pod mentorstvom **doc. dr. Mojce Indihar Štemberger** in skladno s 1. odstavkom 21. člena Zakona o avtorskih in sorodnih pravicah dovolim objavo magistrskega dela na fakultetnih spletnih straneh.

V Ljubljani, dne _____

Podpis: _____

KAZALO VSEBINE

1	UVOD	1
1.1	Problematika magistrskega dela	1
1.2	Namen dela	3
1.3	Cilj dela	3
1.4	Metode in zasnova dela	4
2	SISTEMI PODPRTI Z ZNANJEM	5
2.1	Zgodovinski razvoj metod umetne inteligence	5
2.2	Kaj so sistemi podprti z znanjem?	6
2.3	Ekspertni sistemi	6
2.3.1	<i>Funkcije ekspertnih sistemov</i>	8
2.3.2	<i>Znanje kot del ekspertnega sistema</i>	8
2.4	Skupne značilnosti nalog reševanih s sistemi podprtimi z znanjem	10
2.5	Struktura in delovanje sistema podprtega z znanjem	10
3	UPRAVLJANJE Z ZNANJEM	14
3.1	O čem govorimo, kadar govorimo o znanju in upravljanju z njim?	14
3.2	Opredelitev znanja	16
3.2.1	<i>Izkušnje</i>	17
3.2.2	<i>Resnica</i>	17
3.2.3	<i>Razsodnost</i>	17
3.2.4	<i>Pravila sklepanja in intuicija</i>	18
3.2.5	<i>Vrednote in prepričanja</i>	18
3.3	Različni pogledi na znanje	18
3.4	Učenje in vrste učenja	21
3.4.1	<i>Učeča se organizacija</i>	22
3.5	Pomen znanja v podjetju	24
3.6	Pridobivanje znanja	25
3.7	Kodifikacija in usklajevanje znanja	26
3.7.1	<i>Mapiranje in modeliranje znanja</i>	28
3.8	Oblikovanje kulture znanja v podjetju	28
3.8.1	<i>Inovativna organizacijska kultura in kultura znanja</i>	30
3.9	Prenos znanja	31
3.9.1	<i>Strategije prenosa znanja</i>	34
3.10	Vloge in izkušnje pri upravljanju z znanjem	35
4	BAZA ZNANJA KOT DEL SISTEMA ZA UPRAVLJANJE Z ZNANJEM	39
4.1	Predstavitev znanja	40
4.1.1	<i>Produksijska pravila</i>	41
4.1.2	<i>Okviri</i>	45

4.1.3	<i>Semantične mreže</i>	46
4.1.4	<i>Predikatni račun</i>	48
4.2	Načini organiziranja znanja v bazo znanja	49
4.2.1	<i>Iskanje po celotnem besedilu</i>	50
5	IZGRADNJA SISTEMA ZA UPRAVLJANJE Z ZNANJEM S POMOČJO	
	ORODJA RATIONAL CLEARDDTS	52
5.1	Predstavitev	52
5.2	Uporaba v podjetju	54
5.3	Shranjevanje podatkov	58
5.3.1	<i>Iskanje po bazi</i>	59
5.3.2	<i>Oblikovanje poročil</i>	60
5.4	Gradnja baze znanja	63
5.4.1	<i>Zagon skripte</i>	70
5.5	Primer uporabe baze znanja	74
6	SKLEP	77
7	LITERATURA IN VIRI	80
8	PRILOGE	85

KAZALO SLIK

Slika 1: Predstavitev nivojev pomoči pri podpori produkta	2
Slika 2: Zgradba ekspertnega sistema	12
Slika 3: Povezava med znanjem, informacijami in podatki	14
Slika 4: Vzhodni in zahodni pogled na znanje	19
Slika 5: Hewlett Packardovi forumi znanja.....	32
Slika 6: Rezultati ankete revije Knowledge Management	37
Slika 7: Sestavni deli baze znanja.	39
Slika 8: Pravila sklepanja.....	44
Slika 9: Organizacija vozlišča v sistemu okvirov	46
Slika 10: Izbira razreda in projekta	53
Slika 11: Stanja, med katerimi lahko izbiramo.	54
Slika 12: Odprti klic v ClearDDTS bazi za projekt <i>sd.calls</i>	55
Slika 13: Iskanje po bazi s pomočjo grafičnega vmesnika	59
Slika 14: Iskalni niz	60
Slika 15: Izbiranje iskalnih polij	60
Slika 16: Primer tedenskega poročila	60
Slika 17: Primer mesečnega poročila	63
Slika 18: Vsebina priloge PROBLEM.....	64
Slika 19: Priloga PROBLEM v ClearDDTS bazi.....	65
Slika 20: Primer tabele	65
Slika 21: SQL stavek ddtssql.kdb datoteki.....	66
Slika 22: Vpogled v uporabo ukaza <i>dumpug</i> (man <i>dumpbug</i>)	67
Slika 23: <i>dumpbug.kdb</i> skripta.....	68
Slika 24: Uvodna stran baze znanja Software Distributor produkta	69
Slika 25: Vpogled v direktorij, kjer sem pognala <i>kdb</i> skripto	71
Slika 26: Vsebina datoteke HSLco27558	72
Slika 27: Iskalec.....	73
Slika 28: Končna podoba izdelane baze znanja	74
Slika 29: Sprememba IP naslova na SD strežniku	75
Slika 30: Primer iskanja	75

KAZALO TABEL

Tabela 1: Razlike med znanjem in informacijo	15
Tabela 2: Dejavniki inovativnosti v podjetju	30
Tabela 3: Trenja pri prenosu znanja	34

1 Uvod

1.1 *Problematika magistrskega dela*

Konkurenčne prednosti posameznikov, podjetij in držav so danes postale odvisne od sposobnosti trajnega vzdrževanja znanja in njegovega povečevanja v procesu učenja. Znanje in njegovo uspešno upravljanje sta temelj uspešnosti organizacije, njenega obstoja in poslovne prihodnosti.

Vsak zaposleni v podjetju ima neko znanje. Znanje posameznikov predstavlja enega ključnih virov slovenskega in svetovnega gospodarstva. Danes se podjetja vse bolj ukvarjajo z vprašanjem, kako to vrednoto upravljati in z njo gospodariti, da bodo izdelki ali storitve, ki jih ponujajo na trgu, kvalitetne in konkurenčne.

Vedno pogosteje se srečujemo s pojmom "Knowledge Management". V slovenščini nisem opazila enotnega prevoda, še največkrat sem se srečala s pojmom "upravljanje z znanjem", ki sem ga tudi uporabila v svoji nalogi. Poleg tega sem še zasledila pojme "ravljanje z znanjem", "gospodarjenje z znanjem", "poslovođenje z znanjem" in tudi "upravljanje znanja". Sama sem se odločila uporabiti "upravljanje z znanjem".

K znanju usmerjen prijem zahteva od celotne organizacije ter prav tako njenega vodstva spremembe v načinu vodenja in korenite miselne premike vseh zaposlenih. Vsak tak projekt je namreč projekt spreminjanja kulture ljudi, ki jih je treba osvestiti, da delijo svoje znanje z drugimi zaposlenimi v organizaciji. To je vsekakor težka naloga, saj je znanje tisto, ki daje posamezniku konkurenčno prednost in mu hkrati zagotavlja boljši položaj v podjetju. Mnogokrat se to kaže v nematerialnih in materialnih ugodnostih, pa tudi v ohranjanju delovnega mesta.

Bistvo upravljanja z znanjem je, da se znanje, izkušnje in sposobnosti posameznikov ter strukturno znanje, ki obstaja v obliki procesov, izdelkov, storitev, pretakajo skozi celotno organizacijo, s čimer postanejo le-ti bolj izkoriščeni in z učinki sinergije bolj koristni.

Veliko ali pa že skoraj vsako podjetje ima nekatera znanja shranjena. In sicer v obliki dokumentov, pravil ali postopkih dela. Na nek način imajo svojo bazo znanja.

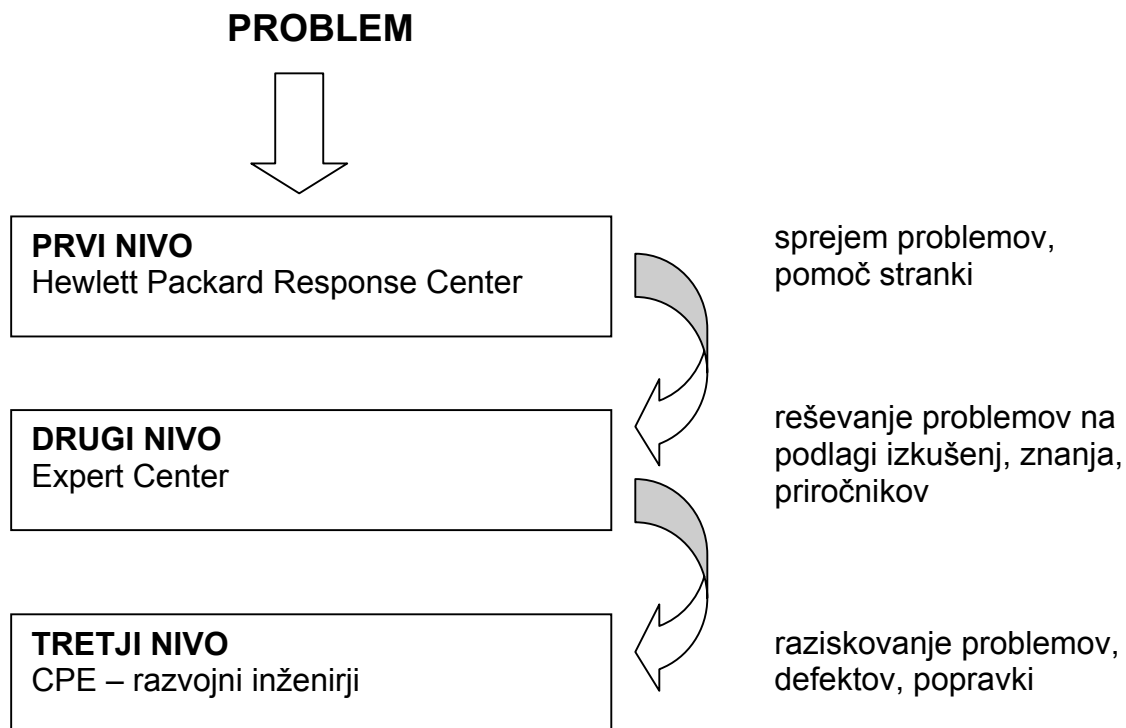
Baza znanja je strukturna komponenta sistemov podprtih z znanjem (Knowledge Based Systems). Kvaliteta sistema podprtega z znanjem je nedvomno v veliki meri odvisna od obsega in kvalitete njegove baze znanja. Baza znanja vsebuje znanje o specifičnem področju ali domeni, ki je ponavadi strokovne ali znanstvene narave.

Vsebuje tudi pravila, ki opisujejo dogodke, relacije med dogodki in metode hevristike za reševanje problemov.

V podjetju Hermes Softlab d.d. uporabljamo orodje Rational ClearDDTS kot vir za interno bazo znanja. V projektnem teamu opravljam funkcijo inženirja za podporo strankam dveh Hewlett Packardovih produktov: OpenView IT/Administration (ITA) in OpenView Software Distribution (SD).

Kadarkoli se pojavi stranki nov problem, ki ga sama ali pa s pomočjo prve stopnje pomoči Hewlett Packard centrov po svetu ne zna rešiti, se obrne na nas. Mi predstavljamo (slika 1) drugi in hkrati tretji nivo podpore strankam. Pod drugi nivo spadajo vsi problemi, ki jih lahko inženirji za podporo strankam rešijo sami (na podlagi znanja, izkušenj ali brskanja po bazi starih problemov), pod tretji nivo pa spadajo problemi, ki zahtevajo pomoč razvojnih (CPE) inženirjev. V slednjem primeru inženir za podporo strankam problem glede na področje dodeli razvojnemu inženirju v nadaljnjo obravnavo.

Slika 1: Predstavitev nivojev pomoči pri podpori produkta



Sodelovanje s strankami oziroma Hewlett Packard centri je pretežno v obliki elektronske pošte. Le redko se zgodi, da stranke prijavijo problem po telefonu. Inženirji v Hewlett Packard centrih ponavadi nimajo specifičnega znanja o posameznih produktih, saj morajo nuditi strankam pomoč pri vseh produktih skupine Hewlett Packard OpenView. Njihova naloga je zbrati čim več koristnih podatkov o problemu ter ga nato posredovati drugemu nivoju pomoči.

Z leti in izkušnjami dela na področju podpore strankam se je naša interna baza znanja dopolnjevala in predstavlja nepogrešljiv vir pri delu inženirja za podporo strankam. Ta baza ni dostopna Hewlett Packard centrom po svetu, zato se dostikrat dogaja, da so centri samo posredniki med nami in strankami.

Da Hewlett Packard centri ne bi bili samo posredniki, bi bilo potrebno določeno znanje o produktu posredovati tudi njim. Njihova naloga je namreč prestreči čim večje število problemov in nam posredovati le težje, zahtevnejše probleme.

K rešitvi problema bi veliko pripomogla priredba naše baze znanja v obliki pogosto zastavljenih vprašanj oziroma javljenih problemov glede na področje samega produkta.

1.2 Namen dela

Namen magistrskega dela je s pomočjo domače in tuje strokovne literature celovito preučiti bazo znanja kot del sistema za upravljanje z znanjem in prikazati izdelavo baze znanja v projektnem teamu ter njeno uporabo.

Moj namen je sestaviti takšno bazo znanja Hewlett Packard centrom, da bi lahko njena uporaba pripomogla k zmanjšanju števila posredovanih problemov na drugi nivo pomoči. Da bi Hewlett Packard centri ne bili samo posredniki, bom iz naše baze znanja zajela opise in rešitve problemov, ki se nanašajo na Hewlett Packardov OpenView produkt Software Distributor. Oboje bom vključila v specifično bazo znanja o Software Distributor produktu. Moj namen pri tem je, da se problemi nanašajo na področje uporabe, kot so na primer pri Software Distributor produktu controller, agent, security, licensing, grafični uporabniški vmesnik in podobno.

Namen naloge je tudi prikazati, kako lahko Hewlett Packard centri in inženirji za podporo strankam pridejo do iskanih informacij. Poleg tega bom prikazala tudi, kako našo bazo dopolnjujemo.

1.3 Cilj dela

Poglavitni cilji magistrskega dela so:

- S pomočjo domače in tuje strokovne literature predstaviti sisteme podprte z znanjem, narediti prehod od ekspertnih sistemov k definiciji in razčlembi znanja. Opisati načine pridobivanja znanja, njegovega učinkovitega prenosa, vloge posameznikov v podjetju in načine kreiranja ter kodiranja znanja. Znanje je sestavni del baze znanja, zato bom podrobneje opisala samo bazo znanja kot del sistema za upravljanje z znanjem

- ❑ Predstaviti bazo znanja, ki jo uporabljamo v našem podjetju, njeno delovanje in način shranjevanja podatkov. Prikazati, kako pridemo do iskanih podatkov in kako bazo dopolnujemo.
- ❑ Zgraditi specifično bazo znanja v obliki pogosto zastavljenih vprašanj in javljenih problemov po določenih produktnih področjih.

1.4 Metode in zasnova dela

Magistrsko delo se opira na teoretična dejstva o sistemih podprtih z znanjem, bazi znanja in upravljanja z znanjem, ki so zajeta iz tuje in domače strokovne literature. Delo sem razdelila na štiri glavna poglavja ter sklep, kjer sem povzela nekatere pomembnejše ugotovitve. V uvodnem delu pa predstavljam ključno problematiko dela, namen in cilje magistrskega dela.

S pomočjo na koncu navedene literature sem v drugem poglavju predstavila sisteme podprte z znanjem, zlasti ekspertne sisteme.

Tretje poglavje se opira na novejšo raziskavo o upravljanju z znanjem (knowledge management) in pomembnosti dobrega upravljanja z znanjem za uspešnost podjetja, medtem ko sem v četrtem poglavju predstavila bazo znanja kot del sistema za upravljanje z znanjem.

V petem poglavju sem se oprla na orodje, ki ga uporabljamo v Hermes Softlabu, na Rational ClearDDTS. Predstavila sem samo orodje, njegovo uporabo v našem projektnem teamu in dodala praktične primere uporabe baze znanja pri iskanju rešitev problemov. Pri tem sem se oprla na praktične izkušnje uporabe Rational ClearDDTS orodja.

Sestavila sem tudi pogosto uporabljena vprašanja oziroma sklop ponavljajočih se problemov o Software Distributor produktu, ki bodo posredovana na Hewlett Packardov intranet. Takšna baza znanja naj bi bila v pomoč Hewlett Packard centrom po svetu in naj bi zmanjšala število dohodnih problemov. Podrobno sem opisala nastajanje in potek dela pri gradnji baze znanja. Po končani gradnji končujem s predstavitvijo njene uporabe.

Magistrsko delo končujem s pomembnejšimi ugotovitvami, do katerih sem prišla s preučevanjem literature in empiričnega dela, s sklepnimi mislimi in predlogi.

2 Sistemi podprti z znanjem

Sistemi podprti z znanjem (knowledge-based systems) so programi, pri katerih potek izvajanja ni vnaprej določen, ampak različne zahteve iz okolja sprožajo različne načine izvajanja. Sistem podprt z znanjem se odloča na osnovi znanja o problemih (zbirke znanja), ki je del sistema. Razvoj sega v 60-ta leta prejšnjega stoletja in je znan tudi pod imenom umetna inteligenca.

2.1 Zgodovinski razvoj metod umetne inteligence

Izraz umetna inteligenca (artificial intelligence, AI) je utemeljil leta 1961 Marvin Minsky (Massachusetts Institute of Technology) v odmevnem članku z naslovom *Steps towards artificial intelligence*. Področje umetne inteligence kot stroke naj bi bil študij duševnih aktivnosti s pomočjo računalniških modelov, rezultat raziskav pa sposobnost računalnikov, da oponašajo človekove načine reševanja nalog. Osnovna predpostavka, na kateri temelji celo področje umetne inteligence, je tako imenovana osrednja dogma, ki pravi, da je delovanje možganov na določenem nivoju neka vrsta računskih postopkov, ki jih je seveda mogoče "ponarediti" z računalnikom (Sistemi podprti z znanjem, 2002).

Za 1960-ta leta je bilo značilno prepričanje, da bo v bližnji prihodnosti mogoče izdelati računalniške programe z inteligenco, podobno človeški. To prepričanje so vsaj na videz potrjevali veliki začetni uspehi metod umetne inteligence. Posebno obetajoči so bili rezultati na nekaterih področjih: teorija iger s programi za igranje šaha, avtomatsko dokazovanje matematičnih teoremov, začetni uspehi analize slik, ki naj bi pripeljali do računalniškega vida, krmiljenje robotov, ki so bili sposobni samostojno zgraditi enostavne strukture iz enostavnih gradnikov, ipd. Dodatni optimizem so vlivali začetni uspehi računalniške implementacije nekaterih teorij jezika. Predvsem je treba omeniti *transformacijsko generativno slovnico* Noama Chomskega, ki je postavila temelje algoritmične obravnave jezikovnih struktur.

Ker je jezik pomembna manifestacija mišljenja, so raziskovalci pravilno sklepali, da bi bilo računalniško razumevanje jezika eden od odločilnih korakov k umetni inteligenci. Tako so bile do srede 70-tih razvite osnovne metode umetne inteligence, kamor sodijo načini računalniške predstavitve znanja, uporaba formalne logike v programih, razpoznavanje vzorcev in odločitvena drevesa pri reševanju problemov. Razvili so se tudi programski jeziki, primerni za razvoj inteligentnih programov (Lisp, Prolog) (Sistemi podprti z znanjem, 2002).

Sredi 70-tih so raziskovalci vzpodbujali sočasen razvoj v treh smereh: splošen razvoj računalniških zmogljivosti, razvoj metod umetne inteligence, predvsem predstavitve znanja in razvoj razumevanja človeškega mišljenja. Predvidevali so, da bo mogoče

kmalu rezultate vseh treh smeri združiti v inteligentnih programih. Konec 70-tih in začetek 80-tih je prinesel prvo veliko razočaranje strokovne javnosti in raziskovalcev umetne inteligence. Z resničnim napredkom vseh treh smeri razvoja se je pokazalo predvsem, da metode umetne inteligence še zdaleč niso sposobne modeliranja človeškega mišljenja. Posledica je bil oster upad financiranja tovrstnih raziskav in dolgotrajno stagniranje teoretičnih področij umetne inteligence. Prišlo je do razkola med "praktiki" in strujo, ki je poskušala modelirati mišljenje. Praktična struja se je usmerila v uporabo dobro raziskanih osnovnih metod umetne inteligence na dobro definiranih problemih. Ta usmeritev, ki je ustrezala tudi preostalim finančnim interesom, je pripeljala do t.i sistemov podprtih z znanjem, med katerimi so najbolj znani *ekspertni sistemi* (Sistemi podprti z znanjem, 2002).

2.2 Kaj so sistemi podprti z znanjem?

Sistemi podprti z znanjem so programi, pri katerih potek izvajanja ni vnaprej določen, ampak različne zahteve iz okolja sprožajo različne načine izvajanja. Sistem podprt z znanjem se odloča na osnovi znanja o problemih (zbirke znanja), ki je del sistema.

Vsakodnevno uporabljan primer enostavnega sistema podprtega z znanjem: slovnični pregledovalnik kot del nekaterih urejevalnikov besedil. Vsebuje znanje o dovoljenih oblikah stavkov, možnih zaporedjih stavčnih členov, ločilih, celo značilnosti različnih stilov. V skladu z značilnostmi posameznih stavkov pregledovanega besedila se sprožajo različna slovnična pravila in uporabniku sporoča ustrezna diagnostika z nasveti (Sistemi podprti z znanjem, 2002).

2.3 Ekspertni sistemi

V sodobnem svetu se morajo podjetja hitro in kvalitetno odločati o poslovnih in strateško pomembnih odločitvah. Pri tem je danes uporaba računalniške programske in strojne opreme nepogrešljiva. Na trgu obstaja veliko število programske opreme v podporo uporabniku za sprejemanje odločitev pri načrtovanju in organizaciji opravil v okviru projektov z upoštevanjem razpoložljivih virov. Sem spadajo:

- ❑ programski paketi za statistično analizo (npr. SPSS, SAS, StatGraph),
- ❑ programski paketi za delo s preglednicami (npr. Microsoft Excel, Quatro Pro, Lotus 1-2-3),
- ❑ programski paketi za modeliranje (npr. IFPS, @Risk),
- ❑ programski paketi za podporo vodenju projektov (npr. Microsoft Project),
- ❑ ekspertni sistemi (vsebujejo bazo znanja),
- ❑ sprotna analitična obdelava podatkov – OLAP (Oracle, DB2) in
- ❑ podatkovno rudarjenje (data mining).

Vir: Damij, Grad, Jaklič, 1995

V nalogi bom obširneje predstavila ekspertne sisteme in sicer uporabo baze znanja kot baze v pomoč uporabnikom softverskega produkta.

Ekspertni sistemi so rezultat raziskav na področju umetne inteligence, kjer je bil poudarek na znanju, ki je temeljil na strokovnosti posameznika (Kerry, 1990, str. 10). Za ekspertne sisteme bi tudi lahko rekli, da predstavljajo utelešenje ekspertnega znanja z določenega področja, ki je izvedeno na tak način, da lahko sistem inteligentno svetuje uporabniku, ali da je na uporabnikovo zahtevo sposoben utemeljiti svoj način "razmišljanja" (Čibej, 1998, str. 14).

Pogoj za razvoj ekspertnih sistemov je bil razvoj metod umetne inteligence in primerna stopnja razvoja računalništva. Ker potrebujejo posebne programske jezike, ki omogočajo gradnjo struktur, ki jih ekspertni sistemi za svoje delovanje rabijo, je bilo potrebno najprej napraviti osnovo za take sisteme. Ekspertni sistemi se razlikujejo od predhodnih stopenj predvsem v tem, da za njihovo delovanje niso potrebni samo algoritmi, temveč te zamenjamo z bazo relacij in operacij (Jarc, 1985).

Glavna predpostavka ekspertnega sistema je, da ekspert svoje znanje in izkušnje uporabi pri oblikovanju pravil shranjenih v bazi, da lahko potem manj izkušeni ljudje s pomočjo mehanizmov sklepanja in pravil v bazi znanja pridejo do ustreznih zaključkov (Fabbri, 1992, str. 466).

2.3.1 Funkcije ekspertnih sistemov

Ekspertni sistem je programska rešitev, ki se vede podobno kot izvedenec za določeno ožje področje uporabe. Ekspertni sistemi morajo biti zmožni reševati probleme, ki zahtevajo posebno področno znanje. Toda prav vsakega sistema, ki vsebuje znanje, ne moremo šteti med ekspertne sisteme. Ekspertni sistem mora znati tudi pojasniti svoje ravnanje in svoje odločitve tako kot živi izvedenci. Taka zmožnost pojasnjevanja je posebno pomembna v negotovih domenah zato, da povečamo uporabnikovo zaupanje v odločitve, ki jo predlaga sistem, ali pa da uporabnik lažje ugotovi napako v sklepanju, po katerem je prišel do svojega vprašljivega zaključka (Bratko, 1989).

Ekspertni sistem mora imeti vgrajen prijazen vmesnik za dialog z uporabnikom, s katerim bo sklepanje, ki se opravi v notranjosti sistema, postalo uporabniku razumljivo.

Dodatna lastnost, ki jo od ekspertnega sistema pogosto pričakujemo je, da lahko smiselno uporablja tudi nezanesljivo in nepopolno informacijo. Podatki o problemu, ki ga rešujemo, so lahko nepopolni ali negotovi, prav tako pa so lahko nezanesljive tudi relacije v domeni uporabe oziroma veljajo samo približno (Bratko, 1989). Uporaba negotovih podatkov in nezanesljivih relacij zahteva verjetnostno sklepanje.

Pri gradnji ekspertnega sistema moramo v splošnem izpeljati naslednje funkcije (Bratko, 1989):

- ❑ **reševanje problemov**, tako da sistem uporablja znanje, ki je značilno za področje uporabe; pogosto je pri tem treba obvladati negotovosti,
- ❑ **dialog z uporabnikom**, ki vključuje pojasnjevanje namer in odločitev sistema tako med procesom reševanja problema kot tudi po njem.

2.3.2 Znanje kot del ekspertnega sistema

Znanje so informacije, ki jih sistem ali program potrebuje za uspešno in učinkovito reševanje problemov. Med znanje štejemo dejstva, ki izražajo neke veljavne ali delno veljavne trditve in pravila v neki domeni.

Znanje glede na različne kriterije delimo na trdo in mehko, znanje o domeni in splošno znanje, znanje o "stvareh" in znanje o znanju (metaznanje) ter na formalno teoretično in neformalno empirično znanje (hevristiko). Te delitve se do določene mere medsebojno prekrivajo.

Trdo in mehko znanje

Ena izmed možnih kategorizacij znanja je njegova delitev na trdo in mehko znanje. Dejstva in pravila niso vedno samo pravilna ali nepravilna, kar velja za trdo ali kategorično znanje. Pogosto so nezanesljiva (nepopolna ali nenatančna) in le delno veljavna. Takšno znanje uvrščamo med mehko ali nezanesljivo znanje. Po istem principu se na trda in mehka delijo tudi problemska področja ali domene. Zelo na splošno lahko rečemo, da so eksaktne znanosti bliže trdim, neeksaktne pa mehkim domenam (Bratko, 1989).

Znanje o domeni in splošno znanje

Znanje o domeni je zbirka specifičnih dejstev in pravil, povezanih z nalogo, ki jo sistem znanja rešuje. Splošno znanje je po drugi strani kontekstno neodvisno znanje, kot so na primer splošna pravila in postopki reševanja problemov (Waterman, 1986). Med splošno znanje štejemo tudi znanje, ki ga ljudje pogosto opisujejo pod pojmom "zdrava pamet". Ločnica med znanjem o domeni in splošnim znanjem je, bolj kot od katerekoli svojstvene lastnosti znanja samega, navadno odvisna od opredelitve nalog, ki jih sistem znanja izvaja (Shapiro, 1992).

Metaznanje

Metaznanje lahko v splošnem opredelimo kot znanje o znanju. Je znanje o širini in organizaciji znanja o sistemu, o tem, kako uporabljati znanje oziroma sklepati, kdaj in kako izvajati določene akcije, itd. Metaznanje usmerja delovanje in odločanje v sistemih znanja (Shapiro, 1992). Preprost primer metapravila (pravila o pravilu) je lahko na primer: "Če je odgovor na splošno vprašanje negativen, ne zastavljaj bolj specifičnih vprašanj." Ali: "Če nimaš na razpolago dejstev, ki bi potrdila pravilo A, uporabi pravilo B."

Hevristika

Za velik del problemov reševanih s sistemi znanja, eksaktne algoritemske rešitve niso znane ali pa so le-te računsko preobsežne. Zato tovrstne, najpogostejše zelo kompleksne, nezanesljivo ali nejasno definirane probleme, rešujemo s hevrističnimi metodami reševanja problemov oziroma s hevrističnim iskanjem. To, v nasprotju z algoritmi, ne zagotavlja vedno rešitve, rešitve tudi niso povsem eksaktne, temveč približne. Približne rešitve se lahko optimalnim zelo približajo, zato so sprejemljive (Shapiro, 1992).

Glavna prednost hevrističnega iskanja pred algoritmi je omejen iskalni prostor. Iskalni prostor je implicitna množica vseh možnih poti, po katerih lahko iskanje rešitve problema poteka. Hevristično iskanje je od začetnega do ciljnega stanja kontrolirano

ali vodeno z eno ali več iskalnimi strategijami ali s hevrističnim znanjem – “bližnjicami”, izpeljanimi iz poznavanja narave in strukture problema (Fenly, 1988).

Hevristično znanje je neformalno opredeljeno empirično znanje, katerega veljavnost potrjujejo izkušnje (Ford, 1990). Je zelo pomemben del znanja strokovnjaka oziroma eksperta. To znanje omogoča, da pridemo do neke zadovoljive rešitve problema, ne da bi preiskali vse možne poti. Hevristične metode so zato posebej primerne za reševanje problemov, pri katerih že manjši obseg hevrističnega znanja pripomore k relativno hitri in skoraj optimalni rešitvi problema.

2.4 Skupne značilnosti nalog reševanih s sistemi podprtimi z znanjem

Skupne značilnosti nalog reševanih s sistemi podprtimi z znanjem lahko strnemo v nekaj točkah:

1. Sistemi podprti z znanjem so primerni za reševanje nalog, katerih narava je hevretilna. Reševanje nalog s pomočjo hevrističnega znanja pa ne zagotavlja vedno eksaktnih rešitev, zato morajo biti tolerirane približne rešitve.
2. Ena izmed lastnosti, ki sisteme podprte z znanjem ločuje od konvencionalnih programov, je simbolično sklepanje. Manipulacija s simboli mora omogočati rešitev naloge.
3. Rešitev naloge mora omogočati kognitivno in ne fizično znanje. Naloga za svojo rešitev tudi ne sme zahtevati širokega obsega splošnega znanja, kreativnosti ali prilagodljivosti.
4. Naloga mora biti dovolj ozka, da jo je možno voditi in nadzirati ter dovolj široka, da ima njena rešitev neko praktično vrednost.
5. Prav tako ne sme biti naloga prezahtevna, saj bi njena kompleksnost pripomogla k zapletenemu in dolgotrajnemu reševanju. Takšne naloge bi bilo dobro razčleniti na več relativno samostojnih in natančno definiranih podnalog ter jih obravnavati kot potencialne kandidate za gradnjo sistema podprtega z znanjem.
6. Naloga mora biti jasno in natančno definirana. Znanje, na katerem temelji, mora biti primerno razčlenjeno.

Problemske domene primerne za aplikacijo sistemov podprtih z znanjem so torej ozke in specifične, omejene, homeogene, formalno potrjene, pogosto rutinske in intenzivno podprte z ustreznim teoretičnim in hevrističnim znanjem (Smith, 1987).

2.5 Struktura in delovanje sistema podprtega z znanjem

Glavna strukturna značilnost sistemov podprtih z znanjem je ločitev znanja o domeni od procedur, ki z znanjem upravljajo. To pomeni tudi ločitev od preostalega znanja v sistemu, kot na primer znanja o interakciji z uporabnikom (Waterman, 1986). Sistemi

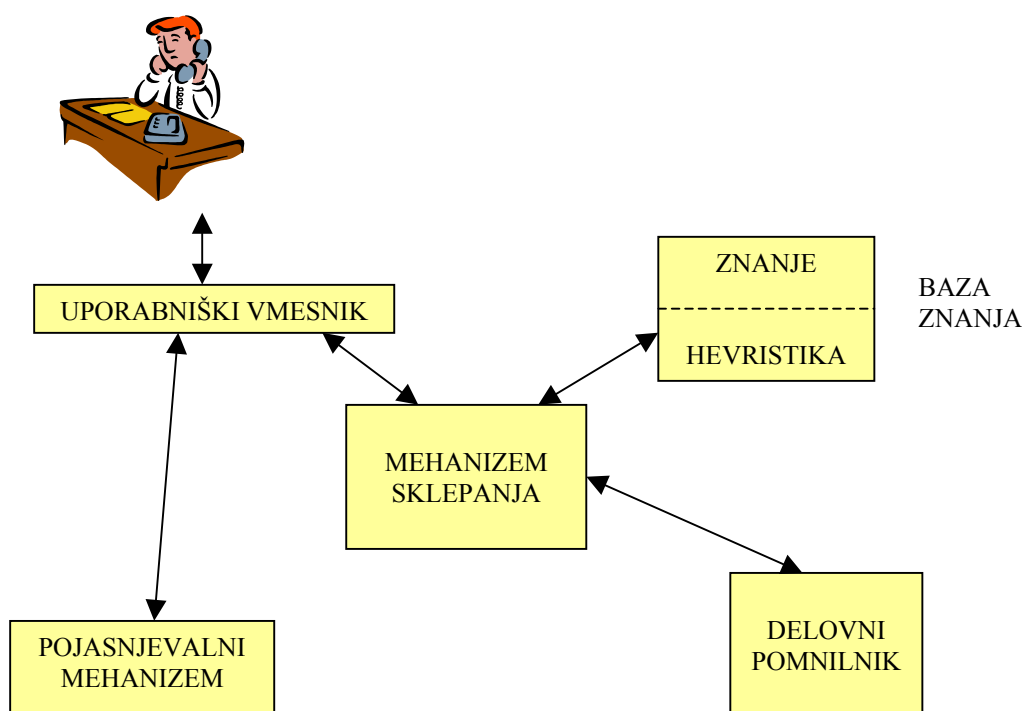
se po svoji zgradbi precej razlikujejo. Arhitekturo sistema namreč v veliki meri pogojuje vrsta naloge, ki jo sistem rešuje.

Ekspertni sistem naj bi zajemal naslednje komponente (Shapiro, 1992):

- ❑ **bazo znanja** (knowledge base),
- ❑ **mehanizem sklepanja** (inference engine/reasoning mechanism),
- ❑ **uporabniški vmesnik** (user interface),
- ❑ **pojasnjevalni mehanizem** (explanation mechanism) in
- ❑ **delovni pomnilnik** (working memory).

Ločitev baze znanja od mehanizma sklepanja bistveno pripomore k modularnosti obeh najpomembnejših komponent sistema. Slika 2 prikazuje zgradbo ekspertnega sistema.

Slika 2: Zgradba ekspertnega sistema



Vir: Avison in Fitzgerald, 1995, str. 95

Mehanizem sklepanja kontrolira proces klicanja pravil, ki se nanašajo na rešitev problema. Uporablja se za sklepanje o informacijah v bazi znanja in na delovnem mestu ter zna aktivno uporabljati to znanje. Mehanizem sklepanja zagotovi navodila in kontrolira procesiranje znanja in pravil, da bi s pomočjo sklepanja prišli do rešitve problema (Avison in Fitzgerald, 1995).

Razlikujemo tri sestavine mehanizma sklepanja. Prva je razlagalec pravil (rule interpreter), ki učinkovito izvaja pravila. Druga sestavina je urnik (scheduler), ki kontrolira čas in vrstni red izvajanja. Tretja pa je preverjanje doslednosti (consistency checker) izvajanja procesov in rešitev.

Uporabniški vmesnik skrbi za udobno sporazumevanje med uporabnikom in sistemom, pri čemer uporabniku omogoča vpogled v proces reševanja problema, ki ga vodi mehanizem sklepanja. Interakcija s sistemom mora biti čimbolj prilagojena laiku in omogočati pojasnjevanje sklepanja oziroma rešitve problema. Najprijaznejša oblika komunikacije sistema z uporabnikom je komunikacija v naravnem jeziku. Za interakcijo v naravnem jeziku sistem potrebuje sposobnost procesiranja naravnega jezika, da lahko generira vprašanja in odgovore ter interpretira odzive uporabnika. Mehanizmi sklepanja omogočajo aktivno uporabo znanja za reševanje problemov. Mehanizme sklepanja in uporabniški vmesnik lahko gledamo tudi kot en sam modul, ki ga navadno imenujemo lupina ekspertnega sistema.

Najsplošnejši tip **pojasnjevalnega mehanizma** obravnava in izvaja retrospektivno sklepanje. Sistem na uporabnikovo zahtevo z metasklepanjem pojasnjuje delovanje s prikazom verige sklepanja. Ta predstavlja pot (izvedene akcije in njihovo zaporedje), po kateri je sistem prišel do rešitve problema. Na ta način sistem ne pojasnjuje samo, kako je prišel do rešitve, ampak tudi zakaj potrebuje določeno informacijo – zakaj je zastavil določeno vprašanje (Bratko, 1989).

Poleg retrospektivnega sklepanja lahko pojasnjevalni mehanizem obravnava tudi hipotetično sklepanje, ki je v nasprotju z dejstvi (counterfactual reasoning) (Waterman, 1986). Pri prvem sistem pojasnjuje, kaj se zgodi, če določeno pravilo spremenimo, pri slednjem pa, zakaj do pričakovane rešitve ni prišlo.

Delovni pomnilnik je v različnih arhitekturah sistema poznan tudi pod imenom globalna podatkovna zbirka (global database) ali delovno mesto (blackboard). V njem je shranjeno trenutno stanje reševanja danega problema: dejstva, ki opisujejo problem, delne ali vmesne rešitve, ki jih lahko sistem kasneje, če jih potrebuje, ponovno uporabi, izpeljana dejstva ali na novo vpeljane vrednosti in končno tudi končna rešitev problema.

Delovni pomnilnik je dinamičen, saj se njegova vsebina stalno spreminja. Vse začasno shranjene informacije v delovnem pomnilniku so sistemu kadarkoli na razpolago za nadaljnjo uporabo v procesu odločanja (Gibb, 1986).

Vsak ekspertni sistem vsebuje tudi **bazo znanja**, sestavljeno iz baze podatkov z dejstvi o sistemu in pravilih sklepanja. V bazi znanja je eksplicitno shranjeno znanje o domeni, ki ga za reševanje problemov uporablja sistem. Vključeno je lahko znanje o vsebinskih konceptih domene in njihovih relacijah, o dejstvih in o zakonitostih ali pravilih problemske domene. Znanje o domeni je lahko v bazi znanja predstavljeno na različne načine. Glavne podatkovne strukture ali formalizmi za shranjevanje znanja so produkcijska pravila, okviri in semantične mreže. Kako in s katerimi formalizmi je znanje o domeni predstavljeno, je odvisno od narave in strukture znanja in problemov, ki jih sistem rešuje. Določen formalizem je namreč primeren za predstavitev določene vrste znanja in za reševanje oziroma sklepanje o določenih tipih problemov (Shapiro, 1992).

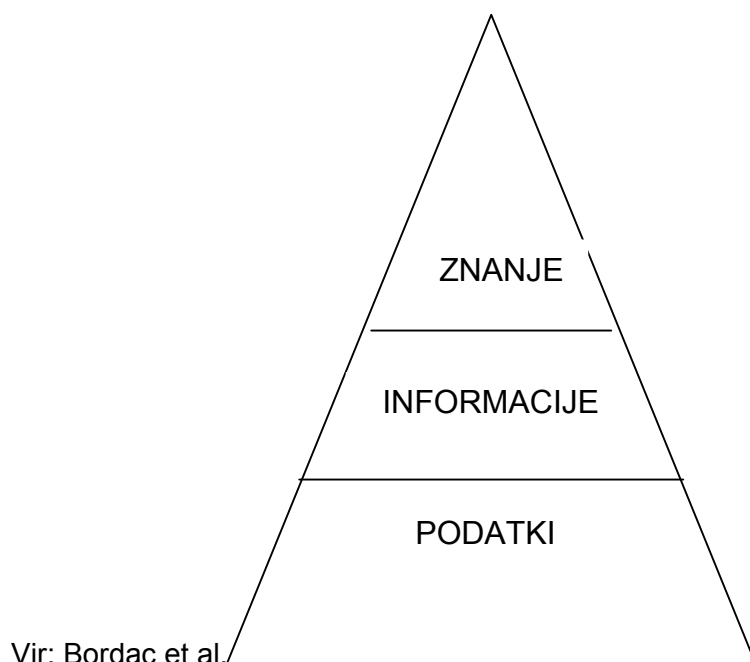
Preden pa se podrobneje lotimo opisa baze znanja, spregovorimo še nekaj besed o znanju in upravljanju z njim.

3 Upravljanje z znanjem

3.1 O čem govorimo, kadar govorimo o znanju in upravljanju z njim?

Podjetja in organizacije nasploh potrebujejo vedno več znanja za uspešno poslovanje. Vse pomembnejši vir znanja so obstoječe baze podatkov. Žal pa so surovi podatki le redko neposredno uporabni in koristni. Resnična vrednost podatkov je pogojena z možnostjo, da iz njih izluščimo informacije, ki prispevajo k razumevanju problemskega področja oziroma so uporabne za podporo odločanju. Povezavo med znanjem, informacijami in podatki lahko prikažemo v obliki piramide, kot prikazuje slika 3.

Slika 3: Povezava med znanjem, informacijami in podatki



Podatki kažejo dejstva, opazovanja in postanejo informacija v kontekstu z drugimi podatki.

Informacija sama po sebi še ni znanje. To postane, ko je analizirana, strukturirana in povezana z drugimi, še nepoznanimi informacijami.

Največkrat se srečujemo z vprašanjem razlikovanja med znanjem in informacijo. Tako za informacije kot tudi za znanje je značilno, da se v nasprotju z drugimi viri ne izčrpata, če ju posredujemo drugim. Ravno nasprotno, mnogokrat se v interakciji z drugimi oplemenitita in nadgradita. Med njima pa vendarle obstajajo določene razlike, ki so prikazane v tabeli 1.

Tabela 1: Razlike med znanjem in informacijo

INFORMACIJA	ZNANJE
otipljiva	neotipljivo – je človeški proces
fizični objekt	mentalni objekt
neodvisna od konteksta	kontekst določa pomen
lahko prenosljivo	prenos je mogoč le z učenjem
lahko se ponovno izvede	ni ga mogoče identično reproducirati

Vir: Amidon in Skyrme, 1997, str. 30.

Znanje in informacija sta torej tesno povezana, nista pa sinonima. Znanje je namreč višja oblika, informacije pa so prvi pogoj za nastanek in širjenje znanja.

Znanje torej niso ne podatki ne informacije. Kljub temu, da je znanje povezano z obema pojmom, med njima obstajajo očitne razlike. Uspeh in neuspeh podjetja je velikokrat odvisen od tega, kaj potrebujemo, kaj imamo na voljo in kaj lahko naredimo s podatki, informacijami ali znanjem, ki jih imamo na voljo. Razumevanje medsebojnih razlik med pojmom in kako prehajati od enega k drugemu je odločilnega pomena za uspešno upravljanje z znanjem (Davenport in Prusak, 1998, str. 1).

Uspešno upravljanje z znanjem v podjetju pomeni ključ, ki vodi do razširitve mnogih poslovnih izzivov in je vir krepitev konkurenčnih prednosti podjetja. Pri podrobnejšem in poglobljenem razumevanju upravljanja z znanjem je le-tega potrebno osvetliti in preučevati iz več perspektiv hkrati. Pojem znanja je namreč zelo kompleksen. Znanje se poraja v glavah posameznikov in je lahko zelo osebno, zaradi česar je potrebno za njegovo upravljanje uporabljati takšne metode, s pomočjo katerih lahko v organizaciji dosežemo primerno okolje in sistematično lotevanje kreiranja, ohranjanja, varovanja in širjenja znanja med zaposlenimi, ki je pomemben vir za poslovni uspeh podjetja.

Za upravljanje z znanjem ni neke enotne definicije, saj je to zelo širok pojem in se nanaša v podjetju na kulturo, organiziranost podjetja, na politiko upravljanja s človeškimi viri, spodbujanje motivacije zaposlenih, izobraževalne procese, nagrajevanje in podobno. V bistvu gre za intelektualni kapital podjetja in je zato odvisno od tega, s katerega vidika ga opredeljujemo.

Najpomembnejše pojme upravljanja z znanjem torej predstavljajo:

- ljudje,
- tehnologija in
- kultura.

Uspešnost podjetja je v veliki meri odvisna od vnovčitve znanj posameznikov in uspešnega upravljanja z njihovim znanjem. Na osnovi tega bi lahko rekli, da upravljanje z znanjem pojmuje kot načrtno usmerjeno aktivnost v podjetju, ki zajema prepoznavanje ključnega znanja v podjetju, oblikovanje novega potrebnega znanja ter aktivno prenašanje le-tega med zaposlenimi v podjetju.

Procesi prenašanja znanja s posameznikov na celotno organizacijo morajo biti aktivno vpleteni v delovanje organizacije, celo postati ključni del kulture in podprti z ustrežno informacijsko tehnologijo. Najpomembneje pa je, da so procesi povezani v ustrežno politiko upravljanja s človeškimi viri in s tem pravilno definirani in uvedeni kadrovskimi procesi v podjetju (medsebojna komunikacija, razvoj, izobraževanje, motivacija, nagrajevanje), saj so zaposleni pomemben del intelektualnega kapitala, na katerega se upravljanje z znanjem navezuje.

K upravljanju z znanjem štejemo tudi oblikovanje baz znanja v podjetju, kar omogoča lažjo razvrstitev znanja – natančno lahko določimo njegovo vsebino in nosilca, kar omogoča večji pregled nad znanjem v podjetju in ga je mogoče poiskati, kadar se po njemu pojavi potreba.

Ko govorimo o znanju, imamo v mislih:

- ☐ podatke/informacije v strukturiranih ali nestrukturiranih bazah podatkov,
- ☐ digitalizirane dokumente v obliki besedil ali slik,
- ☐ kupljene informacije,
- ☐ znanje zaposlenih,
- ☐ vedenje o strankah in poslovnih partnerjih,
- ☐ vedenje in znanje o obnašanju strank, trga in
- ☐ podatke o delovanju in vplivu konkurence na trg.

3.2 Opredelitev znanja

Veliko ljudi bi se strinjalo s tem, da je znanje obsežnejši pojem od podatkov in informacij. Njegov pomen je globlji, obsežnejši in bogatejši v primerjavi z njima. Ljudje govorijo o učenih vsevedih in pri tem mislijo na ljudi, ki so informirani, razgledani, zanesljivi, natančni, izobraženi in inteligentni. Nikogar ne bomo slišali govoriti o vsevednem priročniku, bazi podatkov ali dokumentu, pa čeprav so produkt ljudi oziroma skupin z določenim znanjem. Znanje izhaja iz delovne vneme ljudi (Davenport in Prusak, 1998, str 5).

Po drugi razlagi je znanje mešanica uokvirjenih izkušenj, vrednot, pisnih informacij in ekspertnih pogledov, ki zagotavljajo uspešno absorbiranje novih izkušenj in informacij. Izhaja in uporablja se v zavesti ljudi. V podjetjih pa je večinoma zapisano

ne samo v notranjih dokumentih, ampak tudi v procesih, navadah, delovanju in normah.

V organizaciji je ogromno znanja – v njenem spominu, v glavah ljudi, pogosto neevidentirano tako imenovano tiho znanje, v dokumentih, v proizvodih, storitvah, v organizacijskih strukturah in procesih. Ta znanja so vsebinsko zelo različna. Različno so razpršena, organizirana in imajo različne vrednosti. Nekatera so aktualna, druga že “amortizirana”, nekatera v obtoku, druga skrbno “skladiščena”, drugih je preveč glede na uporabnost, drugih premalo ali nič (Peršak, 2001).

Po vsem tem lahko rečemo, da znanje ni tako enostavno kot se nam zdi na prvi pogled, saj je mešanica več elementov. Znanje izhaja iz informacij in informacije izhajajo iz podatkov. Ljudje so tisti, ki pretvorijo informacije v znanje. Transformacija informacije v znanje se zgodi s pomočjo naslednjih štirih procesov, katerih imena se v angleškem jeziku začnejo s črko c:

- ❑ primerjava (*comparison*): Kako se določena informacija lahko primerja s podobnimi situacijami, ki smo jih doživeli?
- ❑ posledice (*consequences*): Kakšen obseg ima informacija na odločitve in dejavnosti?
- ❑ povezave (*connections*): Kako se ta delček znanja navezuje na druge?
- ❑ komunikacija (*conversation*): Kaj drugi ljudje mislijo o tej informaciji?

Glavne sestavine znanja so izkušnje, resnica, razsodnost in pravila sklepanja.

3.2.1 Izkušnje

Skozi izkušnje, obiskovanje tečajev, branje strokovnih knjig, mentorstvo, se počasi izoblikuje in razvije znanje. Izkušnje se nanašajo na stvari, ki so se nam zgodile v preteklosti in iz katerih smo se nekaj naučili. Izkušenim ljudem pravimo eksperti oziroma strokovnjaki. Vsekakor pa do določenega znanja niso prišli kar čez noč, ampak s časom. S pomočjo izkušenj iz preteklosti lahko v podobnih situacijah v sedanjosti ali prihodnosti hitreje reagiramo.

3.2.2 Resnica

Resnica mogoče ni najboljši prevod besede *truth* iz angleščine. V kontekstu bi bil realnost boljši prevod. Pri tem mislimo na vedenje kaj res deluje in kaj ne deluje. Delitev podrobnosti iz preživetih izkušenj je v resničnem svetu lahko veliko bolj vredno in pomembno kot zanašanje na teorije o istih stvareh.

3.2.3 Razsodnost

Znanje, za razliko od informacij in podatkov, vsebuje mnenje. Ne samo, da lahko precenimo nove situacije in informacije glede na znanje, ampak s pomočjo odziva na nove situacije na novo oblikujemo mnenje oziroma znanje o tej stvari (Davenport in Prusak, 1998, str. 10). So pa nekateri ljudje, ki svojega mnenja ne spremenijo kar tako in bi novim vprašanjem dali vedno isti, star odgovor. V takih primerih bi lahko rekli, da znanje preide v mnenje ali dogmo.

3.2.4 Pravila sklepanja in intuicija

Ljudje se veliko naučimo s pomočjo poskusov in skozi napake, ki jih delamo. V situacijah, ki so podobne tistim, ki smo jih doživeli, bomo hitreje rešili nove probleme. Z znanjem bomo uporabili nam že znane vzorce in na podlagi tega reagirali v določeni situaciji. To je podobno kot pri vožnji avtomobila. Izkušnje nam v veliko situacijah pomagajo. Čisto drugače bo na cesti reagiral voznik, ki ima za seboj 10 let aktivne vožnje, kot pa nekdo, ki je šele naredil vozniški izpit.

3.2.5 Vrednote in prepričanja

Prepričanja in vrednote ljudi imajo velik vpliv na znanje v organizaciji. Navsezadnje so organizacije sestavljene iz ljudi, katerih mnenja in prepričanja vplivajo na njihova dejanja in misli. Zgodovinski razvoj organizacije odraža delovanje ljudi in tudi njihove vrednote in prepričanja.

Vrednote in prepričanja so pomemben del znanja, saj odražajo, kaj posameznik absorbira in sklepa iz posameznih situacij. Ljudje različnih vrednot lahko v isti situaciji vidijo različne stvari.

Nonaka in Takeuchi, avtorja dela *The knowledge-creating company*, pravita, da je znanje, za razliko od informacije, stvar prepričanj in predanosti. Moč znanja organizirati, učiti se in sklepati izhaja iz vrednot in prepričanj v večji meri, kot pa iz informacij in logike (Davenport in Prusak, 1998, str.12).

3.3 Različni pogledi na znanje

Razlikujemo med dvema vrstama znanja, in sicer med:

- **tihim** (tacitnim) **znanjem**, ki temelji na osebnih izkušnjah in je težko prenosljivo in
- **izraženim** (eksplicitnim) **znanjem**, ki je lažje prenosljivo in dokumentirano.

EksPLICITNO znanje lahko izrazimo v besedah in številkah, lahko ga posredujemo, se o njem pogovarjamo, saj ga lahko oblikujemo v znanstvene formule, razpredelnice, kodificirane procedure ali univerzalna načela.

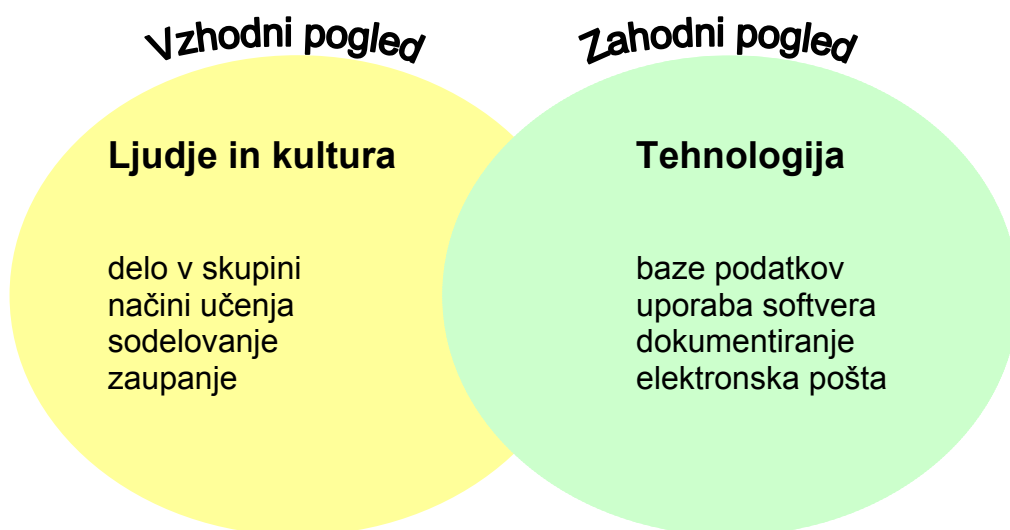
EksPLICITNO znanje lahko opišemo kot vrh ledene gore, saj poleg njega v podjetju obstaja še TIHO znanje, ki ga je Nonaka opredelil kot nekaj težko opredeljivega in težko vidnega. Pravi, da je tiho znanje zelo osebno in ga je skoraj nemogoče oblikovati. Za tovrstno znanje je namreč značilna visoka stopnja intuitivnosti, osebnih uvidov in prepričanj (Nonaka in Takeuchi, 1995, str. 8).

Obe vrsti znanja sta pomembni pri upravljanju z znanjem. Eden izmed ključnih izzivov za vsako podjetje je, kako povezati obe vrsti znanja in pri njuni kombinaciji izkoristiti sinergične učinke.

V svetu glede na uporabo tihega in eksplicitnega znanja obstajajo razlike med vzhodnimi in zahodnimi pogledi na upravljanje z znanjem. Razlike obstajajo predvsem zaradi različnih kultur ljudi, ki se odražajo na načinu dela (Bordac, 2001).

Bordac, Friedman, Glaser in Harding so razlike med Vzhodom in Zahodom predstavili z naslednjo sliko (glej sliko 4):

Slika 4: Vzhodni in zahodni pogled na znanje



Vir: Bordac et al., 2002

Na Vzhodu dajo ljudje večji poudarek medsebojnim odnosom in kulturi. Pomembno je medsebojno zaupanje in sodelovanje. Vpliv na povečanje znanja ima tudi način učenja ter delo v skupinah. Ikujiro Nonaka, Hirotaka Takeuchi, Georg von Krogh in Kazuo Ichijo verjamejo, da so isti faktorji, ki opisujejo učečo organizacijo (Learning Organization) – skrb, zaupanje in pogovarjanje – poglobitveni za pridobivanje tihega znanja iz posameznikov, in da je tiho znanje tisto, ki pelje k inovativnosti podjetja.

Na Zahodu dajejo zaposleni večji poudarek tehnologiji, zlasti bazam podatkov, uporabi različnega softvera. Znanje je največkrat zapisano v dokumentih, a se

prenaša tudi preko elektronske pošte (Bordac, 2001). Zlasti elektronska pošta je danes eden izmed glavnih sredstev za prenos znanja.

V različnih raziskavah je bilo ugotovljeno, da je najboljši način vpeljave upravljanja z znanjem sledenje modelu učeče organizacije pri razvoju zaupanja in skrbi znotraj organizacije in hkrati zagotavljanje različnih programskih orodij zaposlenim za dokumentiranje in prenos znanja (Bordac, 2001). Uspešna bodo tista podjetja, ki se ne bodo nagibala ne k vzhodnem ne k zahodnemu pogledu na upravljanje z znanjem, temveč bodo uporabljala mešanico obeh pogledov na znanje.

3.4 Učenje in vrste učenja

Učenje je proces pridobivanja novih, nadgrajevanja že obstoječih znanj, pridobivanja novih veščin in sposobnosti. To je dinamičen proces, s katerim se soočamo celo življenje. Prav tako kot ljudje se z nenehnim učenjem soočajo tudi podjetja.

Učenje je zelo pomembno pri razvoju sposobnosti posameznika. V organizacijah poteka nenehen proces učenja, ki ga podjetja ne vidijo več samo kot strošek, temveč kot investicijo za prihodnost. Učinkovite naložbe v proces učenja povečajo vrednost posameznikov in s tem organizacije.

V organizaciji ločimo tri vrste učenja: individualno, teamsko in organizacijsko učenje (Pirc, 2000, str. 16):

Individualno učenje

To je proces v katerem posamezniki na različne načine pridobivajo nova znanja, stališča, vrednote in spretnosti. Individualno učenje bi lahko opredelili kot zavestno spreminjanje posameznikovega vedenja v smeri lažjega doseganja zastavljenih ciljev.

Najpogosteje se v organizaciji srečamo z dvema načinoma individualnega učenja. Učimo se lahko od sodelavcev ali na podlagi lastnih izkušenj.

Učenje od sodelavcev v podjetju lahko poteka po formalnih organizacijskih metodah. Sem spadajo razna usposabljanja, oblikovanja seminarjev, razni treningi na delovnem mestu, prebiranje strokovnih člankov, učbenikov, vodičev. Ta proces se začne že s samim prihodom novo zaposlenega v podjetje. Novi sodelavci ponavadi dobijo mentorja, ki jih uvaja in uči postopkov delovanja. K učenju po neformalnih metodah pa štejemo opazovanje sodelavcev pri delu, strokovne razgovore in razreševanje strokovnih dilem v komunikaciji s sodelavci.

Teamsko učenje

Pri tej vrsti učenja gre za pridobivanje znanja, spretnosti in vrednot v okviru celega tima in ne le posameznika. Bistveno je, da se člani tima učijo drug od drugega, tako skupaj dosegajo boljše rezultate in se hitreje razvijajo kot bi se v primeru individualnega učenja.

Pri implementiranju upravljanja z znanjem v podjetja je oblikovanje timov in njihovo delovanje ena izmed osrednjih točk. Za teamsko delo, predvsem projektno delo, je namreč značilno, da pride pretok znanja še posebno do izraza. O najučinkovitejšem

oblikovanju in vodenju teamov sicer obstaja mnogo modelov in predlogov (Planko, 2001, str. 52).

Organizacijsko učenje

Organizacijsko učenje se nanaša na veščine in procese učenja v organizaciji kot celoti in predstavlja skupinski proces učenja, ki se odvija skozi interakcijo posameznikov znotraj organizacije.

Organizacijsko učenje predstavlja spreminjanje organizacijskega vedenja in je pogojeno od učenja vsakega posameznika znotraj organizacije (Pirc, 2000, str. 26).

Vse tri vrste učenja – individualno, teamsko in organizacijsko učenje - so med seboj tesno povezane in medsebojno odvisne. Organizacijsko učenje se odraža tudi skozi individualno in teamsko učenje in daje boljše rezultate, kot bi jih lahko pričakovali samo znotraj ene od navedenih dveh vrst učenja.

3.4.1 Učeča se organizacija

Pojem učeče se organizacije se je v zadnjih desetih letih močno razširil, prvi pa so pojem uporabili leta 1988 v Ameriki avtorji Hayes, Wheelwright in Clarc (Kavčič, 1994, str. 424). Pri teoriji o učeči se organizaciji se poudarja pomen nenehnega učenja in vključevanja vseh zaposlenih v procese usposabljanja in učenja. Pri tem je pomembna fleksibilnost organizacije in njena odprtost do ožjega in širšega okolja.

Za učeče se organizacije je Garvin navedel naslednje prepoznavne dejavnike (Kavčič, 1994, str. 425):

- ❑ sistematično reševanje problemov,
- ❑ eksperimentiranje z novimi pristopi,
- ❑ učenje iz lastnih izkušenj in
- ❑ učenje od drugih (benchmarking).

Nekatere skupne, stične točke so naslednje:

- ❑ pridobivanje novega znanja je konstanten in dolgotrajen proces in ne posamezna akcija,
- ❑ stalno pridobivanje novega znanja in spreminjanje aktivnosti organizacije na tej podlagi je sestavina strategije organizacije,
- ❑ učenje v organizaciji ustvarja skupno videnje razvoja organizacije v prihodnje,
- ❑ učeča se organizacija je fleksibilna in odprta do okolja in
- ❑ nenehno se spodbuja učenje tako na individualni kot tudi skupinski ravni.

Pri učeči se organizaciji je pomemben celovit pristop, ki vključuje tudi spremembo vedenjskih vzorcev in ne ostaja le na spoznavni fazi, kar vodi do izboljšanih poslovnih uspehov podjetja. Gre za spreminjanje celotne kulture podjetja, kar pomeni, da gre za dolgotrajnejši proces, ki ga lahko v organizaciji izvedejo le s podporo vodstva podjetja in vključenostjo vseh zaposlenih (Planko, 2001, str. 33).

V praksi je opisane spremembe težko izvesti in uspešno uvajati. Razlogi temu so neobveščenost zaposlenih o spremembah, strah pred ekonomskimi in socialnimi posledicami uvedenih sprememb in strah pred neznanim. Zato je ključno, da se podjetja tovrstnega uvajanja lotijo celovito in načrtno.

V primerjavi z osnovnimi opredelitvami upravljanja z znanjem se le-ta v veliki meri ujemajo z osnovnimi smernicami teorije o učeči se organizaciji. Vendar upravljanje z znanjem gleda na znanje in njegovo upravljanje v organizaciji celoviteje, širše, saj je pri upravljanju z znanjem pomemben predvsem pretok posameznega znanja po organizaciji in tudi zunaj njenih meja.

Ob primerjavi izhodišč teorije učeče se organizacije in upravljanja z znanjem bi lahko ugotovili, da pomeni slednje določeno nadgradnjo teorije o učeči se organizaciji. Uvajanje upravljanja z znanjem v podjetje pomeni strateško odločitev, ki se usmerja in kontrolira prek najvišjega vodstva podjetja in tako pomeni ne le pridobivanje temveč tudi ustrezno ravnanje in upravljanje s pridobljenim znanjem. Vsaka organizacija, ki uvaja in se ravna po principih upravljanja z znanjem, je hkrati učeča se organizacija.

3.5 Pomen znanja v podjetju

V času, ko intelektualna lastnina postaja vse bolj pomembna, se podjetja vse bolj ukvarjajo z vprašanjem, kako to vrednoto upravljati in z njo gospodariti, da bi bil izdelek, ki ga ponujajo na trgu, kvaliteten in konkurenčen. Intelektualna lastnina vpliva tudi na ugled in konkurenčno vrednost podjetja na trgu.

Če se podjetje odloči, da bo začelo upravljati z znanjem, mora biti ta odločitev del strategije podjetja, saj pomeni, da se podjetje zaveda, kako zelo pomemben je lastni intelektualni kapital. Podjetje se mora odločiti, kako bo z upravljanjem intelektualne lastnine vplivalo na poslovne cilje. Upravljanje znanja zahteva takšno kulturo podjetja, ki promovira in podpira izmenjavo znanja, sodelovanje med zaposlenimi in med poslovnimi enotami. Jasna mora biti usmeritev k inovativnosti (Zakrajšek in Gašperič, 2002).

Glede na cilje, ki jih želimo doseči z uvedbo upravljanja z znanjem, izbiramo tudi primerno tehnologijo, da bo ustrezno podprla sodelovanje in izmenjavo znanja. Eden osnovnih ciljev sistema upravljanja z znanjem je omogočiti zaposlenim, da naloge opravljajo inovativno in kreativno, kar pomeni, da jim zagotovimo relevantne podatke in informacije ter jih ne obremenjujemo z aktivnostmi in s podatki, ki niso usmerjeni k izpolnitvi naloge.

Podjetja, ki se želijo konkurenčno postaviti ob bok močnim konkurentom, se zavedajo velikega pomena znanja in njegovega upravljanja. Ravno intelektualni kapital in njegovo ustrezno upravljanje sta tista, ki ločujeta dobra podjetja od slabih.

Vsaka organizacija ima svojo zakladnico znanja, ki jo skozi mrežo komunikacij dan za dnem odpira svojim zaposlenim in partnerjem ter jo vsak dan zavestno ali po naključju dopolnjuje z novimi spoznanji in informacijami. Podjetje je tako rekoč vsak dan na "borzi znanja", kjer se sooča s ponudbo, povpraševanjem in cenovnimi nihanjem (Peršak, 2001).

Za takšen nastop mora obvladati dve veščini:

- poznati in oblikovati ceno znanja,
- znati zaščititi tisto znanje, ki ji nudi tržno prednost.

Znanje ima absolutno vrednost in v splošnem pomeni strateško vlogo razvojnega faktorja pa tudi operativnega preživetja. Znanje pridobiva na ceni glede na povpraševanje na globalni borzi znanja pa tudi glede na aktualnost v specifičnih poslovnih okvirih posamezne organizacije.

Bolj ko bodo podjetja znala razumevati ceno znanja, bolj bo prihajala do izraza večina gospodarjenja z znanjem in seveda zaščita znanja. Česar ne moremo ceniti, tega ne bomo niti varovali niti z njim gospodarno ravnali, tako z vidika stroškov, kakor z vidika investiranja in pričakovanih donosov (Peršak, 2001)

Glavni principi upravljanja z znanjem (Davenport in Prusak, 1998, str. 24):

- ❑ znanje izvira in se nahaja v glavah ljudi,
- ❑ prenosu znanja je potrebno zaupati,
- ❑ tehnologija omogoča nove oblike znanj,
- ❑ prenos znanja je potrebno spodbuditi in nagraditi,
- ❑ potrebna je podpora managementa in virov,
- ❑ uvajanje znanja naj se najprej izvede s poskusnim programom,
- ❑ količinske in kvalitetne meritve naj ocenijo uvajanje znanja in
- ❑ znanje je kreativno in njegova nepričakovana smer razvoja mora biti vzpodbujena.

3.6 Pridobivanje znanja

Vsa uspešna podjetja ustvarjajo (knowledge generation) in uporabljajo znanje. Z delovanjem v okolju podjetja pridobivajo informacije, jih spremenijo v znanje in ukrepajo v povezavi z izkušnjami, vrednotami in notranjimi pravili. Čutijo in ukrepajo. Brez znanja bi podjetja ne mogla delovati in se tudi ne ustrezno organizirati. Znanje podjetja pridobivajo na različne načine, odvisno od interesov, potreb, stroškov, možnosti in racionalnosti pridobivanja posameznega znanja.

V tem poglavju bom opisala katerih dejavnosti se podjetja lotijo z namenom povečanja znanja v podjetju. Nekatera podjetja zaradi tega zaposlijo "pametne" ljudi in jim dajo proste roke pri delu. Davenport in Prusak pa sta v svojih raziskavah ugotovila, da niso najuspešnejša tista podjetja, ki se ukvarjajo samo z ustvarjanjem znanja, temveč tista, ki hkrati nudijo tudi dobre delovne pogoje, vključno z lokacijo in sestavo teamov (Davenport in Prusak, 1998, str. 52).

Podjetja lahko znanje pridobijo na različne načine:

- ❑ **Akvizicija.** Najbolj neposreden in učinkovit način pridobitve znanja je njegov nakup. Podjetje lahko pridobi znanje z nakupom drugega podjetja ali z najemom posameznikov, ki obvladajo določeno znanje, ki ga podjetje potrebuje. Podjetje, ki kupi drugo podjetje, v bistvu kupuje ljudi, torej znanje, ki ga imajo v sebi. Poleg znanja zaposlenih, kupuje tudi nekatero dokumentirano znanje, procese in spretnosti.

Vedno več podjetij se odloča za zunanjo strokovno pomoč, ki omogoča da se določeni projekti učinkoviteje uvedejo in prinašajo v podjetje nove izkušnje in neobremenjene, nepristranske vpogleda v situacijo v organizaciji.

- **Najem znanja.** Podjetja lahko določen del poslovanja zaupajo drugemu podjetju. Na primer podjetja lahko najamejo finančne organizacije za pomoč pri obračunavanju stroškov, lahko najamejo podjetje za raziskavo o zadovoljstvu kupcev. Na ta način se odrečejo določenemu nadzoru ter privarčujejo denar in organizacijske stroške.

S tem, ko podjetje najame projektanta, plača samo njegove storitve, njegovo znanje, ne pa tudi stroškov malice, prevoza, zdravstvenega zavarovanja in podobno.

- **Oblikovanje posebnih oddelkov.** Podjetja lahko ustanovijo razvojno-raziskovalne oddelke (R&D – Research & Development; primer je Andersen Consulting), inovacijske oddelke (Ernst & Young), posebne učne centre in knjižnice.
- **Fuzija.** Združevanje ljudi z različnimi spretnostmi, idejami in vrednotami lahko pripelje do kreativnih zaključkov. Za fuzijo je značilno, da so v projekte vključeni po sposobnostih različni ljudje, ki morajo priti do skupnega rezultata. Gre za projektno delo, oblikovanje večdimenzionalnih teamov in izkoriščanje sinergičnih učinkov konstruktivnih konfliktnih situacij.
- **Adaptacija.** Novi konkurenčni proizvodi, nove tehnologije, družbene in gospodarske spremembe vplivajo na pridobivanje znanja, kajti podjetja, ki se niso sposobna prilagoditi, ne morejo biti uspešna.

Zmožnost prilagajanja podjetja je odvisna od dveh stvari:

- ✓ obstoječih sredstev in zmogljivosti, ki jih lahko uporabijo na nov način in
- ✓ naklonjenosti spremembam.

- **Povezovanje v neformalne mreže znanja.** Znanje se generira tudi zunaj podjetja, na raznih srečanjih, ob druženju z ljudmi podobnih interesov, na poslovnih sestankih, ob sodelovanju s strankami, zunanjimi svetovalci in podobno. Gre za preseganje meja organizacije in sodelovanje s širšim okoljem podjetja (Davenport in Prusak, 1998, str. 66).

3.7 Kodifikacija in usklajevanje znanja

Pri uvajanju upravljanja z znanjem se srečujemo s procesi, ki so ključni in nepogrešljivi pri uvajanju in razvoju upravljanja z znanjem za posamezno podjetje. Med najpomembnejše procese uvajanja upravljanja z znanjem sodijo procesi identificiranja pomembnega znanja v podjetju in določitev virov znanja, kreiranje znanja in pretok znanja med zaposlenimi, oblikovanje baze najpomembnejšega znanja v podjetju in nenazadnje vzpostavitev takšnih sistemov in mehanizmov v podjetju, ki pridobljeno znanje v podjetju tudi ustrezno varujejo pred konkurenco.

Cilj kodifikacije je prenesti znanje v takšno obliko, da bo dosegljivo tistemu, ki ga potrebuje. Znanje oblikujemo v kodo, čeprav ne nujno računalniško ter ga razumljivo, nedvoumno oblikujemo. Primer kodifikacije predstavljajo zakoni v našem pravnem sistemu.

Pred procesom kodifikacije je potrebno prepoznati ključno znanje za podjetje, ga locirati, določiti njegove nosilce in oceniti njegov obseg.

Kodifikacija znanja pomeni organizirati in shraniti znanje v takšno obliko, da jo bodo zaposleni lahko razumeli, vrednotili in prenašali po organizaciji. Znanje mora biti shranjeno tako, da ga lahko po potrebi hitro najdemo in uporabimo.

Konkurenčne prednosti posameznikov, podjetij in držav so danes postale odvisne od sposobnosti trajnega vzdrževanja znanja in njegovega povečevanja v procesu učenja. Pri tem je ključna vloga tihega znanja v podjetju in sposobnost njegove pretvorbe s pomočjo sodelovanja, komuniciranja, distribucije in strokovnega izpopolnjevanja v eksplicitno obliko – kodificirano obliko (Tomassini, 2001).

Kot že omenjeno obstaja več različnih vrst znanja. Nekatera znanja je za razliko od drugih lažje kodificirati. Med ta znanja sodita strukturno in eksplicitno znanje. Primer znanja, ki ga težje kodificiramo pa je tiho oziroma tacitno znanje (Davenport in Prusak, 1998, str. 69).

Dr. Maja Bučar iz Centra za mednarodne odnose na Fakulteti za družbene vede je na mednarodnem simpoziju "Tiho znanje – velik kapital v podjetju" spregovorila o podpornih okoljih, ki omogočajo nadgradnjo tihega znanja. Pri tem niso ključna orodja (informacijska tehnologija) temveč je v osnovi najprej potrebno opredeliti željeno vsebino znanja, ki ga podjetje potrebuje, organizacijsko kulturo, strukture in procese. Delo s tihim znanjem vsebuje:

- ❑ faze zbiranja – knjižnice, koordinator znanja,
- ❑ faze povezovanja – pogostost in kakovost stikov nosilcev znanja in
- ❑ faze opredeljevanja znanja – zakaj delati s tihim znanjem.

Tiho znanje ima pomembno poslovno vrednost, za njegovo aktiviranje je pomembna ustrezna organizacijska kultura, ki podpira pripadnost skupnim ciljem, izmenjavo informacij, medsebojno zaupanje in sodelovanje.

Načela, ki naj se jih držijo podjetja, ki želijo svoje znanje kodificirati (Davenport in Prusak, 1998, str 69):

1. managerji se morajo odločiti katerim poslovnim ciljem bo kodificiranje znanja služilo,
2. managerji morajo biti sposobni identificirati primerne oblike znanja za doseganje ciljev,
3. managerji znanja morajo oceniti koristnost znanja in primernost kodifikacije,
4. zapisovalci zakonov (koderji) morajo oceniti primerno sredstvo za kodifikacijo in distribucijo.

3.7.1 Mapiranje in modeliranje znanja

Razvijanje "zemljevida" oziroma oris modela znanja vključuje razvrščanje pomembnega znanja v organizaciji in objavo neke vrste slike ali listine, ki kaže, kje se to znanje nahaja. Tako imenovani zemljevidi znanja kažejo tako na zaposlene kot tudi na dokumentacijo in baze podatkov. Omogočajo vizualizacijo znanja v podjetju, kar je bolj oprijemljivo, lažje predstavljivo in manj abstraktno. Takšen način omogoča, da imajo v podjetju nadzor nad tem, katero znanje je že v podjetju in katero bi še bilo potrebno pridobiti.

Podatki potrebni za izgradnjo zemljevida znanja so v organizaciji pogosto že obstoječi, vendar pa se nahajajo v razkrojeni in nenapisani obliki. Vsak zaposleni ima v svoji glavi delček zemljevida, saj ve, katera znanja ima in kam se po potrebi obrniti po pomoč. Za pridobitev zemljevida znanja je potrebno individualne "mini zemljevide" sestaviti v celoto. Podjetje mora najti način, kako od zaposlenih pridobiti čimveč informacij. En način, ki ga podjetja uporabljajo, so ankete.

Kodificiranje da neko stalnost znanju, ki bi sicer obstajala samo v glavah posameznikov. Znanje oblikuje in vtakne v takšne oblike, da se lahko znanje na različne načine deli, shranjuje, kombinira in manipulira.

3.8 Oblikovanje kulture znanja v podjetju

Organizacijska kultura pomeni identiteto posamezne organizacije; sestavljajo jo vrednote, filozofija, posebnosti podjetja in s tem tudi posamezni vzorci delovanja in razmišljanja zaposlenih v podjetju.

Odnos do znanja v podjetju je v določeni meri povezan tudi z obstoječo organizacijsko kulturo.

Upravljanje z znanjem je torej proces, ki se navezuje na spreminjanje organizacijske kulture, na njeno preoblikovanje oziroma dodajanje novega poudarka, v našem primeru znanja. Če želimo razumeti vplive in procese spreminjanja kulture v organizaciji, se moramo najprej seznaniti z njenimi osnovnimi opredelitvami in značilnostmi.

Organizacijska kultura je zelo heterogen pojem, ki združuje različne dimenzije znotraj organizacije, zato obstaja zelo veliko definicij, ki ta pojem opredeljujejo. Navedimo le nekatere (Možina et al, 1994, str. 175):

- ❑ Kultura je vzorec skupnih prepričanj in vrednot, ki oblikuje pomen institucije za njene člane in oblikuje pravila njihovega vedenja.
- ❑ Kultura je skupna filozofija članov organizacije.
- ❑ Kultura je tisto, kar resnično omogoča razumeti bistvo in dušo organizacije, kar je torej globlje od organigramov, pravil, strojev in zgradb.
- ❑ Kultura so tista prepričanja in vrednote, ki jih člani sprejemajo kot svoje, ki so jih internalizirali.

V strokovni literaturi je moč najpogosteje zaslediti definicijo, ki jo je predlagal Schein (Možina et al., 1994, str. 177):

“Organizacijska kultura je vzorec temeljnih domnev, ki jih je kaka skupina iznašla, odkrila ali razvila, ko se je učila spopadati se s problemi eksterne adaptacije in interne integracije; vzorec se je pokazal kot dovolj dober, da ga ocenjujejo kot validnega, zato nove člane učijo po tem vzorcu dojemati, misliti in čutiti te probleme.”

Kljub izčrpnosti, ki jo premore navedena opredelitev pojma organizacijske kulture, pa le-te ne gre jemati kot dokončno, saj organizacijska kultura posega v mnoge in različne vsebine. Kultura je od organizacije do organizacije zelo različna, zato lahko pojem organizacijske kulture krajše in plastično zajamemo z opredelitvijo, da je to način, kako stvari v posamezni organizaciji potekajo in kar organizacije med seboj razlikuje.

Velike korporacije, tiste z nekaj deset tisoč zaposlenimi in s podružnicami po vsem svetu, imajo močno notranjo hierarhijo, kjer se vsak trenutek ve, kdo je za kaj odgovoren. Presenetljivo tudi ni, da v teh podjetjih in podružnicah veljajo – čeprav zaposleni tega ne priznajo radi – stroga notranja pravila vedenja, oblačenja in komuniciranja. Slovenske podružnice tujih nadnacionalk v poslovni kulturi niso nič drugačne (Humar, 2002).

Preoblikovanje organizacijske kulture v tako imenovano kulturo znanja je ključni in hkrati najtežji izziv za posamezna podjetja, ki želijo uvajati upravljanje z znanjem v prakso. Težko je namreč spremeniti mišljenje posameznih zaposlenih v podjetju, ki se ravna po načelu "znanje je moč". Najtežje je namreč doseči ravno miselne premike in spremembo ustaljenih vzorcev obnašanja in mišljenja zaposlenih. Hkrati pa pomeni tovrstno preseganje ozke naravnosti zaposlenih nujen korak za oblikovanje in negovanje organizacijske kulture, kjer je poudarjeno medsebojno sodelovanje, povezovanje, komuniciranje, kreativnost zaposlenih in predvsem pripravljenost na posredovanje in prenašanje znanja.

3.8.1 Inovativna organizacijska kultura in kultura znanja

Inovativnost je širok pojem, ki ne obsega le vidnih inovacij in tehnologije. Da bi bili zaposleni ustvarjalni in kreativni mora podjetje zagotoviti nekatere pogoje. Pomembno je, da imajo posamezniki podporo v skupini, da se ne čutijo izolirane, in da je podjetje do določene mere pripravljeno sprejemati tudi tveganja. Organizacija kot celota lahko postane kreativnejša, če je njena organizacijska struktura fleksibilna in ne toga, hierarhična s strogim stilom vodenja, kjer se od posameznikov pričakuje konformizem in prilagajanje ustaljenemu mnenju večine. Tabela 2 podrobneje navaja dejavnike inovativnosti v podjetju.

Tabela 2: Dejavniki inovativnosti v podjetju

PROCESI	ORGANIZACIJSKA KULTURA
osredotočenje na aktivno partnerstvo, komunikacijo s strankami, dobavitelji in pridobivanje njihovih mnenj ter pričakovanj	oblikovanje odprtega, kooperativnega stila vodenja, zmanjševanje pomena klasične hierarhije.
sistem motiviranja in nagrajevanja za inovativnost	zagotavljanje ustrezne klime
sprotno preverjanje zadovoljstva strank, položaja na trgu, tržnega deleža in števila novih proizvodov	prevzemanje tveganja do določene mere
skrb za profesionalni razvoj, nadgrajevanje, izpopolnjevanje zaposlenih	oblikovanje vizije s poudarkom na izboljšavah, inovativnosti in kakovosti

Vir: Amidon in Skyrme, 1997, str. 277.

Uvajanje sprememb v podjetje je za vodstvo podjetja zelo težaven in tudi dolgotrajen proces. Spreminjanje organizacijske kulture pa je še posebno zahtevna naloga, saj je kultura v podjetju globoko ukoreninjena in zajema prepričanja, stališča in vrednote posameznikov. Najbolj ukoreninjena so prepričanja, saj so nezavedna in jih ne moremo opazovati. Po drugi strani pa so najmanj ukoreninjeni vedenjski vzorci, ki jih je tudi možno opazovati in ocenjevati.

Vodilni v podjetju naj bi bili drugim za zgled, njihova dejanja so pomembnejša od govorjenja. V podjetju mora biti zelo dobro razvito komuniciranje, saj na ta način odpravljamo blokade, nezaupanje, izoliranost in tako vanj vnašamo klimo povezovanja in nenehnega pretoka idej, informacij in znanja. Na razvoj kulture znanja v podjetju vplivajo mnogi dejavniki – nekateri spodbujevalno, drugi zaviralno. Pomembno je, da se jih zavedamo in poznamo njihov učinek ter ga poskušamo odpraviti ali vsaj oslabiti.

Spodbujevalci kulture znanja so lahko:

- ❑ medsebojno zaupanje,
- ❑ jasni, natančno določeni cilji,
- ❑ nagrajevanje dobrega dela,
- ❑ možnost sodelovanja na usposabljanjih, tečajih,
- ❑ dovolj časa za interakcije med zaposlenimi in
- ❑ pozitiven odnos do sprememb in njihovega uvajanja.

Zaviralci kulture znanja pa so lahko naslednji:

- ❑ slaba komunikacija vodstva z zaposlenimi,
- ❑ skriti interesi vodstva,
- ❑ nezaupanje in nepripravljenost na spremembe,
- ❑ zaposleni vidijo uvajanje upravljanja z znanjem kot dodatno obremenitev,
- ❑ neusklajenost ciljev med posamezniki in podjetjem in
- ❑ podcenjevanje vrednosti zaposlenih.

Podjetje, ki želi uvesti in delovati v smeri uvajanja upravljanja z znanjem, mora oblikovati takšno organizacijsko kulturo, kjer bodo prevladovali pozitivni dejavniki in minimizirani negativni.

3.9 Prenos znanja

Prenos znanja po organizaciji in med zaposlenimi predstavlja zelo pomemben del upravljanja z znanjem. Samo kodificiranje, identificiranje in generiranje znanja namreč niso dovolj za maksimalen izkoristek učinkov znanja v podjetju.

Ne glede na to ali obvladujemo proces prenosa znanja znotraj podjetja, smo lahko priča takšnemu prenosu. Že, če samo vprašamo sodelavca, kako se katera stvar naredi, sprašujemo po njegovem znanju z namenom, da bi se tudi sami nekaj naučili. Večje kot je podjetje, večja je verjetnost, da se iskano znanje nahaja znotraj podjetja. Hkrati pa z velikostjo pada verjetnost, da bomo vedeli kje najti iskano znanje.

Na primer podjetje Hewlett Packard v ta namen uporablja tako imenovane "Discussion" forume. Slika 5 prikazuje uvodno stran Hewlett Packardovih forumov znanja.

Slika 5: Hewlett Packardovi forumi znanja

AMERICAS / ASIA PACIFIC

hp home products & services support solutions how to buy

hp
invent

IT resource center forums

This is the gathering place for IT Professionals to solve problems, exchange ideas, and learn lessons from the 500,000 fellow peers who also use the IT Resource Center. HP engineers participate in all of these forums to share their advice; however, these forums are intended as a peer-to-peer resource. Select your community:

IT resource center

- search
- online help
- contact hp
- IT resource center home
- my profile
- login
- maintenance and support forums
- training and education
- planning, design, and implementation
- site map

Forums Update

- * jim ferguson is [first ITRC Forums Olympian!](#)
- * stuart elflett is [new forums printer expert](#)
- * post of the month: [alan casey lends a big hand and gets his first hat!](#)
- * NA contract customers [archive your software call manager data](#)
- * get the most out of a [support contract](#)
- * [post your forums successes here](#) for others to read!

areas of peer problem solving

- **business related issues**
 - business recovery planning
- **digital imaging**
 - capshare
 - digital senders
 - photo systems
- **hp-ux**
 - databases
 - LVM
 - networking
 - system administration
 - more...
- **Linux**
 - networking
 - planning
 - secure OS software for Linux
 - system administration
 - more...
- **management software**
 - OpenView applications
 - OpenView Network Node Manager
 - OpenView Omniback
 - OpenView Operations (ITO)
 - more...
- **Microsoft**
 - BackOffice products
 - Windows 2000
- **multifunction, copiers, & fax products**
 - black & white multifunction
 - color multifunction
 - fax products
 - personal copiers
 - more...
- **print servers & peripheral sharing**
 - installation and management software
 - jetsend solutions
 - printer servers
- **printers**
 - black and white
 - color
 - large format
 - software
 - more...
- **servers**
 - 9000
 - e3000
 - netservers
 - planning
 - more...
- **storage**
 - disk
 - disk array
 - storage area networks (SAN)
 - tape libraries
 - more...

top users

- 1 [James R. Ferguson](#) 20383 pts.
- 2 [A. Clay Stephenson](#) 18596 pts.
- 3 [Sanjay](#) 9968 pts.
- 4 [Patrick Wallek](#) 9407 pts.
- 5 [Stefan Farrelly](#) 9111 pts.
- 6 [harry d brown jr](#) 8006 pts.
- 7 [Alex Glennie](#) 7356 pts.
- 8 [Michael Tully](#) 6270 pts.
- 9 [Wodisch](#) 5927 pts.
- 10 [Sridhar Bhaskarla](#) 5899 pts.
- 11 [Bill Hassell](#) 4732 pts.
- 12 [Rita C Workman](#) 4702 pts.
- 13 [Bill McNAMARA](#) 4650 pts.
- 14 [Andreas Voss](#) 4477 pts.
- 15 [Steven Sim Kok Leong](#) 4472 pts.
- 16 [John Palmer](#) 4387 pts.
- 17 [melvyn burnard](#) 4154 pts.
- 18 [Ian](#) 4140 pts.
- 19 [Santosh Nair](#) 3721 pts.
- 20 [Ramesh Upadhyayula](#) 3682 pts.
- 21 [Victor Berridge](#) 3411 pts.
- 22 [Kathy Brockway](#) 3383 pts.
- 23 [RajMan](#) 3355 pts.
- 24 [Vincenzo Restuccia](#) 3170 pts.
- 25 [Dan Hetzel](#) 3137 pts.

→ all forum members
→ more about the point system

most active forums

- [hp-ux - system administration](#) 51138

Vir: Hewlett Packard, 2002

Forumi so odprti javnosti in omogočajo vsem, katerim se pojavi kakšno vprašanje ali problem, obrniti se nanje. Forumi zajemajo različna področja, od operacijskih sistemov Windows, Linux, HP-UX, tiskanja, do uporabe Hewlett Packard OpenView produktov. Ljudje, ki odgovarjajo na vprašanja, izhajajo tako iz vrst razvijalcev,

managerjev kot uporabnikov. Sodelovanje v forumu ni omejeno in ni potrebno biti zaposlen v podjetju Hewlett Packard. Branje odgovorov je zastoj. Za sodelovanje, torej pri postavljanju vprašanj in odgovarjanju na vprašanja, pa se je najprej potrebno identificirati. Da pa so spodbudili pisanje odgovorov, lahko tisti, ki zastavi vprašanje, odgovor oceni s točkami do deset. Posamezniki tako med seboj tekmujejo v zbiranju točk. Njihovo sodelovanje v procesu pa je seveda opaženo in nagrajeno tudi s strani njihovih nadrejenih.

Raziskava Inštituta USP med slovenskimi podjetji je pokazala, da imajo slovenska podjetja veliko znanja v posameznikih, rezervo pa predstavlja udejanjanje tega znanja v prakso. Ker znanje predstavlja enega od ključnih virov slovenskega gospodarstva, je ravno pospeševanje prenosa znanja nadvse pomembno za dvig konkurenčne sposobnosti slovenskega gospodarstva (Peršak in Češnovar, 2001).

Za uspešen prenos znanja je potrebno graditi in vzpostaviti mehanizme pretakanja in prenosa znanja med posameznimi nosilci znanja. Glede na to, da v organizaciji obstajajo različne vrste znanja, se nekatere lažje prenašajo kot druge. Znanje zapisano v dokumentaciji (eksplicitno znanje) ali poročilih je lažje prenosljivo. Medtem ko je znanje, ki se nahaja v glavah posameznikov (tiho znanje) težje prenosljivo. Takšno znanje lahko prenašamo večinoma preko komunikacije (Davenport in Prusak, 1998, str. 69).

Pri prenosu znanja v prakso je ključna vloga managementa, ki mora preko elementov strateškega managementa oblikovati temu primerno organizacijsko strukturo, procese in kulturo. V praksi se to odraža v ustreznih sistemih nagrajevanja prenosa znanja in v odprtih komunikacijah, na kar poleg ustreznih procesov vpliva tudi stil poslovanja (Peršak in Češnovar, 2001).

Nonaka in Takeuchi menita, da se znanje ob procesu inverzije preoblikuje iz ene oblike v drugo. Pri tem razlikujeta med tihim oziroma tacitnim in izraženim oziroma eksplicitnim znanjem. Razlikujeta štiri glavne prehode iz enega v drugo znanje (Nonaka in Takeuchi, 1995, str. 45):

- **Internalizacija.** Izraženo znanje prehaja v tiho. Posamezniki preko pisnih virov, baz podatkov in učenja med delom osvojijo znanje, ki postane del njih.
- **Socializacija.** Z druženjem prehaja tiho znanje posameznikov tiho na druge. Tu gre predvsem za neposredni način komuniciranja, tako besedno kot z zgledom.
- **Kombinacija.** Izraženo znanje se oblikuje in ureja v različne eksplicitne oblike podatkovnih baz in dokumentov.

- **Eksternalizacija.** Za ta prehod je značilno prehajanje tihega znanja v eksplicitno. To se lahko zgodi preko dialogov in z oblikovanjem v oprijemljivejše, materialne oblike.

Pri tem Nonaka in Takeuchi poudarjata, da so ti prehodi ključni pri posredovanju širjenja znanja po organizaciji. Poseben izziv sta mobilizacija in preobrazba tihega znanja v izraženo znanje. Kreiranje znanja se prenaša v obliki spirale – posamezniki osvajajo določeno znanje, se v fazi socializacije pogovarjajo z drugimi, nato izrekajo novo nastale ideje v fazi eksternalizacije, kjer tiho znanje prehaja v izraženo in se tako nadaljuje v procese kombinacije in internalizacije. Znanje na ta način prehaja od ravni posameznika, preko teamov, organizacijske do medorganizacijske ravni.

3.9.1 Strategije prenosa znanja

Za uspeh podjetja je pomemben spontan in nestrukturiran prenos znanja. Naloga managementa znanja je razviti takšne strategije, ki bi omogočale spontane prenose znanja med posamezniki. Predvsem je tak način prenosa znanja pomemben za tiste organizacije, katerih primarna naloga je generiranje znanja.

Obstaja več tehnik, ki jih podjetja uporabljajo za prenos znanja. Podjetja uporabljajo zveze, pooblaščenke, delavnice, trening, tehnična poročila, zunanje sodelavce in mnogo drugih tehnik. Veliko japonskih podjetij je ustanovilo tako imenovane "pogovorne sobe", da bi na ta način pripomoglo k nepredvidljivemu prenosu znanja med sodelavci. V teh sobah lahko zaposleni pijejo tradicionalni zeleni čaj in se pogovarjajo o tekočih projektih, nalogah in ciljih. Pričakovano je, da bodo govorili o delu ter na ta način pripomogli k dvigovanju vrednosti podjetja. Vendar pa se metode prenosa znanja med organizacijami razlikujejo. Odvisne so od nacionalne kulture, pa tudi od kulture podjetja.

Nekatera podjetja so se z velikim uspehom lotila sejmov znanja (knowledge fairs), s pomočjo katerih se zberejo zaposleni razpršeni po celi državi ali celo po svetu. Sodelujoči lahko sproščeno hodijo naokrog in se udeležijo predavanj, ki jih zanimajo. Hkrati se lahko spoznajo in govorijo z ljudmi, s katerimi so bili doslej v stiku samo preko elektronske pošte (Davenport in Prusak, 1998, str. 94). Po ameriškem zgledu tudi pri nas podjetja organizirajo tako imenovane treninge za krepitev skupinskega dela (team building).

Po drugi strani pa obstaja veliko kulturnih faktorjev, ki ovirajo prenose znanja. Takšne ovire imenujemo trenja, ker upočasnijo ali onemogočijo prenos znanja po organizaciji. Najpogostejša trenja in načini njihovih rešitev so navedeni v tabeli 3.

Tabela 3: Trenja pri prenosu znanja

TRENJE	MOŽNA REŠITEV
pomanjkanje zaupanja	graditev zaupanja in odnosov z osebnimi stiki, sestanki
drugačne navade, jezik in način obnašanja	kreiranje enakopravnosti s pomočjo izobraževanj, razprav, publikacij, teamskega dela in rotacijo delovnih mest
pomanjkanje časa, prostorov za druženje; vplivajo na produktivnost pri delu	uvedba časa in ureditev prostora za prenos znanja: sejmi znanja, razpravljalne sobe, konferenčna poročila
položaj in nagrade gredo "lastnikom" znanja	ocenitev dela in delitev nagrad zaslužnim
pomanjkanje vpojne zmožnosti sprejemnikov	izobraževanje zaposlenih zaradi fleksibilnosti; zagotoviti čas za učenje in biti odprt do novih idej
prepričanje, da je znanje privilegirano določenim skupinam	spodbuditi nehierarhičen dostop do znanja; kvalitetne ideje morajo biti pomembnejše od statusa izvora
nestrpnost do napak ali potrebe po pomoči	sprejemanje in nagrajevanje kreativnih napak in sodelovanja; ne nazaduj nekoga, če ne ve vsega

Vir: Davenport in Prusak, 1998, str. 97.

Prenos znanja je sestavljen iz dveh akcij: oddaje znanja na potencialnega sprejemnika in vsrkanja znanja s strani te osebe ali skupine. Če znanje ni absorbirano, ne moremo govoriti o njegovem prenosu.

PRENOS ZNANJA = ODDAJA ZNANJA + VSRKANJE (IN UPORABA) ZNANJA

Cilj prenašanja znanja je izboljšanje zmožnosti opravljanja nalog organizacije in hkrati povečanje njene vrednosti. Tudi oddajanje in absorpcija znanja skupaj nimata nobene vrednosti, če novo znanje ne vpliva na spremembo obnašanja ali pelje k razvoju nove ideje, ki vodi k novemu obnašanju organizacije (Davenport in Prusak, 1998, str.101).

3.10 Vloge in izkušnje pri upravljanju z znanjem

Če hočemo, da bo upravljanje z znanjem v podjetju uspešno, morajo podjetja kreirati in določiti vloge in izkušnje pri zajemanju, pretoku in uporabi znanja. Ljudje so tisti, ki dodajo vrednost podatkom in informacijam ter jih pretvorijo v znanje.

Nekatere novejšje vloge v podjetju, ki so povezane z znanjem, so povezovalec znanja (knowledge integrator), poročevalec, urednik, vratar znanja, inženir in analitik znanja. Malo organizacij ima veliko število zaposlenih, ki so vešči oblikovati in strukturirati svoje znanje. Še manj pa je takih, ki imajo čas, da se usedejo in svoje znanje vpišejo v sistem. Team inženirjev lahko razvije nek zelo dober produkt, ampak nobeden od inženirjev nima časa ali znanja, da bi opisal potek dela na projektu. Podjetja zato potrebujejo ljudi, ki bodo izluščili znanje od ljudi, ki ga imajo, ga strukturirali in nenehno dopolnjevali (Davenport in Prusak, 1998, str. 110).

Dobri zaposleni, ki opravljajo dela v povezavi z znanjem na katerem koli nivoju v podjetju, naj bi imeli kombinacijo strukturiranega, tehničnega znanja in izkušenj ter občutek za kulturne politične in osebne vidike znanja. Takšne značilnosti sicer niso pogoj, vendar so pomembne pri delu z uporabniki znanja.

Poročevalce, urejevalce in knjižničarje lahko združimo v skupino. Takšnim ljudem bi preprosto lahko rekli inženirji znanja. V nižji sloj strukture organizacije na področju znanja spadajo:

- **Inženir znanja.** Naloga inženirja znanja je ugotavljati vrzeli znanja v podjetju in na osnovi tega skrbeti za razvoj potrebnih veščin. Seznanjen mora biti tudi z "zemljevidom" znanja ter mora natančno vedeti pri katerem posamezniku oziroma teamu je določeno znanje.
- **Urednik znanja.** Urednik znanja skrbi za urejanje izraženega (eksplicitnega) znanja v oblike, ki so razumljive in dostopne uporabnikom. Urednik znanja ureja in zbira informacije tudi iz zunanjih virov, ki jih prevede v razumljiv jezik in logično uredi.
- **Analitik znanja.** Analitik povezuje uporabnika in bazo znanja. Usklajuje tudi potrebe in povpraševanja obeh strani, zato so za to delovno mesto pomembne sposobnosti za medsebojno komuniciranje, empatija in pripravljenost za pomoč.
- **Navigators znanja.** Posebnost navigatorja znanja je v poznavanju in identificiranju virov posameznih informacij in znanja v organizaciji.
- **Vratar znanja.** Vratar znanja je v vlogi kanala, po katerem se pretaka znanje v organizacijo ali team in obratno. Nadzoruje pritok in odtok znanja iz podjetja, kar je še posebej pomembno.
- **Broker znanja.** Broker združuje vlogi vratarja in navigatorja znanja. Ima zelo dobro razvito interno in eksterno kontaktno mrežo.

Srednji sloj v strukturi organizacije na področju znanja zaseda **vodja projekta znanja**. Naloge vodje so naslednje:

- ❑ razvoj ciljev projekta,
- ❑ zbiranje in vodenje teamov,
- ❑ ugotavljanje in upravljanje z zahtevami strank,
- ❑ spremljanje proračuna (prihodkov in odhodkov) projekta in planov,
- ❑ identifikacija in reševanje projektnih problemov.

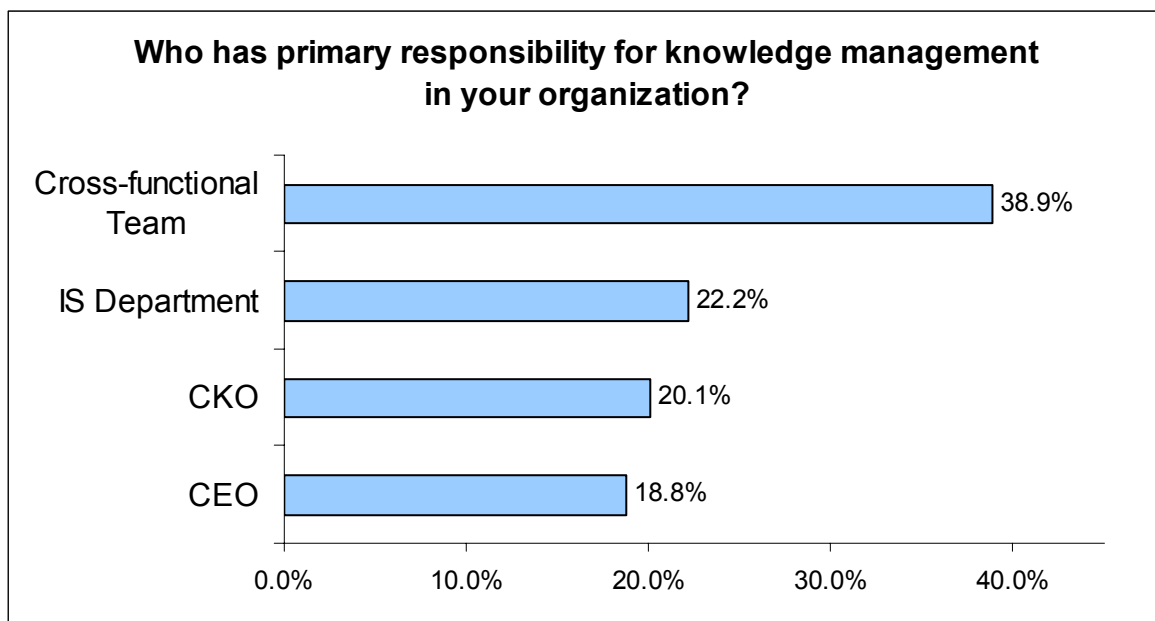
Nekaj podjetij v Evropi in pa dokaj veliko v združenih državah Amerike je oblikovalo posebno delovno mesto za **direktorja upravljanja znanj** (Chief Knowledge Officer – CKO). Vloga direktorja upravljanja znanj je zelo kompleksna in vsestranska, sodi pa v višji sloj strukture organizacije podjetja na področju znanja. Po Davenportu, mora imeti direktor upravljanja znanja tri kritične odgovornosti in naloge:

- ❑ Poskrbeti mora za oblikovanje infrastrukture za upravljanje z znanjem. Ta naloga vključuje skrb za računalniško opremo, omrežje, baze podatkov, iskalcev podatkov in zagotovitve potrebnih softverskih orodij.
- ❑ Poskrbeti mora za gradnjo kulture znanja v podjetju.
- ❑ Ugotoviti mora, kako bo boljše upravljanje z znanjem pripomoglo podjetju. Ali bodo podjetja z upravljanjem znanja bolj produktivna in varčna? Direktor upravljanja znanja bo moral prikazati povrnitev vlaganj v znanje v ekonomskem smislu, v denarju.

Na internetu je bila pod okriljem revije *Knowledge Management* izvedena anketa, v kateri so ljudi spraševali, kdo ima ključno vlogo pri upravljanju znanja v podjetju, v katerem so zaposleni. Na voljo so bili štirje odgovori, rezultati pa so naslednji (grafični prikaz rezultatov prikazuje slika 6):

- ❑ direktor – Chief Executive Officer (CEO),
- ❑ direktor upravljanja z znanjem – Chief Knowledge Officer (CKO),
- ❑ oddelek informacijskih sistemov – Information Systems Department in
- ❑ raznolik team iz predstavnikov različnih organizacijskih enot – Cross-functional Team.

Slika 6: Rezultati ankete revije Knowledge Management



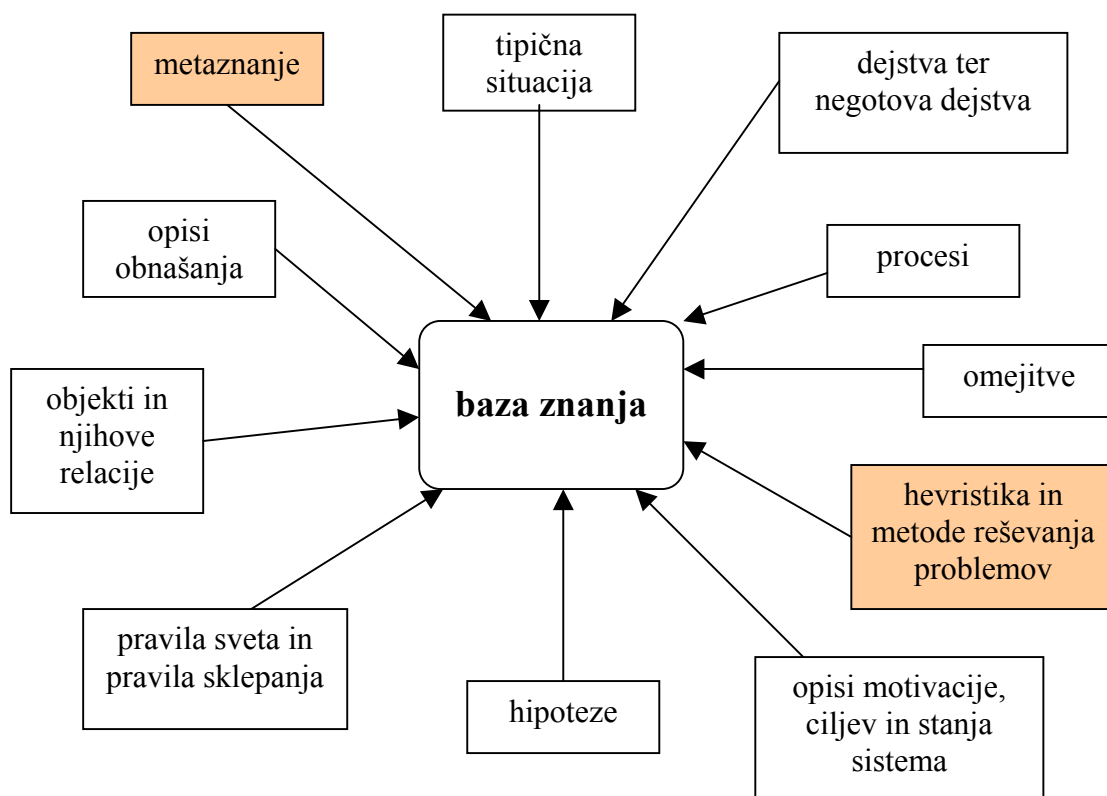
Vir: Knowledge Management, 2002

Zanimivi rezultati. Naj bo s tem dovolj znanju in se podrobneje lotimo še opisa baze znanja kot dela sistema za upravljanje z znanjem.

4 Baza znanja kot del sistema za upravljanje z znanjem

Kvaliteta ekspertnega sistema je v glavnem funkcija obsega in kvalitete njegove baze znanja. (Harmon in King, 1985, str. 240). Baza znanja vsebuje znanje o specifičnem področju, ki je ponavadi strokovne ali znanstvene narave. Potem vsebuje pravila, ki opisujejo dogodke, relacije med dogodki in metode ter hevrstike za reševanje problemov (Kukovica, 1993). Slika 7 predstavlja sestavne dele baze znanja.

Slika 7: Sestavni deli baze znanja.



Vir: Budin et al., 2001.

V bazi je vsebovano znanje dveh vrst:

- splošno priznana in v posamezni stroki znana dejstva (znanje iz problemske domene) in
- ekspertna pravila sklepanja in odločanja, hevrstike (znanje o načinu reševanja problemov).

K znanju iz problemske domene spadajo vsi na sliki 2 naštetih sestavni deli baze znanja, razen metaznanja in hevrstike ter metod reševanja problemov.

Eden glavnih problemov raziskav in razvoja baze znanja je, kako predstaviti velike količine znanja v obliki, ki bi omogočala njeno učinkovito uporabo (Schmoldt in Rauscher, 1996). Ta problem je znan kot predstavitveni problem (knowledge representation problem). Pri ekspertnih sistemih se odločamo za tiste, ki podpirajo modularnost, inkrementalnost in enostavno spremenljivost baze znanja (Bratko, 1986). Predstavitev mora biti takšna, da je lahko shranjena in obdelana s strani računalnika. Kodiranje formalizira znanje in pripomore k ponavljanju in zanesljivosti sklepanja. Prav tako pomaga zagotoviti trdnost/trajnost med skupino strokovnjakov (Schmoldt in Rauscher, 1996).

4.1 Predstavitev znanja

Največji problem gradnje sistema podprtega z znanjem je formalizacija znanja – zapisovanje znanja v obliki, ki jo lahko razumejo programi. Formalizacija je tudi sicer postopek, ki v razvoju civilizacije poteka vzporedno z razvojem znanja. Od začetkov mišljenja, ki so pripeljali do verbalnega sporočanja (osnovna formalizacija), preko odkritja zapisovanja (formalizacija z abstraktnimi simboli), do sporočanja znanstvenih odkritij, ki mora biti izvedeno tako, da ga razume strokovnjak kjerkoli, neodvisno od kulturnega ozadja (na primer matematični jezik) (Sistemi podprti z znanjem, 2002).

Formalizirati je veliko lažje trdo, kot mehko znanje. Proces formalizacije navadno poteka preko razbijanja problema na podprobleme, dokler niso dovolj majhni, da so razvidni in jih je mogoče zapisati v obliki, ki jo razume program (v nekem formalnem jeziku). Strokovnjak, ki operira z mehkim znanjem, se razgradljivosti pogosto ne zaveda (odločanje na podlagi "izkušenj in intuicije").

Formalni jeziki so umetni jeziki. Naravni jeziki niso formalni, ker ni pravil, ki bi določala vse možne stavke (trditve) v naravnem jeziku, in ker je pomen trditev odvisen tudi od konteksta (Sistemi podprti z znanjem, 2002).

Rešitve problemov reševanih s sistemi podprtimi z znanjem so odvisne tudi od načina formalizacije znanja v bazi znanja in od metod za njeno uporabo.

Področje, ki se ukvarja z raziskovanjem načinov, kako formulirati znanje, da bo čimbolj uporabno v procesu sklepanja, imenujemo predstavitev znanja (knowledge representation), posamezne metode pa predstavitvene metode, sheme ali formalizmi za predstavitev in shranjevanje znanja o domeni. Cilj predstavitve znanja je čimbolj učinkovito iskanje in sklepanje iz tako shranjenega znanja (Shapiro, 1992).

Razlikujemo več različnih metod za predstavitev znanja, npr. okviri, semantične mreže, predikatni račun in produkcijska pravila. Na kratko bom predstavila prednosti in pomanjkljivosti vsake izmed njih. V ekspertnih sistemih najpogosteje uporabljamo

formalizem produkcijskih pravil. Zaželeno je, da izbrani formalizem predstavi znanje tako, da je uporabniku razumljivo ter da ga je mogoče enostavno preverjati, dopolnjevati in spreminjati (Kukovica, 1993).

4.1.1 Produkcijska pravila

Produkcijska pravila so najpogostejše uporabljena metoda. V začetku se je sicer uporabljal izraz produkcijski sistemi. Formalizem produkcijskih sistemov je leta 1943 izdelal Emil Post (Schmoldt in Rauscher, 1996).

Logične relacije med koncepti problemskega področja pogosto opišemo s pravili oblike ČE-POTEM (if-then). Torej, če velja pogoj P, potem velja sklep S s faktorjem zaupanja G.

Primer logične relacije:

ČE 1 je infekcija primarno bakteremija IN
 2 je izvor kulture sterilno mesto IN
 3 sumimo, da izvira organizem v kulturi iz gastrointestinalnega trakta,
POTEM lahko z verjetnostjo 0.7 trdimo, da je organizem v kulturi bakterija.

Vir: Sistemi podprti z znanjem, 2002.

Posplošeno: Če A, potem B.

Leva stran pravila (A) predstavlja pogoj ali neko situacijo, v kateri je pravilo uporabno. Desna stran (B) pa specifikira posledico, sklep ali akcijo pravila. Vsaka stran lahko vsebuje več členov, ki so med seboj povezani z operatorji IN (AND), ALI (OR) ter redkeje NE (NOT) (Gibb, 1986).

Produkcijsko pravilo je sestavljeno iz procedure, imenovane jedro pravila in iz vzorca. Vzorec je tisti del pravila, ki ga sistem primerja z dejstvi v podatkovni zbirki, jedro pa del, iz katerega izpeljuje nova dejstva (Shapiro, 1992). Pri sklepanju v smeri naprej je jedro desna, vzorec pa leva stran pravila. Pri sklepanju nazaj je ravno obratno.

Produkcijsko pravilo interpretiramo z načelom imenovanim "*modus ponens*". Ta se glasi: Če vemo, da velja A in da velja, da iz A sledi B, lahko sklepamo, da velja tudi B. To načelo je osnova izpeljevanja dejstev iz aktiviranih produkcijskih pravil. Aktiviranje produkcijskih pravil imenujemo veriženje ali sklepanje (chaining). Sklepanje je proces, sestavljen iz dveh faz, faze selekcije in faze aktivacije. V fazi selekcije sistem ugotovi, katera pravila so primerna ter eno izmed njih izbere za aktivacijo. V fazi aktivacije interpretira izbrano pravilo in iz njegovega jedra izpelje dejstvo, ki ga doda v podatkovno zbirko. Vsa komunikacija poteka preko dejstev v

podatkovni zbirki, saj pravila ne morejo neposredno sprožiti drugih pravil (Shapiro, 1992).

Osnovni metodi sklepanja sta:

- ❑ **sklepanje v smeri naprej** (forward chaining) in
- ❑ **sklepanje v smeri nazaj** (backward chaining).

Sklepanje v smeri naprej je podatkovno vodena metoda (data-driven) iskanja rešitve (Fenly, 1988). Mehanizem sklepanja začne iskati končni cilj – neko zadovoljivo rešitev z znanim dejstvom. Tega primerja z vzorci produkcijskih pravil, ki so pri sklepanju v smeri naprej leva strani pravil. Če se leva stran ujema ali izenači z dejstvom v podatkovni zbirki, se pravilo lahko aktivira. Aktivirano pravilo podatkovno zbirko obogati z novim dejstvom, izpeljanim iz jedra oziroma desne strani pravila. Izpeljano dejstvo sedaj nastopa v podatkovni zbirki enakovredno in pri ponovitvi postopka se lahko sproži pravilo, ki ima na levi strani prav to izpeljano dejstvo. Opisani postopek se ponavlja vse dotlej, dokler v zbirki produkcijskih pravil obstajajo pravila, ki se še lahko aktivirajo ali dokler mehanizem sklepanja ne izpelje dejstva, ki predstavlja zadovoljivo rešitev (Waterman, 1986).

Cilj sklepanja v smeri nazaj je potrditi ali zavreči pravilnost ciljne hipoteze. Mehanizem sklepanja najprej preveri ali lahko ciljno hipotezo potrdi z dejstvom v podatkovni zbirki. Če ne, išče pravilo, ki hipotezo lahko potrdi. Takšno pravilo ima na svoji desni vzorec, ki je enak ciljni hipotezi, levo stran pa potrjujejo dejstva iz podatkovne zbirke. Če se leva stran pravila izenači z dejstvi iz podatkovne zbirke, je sklepanje končano, mehanizem sklepanja pa potrdi pravilnost ciljne hipoteze. V nasprotnem primeru levo stran razume kot podcilj, ki ga poskuša potrditi na enak način kot ciljno hipotezo. Torej išče pravila, ki imajo na desni strani podcilj. Postopek se ponavlja, dokler mehanizem sklepanja ne potrdi vseh podciljev ali dokler ne zmanjka pravil, s katerimi bi lahko podcilje potrdil. Pravilnost hipoteze je dokazana, ko so potrjeni vsi podcilji, sicer ciljne hipoteze ni možno potrditi (Waterman, 1986).

Sklepanje v smeri nazaj izhaja iz ciljne hipoteze, zato je to ciljno vodena (goal or hypothesis driven) metoda iskanja rešitve (Fenly, 1988). Sklepanje v smeri naprej ugotavlja, do kakšnih zaključkov pridemo glede na dana dejstva in se uporablja, ko ni mogoče vnaprej predvideti, kateri izmed številnih možnih zaključkov je najprimernejši. Sklepanje v smeri nazaj pa ugotavlja, katera ciljna hipoteza je pravilna in se uporablja, kadar obstaja manjše število ciljev oziroma zaključkov, ki jih je možno določiti vnaprej (Brooks, 1987). Pogosto je koristno kombiniranje obeh metod na različnih nivojih reševanja problema, kar imenujemo *kombinirano sklepanje* (mixed chaining).

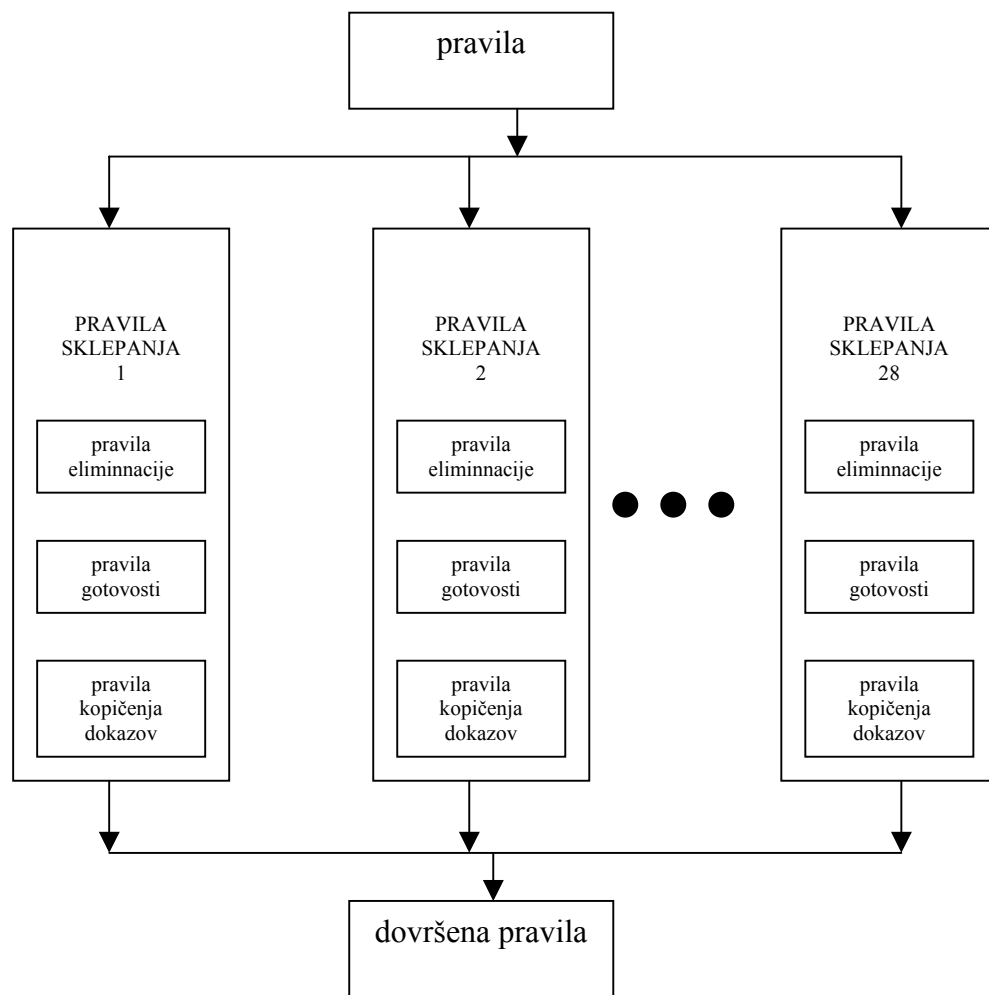
Pri obeh metodah sklepanja je lahko v danem koraku za aktiviranje primernih več produkcijskih pravil hkrati. Takšno množico imenujemo *konfliktna množica* (conflict set). Dilemo, katero pravilo sprožiti najprej, rešuje mehanizem sklepanja z uporabo tehnik reševanja konflikta (conflict-resolution techniques). Preprosta tehnika reševanja konflikta je izbor pravega pravila iz določenega zaporedja pravil v konfliktni množici.

Učinkovitejši tehniki, podprti z metapravili ali hevrstiko, sta tehniki, pri kateri se najprej aktivira pravilo, ki je najbolj specifično ali omejeno glede na dani problem ali izbor pravila, in ki je najbližje dobljenemu rezultatu aktiviranega pravila (Prerau, 1990).

Poleg različnih tehnik reševanja konflikta se lahko produkcijski sistemi med seboj razlikujejo tudi po funkcijah, ki določajo, kdaj je vzorec pravila izenačen z dejstvom v podatkovni zbirki. Pri tem ni nujno, da je ujemanje natančno.

Primer pravila sklepanja prikazuje slika 8.

Slika 8: Pravila sklepanja



Vir: Schmoldt in Rauscher, 1996, str. 39

Baza znanja zgrajena na podlagi pravil lahko vsebuje različne tipe pravil. Pravila so lahko strateška, pravila eliminacije, dokazna pravila in podobno.

Prednosti pravil (Rajkovič, 2001):

- ❑ vsako pravilo predstavlja majhen, relativno neodvisen del znanja,
- ❑ dodajanje in spreminjanje pravil je relativno enostavno in
- ❑ pravilo podpira transparentnost sistema, ker omogoča odgovarjanje na vprašanja tipa "kako" in "zakaj".

Pomanjkljivosti pravil so naslednje (Schmoldt in Rauscher, 1996, str.40):

- ❑ razlikovanje med zankami in ponavljanji je lahko zapleteno,
- ❑ težko je predvideti vrstni red izvajanja pravil; to je lahko kritično pri dodajanju novih pravil v sistem,
- ❑ podlaga pravil mora biti previdno razvrščena in strukturirana, saj lahko ob več sto pravilih pride do zmešnjave in
- ❑ pravila ne morejo nadomestiti naravne hierarhične strukture domene znanja.

4.1.2 Okviri

Okvire je leta 1975 razvil Marvin Minsky. Predstavljajo osnovo za razumevanje vizualnega dojetja, oblikovanje dialoga v naravnem jeziku in drugih kompleksnih načinov obnašanja. Razvili so se kot metoda organiziranja obširnega znanja za reševanje navedenih nalog. Njihov osnovni namen je, da predstavimo znanje o tipičnih objektih in dogodkih v specifičnih situacijah. Nove informacije potem interpretiramo z uporabo znanih konceptov s "procesiranjem na osnovi pričakovanj" (Schmoldt, Rauscher, 1996, str.41).

Okvir je opis objekta, v katerem je prostor za vsako informacijo vezano na ta objekt. Kot okvir lahko štejemo zbirko med seboj povezanega znanja o določeni temi. Takšno znanje lahko vsebuje dejstva, relacije, pravila, dogodke ali privzete vrednosti.

Teorija okvirov izhaja iz spoznanja, da človek, ko se nahaja v neki novi situaciji, poseže po znanju, ki je plod izkušenj, povezanih z določenimi situacijami, objekti in podobno. Namesto temeljitega preiskovanja in analiziranja nove situacije ter izgrajevanja znanja o njej si priključuje v spomin podobne, že poznane situacije (proto ali stereotipne situacije). Elemente nove situacije nato poskuša vključiti v že obstoječo strukturo vedenja o nečem, v t.i. okvire (Ribarič, 1990).

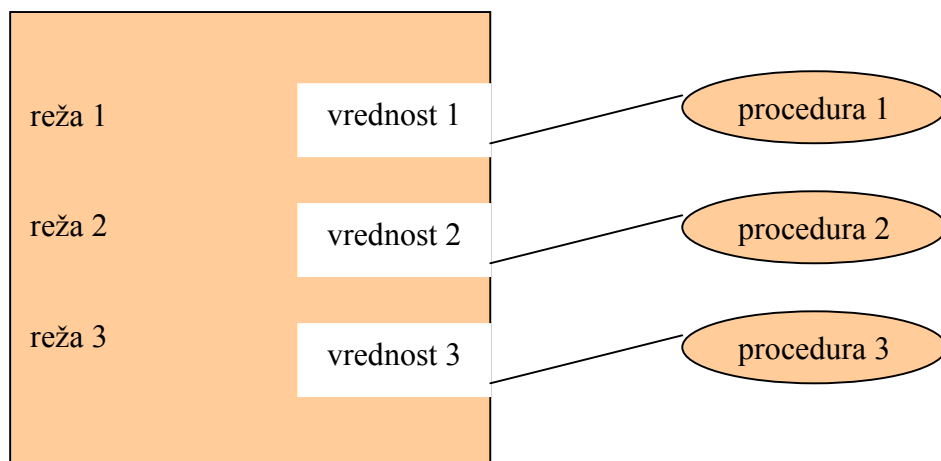
Sistem okvirov je organiziran na podoben način kot semantična mreža. Oba formalizma pravzaprav sodita v sisteme, osnovane na okvirih (frame-based systems) (Waterman, 1986) ali po drugih avtorjih med strukture "polnjenja rež" (slot and filler structures) (Frost, 1986).

Mreža je tudi tukaj sestavljena iz vozlišč in relacij, organiziranih v hierarhično strukturo. Okviri konstituirajo vozlišča, ki na višjih nivojih hierarhične organizacije predstavljajo splošne koncepte, na nižjih pa specifične primere teh konceptov. Za razliko od semantičnih mrež je koncept vsakega vozlišča nadalje definiran z zbirko atributov, imenovanih reže (slots). Atributi lahko imajo svoje vrednosti, to so informacije, s katerimi se reže zapolnjujejo (Waterman 1986). Primer koncepta okvirov prikazuje slika 9.

Z vsako režo so lahko procedure, ki se v določenih situacijah aktivirajo.

Slika 9: Organizacija vozlišča v sistemu okvirov

KONCEPT



Vir: Waterman, 1986.

Trije pogosto uporabljeni tipi procedur, povezani z režami, so (Waterman, 1986):

- ❑ procedura, “če je dodano” (if-added) se aktivira, ko je v režo vstavljena neka vrednost,
- ❑ procedura “če je potrebno” (if-needed) se aktivira, ko je vrednost reže potrebno determinirati, zareza pa je prazna,
- ❑ procedura “če je odstranjeno” (if-removed) se aktivira, ko je vrednost zareze zbrisana, odstranjena.

Med prednosti okvirov zagotovo sodi, da strukturno osnovo znanja lahko prikažemo grafično, kar lahko močno poveča njeno razumljivost.

Po drugi strani pa lahko fleksibilnost in nadzor nad pametnim procesom vplivata na njegovo razumevanje in predpostavljeno obnašanje. Sisteme zgrajene na podlagi okvirov je ponavadi težje planirati, razviti in implementirati kot sisteme zgrajene na podlagi pravil (Schmoldt in Rauscher, 1996, str.44).

4.1.3 Semantične mreže

Semantične mreže so eden od načinov, s katerimi podajamo računalniku znanje v hranitev in obdelavo. Prve semantične mreže so se pojavile v 3. stoletju n.š. in so že takrat grafično predstavljale znanje o svetu, ki ga je opisal Aristotel. Še danes je

najmočnejši adut semantičnih mrež v primerjavi z ostalimi predstavitvami znanja prav njihova vizualna predstava, ki je blizu človeku.

Definicija semantične mreže (John F. Sowa): Semantična mreža (Semantic Network) je struktura, ki predstavlja človekovo znanje tako, da ga organizira v vozlišča, medsebojno povezana s povezavami (Shapiro, 1992).

Prve semantične mreže, ki so jih uporabili v umetni inteligenci, so bile namenjene prepoznavanju jezikov, od začetka strojnih, kasneje pa vse bolj tudi naravnih, od tod tudi dileme znanstvenikov o njihovem poimenovanju. Ime "semantične mreže" je nastalo ob raziskavah grafičnega zapisa znanja, ko je Quillian želel predstaviti pomen (semantiko) angleškega jezika z risanjem vozlišč in povezav med njimi. To poimenovanje je uporabila leta 1961 tudi Margaret Masterman. Do danes se je pojavilo še nekaj drugih imen – asociativne mreže (Associative Networks), konceptualne mreže (Conceptual Networks), semantični spomin (Semantic Memory), mrežna predstavitev (network representation). Nova poimenovanja kažejo na to, da semantičnih mrež ne uporabljamo več le za opis naravnih in umetnih jezikov, ampak tudi za zapis drugih znanj, še posebej tistih, ki temeljijo na simbolni strukturi (Shapiro, 1992).

V formalizmu mrež je znanje predstavljeno z vozli povezanimi med seboj. Vozli označujejo objekte, ki so lahko fizični, množice, situacije ali celo same relacije. Povezave pa označujejo binarne relacije med objekti.

Znane so različne oblike mrež, ki se ločujejo po izrazni moči in tipih procedur za manipulacijo z njimi. Za vse oblike je značilno, da struktura, ki je potrebna za kodiranje znanja, je potrebna tudi za vodenje pri iskanju medsebojnih povezav. S pomočjo označb na vozlih in povezavah lahko enostavno pridemo tja, kamor želimo (Kukovica, 1993, str. 85).

Mreže uporabljamo predvsem za naravno opisovanje enostavnih relacij in ne za izražanje kompleksnih formul in povezav. Zaradi tega se predstavitev znanja v obliki mrež uporablja predvsem v sistemih za obravnavanje naravnega jezika in sklepanje na področju množic. Pogosto se v obliki mrež predstavljajo tudi produkcijski sistemi, pri katerih mreža modelira zveze med produkcijskimi pravili.

Med prednosti semantičnih mrež zagotovo sodi predvsem njihova enkratna moč predstavitve, edinstvena med vsemi načini predstavljanja znanja, ki je blizu človeku (Sowa, 1991). V splošnem lahko pretvorimo semantične mreže v negrafični jezik in, pomembnejše, jezik lahko zapišemo s semantično mrežo (Shapiro, 1992). Semantične mreže tudi lahko pretvorimo v kompleksne linearne notacije, ne da bi zmanjšali njihovo izrazno moč.

Semantične mreže pomagajo eksplicitno zapisati označbe iz logike, psihologije, lingvistike, ki so prej bile izražene implicitno (Sowa, 1991).

Semantične mreže so učinkovit pripomoček za opisovanje človekovega naravnega jezika in njegovega razumevanja – orodja, ki ga ljudje uporabljamo vsak dan. S semantičnimi mrežami lahko opisujemo zelo bogate naravne jezike, pa tudi umetne tvorbe, npr. programske jezike. Z njimi preverjamo sintakso in semantično pravilnost vhodnih izrazov za določen jezik.

Norm Sondheimer ugotavlja, da so različne notacije semantičnih mrež tiste, ki znanstvenike ločujejo. Naslednja slabost, ki jo semantičnim mrežam najpogosteje očitajo (Sowa, 1991), je njihova preširoka zasnova. Bill Woods, eden kritikov, poskuša prikazati dvome o semantičnih mrežah preko metafore o gozdu in drevesu: vidi veliko zanimivih dreves, ne ve pa, kakšen je ves gozd. Sprašuje se tudi, kakšno pot po gozdu je ubral, naključno ali je šel po robu gozda. Glede na zelo različna znanja, ki jih prikazujemo s semantičnimi mrežami, misli, da nam primanjkuje algoritmov, s katerimi bi te podatke obdelovali (Sowa, 1991).

Naslednji problem predstavlja predstavitev človekove “zdrave pameti”:

- ❑ modeliranje človekovih namenov,
- ❑ gradnja modelov, ki se spreminjajo s časom,
- ❑ hipotetično sklepanje in
- ❑ razumevanje konteksta.

Področje, ki bo potrebovalo obsežne raziskave in ki je povezano z “zdravo pametjo”, je razumevanje naravnega jezika, ki je preraslo razpoznavanje sintakse, semantike in preprostega sklepanja in si zadalo nalogo, da povezuje različne vrste znanj v eno samo – globalno in tako tvori integrirano zbirko znanja. Naravni jezik je pogojen z zgodovinskim razvojem, sestavljen je iz pravil in izjem, ki jih je skoraj nemogoče obvladovati.

4.1.4 Predikatni račun

Za predstavitev znanja je možno uporabiti tudi matematično logiko oziroma njen podsistem, kar je predikatni račun prvega reda. Dejstva in pravila zapišemo v skladu s sintaktičnimi pravili tega podsistema. Znana dejstva in pravila imamo za aksiome, rešitev ali odgovor na zastavljeno vprašanje pa je izrek, ki ga izpeljemo iz dejstev in zakonitosti.

Prednost predikatnega računa so hitri algoritmi za dokazovanje izrekov, ki temeljijo na tako imenovanem principu resolucije oziroma ločljivosti. V tem formalizmu je tudi enostavno definirati relacije in strukturirati podatke.

Pomanjkljivost je pomanjkanje mehanizmov za modeliranje mehkega znanja (Rajkovič, 2001).

4.2 Načini organiziranja znanja v bazo znanja

Uporabniki postavljajo različna vprašanja, na katera naj bi bil sistem sposoben odgovoriti. Celotna organiziranost sistema mora biti namreč usmerjena k uporabnikom, k njihovim potrebam in željam. Med osnovne tipe vprašanj, ki naj jih sistem zna obdelati sodijo (Siegl, 2002, str. 36):

- ❑ **Zahteve po informacijah.** Uporabnik ve, kaj išče, sistem pa mora znati poiskati ustrezno informacijo oziroma ugotoviti, da je nima. Med vire informacije sodijo uporabniški priročniki, pa tudi dokumenti na spletnih straneh.
- ❑ **Povpraševanje po postopkih.** Tudi tukaj uporabnik ve, kaj išče, vendar ne ve, kako neko stvar opraviti. Uporabnik je, na primer, napisal pismo, a ne ve kako ga je potrebno pravilno oblikovati.
- ❑ **Napaka/znani simptomi.** Uporabnik je odkril neke simptome napake, sistem pa ga mora voditi po bazi znanja do rešitve.
- ❑ **Oblikovanje vprašanja z reševanjem težav.** Uporabnik ve samo, da potrebuje informacijo ali da je nekaj narobe, potrebuje pa pomoč že pri oblikovanju vprašanja ali pri identifikaciji simptoma, preden lahko začne z iskanjem odgovora.

V sistemih podprtih z znanjem se največkrat pojavljajo vsi naštetimi tipi vprašanj. Nekateri se pojavljajo pogostejše kot drugi, zato je od njihove pogostosti odvisno, kakšno organiziranost baze znanja bomo uporabili. Razlikujemo med petimi načini organiziranja baz znanja in sicer (Siegl, 2002):

- ❑ zbirko dokumentov z iskanjem po celotnem besedilu,
- ❑ klasifikacijo v kategorije,
- ❑ odločitvena drevesa,
- ❑ sklepanje na osnovi primerov in
- ❑ nevronske mreže in ostale metode umetne inteligence.

Podrobneje bom opisala samo organiziranje baze znanja z zbirko dokumentov z iskanjem po celotnem besedilu. Le-to sem tudi uporabila v praktičnem delu magistrskega dela.

4.2.1 Iskanje po celotnem besedilu

Zbirka dokumentov z iskanjem po celotnem besedilu je najenostavnejši način organiziranja znanja, če pri tem o organiziranju sploh lahko govorimo. Baza znanja je torej kar zbirka dokumentov v naravnem jeziku, pri čemer dokumenti lahko vsebujejo poleg teksta tudi slike, grafe, preglednice in podobno. Dokumenti morajo imeti vsaj naslov, da je sploh možno zaporedno pregledovanje, lahko pa vsebujejo tudi povezave na druge dokumente (Siegl, 2002).

Takšen način organiziranja baze je primeren le za baze z majhnim številom dokumentov, za ostale pa je potrebno imeti nek mehanizem iskanja. Iskanje po bazah z večjim številom dokumentov navadno deluje na osnovi indeksov, ki se gradijo avtomatsko ob spremembi vsebine ali ob določenih časovnih intervalih. To omogoča relativno hitro iskanje znotraj indeksa, ki kaže na ustrezne dokumente. Na tak način deluje večina spletnih iskalnikov (Google, AltaVista, HotBot in ostali) ter tudi več orodij za upravljanje baz (Lotus Notes).

Mehanizmi iskanja po celotnem besedilu delujejo podobno kot iskalniki po svetovnem spletu. Da bi bili z najdenimi rezultati zadovoljni, so potrebne dodatne lastnosti iskanja, ki pomagajo natančneje definirati, kaj iščemo. Sem spadajo (Siegl, 2002):

- ❑ iskanje po korenih besed (po ključnih besedah),
- ❑ možnost iskanja fraze,
- ❑ možnost uporabe logičnih izrazov IN, ALI, NE z možnostjo določanja prioritete z oklepaji,
- ❑ izpuščanje nepomembnih besed (stop word), na primer veznikov in predlogov in
- ❑ uporaba tezavra za sinonime ali besede različnih jezikov (če so dokumenti v različnih jezikih).

Poleg naštetega pa je pomembno, da ima uporabnik vpliv na razvrščanje rezultatov, ki je lahko (Siegl, 2002):

- ❑ glede na pogostost pojavljanja besed, ki jih iščemo, glede na mesto pojavljanja ali glede na poseben nabor besed,
- ❑ glede na informativnost besed (besede, ki se v bazi manjkrat pojavijo, imajo večjo informativnost,
- ❑ glede na podobnost s trenutnim dokumentom,
- ❑ glede na meta podatke (datum, avtor, število dostopov do dokumenta in podobno) in
- ❑ možnost iskanja znotraj iskanja.

V sistemih podpore uporabnikom so take baze znanja dokaj pogoste. Takšen način je hitra, poceni in tudi dovolj dobra rešitev. Tudi sama sem tak način uporabila pri izdelavi baze znanja za pomoč uporabnikom Software Distributor produkta. Izgradnja baze je podrobneje predstavljena v naslednjem poglavju.

5 Izgradnja sistema za upravljanje z znanjem s pomočjo orodja Rational ClearDDTS

Na trgu je več različnih orodij, ki služijo in pomagajo podjetjem pri upravljanju z znanjem. Od podjetja samega in od namena uporabe je odvisno, za nakup katerega orodja se bo podjetje odločilo. V našem primeru, vsaj menim, je v veliki meri vplival način dela v ameriškem Hewlett Packardu. Tako so se pri nas odločili za uporabo programskega orodja Rational ClearDDTS. Kljub temu, pa po nekaterih podatkih v podjetju Hewlett Packard uporabljajo 5 različnih, po funkcionalnosti sorodnih orodij, ki služijo upravljanju znanja.

Rational's Distributed Defect Tracking System (Rational ClearDDTS™) je orodje, ki omogoča sledenje in upravljanje s problemi (defects/bugs) in predlogi za izboljšavo produkta (enhancement requests) skozi celotno življenjsko dobo projekta.

5.1 Predstavitev

Kot že rečeno, je ClearDDTS namenjen spremljanju problemov in predlogov za izboljšavo produkta skozi življenjsko dobo produkta. Njegova uporaba omogoča inženirjem hitrejši vpogled v posamezne probleme, možne pomožne ali alternativne rešitve (workarounds) ali rešitve.

ClearDDTS razporedi probleme v projekte (projects) in razrede (classes). Vsak projekt se identificira z edinstvenim imenom in se navezuje na določen razred (izbira razreda in projekta je prikazana na sliki 10). Vsak projekt je tudi vezan na svoj "domači" sistem. To pomeni, da se vsi problemi, ki se nanašajo na določen produkt, nahajajo na tem sistemu. Uporabniki tega produkta kjerkoli po svetu lahko vpišejo nov problem. ClearDDTS nato preko elektronske pošte obvesti razvijalce produkta o prispelem problemu.

Razvijalci programske opreme in testni inženirji uporabljajo ClearDDTS na podoben način. Vanj vpišejo vsak interno najden defekt ali pomanjkljivost v delovanju produkta. ClearDDTS omogoča nadzor nad številom aktivnih problemov, postopki ponovitve problema, koraki reševanja in podobno.

Slika 10: Izbira razreda in projekta

Select Project, Class & State

Which Class

Class Name:

Which Projects & States

Projects	States
MPP-testcal Test ONLY project for MPP	S Submit
calls This project tracks customer telephone calls.	N New
ita.calls IT-Administration call tracking project	A Assign
sd.calls Software Distributor call tracking project	O Open
	R Resolve
	V Verify

Clear Select All

Clear Select All Select Unresolved

Which Users

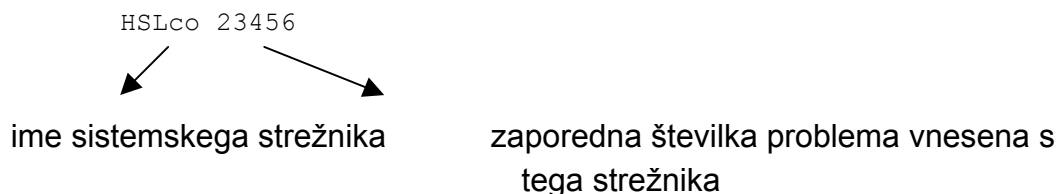
☒ Assigned To ☐ Submitted By ☒ Users: ☐ Myself

Users' Names:

OK Dismiss Save As Default Reload From Default Help

Vir: Hermes SoftLab, 2002

Problemi se med seboj razlikujejo po identifikacijskih številkah. Primer takšne številke je HSLco23456. Pomen identifikacijske številke je naslednji:



Od vpisa pa do razrešitve problema gre le-ta skozi več stanj (glej sliko 11). Pri vpisu v ClearDDTS je problem v stanju "N" (new). ClearDDTS administrator projektnega teama nato glede na problemsko področje problem dodeli razvojnemu inženirju v obdelavo. Problem je sedaj v stanju "A" (assigned). Inženir dobi preko elektronske pošte obvestilo, da mu je bil dodeljen problem v obdelavo. S pričetkom obdelave problema se njegovo stanje spremeni v "O" (open). Ob njegovi razrešitvi gre v stanje "R" (resolved), nato v stanje "V" (verified). Ponavadi mora nek drug inženir preveriti, če je opisan problem res rešen. Vsi verificirani problemi so vključeni v novo verzijo produkta. Ob izidu nove verzije tudi preidejo v stanje "Z" (released).

Slika 11: Stanja, med katerimi lahko izbiramo.

The screenshot shows a window titled "Record" with a menu bar containing "Modify", "Change_State", "Links", "CM", and "Help". A dropdown menu is open under "Change_State", listing the following options: "S Submit", "N New", "A Assign", "O Open", "R Resolve", "V Verify", "C Continue", "E End_call", "D Duplicate", "P Postpone", "U Update_time", "B Batch_update", and "F Forward". The "A Assign" option is currently selected.

The main area of the window displays the following information:

Call ID:	HP-UX	Submitted	020115
Status:	version A.02.10.07	Assigned	020115
	3 enclosures	Opened	020115

Below the call information, there are several sections:

- PROBLEM INFORMATION**
 - Call type: Incoming call
 - Call result: Contacted person
 - Severity: 3
- ASSIGNMENT INFORMATION**
 - Engineer: matejab
 - Est. fix date: 020115
- CALL STATISTICS**
 - Phone time: 60 minutes
 - Other time: 0 minutes
 - Total time: 60 minutes
 - Last Contacted: 020115 100026

At the bottom, there is a section for "SUPPORT INFORMATION" with the following details:

- Support: Mateja Bezjak
- Branch: Hermes Softlab
- Phone: +386-1-5865-480
- Address: @dirlija

Vir: Hermes SoftLab, 2002

V primeru, da se management odloči, da katerega problema ne bo rešil, se le-ta zapre in preide v stanje "C" (closed). Možno stanje problema je še "P" (postponed), ki se dodeli problemu takrat, ko management meni, da rešitev ne bo na voljo v novi verziji produkta ampak šele v njegovi naslednici.

5.2 Uporaba v podjetju

Vsak projektni team v podjetju pri razvoju softvera uporablja ClearDDTS. Njegova uporaba pa se med teami nekoliko razlikuje. Vsi teami, ki razvijamo in podpiramo Hewlett Packardove produkte imamo bolj ali manj poenoten način uporabe ClearDDTS orodja. Še vedno obstajajo razlike, vendar je težnja managementa in inženirjev, da bi bili postopki med seboj enaki.

Glede na to, da delam kot inženir za podporo strankam pri uporabi Hewlett Packard OpenView Software Distributor (SD) produkta, bo prikaz uporabe temeljil na področju mojega dela.

Software Distributor je produkt, ki omogoča centralen nadzor softvera nad sistemi razpršenimi po različnih lokacijah, pa tudi po celem svetu. Administratorju tudi omogoča omejiti dostop uporabnikom do sistemskih programov. To lahko določi s pomočjo varnostnih ukazov, ki jih Software Distributor omogoča (swacl). Izvirno je produkt narejen za operacijske sisteme HP-UX, kasneje pa je bil prirejen uporabi tudi

na drugih operacijskih sistemih, kot so Windows, AIX in Solaris. Najbolj znana ukaza, po katerih se produkt spozna sta *swinstall* in *swremove*.

Software Distributor team ima svoj elektronski naslov, preko katerega poteka komunikacija s strankami. Kadarkoli se pojavi stranki nov problem, ki ga sama ali pa s pomočjo prve stopnje pomoči Hewlett Packard centrov po svetu ne zna rešiti, se obrne na nas. Mi predstavljamo drugi in hkrati tretji nivo podpore strankam. Pod drugi nivo spadajo vsi problemi, ki jih lahko inženirji za podporo strankam sami rešijo (naj bo to na podlagi znanja, izkušenj ali brskanja po bazi starih problemov), pod tretji nivo pa spadajo problemi, ki zahtevajo pomoč razvojnih (CPE) inženirjev. V slednjem primeru inženir za podporo strankam problem glede na področje dodeli razvojnemu inženirju v nadaljnjo obravnavo.

Sodelovanje s strankami oziroma Hewlett Packard centri je pretežno v obliki elektronske pošte. Le redko se zgodi, da stranke prijavijo problem po telefonu. Inženirji v Hewlett Packard centrih ponavadi nimajo specifičnega znanja o posameznih produktih, saj morajo nuditi strankam pomoč pri vseh Hewlett Packard OpenView produktih. Teh produktov pa je veliko. Hewlett Packard inženir mora vedeti kateremu OpenView produktu problem pripada. Njegova naloga je potem zbrati čimveč koristnih podatkov o problemu ter jih posredovati drugemu nivoju pomoči oziroma nam.

Vsak problem, ki ga naše stranke prijavijo je vpisan v ClearDDTS bazo. SD podpora strankam uporablja razred *calls* in projekt *sd.calls*. Vsak problem pri vnosu v bazo dobi svojo identifikacijsko številko (npr. HSLco30345). Pri vnosu je potrebno zajeti podatke, kot so:

- ❑ naslov,
- ❑ ime stranke,
- ❑ verzija SD produkta,
- ❑ prioriteto,
- ❑ operacijski sistem, na katerem se problem pojavlja in
- ❑ podroben opis problema.

Prispeli problem nato inženir za podporo strankam začne reševati ter njegovo stanje spremeni v "O". Vsa nadaljnja komunikacija med inženirjem in stranko se kot priloga (enclosure) doda v bazo, kar prikazuje slika 12.

Slika 12: Odprti klic v ClearDDTS bazi za projekt *sd.calls*

Xddts

FileSelectMetricsOptionsSpecial QueriesTo Do ListHelp

'calls' Records [0] [sd.calls]2 records

1 0 Sv3 020108 Luc PioroSD-OV 2.11 on HPUX 11.0, wrong status re +4
2 0 Sv3 020103 Bill IngramProblem with swpackage.compress=true opt +3

SubmitCommitCloneRefreshClearNextPrevPrint...Gripe

Record

ModifyChange_StateLinksCMHelp

Call ID: HSLco31395HP-UXSubmitted 011227
Status: OPEN callversion A.02.10.07Assigned 011227
3 enclosuresOpened 011227
"Problem with swpackage.compress=true option"

CALLER INFORMATIONPROBLEM INFORMATION

Name: Bill IngramCall type: Incoming call
Cphone:Call result: Contacted person
MS:Severity: 3
E-mail:ASSIGNMENT INFORMATION
Other:Engineer: matejab
Other:Est. fix date: 011227
Company: Agilent
Phone:
FAX:
Address:
Address:
Address:
Address:

SUPPORT INFORMATIONCALL STATISTICS

Support: Mateja BezjakPhone time: 300 minutes
Branch: Hermes SoftlabOther time: 0 minutes
Phone: +386-1-5865-480Total time: 300 minutes
Address: @dirlijaLast Contacted: 011227 083155

Enclosures






HistoryConversation bill:21/12/01 to bill:24/12/01

Vir: Hermes SoftLab, 2002

Naš način shranjanja podatkov je takšen, da vsako poslano in dobljeno elektronsko pošto shranimo v posebni prilogi. To kasneje drugim inženirjem omogoča lažji pregled dogajanja in faze v procesu reševanja samega problema. Ta metoda se je že pokazala kot uspešna pri prenosu znanja na nove inženirje.

Razlikujemo med problemi in defekti. Problem je vsak prihajajoč klic; vsako poslano elektronsko poročilo na naš elektronski naslov za podporo produkta. Problemi so shranjeni v razredu *calls*. Defekti (imenovani Change Requests ali CRs) se vpisujejo v razred *Supernova*. Nanašajo se na napake v kodi samega produkta, medtem ko se problemi lahko nanašajo na napačno uporabo produkta, vprašanja stranke in podobno.

Dokler problem ni rešen, se nahaja v "O" stanju. Ob njegovi rešitvi in soglašanju stranke ga lahko zapremo. Iz "O" stanja preide v "R" stanje. Takšen je splošni postopek za probleme, ki se ne nanašajo na defekte produkta, ampak na probleme strank pri njegovi uporabi. V primeru, da stranka odkrije defekt in ga mi verificiramo, ga mora stranka prijaviti v ClearDDTS bazo. V takšnih primerih uporabljamo razred *Supernova* in ime trenutno tekočega projekta (npr. *HSL_Gator*). Vsaka verzija produkta ima namreč svoje ime. Hkrati inženir za podporo strankam klic zapre in doda prilogo, ki kaže na identifikacijsko številko odkritega defekta. Ime priloge je številka vpisanega defekta.

Pri prehajanju defekta v različna stanja je potrebno problemu dodati posebne priloge (enclosures). Ob prijavi defekta je potrebno v prilogo *Problem* napisati:

- ❑ kaj je problem,
- ❑ kako se pojavlja,
- ❑ kdaj se pojavlja (občasno, stalno) in
- ❑ verzijo produkta, ki jo stranka uporablja.

V primeru, da obstaja alternativna rešitev (workaround) problema mora inženir podrobne korake opisati v prilogi *Workaround*. Takšnih rešitev je lahko več. Potrebno je navesti vse možne rešitve, na podlagi katerih se lahko stranka odloči, če ji katera odgovarja. Dosti primerov je takšnih, kjer alternativna rešitev ne obstaja. Takrat seveda tudi priloga *Workaround* ne obstaja.

Ko je defekt prijavljen, morajo inženirji preučiti, v kakšnem časovnem okviru bi ga bilo možno odpraviti. Odvisno od časa in zahtevnosti samega defekta se management odloči ali bo vložil sredstva v njegovo odpravo. V veliki meri vpliva na takšne odločitve prioriteta defekta in onemogočanje določene funkcionalnosti, ki jo produkt prinaša. Odločitev management zabeleži v prilogi *Marketing*.

Prilogo *Fix* dodamo defektu takrat, ko je bil popravljen. Istočasno preide defekt v stanje "R" (resolved). Priloga *Fix* mora vsebovati podatke o reševanju defekta, kaj je bilo potrebno spremeniti da stvar deluje. Hkrati s prilogo *Fix* se dostikrat pojavlja priloga *PatchInfo*, ki kaže na to, da je rešitev defekta vključena v sistemskem oziroma produktne popravi (patchu). Vsebina *PatchInfo* priloge je številka sistema popravljenega (npr. PHCO_23456).

V primeru, da se management odloči, da rešitev nekega defekta ne bo podpiral, se le-ta zapre. Razlogi za takšno odločitev morajo biti vpisani v prilogo *Closed*.

5.3 Shranjevanje podatkov

V ClearDDTS bazo shranjujemo vse podatke, ki so potrebni za razumevanje problema in za iskanje njegove rešitve. Problemi se dostikrat ponavljajo, zato so inženirju lahko v veliko oporo stari problemi.

Ob vnosu novega klica v ClearDDTS bazi je inženir povprašan o verziji produkta, operacijskem sistemu, opisu problema in podatkih o stranki. Vsi ti podatki so zelo pomembni pri iskanju rešitve. Inženir nato vnese tudi vse izmenjane elektronske pošte med nami in stranko. Tako imamo evidenco o vsakem koraku na poti od prijave problema do njegove rešitve. Takšen način shranjevanja podatkov o problemih omogoča vsemu teamu, da si s pomočjo baze pomaga pri različnih problemskih situacijah. Je pa to tudi odličen vir znanja za vsakega novega inženirja, ko se začne spoznavati s produktom.

Prav tako uporabljamo ClearDDTS bazo za evidenco vseh defektov. Njena uporaba je kar precej različna od uporabe baze pri problemih. Najbolj se razlikuje uporaba prilog.

5.3.1 Iskanje po bazi

Možnih je več načinov iskanja po bazi. Iščemo lahko po samo enem ali pa več projektih hkrati. Iščemo lahko vse probleme, ki jih je imela neka stranka v določenem časovnem obdobju z našim produktom. Iskanje pa je lahko tudi bolj splošno. V tem primeru izberemo iskalno polje (field selection) "enclosure_text" in vpišemo iskano besedo. Rezultat iskanja bodo vsi tisti problemi, katerih prologa vsebuje iskano besedo. Pomanjkljivost takšnega iskanja je veliko število dobljenih rezultatov, ki se tudi dostikrat ne nanašajo na rešitev našega problema. Zato je bolje, če iskanje omejimo.

Slika 13: Iskanje po bazi s pomočjo grafičnega vmesnika

Complex Query & Sort

Search String

I

Selected Projects: sd.calls

Selected States: SNAORVCEDPUBF

Sorting Fields:

Clear String Delete Item Insert In History Get First From History Select Project, Class & State... Select Sorting Criteria...

Field Selections

patchavail	phone	postponed_on	postponed_out	previousss	priority
problem_type	product	project	qaengineer	released_on	released_version
releaser_id	relvehicle	request_type	requestedBy	res_period	resolution
resolved_on	resolved_out	resolver_id	return_path	security_token	severity
severitycr	showstopper	software	sr_id	status	subcomponent

Operator Selections

...	>	>...	has	&&	(
...	<	<...	hasno		)

Selection Value

History

OK Dismiss Save As Default Reload From Default Delete First History Help

Vir: Hermes SoftLab, 2002

ClearDDTS omogoča iskanje preko grafičnega vmesnika (slika 13), kot tudi uporabo SQL stavkov s pomočjo ukaznih vrstic. Pri iskanju je potrebno vnesti ime projekta. V našem primeru je to *sd.calls*. Označiti je tudi potrebno, katera stanja problemov nas zanimajo. Če hočemo iskati samo med problemi, ki so trenutno odprti, izberemo stanje "O".

Slika 14: Iskalni niz

Search String	
resolved_on >= "000101" && enclosure_name == "PROBLEM"	
Selected Projects	sd.calls
Selected States	R

Vir: Hermes SoftLab, 2002

Slika 14 kaže primer iskanja vseh rešenih klicev (Klici v stanju "R" resolved), ki vsebujejo prilogo z imenom PROBLEM. Klici so morali biti rešeni po 1. januarju 2000. Iskalni niz se nanaša na project *sd.calls*.

Slika 15: Izbiranje iskalnih polij

Field Selections					
non_call_time	notify_submitter	numofdups	opened_on	opened_out	origin
os	os_language	os_version	osvendor	parent_id	parents
patchavail	phone	postponed_on	postponed_out	previousss	priority
problem_type	product	project	qaengineer	released_on	released_version
releaser_id	relvehicle	request_type	requestedBy	res_period	resolution

Vir: Hermes SoftLab, 2002

V prejšnjem primeru sem izbrala le dve iskalni polji; ime priloge in datum rešitve klica. ClearDDTS omogoča celo vrsto polj (slika 15), s pomočjo katerih lahko iščemo po njegovi bazi. Lahko se omejimo samo na ime stranke, prioriteto in podobno. Zgornja slika prikazuje en del polj med katerimi lahko izbiramo.

Uporaba grafičnega uporabniškega vmesnika je dokaj enostavna in tudi zelo primerna za vsakega uporabnika. Zlasti prav pride tistim, ki nikoli niso uporabljali SQL stavkov ali pa so pri njihovi uporabi bolj okorni.

5.3.2 Oblikovanje poročil

Uporaba ClearDDTS je projektnemu vodji in njegovim nadrejenim v veliko pomoč pri oblikovanju mesečnih in tedenskih poročil. S pomočjo poročil dobijo vpogled v dogajanje na projektu in potrebne podatke za pridobivanje statističnih podatkov potrebnih za strateške odločitve. S pomočjo ClearDDTS v mojem projektnem teamu generiramo tedenska in mesečna poročila o številu na novo prispelih klicev, številu skupaj odprtih klicev, o odzivnem času na prispeli klic. Slika 16 prikazuje primer tedenskega poročila.

Slika 16: Primer tedenskega poročila

1. Calls statistics:

1.1 Calls opened this week - from 01/11/2002 to 01/18/2002 TOTAL: 2

HSLco31618 - Different behaviour on error when using 1 or more targets.

SUBMIT DATE: 01/11/2002
INITIAL RESPONSE(min): 60
PRIORITY: 3
OWNER: matejab
CUSTOMER: Oberfinanzdirek
STATUS: O
TIER3 SUBMIT:

HSLco31666 - swpackage question

SUBMIT DATE: 01/15/2002
INITIAL RESPONSE(min): 60
PRIORITY: 3
OWNER: matejab
CUSTOMER: HPO R&D
STATUS: O
TIER3 SUBMIT:

1.2 Old opened calls - up to 01/11/2002 TOTAL: 2

HSLco31395 - Problem with swpackage.compress=true option

SUBMIT DATE: 12/27/2001
INITIAL RESPONSE(min): 300
PRIORITY: 3
OWNER: matejab
CUSTOMER: Agilent
STATUS: O
TIER3 SUBMIT:

HSLco31503 - SD-OV 2.11 on HPUX 11.0, wrong status report of completed job in SD-OV

SUBMIT DATE: 01/08/2002
INITIAL RESPONSE(min): 180
PRIORITY: 3
OWNER: matejab
CUSTOMER: Oberfinanzdirek
STATUS: O
TIER3 SUBMIT:

1.3 Calls resolved this week - from 01/11/2002 to 01/18/2002 TOTAL: 0

Vir: Hermes Softlab, 2002

Kot piše v tedenskem poročilu na sliki 16, smo v tednu od 11.1. do 18.1. 2002, preko elektronske pošte prejeli dva problema. Oba sta bila tretje prioritete, kar pomeni da smo imeli 24 ur časa za odgovor. Razlikujemo med prioritetami problemov:

- ❑ prioriteta 1 ali eskalacija,
- ❑ prioriteta 2 ali kritična prioriteta in
- ❑ prioriteta 3 ali normalna prioriteta.

Za normalno prioriteto ima inženir na voljo en dan za odgovor, za kritično prioriteto ima na voljo samo štiri ure, medtem ko ima pri eskalaciji na voljo samo eno uro.

Prav zaradi statistike in nadzora nad časom je v poročilu omenjen *Initial response* ali po slovensko odzivni čas na prispeli problem.

Tedenska poročila se generirajo vsak teden in so poslana preko elektronske pošte projektnemu vodji in inženirju za podporo strankam. Pri oblikovanju smo poleg ClearDDTS programskega orodja uporabili tudi ClearDDTS SQL, shell in Perl skripte.

Mesečna poročila se od tedenskih kar precej razlikujejo. Največji pomen pri tej statistiki ima odzivni čas na prispeli problem. Mesečno poročilo izpiše vse prispele probleme, razvrščene glede na prioriteto in njihove odzivne čase. Hkrati izpiše v koliko odstotkih je bil odzivni čas v okviru dovoljenega.

Slika 17 prikazuje primer mesečnega poročila.

Slika 17: Primer mesečnega poročila

Monthly Expert Center report

1 Response time by priority for every calls submitted from 12/19/2001 to 01/18/2002

Priority 1

Priority 2

Priority 3

HSLco31618	01/11/2002	IRT in minutes	60
HSLco31666	01/15/2002	IRT in minutes	60
HSLco31395	12/27/2001	IRT in minutes	300
HSLco31503	01/08/2002	IRT in minutes	180

2 Percentage of calls that met the response time requirement

Priority 1: Info not available

Priority 2: Info not available

Priority 3: 100 %

Vir: Hermes Softlab, 2002

5.4 Gradnja baze znanja

Hewlett Packard centri nimajo omogočenega dostopa do naše ClearDDTS baze. Baza je interna in je v večji meri dostopna samo zaposlenim v podjetju. Seveda obstajajo izjeme, ki pa so odvisne od imena razreda in njegovega namena. Prav zaradi sodelovanja s Hewlett Packardom so nekateri razredi vidni znotraj obeh podjetij.

Glede na to, da inženirji v Hewlett Packard centrih nimajo dostopa do baze, je potrebno čimveč znanja o produktu posredovati tudi njim. Naloga Hewlett Packard centrov je prestoreti čim večje število problemov ter nam posredovati le težje, zahtevnejše probleme. Dostikrat se namreč dogaja, da so Hewlett Packard centri le posredniki med nami in zunanjimi strankami. Da bi se temu izognili je potrebno določeno znanje o produktu prenesti tudi nanje. Poleg tečajev so lahko pomemben vir znanja tudi stari problemi in njihovi načini odpravitve. Moj namen je pripraviti bazo znanja, ki bo vsebovala probleme razvrščene po produktnih področjih.

Končna baza mora biti dostopna preko interneta, torej v HTML formatu. ClearDDTS se nahaja na UNIX strežniku, zato sem se odločila s pomočjo Korn shell skripte poiskati spisek vseh problemov, ki bodo vsebovani v bazi znanja. Vsi problemi nikakor ne pridejo v poštev, zato jih je bilo najprej potrebno razvrstiti. Glede na to, da v teamu že kar nekaj časa beležimo težje probleme s prilogo PROBLEM, je bilo razvrščanje lažji del naloge. Priloga PROBLEM je pri vseh izbranih problemih zadnja priloga. ClearDDTS omogoča iskanje po imenu priloge, kar olajša izločitev vseh problemov s prilogo PROBLEM.

Enotnost beleženja je zagotovljena z vzorcem priloge PROBLEM. Le-ta je prikazana na sliki 18.

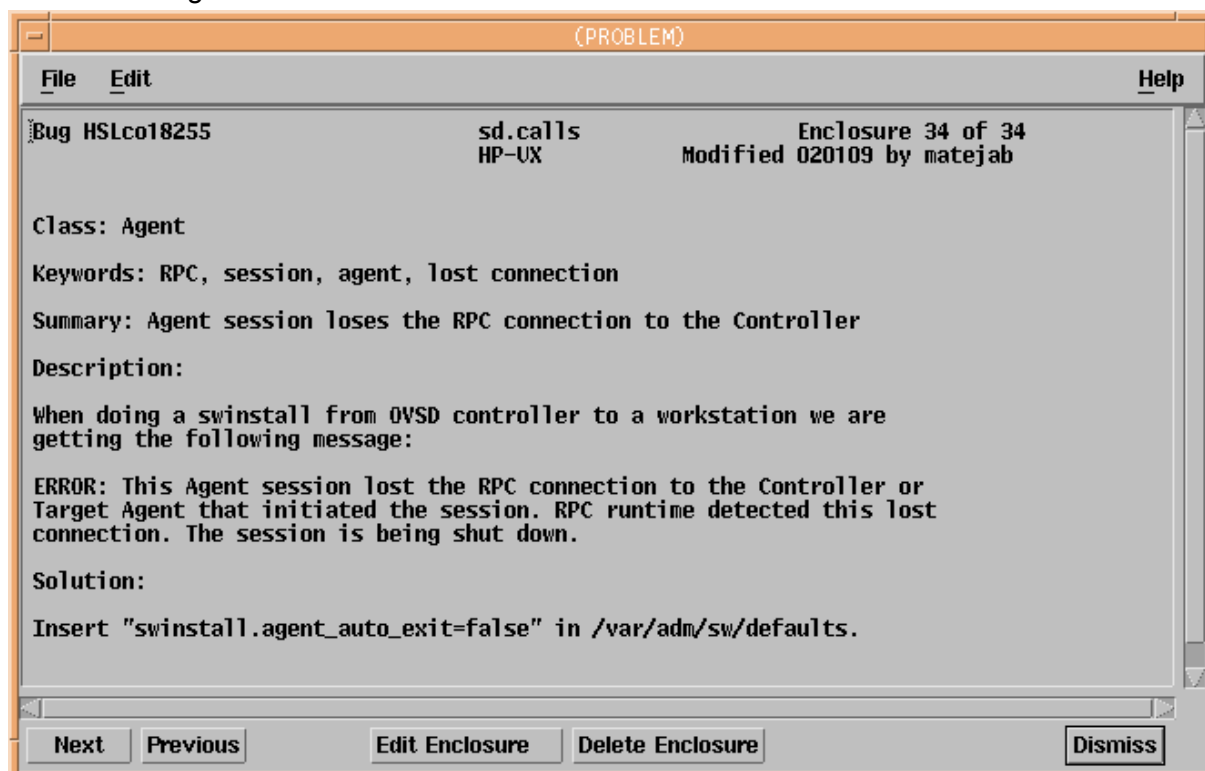
Slika 18: Vsebina priloge PROBLEM

```
Class: Installation, GUI, Controller, Agent, Security, PC Controller,  
Licensing, NT Agent, Other (leave one, delete other text under Class)  
  
Keywords:  
  
Summary:  
  
Description:  
  
Solution:  
  
Related SR:
```

Vir: Hermes SoftLab, 2002

Vsak uporabnik ClearDDTS orodja v teamu ima v svojem direktoriju prilogo PROBLEM, ki jo uporabi, če želi rešitev nekega problema prikazati drugim uporabnikom. Slika 19 kaže končno podobo priloge PROBLEM v ClearDDTS bazi.

Slika 19: Priloga PROBLEM v ClearDDTS bazi



Vir: Hermes SoftLab, 2002

Vse zapise v ClearDDTS bazi s prilogo PROBLEM je nato potrebno zbrati v posebno datoteko, iz katere je mogoče razpoznati identifikacijske številke vseh klicev z odgovarjajočo prilogo. To sem storila s pomočjo SQL stavka.

ClearDDTS je dokaj majhna baza z le nekaj standardnimi tabelami. Vsaka tabela se sestoji iz velikega števila stolpcev, ki vsebujejo specifične podatke o vnesenem problemu. Slika 20 kaže poenostavljen primer tabele (vsi stolpci niso prikazani):

Slika 20: Primer tabele

ime tabele	→	defects		
stolpci	→	identifier	Headline	Engineer
	→	HSLco00001	Defect record 1	Roberts
	→	HSLco00002	Defect record 2	Smith
	→	HSLco00003	Defect record 3	Jones

Vir: Rational Software, 1999, str. 34.

ClearDDTS SQL stavki uporabljajo le en del standardnih SQL stavkov, saj nekateri ukazi niso podprti. Sem spadajo na primer gnezdeni stavki, virtualne tabele, operator NOT je lahko uporabljen samo skupaj z Like (NOT LIKE) ali z Null (NOT NULL).

Kljub temu, da je SQL nekoliko omejen, v več kot zadostni meri zadostuje vsem našim iskalnim nizom.

V datoteko, ki sem jo poimenovala, *ddtssql.kdb* sem napisala SQL stavek, ki iz tabel *enclosures* (priloge) in *defects* išče identifikacijske številke klicev s prilogo PROBLEM. Pripadajoči projekt je *sd.calls*. Slika 21 prikazuje zapis v *ddtssql.kdb* datoteki.

Slika 21: SQL stavek *ddtssql.kdb* datoteki

```
export PATH=$PATH:/ddts/bin
ddtssql -noheader -f - << _E_O_F
SELECT D.identifier
FROM enclosures E, defects D
WHERE E.name = 'PROBLEM'
AND E.identifier = D.identifier
AND D.project = 'sd.calls';
_E_O_F
```

Vir: Hermes SoftLab, 2002

Naslednji korak je bil zbrati identifikacijske številke problemov v posebno datoteko. To datoteko sem poimenovala indeks. Za vsak problem iz index datoteke je nato bilo potrebno razpoznati pripadajoče besedilo oziroma podatke, ki jih nosi s seboj. V ClearDDTSu za to obstaja ukaz *dumpbug* (slika 22), katerega naloga je natisniti in oblikovati zapis problema.

Slika 22: Vpogled v uporabo ukaza *dumpbug* (man dumpbug)

```
DUMPBUG(1)                                Rational Software                                DUMPBUG(1)
November 6, 1997

NAME
  dumpbug - format and print a defect record

SYNOPSIS
  dumpbug [-fmnrhR] [-i ctemplate] [-t template] [id ... | -b bugbox]

DESCRIPTION
  Use dumpbug to print defect records to the standard output. The
  record ids of the defect record(s) can be supplied on the command
  line, from a specified bugbox file by specifying the -b option. If no
  -b option and no ids are specified, the list of record ids to be
  printed is read from the standard input.

  By default, for each id dumpbug writes a row of asterisks, the record
  formatted according to a default template, followed by any enclosures
  attached to the defect record, followed by a formfeed if there are any
  more ids to process. An enclosure is a named, timestamped,
  encapsulated section of text that can be attached to a defect record.

  -f      suppresses the formfeeds that are normally printed before the
          output for the second and subsequent defect records.

  -i ctemplate
          specifies a directory, located in the class directory associated
          with the bug, that is to be used for finding template files to
          print the bug. The directory must be in the associated class
          directory (~ddts/class/<classname>/ctemplate) and the template
          files in that directory must be named for the states associated
          with that bug class. For example, to print a one line
          description of a bug exactly like bugs(1) and xddts(1) you could
          do:

              dumpbug -i user.index XXXxx12345

  In this case dumpbug would find the class and state of the bug.
  If the class were "software" and the state was "0" then dumpbug
  would use the ~ddts/class/software/user.index/0 template file to
  print the bug.
```

Vir: Hermes SoftLab, 2002

Ukaz sem uporabila v skripti *dumpbug.kdb*, kjer sem na domačem direktoriju s pomočjo skripte zbrala potrebne informacije za nadaljnje delo.

Slika 23: *dumpbug.kdb* skripta

```
#!/usr/bin/ksh

INDEX_FILE=/home/sdsupp/kdb/index

# find all bug ids with PROBLEM enclosure
/home/sdsupp/kdb/ddtssql.kdb 2>/dev/null 1>$INDEX_FILE

cat $INDEX_FILE |
while read BUGID
do
    /ddts/bin/dumpbug $BUGID > /home/sdsupp/kdb/$BUGID
done
```

Vir: Hermes SoftLab, 2002

Rezultat skripte so problemi razporejeni po datotekah z imenom identifikacijske številke klica. Trenutno je okrog 70 datotek, torej 70 klicev, zapisanih v tekstovnih datotekah. Vse datoteke je sedaj potrebno pretvoriti v HTML obliko.

Stran, ki sem si jo zamislila bo vsebovala okvire in bo izdelana s pomočjo Microsoft FrontPage urejevalnika. Zgornji okvir bo vseboval naslov Software Distributor's Knowledge Database, logotip podjetja Hewlett Packard in skupine produktov OpenView, kamor spada tudi Software Distributor. Software Distributor se deli na več področij:

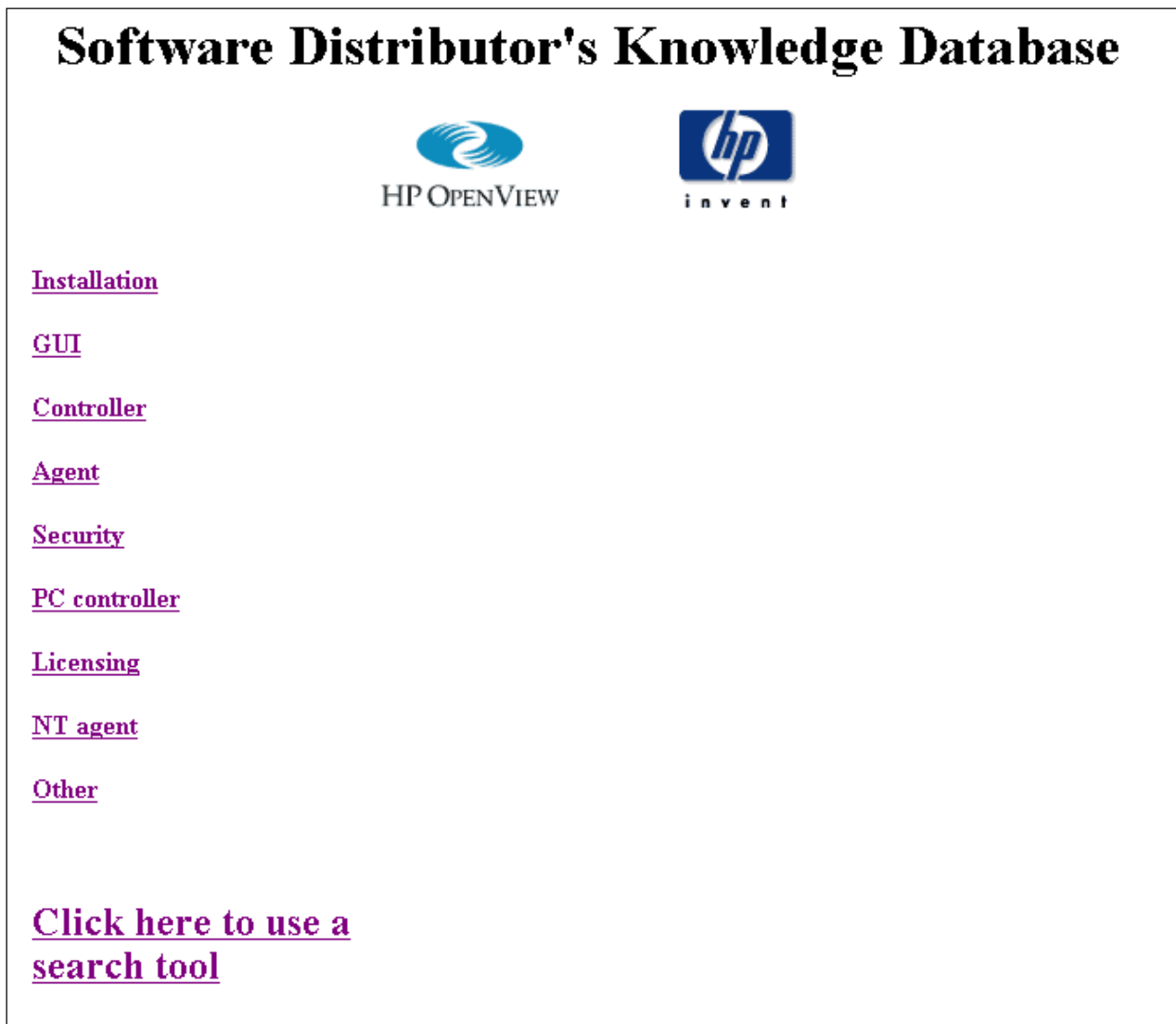
- ☐ Installation
- ☐ GUI
- ☐ Controller
- ☐ Agent
- ☐ Security
- ☐ PC controller
- ☐ Licensing
- ☐ NT agent

Ta področja bodo del levega okvira strani in bodo vsebovala povezave na probleme, ki spadajo pod določeno področje. Včasih se zgodi, da se ne da točno določiti pod katero področje spadata problem. V ta namen bom dodala področje Other (razno).

Kljub temu, da področja olajšajo iskanje rešitev, je potrebno stran dopolniti z iskalcem, ki bo iskal po ključnih besedah. Ključne besede je inženir vnesel v prilogo PROBLEM.

Slika 24 prikazuje izgled zamišljene uvodne strani baze znanja.

Slika 24: Uvodna stran baze znanja Software Distributor produkta



Vir: Hermes SoftLab, 2002

Prikazana sta levi in zgornji okvir. Vsebina desnega okvira bodo klici razvrščeni po identifikacijskih številkah in hkrati po področjih produkta Software Distributor. Le-te je potrebno najprej generirati.

Najpomembnejšo skripto, s pomočjo katere najdemo vse klice v projektu *sd.calls* in s statusom "R" (resolved) ter prilogo PROBLEM, sem poimenovala *kdb*. Za ime sem izbrala kar kratico od imena "Knowledge Database".

Glavna naloga te skripte je poiskati odgovarjajoče klice in hkrati shraniti kopijo izpisa tekstovne datoteke v Rational ClearCase bazi. Rational ClearCase je orodje, ki ga v podjetju uporabljamo za shranjevanje podatkov, predvsem pomembnih datotek, dokumentov in vseh drugih za projekte pomembnih stvari. Tako je bilo potrebno

shraniti tudi bazo znanja, da bi v primeru okvare sistema ne izgubili pomembnih podatkov.

Kot že rečeno, je najpomembnejša skripta projekta izgradnje baze znanja za pomoč uporabnikom *kdb*. V njej sem uporabila ukaze, ki se nanašajo na shranjevanje podatkov v ClearCase bazi.

Celotna vsebina skripte *kdb* je prikazana v prilogi na koncu magistrskega dela. Skripta zajame posamezne klice in jih iz tekstovne oblike pretvori v HTML. To naredi za vsak klic posebej in za vsakega kreira datoteko z imenom identifikacijske številke klica (npr. HSLco23456, HSLco25631, HSLco25678, in podobno).

ClearCase je datoteke poimenoval *elementi*. Dobra lastnost ClearCase orodja je, da v primeru da se stanje oziroma zapis posameznega elementa ne spremeni, onemogoči shranjenje identične verzije elementa. Tako ne pride do podvajanj in nepotrebnega večkratnega shranjevanja istih verzij v drevesu elementa.

V primeru, da inženir vnese nov klic s prilogo PROBLEM v ClearDDTS bazo, je naloga *kdb* skripte kreirati nov element ter ga shraniti.

5.4.1 Zagon skripte

S tem, ko sem *kdb* skripto napisala, sem že bila bližje zastavljenemu cilju. Seveda, je bilo pred tem potrebno skripto še testirati in odpraviti morebitne napake. Prvi rezultati pa so že bili vidni.

Pravilno izvajanje skripte je moralo prinesiti v standardni izpis (standard output) datoteke z imenom identifikacijske številke klica z vsebino priloge PROBLEM. Direktorij na katerem se skripto pognala je prikazan na sliki 25.

Slika 25: Vpogled v direktorij, kjer sem pognala *kdb* skripto

```
$ ls
DATE                HSLco05724.htm      HSLco10794.htm      HSLco21255.htm
DDTS                HSLco05784.htm      HSLco11139.htm      HSLco21932.htm
HSLco03984.htm      HSLco05807.htm      HSLco11434.htm      HSLco26814.htm
HSLco03985.htm      HSLco05809.htm      HSLco11605.htm      HSLco27558.htm
HSLco04050.htm      HSLco05839.htm      HSLco11608.htm      c_hpovlogo.gif
HSLco04081.htm      HSLco05989.htm      HSLco11881.htm      cg_rpm.gif
HSLco04094.htm      HSLco06061.htm      HSLco11994.htm      frbottom.htm
HSLco04195.htm      HSLco06299.htm      HSLco12062.htm      frcont02.htm
HSLco04196.htm      HSLco06597.htm      HSLco12140.htm      frmmain02.htm
HSLco04401.htm      HSLco06635.htm      HSLco12533.htm      frmmain02.htm.old
HSLco04515.htm      HSLco06720.htm      HSLco12535.htm      frmmain02.htm.tmp
HSLco04607.htm      HSLco06986.htm      HSLco13594.htm      frtop.htm
HSLco04612.htm      HSLco08133.htm      HSLco13727.htm      frtop.htm.old
HSLco04687.htm      HSLco08134.htm      HSLco13897.htm      h_hplogo.gif
HSLco04819.htm      HSLco08165.htm      HSLco14237.htm      httpd-pid
HSLco04855.htm      HSLco08168.htm      HSLco15341.htm      kdb
HSLco04858.htm      HSLco08402.htm      HSLco15631.htm      kdb.clearcase
HSLco04879.htm      HSLco09026.htm      HSLco15895.htm      kdb.htm
HSLco04899.htm      HSLco09139.htm      HSLco15938.htm      kdb.orig
HSLco04947.htm      HSLco09149.htm      HSLco16575.htm      kdb.saved
HSLco05075.htm      HSLco09606.htm      HSLco16576.htm      logo.gif
HSLco05253.htm      HSLco09607.htm      HSLco16577.htm      main_template.htm
HSLco05375.htm      HSLco09913.htm      HSLco16764.htm      mkindex
HSLco05485.htm      HSLco10325.htm      HSLco16842.htm      ov_logo4.gif
HSLco05506.htm      HSLco10448.htm      HSLco16844.htm
HSLco05547.htm      HSLco10715.htm      HSLco18255.htm
HSLco05706.htm      HSLco10762.htm      HSLco20305.htm
$
```

Vir: Hermes SoftLab, 2002

Vidne so HTML datoteke, ki jih je *kdb* skripta generirala. Urejene so v pravilnem vrstnem redu, pa tudi pretvorba iz tekstovne v HTML obliko je bila uspešna. Slika 26 prikazuje primer zapisa iz ene od datotek, iz HSLco27558.

Call ID:HSLco27558

Date:011120

Class: GUI

Keywords: Job Title, swcopy, preview

Summary: swcopy doesn't go through the execution phase

Description:

Customer was running swcopy (in GUI) as a preview first. Then he selected the same job and run swcopy again (deselected preview tick). swcopy finished in the analysis phase and didn't go through the execution even if the preview tick was removed.

Solution:

The problem was caused by the Job Title string longer than 36 characters. This is a workaround, the fix will be available in the next SD-UX patch (spring 2002).

To limit the Job Title string edit:

```
/usr/lib/sw/ui/C/smc_sched.ui
```

```
-----
```

```
text_edit SMC_SCHED_TITLE
{
label "Job Title:"
reset false
width 40 # >> change this number to 36
optional true
help "sduhelp.hv:SDU-DIALOG-SCHEDULE"
}
} /* end dialog */
```

```
-----
```

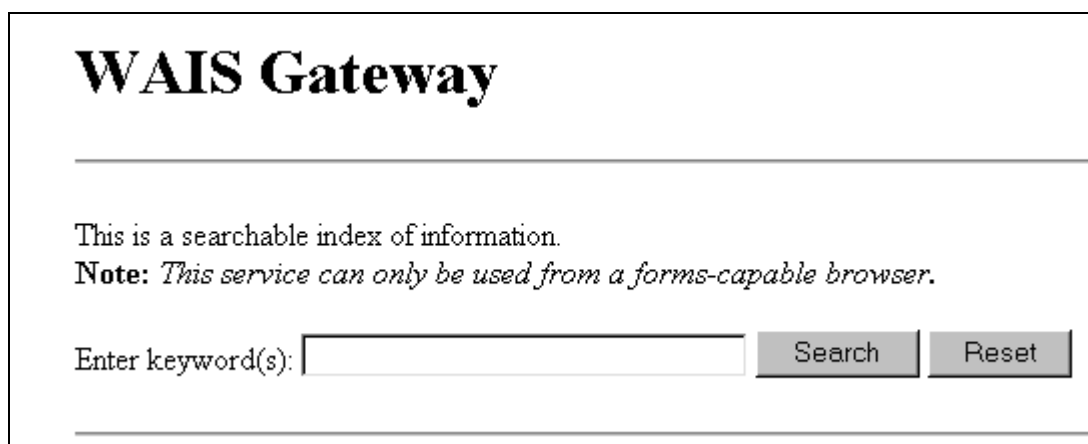
Related CR: JAGad83071, JAGad92694 (SD-UX)

Določeno predhodno znanje produkta Software Distributor ni pogoj za boljše razumevanje besedila v bazi znanja. Predpostavljeno pa je predhodno osnovno izobraževanje Hewlett Packardovih inženirjev, da so seznanjeni vsaj z ukazi, ki jih Software Distributor izvaja.

S tem, ko je generiranje datotek uspelo in je njihova vsebina na želenem direktoriju, je bil čas za moj naslednji korak. Namreč vsebino je bilo potrebno prikazati še v desnem okviru strani. Vsebino datotek je bilo potrebno vključiti v kdb.htm stran. Po nekaj poskusih je bila stvar rešena.

Stran je delovala v meji pričakovanj. Potrebno je bilo odpraviti določene pomanjkljivosti in napake. Potrebovala sem še iskalec, ki bi uporabnikom omogočal iskanje po bazi s pomočjo ključnih besed. Na internetu je moč najti več vrst iskalcev. Izbrala sem WAIS (Wide Area Information System, prikazan na sliki 27) in ga prilagodila naši strani.

Slika 27: Iskalec

The image shows a web browser window displaying the 'WAIS Gateway' search interface. At the top, the title 'WAIS Gateway' is in a large, bold, serif font. Below the title is a horizontal line. Underneath the line, the text 'This is a searchable index of information.' is displayed. Below that, a note in italics reads: 'Note: This service can only be used from a forms-capable browser.' At the bottom of the interface, there is a search form. It consists of a text input field with the placeholder text 'Enter keyword(s):', followed by a 'Search' button and a 'Reset' button. The entire interface is enclosed in a simple rectangular border.

Vir: Hermes SoftLab, 2002

WAIS iskalec po svoji bazi išče dokumente, ki odgovarjajo iskani besedi ali frazi. Dobljeni rezultati so ocenjeni med 1 in 1000, glede na pomembnost v dokumentu. Pomembnost je odvisna od tega, če se iskana beseda pojavlja v naslovu, pa tudi, kako pogosto se pojavlja v besedilu dokumenta. Rezultat, ki je najbližje našemu iskanju je ocenjen s 1000 točkami.

WAIS lahko uporablja naravne jezike, kajti iskalec vzame besedo ali frazo in išče dokumente, v katerih se le-ta pojavlja. Pri iskanju si lahko pomagamo z logičnimi izrazi AND, OR, NOT in ADJ. Uporabnik lahko, kadar hoče iskati po korenu besede, uporabi tudi t.i. jokerja oziroma "*" (wild card). Na primer, rezultat iskanja po korenu besede "geo*" lahko vrne rezultate kot so geografija, geologija, geometrija in podobno.

Končna podoba spletne strani je prikazana na sliki 28.

Slika 28: Končna podoba izdelane baze znanja

Software Distributor's Knowledge Database




Installation GUI Controller Agent Security PC controller Licensing NT agent Other Click here to use a search tool	Installation • HSLco16575 Summary: pushAgent ERROR when installing to HP-UX 10.20 agent system • HSLco12062 Summary: installation of SW to one agent from multiple controllers • HSLco12140 Summary: Installation from tared depot. • HSLco14237 Summary: Since patchinstall swinstall takes 6 times longer. • HSLco13727 Summary: ERROR when installing to a depot • GUI • HSLco08168 Summary: GUI error when unmarking PC target • HSLco10762 Summary: DT Hertz Special: Can't monitor status of active jobs. • HSLco27558 Summary: swcopy doesn't go through the execution phase • HSLco11434 Summary: Internal error when monitoring job results. • HSLco21932 Summary: on a dterm (or hpterm) with DISPLAY not set to "" so that a • HSLco11608 Summary: SetValue AT_ytop failed error occurs when usr/sbin/sd • HSLco09606
--	--

Vir: Hermes SoftLab, 2002

5.5 Primer uporabe baze znanja

Ko se pojavi konkreten problem, ki ga inženir ne zna rešiti, lahko išče po bazi znanja s pomočjo iskalca ali pa po področju uporabe. V primeru, da se problem pojavi pri uporabi dovoljenja za uporabo programa, lahko inženir preveri, če je podoben problem že nekoč obstajal. V tem primeru išče po področju "licensing".

Kar nekaj problemov se je pojavilo pri napačni uporabi dovoljenja za uporabo programa. Dovoljenje za uporabo je vsebovano v datoteki `/var/adm/sw/.sdkey`. Problemi pa so bili različni; od tega, da je dovoljenje poteklo, da je stranka spremenila IP (Internet Protocol) naslov računalnika ali po pomoti zbrisana datoteka.

V levem okviru izberemo "Licensing" in izberemo ustrezen odgovor. Sama sem izbrala primer, kako ravnati, če zamenjamo IP naslov od Software Distributor strežnika (slika 29).

Slika 29: Sprememba IP naslova na SD strežniku

Call ID: HSLco04855
Date: 990927
Class: Licensing
Keywords: license, IP address
Summary: New IP Address of SD controller
Description: If you are forced to change the Unix SD Controller IP Address then license problem will occur because license is IP based. (applies to SD versions older than A.02.01)
Solution: For urgent cases contact SD support and you will be delivered a temporary license. In this case you have to send us new IP address. For permanent license contact PDC (Password Delivery Center). The change of IP won't affect the working of SD A.02.01.

Vir: Hermes SoftLab, 2002

Dostikrat se zgodi, da inženir ne ve v katero področje se uvršča določen problem. Takrat je priporočljiva uporaba iskalca. V polje iskalca vpišemo ključno besedo. Sama sem izbrala iskanje problemov, ki vsebujejo ključno besedo "daemon". Daemon ali swagentd se lahko nanaša na problem pri namestitvi programa, kot tudi pri varnosti, uporabi grafičnega vmesnika in podobno. swagentd je proces, ki je nujen za uporabo Software Distributorja in mora ves čas teči. Slika 30 prikazuje rezultat iskanja s ključno besedo "daemon":

Slika 30: Primer iskanja

WAIS Gateway

This is a searchable index of information.

Note: *This service can only be used from a forms-capable browser.*

Enter keyword(s):

Here is the result of your search using the keyword(s) "**daemon**":

- 1: [HSLco13897](#)
Score: **1000**, Size: **704 bytes**, Type: **HTML file**
- 2: [HSLco20305](#)
Score: **702**, Size: **977 bytes**, Type: **HTML file**
- 3: [HSLco04094](#)
Score: **575**, Size: **1 kbytes**, Type: **HTML file**
- 4: [HSLco11994](#)
Score: **365**, Size: **1 kbytes**, Type: **HTML file**
- 5: [HSLco06299](#)
Score: **235**, Size: **2 kbytes**, Type: **HTML file**
- 6: [HSLco16764](#)
Score: **227**, Size: **2 kbytes**, Type: **HTML file**
- 7: [HSLco06635](#)
Score: **224**, Size: **2 kbytes**, Type: **HTML file**
- 8: [HSLco04899](#)
Score: **138**, Size: **4 kbytes**, Type: **HTML file**
- 9: [Main Frame in kdb](#)
Score: **88**, Size: **11 kbytes**, Type: **HTML file**

Vir: Hermes SoftLab, 2002

Končna podoba Software Distributor baze znanja torej zajema opis problemov vključno z opisi rešitve teh problemov. Baza znanja bo služila tako našemu teamu, kot tudi Hewlett Packard centrom. Trenutno omogoča pregled 70 znanih problemov. Potrebno jo bo sproti dopolnjevati in izboljševati. Mislim, da bo dobrodošla pri inženirjih v Hewlett Packard centrih in bo pripomogla k povečanju njihove učinkovitosti reševanja Software Distributor problemov.

6 Sklep

Na osnovi preučevane literature lahko ugotovimo, kako velik in kompleksen pojem je znanje, zakaj je pomembno pri reševanju problemov s pomočjo sistemov podprtih z znanjem in zakaj je pomembno dobro in učinkovito upravljanje z njim.

Upravljanje z znanjem pomeni zelo širok spekter razsežnosti in aktivnosti v podjetju, ki se navezujejo na ugotavljanje potreb po posameznem znanju, na ustvarjanje pogojev za pridobivanje novega in izkoriščanje že obstoječega znanja v podjetju ter njegovo prehajanje na zaposlene znotraj podjetja.

Pri pojmovanju znanja kot dela ekspertnih sistemov in upravljanja z njim obstajajo določene razlike. Ekspertni sistemi znanje glede na različne kriterije delijo na trdo in mehko, znanje o domeni in splošno znanje, znanje o stvareh in metaznanje ter na formalno teoretično in neformalno empirično znanje (hevrstiko). Pri upravljanju z znanjem je poglobljeno razlikovanje med tihim (tacitnim) in izraženim (eksplicitnim) znanjem. Tiho znanje je precej podobno mehкому, eksplicitno pa trdemu. Prvega je težko formalizirati in prenašati, drugega precej lažje.

Vedno več podjetij se danes zaveda pomena, ki ga ima intelektualni kapital za razvoj in konkurenčno prednost podjetja. Podjetja iščejo načine in metode, ki jih ločijo od konkurentov in jim omogočajo boljši izhodiščni položaj, kot ga imajo tekmeci na trgu. Podjetja, ki so se odločila za ne ravno preprosto pot implementacije upravljanja z znanjem v praksi, se morajo soočiti z mnogimi izzivi in spremembami, ki so ključne za uspešno funkcioniranje in življenje koncepta upravljanja z znanjem v podjetju.

Velik pomen pri pridobivanju novih in nadgrajevanju že obstoječih znanj, veščin in sposobnosti ima proces učenja. To je kontinuiran proces tako posameznikov kot podjetij. Razlikujemo med več vrstami učenja, kot so individualno, teamsko in organizacijsko učenje.

V zadnjih desetih letih se je močno razširil pojem učeče se organizacije. Teorija učeče se organizacije poudarja pomen nenehnega učenja in vključevanja zaposlenih v procese usposabljanja in učenja.

Ob primerjavi izhodišč učeče se organizacije in upravljanja z znanjem bi lahko ugotovili, da pomeni slednje določeno nadgradnjo teorije o učeči se organizaciji. Upravljanje z znanjem je širši pojem, saj poleg učenja zajema tudi pretok posameznih znanj po organizaciji in tudi zunaj njenih meja.

Uvajanje upravljanja z znanjem v podjetju je strateška odločitev, ki se usmerja in kontrolira prek najvišjega vodstva podjetja. Podjetja, ki se želijo konkurenčno

postaviti ob bok močnim konkurentom, se zavedajo velikega pomena znanja in njegovega upravljanja.

Vsa uspešna podjetja pridobivajo in uporabljajo znanje. Na podlagi znanja delujejo in ukrepajo v nastali situaciji na trgu. Poleg dobrega upravljanja z znanjem je uspeh podjetja v veliki meri pogojen tudi z delovnimi pogoji, lokacijo in sestavo tima. Obstaja več načinov, s katerimi podjetja pridobivajo znanje: z akvizicijo, najemom znanja, oblikovanjem posebnih oddelkov, fuzijo, adaptacijo ali s povezovanjem v neformalne mreže znanja.

Podjetja torej imajo določeno znanje, saj vsak posameznik doprinese določeno specifično znanje. S pomočjo kodifikacije želimo znanje prenesti v takšno obliko, da bo dosegljivo tistemu, ki ga potrebuje. Lažje je sicer kodificirati eksplicitno znanje, vendar je tiho znanje tisto, ki je kapital podjetja.

Da bi bilo upravljanje z znanjem uspešno, je pomembna organizacijska kultura podjetja. Poudarjeno mora biti medsebojno sodelovanje, povezovanje, kreativnost zaposlenih in pripravljenost na posredovanje in prenašanje znanja. Na organizacijsko kulturo vplivajo tako spodbujevalci kulture znanja (jasni cilji, nagrajevanje, medsebojno sodelovanje) kot njeni zaviralci (nezaupanje, podcenjevanje, slaba komunikacija). Naloga vodstva, ki želi uvesti in delovati v smeri upravljanja z znanjem, je oblikovati organizacijsko kulturo, v kateri bodo prevladovali spodbujevalci kulture znanja in minimizirani zaviralci.

Še eden dejavnik, ki je zelo pomemben pri upravljanju z znanjem, je prenos znanja. Obstaja več metod prenosa znanja, ki so odvisne tudi od nacionalne kulture, pa tudi od same kulture podjetja. Podjetja za prenos znanja uporabljajo treninge, delavnice, tehnična poročila, razno dokumentacijo in podobno. Za uspešno upravljanje z znanjem so ključne tudi vloge in izkušnje posameznikov. Le-te se delijo na tri sloje: nižji (analitik znanja, urednik, inženir znanja), srednji (vodja projekta znanja) in višji sloj (direktor upravljanja z znanjem).

Podjetja danes dajo vedno večji poudarek znanju, oblikujejo in urejajo svoje baze znanja. Baza znanja vsebuje znanje o specifičnem področju, ki je ponavadi strokovne narave. Potem vsebuje pravila, ki povezujejo dogodke, relacije med dogodki in metode ter heuristike za reševanje problemov. Baza torej vsebuje znanje dveh vrst: znanje iz problemske domene in o načinu reševanja problemov.

Velik problem graditve baze znanja je predstavitev oziroma zapis znanja v obliki, ki jo razumejo programi. Znanih je več metod predstavitve znanja: produkcijska pravila, okviri, semantične mreže in predikatni račun. Vsaka od njih ima določene prednosti in pomanjkljivosti. Pravtako obstaja tudi več načinov organiziranja znanja pri podpori

uporabnikov. Eden takih, uporabljen v praktičnem delu naloge, je zbirka dokumentov z iskanjem po celotnem besedilu.

Cilj izgradnje baze znanja produkta Software Distributor je bil prenesti določeno znanje o problemih in vprašanjih v takšno obliko, da bi bila dostopna tudi ljudem izven Software Distributor teama. Problemi so razporejeni po glavnih produktnih področjih. Inženirjem takšna baza omogoča tudi z iskanjem po ključnih besedah priti do rešitve problema. V kolikor je podoben problem že obstajal, baza znanja dobro služi svojemu namenu in zmanjša število dohodnih klicev v naš team. V kolikor baza ne vsebuje rešitve strankinega problema, pa se Hewlett Packard centri še vedno lahko obrnejo na nas. To seveda ni naš namen, saj bi radi zmanjšali število dohodnih klicev. Je pa vsekakor začetek daljšega procesa, ki bo sčasoma dosegel raven naših pričakovanj. Zato je naša naloga stalno izpopolnjevati bazo in sčasoma doseči njeno izpopolnjenje do zadovoljive meje.

7 Literatura in viri

Literatura:

1. Amidon M. Debra, Skyrme J. David: Creating the Knowledge-based Business. Wimbeldon: Business Intelligence Ltd., 1997. 524 str.
2. Avison D. E., Fitzgerald G.: Information Systems Development: Methodologies, Techniques and Tools. London: McGraw-Hill, 1995. 505 str.
3. Bach, Oesterle, Vogler: Business Knowledge Management in der Praxis. Heidelberg: Springer-Verlag, 2000.
4. Bordac et al.: Knowledge Management and Sharing. [URL: <http://www.beedu.org>], 21.02.2002.
5. Brasza Henry P.: Expert Systems Technology. White Paper #1. Churchill Systems Inc. [URL: <http://churchillsys.com/WP1-EXSY.html>].
6. Bratko Ivan: Inteligentni informacijski sistemi. Ljubljana: Fakulteta za elektrotehniko, 1984. 106 str.
7. Bratko Ivan: Ekspertni sistemi in umetna inteligenca. Ljubljana: Fakulteta za strojništvo, 1991. 223 str.
8. Budin Leo et al.: Inteligentni sustavi. Opatija: MIPRO, 2001. [URL: <http://www.isustavi.com>], maj 2001.
9. Chorafas Dimitris N.: Knowledge Engineering. New York: Van Nostrand Reinhold, 1990. 380 str.
10. Čater Tomaž: Znanje kot vir konkurenčne prednosti in management znanja. Ljubljana: Naše gospodarstvo, št. 4, 2000, str. 505-520.
11. Česnovar Tone: Ključna vloga tihega znanja. Socius, 2001.
12. Čibej Jože Andrej: Sistemi za podporo poslovnemu odločanju. Zapiski predavanj. Ljubljana: Ekonomska fakulteta, 1998. 141 str.
13. Damij Talib, Grad Janez, Jaklič Jurij: Izbrane teme iz informacijske tehnologije. Ljubljana: Ekonomska fakulteta, 1995. 316 str.
14. Davenport Thomas H.: "If only HP knew what HP knows...". Center for Business Innovation. [URL: <http://cbi.cgey.com/journal/issue1/features/ifonly/index.html>], 26.11.2001.

15. Davenport Thomas H., Prusak Lawrence: Working Knowledge – How Organizations Manage What They Know, Boston: MA, Harvard Business School Press, 1998. 240 str.
16. Debenham John: Knowledge Engineering. Unifying Knowledge Base and database Design. Berlin: Springer, 1998. 465 str.
17. Deen S. M., Thomas G. P.: Data and Knowledge Base Integration. London: Pitman Publishing, 1990. 326 str.
18. Everest Gordon C.: Database Management. Objectives, System Functions, and Administration. New York: McGraw-Hill Series in Management Information Systems, 1986. 816 str.
19. Fabbri Anthony J., Schwab Robert A.: Practical database management. Boston: PWS-KENT, 1992. 498 str.
20. Fischer Michael D.: Expert Systems and Anthropological Analysis. BICA Issue No. 4. University of Florida. [URL: http://lucy.ukc.ac.uk/bicaweb/b4_/expert.html], marec 1986.
21. Gibb F.: Expert systems in libraries. Proceedings of a conference of the Library Asociation Information Technology Group and the Library and Information Research Group. London: Taylor Graham, 1985. 97 str.
22. Grad Janez, Jaklič Jurij: Baze podatkov. Ljubljana: Ekonomska fakulteta, 1996. 254 str.
23. Humar Dekleva Luka: Zabrisana meja med fenomenom in kultom. Nedelo. Ljubljana, 10.3.2002, str. 9.
24. Hunt Daniel V.: Artificial Intelligence & Expert Systems Sourcebook. New York: Chapman & Hall, 1986. 315 str.
25. Jarc Valentin: Znanstveno tehnična informatika. Ljubljana: Informacijski center, 1985.
26. Kavčič Bogdan: Učeča se organizacija. Ljubljana: Slovenska ekonomska revija, letnik 45, št. 5, 1994, str. 424 – 432.
27. Kerry Ruth: Integrating knowledge-based and database management systems. Chichester: Ellis Horwood, 1990. 379 str.
28. Krogh Georg Von, Kazuo Ichijo, Ikujiro Nonaka: Enabling Knowledge Creation – How to Unlock the Mystery of Tacit Knowledge and Release the Power of Innovation. Oxford: Oxford University Press, 2000.

29. Kukovica Peter: Možnost uporabe ekspertnih sistemov pri tržnih informacijskih sistemih. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 1993. 174 str.
30. Macdonald John: Understanding Knowledge Management in a week. London: Hodder&Stoughton, 1999. 95 str.
31. Možina Stane, et. al.: Management. Radovljica: Didakta, 1994. 1072 str.
32. Newman Brian D.: Knowledge Management vs. Knowledge Engineering. [URL: <http://revolution.3-cities.com/~bonewman/kmvske.htm>], 5.1.1996.
33. Nonaka Ikujiro, Takeuchi Hirotaka: The knowledge – creating company: how Japanese companies create the dynamics of innovation. Oxford: Oxford University Press, 1995. 284 str.
34. Novins Peter, Armstrong Richard: Choosing Your Spots for Knowledge Management. A Blueprint for Change, 26.11.2001. 7 str.
35. Peršak Marjan: Gospodarjenje z znanjem. Socius. 2001.
36. Peršak Marjan, Češnovar Tone: Prenos znanja v prakso slovenskih podjetij. 2002.
37. Pirc Aleša Saša: Organizational learning and knowledge management. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2000. 169 str.
38. Planko Sergeja: Upravljanje z znanjem v organizaciji. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2001. 135 str.
39. Prusak L.: Where did knowledge come from? Knowledge Management: Volumen 40, 2001, številka 4.
40. Rajkovič Vladislav: Poslovni ekspertni sistemi. [URL: <http://lopes1.fov.uni-mb.si/pub/PES>], 2001.
41. Rational Software: ClearDDTS, User's Guide, Version 4.1. Rational Software, 1998. 76 str.
42. Rational Software: Rational ClearDDTS, Administrator's Guide, Version 4.5. Rational Software, 1999. 213 str.
43. Ribarič S.: Arhitektura računalna. Zagreb: Školska knjiga, 1990.
44. Schmoldt Daniel L., Rauscher Michael H.: Building Knowledge-Based Systems for Natural Resource Management. New York: Chapman & Hall, 1996. 386 str.

45. Shapiro Stuart C.: Encyclopedia of Artificial Intelligence. New York: Willey-Interscience Publication, John Willey & Sons, Inc., 1992.
46. Siegl Matjaž: Upravljanje znanja pri izvajanju podpore uporabnikom. Magistrsko delo. Ljubljana: Ekonomska fakulteta, 2002. 88 str.
47. Sowa John F.: Principles of Semantic Networks. San Mateo: Morgan Kaufmann Publishers, 1991.
48. Sproull Lee, Kiesler Sara: Connections: New Ways of Working in the Networked Organization. Cambridge, Massachusetts: MIT Press, 1998. 212 str.
49. Waterman D. A.: A guide to expert systems. Reading, Addison-Wesley, 1986.
50. Winston Patrick Henry, Brown Richard Henry: Artificial Intelligence: An MIT Perspective, Volume I: Expert Problem Solving, Natural Language Understanding, Intelligent computer Coaches representation and Learning. Cambridge: The MIT Press, 1984. 492 str.
51. Zack Michael H.: Managing Codified Knowledge: Knowledge Management and Information Technologies. College of Business Administration Northeastern University. [URL: <http://web.cba.neu.edu/~mzack/articles/kmarch/kmarch.htm>], september 1998.
52. Zakrajšek Meta, Gašperšič Andreja: Upravljanje znanja. [URL: <http://www.src.si>], 15.02.2002.

Viri:

1. Answers to Frequently Asked Questions. [URL: <http://www.bus.utexas.edu/kman/answers.htm>], 26.11.2001.
2. Brezovnik Robi: [URL: http://www.pfmb.uni-mb.si/didgradiva/2000/projekti96/Robi-Brezovnik/umetna_inteligenca-desno.htm], 26.11.2001.
3. Bučar Maja: Mednarodni simpozij Tiho znanje – velik kapital v podjetju, 2001.
4. Hermes SoftLab, d.d. Ljubljana, ClearDDTS interni dokumenti.
5. Hermes SoftLab, d.d. Ljubljana, intranet.
6. Hewlett Packard. IT resource center forums. [URL: <http://forums.itrc.hp.com/cm/>], 14.2.2002.

7. Knowledge Management. Bordac, Friedman, Glaser, Harding 2001. [URL: <http://www.beedu.org/omaet/ed634/km/litreview.html>], 20.11.2001.
8. Knowledge Management. [URL: <http://www.destinationcrm.com>], 12.2.2002.
9. Navodila za Arity/Expert. [URL: <http://lisa.uni-mb.si/student/predmeti/es/aexp.htm>], 10.09.2001.
10. Sistemi podprti z znanjem (knowledge based systems). [URL: http://www.mf.uni-lj.si/~jure/pred_bib/i1/kbs/kbs.html], 15.1.2002.
11. Tomassini Massimo: Mednarodni simpozij Tiho znanje – velik kapital v podjetju, 2001.

8 Priloge

- Vsebina kdb datoteke

Vsebina kdb datoteke (skripte):

```
# kdb - Knowledge Database script
# set -x
# find all bugs of sd.calls project in status Resolved
BUGS=`remsh dirlija -l sdsupp "cat /home/sdsupp/kdb/index"`

# check out the directory under which KDB is stored
cd /udl/sd-doc/sdsupp
/usr/atria/bin/cleartool co -nc .

# Change directory to the KDB directory under WWW root
#cd /home/sdsupp/WWW

# check out main frame
/usr/atria/bin/cleartool co -nc frmain02.htm
rc=$?
if [ ${rc} -ne 0 ]
then
    echo "CO ne dela!"
    /usr/atria/bin/cleartool unco .
    exit 1
fi

# back up the old frame
cp frmain02.htm frmain02.htm.old 1>/dev/null 2>&1

# use template
cp main_template.htm frmain02.htm

for RESOLVED in $BUGS
do
    # take the last line of history enclosure that has strings
    # (PROBLEM) and (added or eddited) !we don't want string
    # 'deleted'

    PROB=`remsh dirlija -l sdsupp "cat /home/sdsupp/kdb/$RESOLVED \
    | grep xddts | grep PROBLEM | grep -e added -e edited -e \
    deleted | tail -1"`

    PROBL=`echo $PROB | awk '{print $6}'`

    # enclosure PROBLEM was deleted, remove the html file from CC
    if [ "$PROBL" = "deleted" ]
    then
        /usr/atria/bin/cleartool rmname $RESOLVED.htm
        rm $RESOLVED.htm
        echo "$RESOLVED deleted."
    fi

    if [ "$PROBL" = "edited" -o "$PROBL" = "added" ]
    then
```

```

#parse out the PROBLEM enclosure
TEXT=`remsh dirlija -l sdsupp "cat /home/sdsupp/kdb/$RESOLVED \
| awk /'PROBLEM \('/,/History/'`

#delete the first line
TEXT=`echo "$TEXT"|sed "1d"`

# Delete also the first line of History enclosure, which is
# also parsed
TEXT=`echo "$TEXT"|sed "/** History **/d"`

# Add date and call ID before the PROBLEM enclosure's text
TEXT="Date:"`echo $PROB|cut -f2 -d" "`"\n"$TEXT
TEXT="Call ID:"$RESOLVED"\n"$TEXT

# get class from PROBLEM enclosure
CLASS=`echo "$TEXT"|grep "Class:"|awk -F':' '{print $2}' \
| sed "s/^ */g"`

# get summary from PROBLEM enclosure
SUMMARY=`echo "$TEXT" | grep "Summary:" | cut -c9- | tr \
"\/" " " | sed s/\&/and/g`

# Change text to HTML
TEXT="<HTML>\n""<title>$RESOLVED</title>\n""$TEXT""\n</HTML>"
TEXT=`echo "$TEXT"|sed "s/Call ID:/<strong>Call ID:</strong>/g"\
|sed "s/Date:/<br><strong>Date:</strong>/g"`
TEXT=`echo "$TEXT"|sed "s/Class:/<br><strong>Class:</strong>/g"\
|sed "s/Keywords:/<br><strong>Keywords:</strong>/g"`
TEXT=`echo "$TEXT"|sed "s/Summary:/<br><strong>Summary:</strong>\
/g"|sed "s/Description:/<br><strong>Description:</strong>/g"`
TEXT=`echo "$TEXT"|sed "s/Solution:/<br><strong>Solution:</strong>\
/g"|sed "s/Related SR:/<br><strong>Related SR:</strong>/g"\
| awk '{ print $0"<br>"}'`

#
# Now move the text to html file named <CallID>.htm
# Assumption here is that the script is run in view sdsuppweb
# under CC html file is moved to CC
#

CO=`/usr/atria/bin/cleartool ls -long | grep -v "view private" \
|grep "${RESOLVED}.htm"`
echo $CO
if [ -z "$CO" ]
then
# ga se ni pod CC
/usr/atria/bin/cleartool mkelem -nc $RESOLVED.htm
rc=$?
if [ ${rc} -ne 0 ]
then

```

```

        echo "Ne morem kreirati elementa!"
        /usr/atria/bin/cleartool unco .
        exit 1
    fi
else
    /usr/atria/bin/cleartool co -nc ${RESOLVED}.htm
    rc=$?
    if [ ${rc} -ne 0 ]
    then
        echo "CO ne dela!"
        /usr/atria/bin/cleartool unco .
        exit 1
    fi
fi
echo "$TEXT" > $RESOLVED.htm
ERR_TEXT=`/usr/atria/bin/cleartool ci -nc $RESOLVED.htm 2>&1`
rc=$?
ERR_TEXT=`echo $ERR_TEXT | grep "Error: By default, won't create \
    version with data identical"`
if [ ${rc} -ne 0 ]
then
    if [ -z "${ERR_TEXT}" ]
    then
        echo "CI ne dela!"
        /usr/atria/bin/cleartool unco .
        exit 1
    else
        /usr/atria/bin/cleartool unco -rm $RESOLVED.htm
    fi
fi

#
# rebuild main frame frmain02.htm
# every record's Call ID and Summary are put under proper class
#

# insert call id and summary into main frame
sed "s/$CLASS<\/strong><\/font><\/a><\/p><ul>/$CLASS<\/strong><\/font><\/a><\/p><ul><li><a href=\"\$RESOLVED\.htm\">$RESOLVED\
<\/a><br><strong>Summary:<\/strong>$SUMMARY\
<\/li>/g" frmain02.htm > frmain02.htm.tmp

# replace frmain02.htm with saved output of sed command
cp frmain02.htm.tmp frmain02.htm

fi # end of statement if enclosure PROBLEM exists
done

# check in rebuild main frame
ERR_TEXT=`/usr/atria/bin/cleartool ci -nc frmain02.htm 2>&1`
rc=$?
ERR_TEXT=`echo $ERR_TEXT | grep "Error: By default, won't create version\

```

```

        with data identical"`
if [ ${rc} -ne 0 ]
then
    if [ -z "${ERR_TEXT}" ]
    then
        echo "CI ne dela!"
        /usr/atria/bin/cleartool unco .
        exit 1
    else
        /usr/atria/bin/cleartool unco -rm frmain02.htm
    fi
fi
/usr/atria/bin/cleartool ci -nc .

#
# Rebuild search engine's index
#

# change directory to the cgi-bin dir
cd /opt/httpd/cgi-bin/
./swish -c swish.conf

```

Slovarček tujih izrazov

artificial intelligence (AI)	umetna inteligenca
associative networks	asociativne mreže
backward chaining	sklepanje v smeri nazaj
blackboard	delovno mesto
bug/defect	hrošč/defekt
chaining	veriženje ali sklepanje
Chief Executive Officer (CEO)	direktor
Chief Knowledge Officer (CKO)	direktor upravljanja znanj
class	razred
conceptual networks	konceptualne mreže
conflict-resolution techniques	tehnike reševanja konflikta
conflict set	konfliktna množica
consistency checker	preverjanje doslednosti
counterfactual reasoning	sklepanje v nasprotju z dejstvi
cross-functional team	raznolik team iz predstavnikov različnih org. enot
data driven method	podatkovno vodena metoda
enclosure	priloga
enhancement request	predlog za izboljšavo produkta
expert systems	ekspertni sistemi
explanation mechanism	pojasnjevalni mehanizem
field selection	izbira polja
forward chaining	sklepanje v smeri naprej
frame-based systems	sistemi osnovani na okvirih
global database	globalna podatkovna zbirka
goal driven method	ciljno vodena metoda
inference engine/reasoning mechanism	mehanizem sklepanja
Information Systems department	oddelek informacijskih sistemov
knowledge base	baza znanja
knowledge based systems	sistemi podprti z znanjem
knowledge generation	pridobivanje, generiranje znanja
knowledge integrator	povezovalac znanja
knowledge management	upravljanje z znanjem
knowledge map	zemljevid, načrt znanja
knowledge representation problem	predstavitveni problem znanja
learning organization	učeča se organizacija
mixed chaining	kombinirano sklepanje
network representation	mrežna predstavitev
patch	sistemski, produktni popravek
project	projekt
Research & Development (R&D)	razvojno raziskovalni oddelek
rule interpreter	razlagalec pravil

scheduler	urnik
semantic memory	semantični spomin
slot	reža
team building	treningi za krepitev skupinskega duha
truth	resnica
user interface	uporabniški vmesnik
workaround	pomožna, alternativna rešitev
working memory	delovni pomnilnik