

# UREJANJE IN PREISKOVANJE BAZ PODATKOV NA ASOCIATIVNI OSNOVI

Iztok Lajovic  
KreS, Kreativni sistemi, d.o.o., Ljubljana

Današnja stopnja razvoja računalništva zahteva veliko znanj za preiskovanje kompleksnih informacij. Običajni uporabnik računalnika je odvisen od specializiranih, namenskih programskih rešitev in za izdelavo le-teh so seveda potrebni računalniški strokovnjaki. Namenske rešitve zadovoljujejo obdelavo podatkov v načrtovanem obsegu problemskega prostora. Uporabniki pa želijo še nekaj več - nekaj, kar bi obvladovalo tisti del njihovih želja po obdelavi, ki jih namenske rešitve ne pokrivajo.

## Asociativno razmišljanje

Človekova zmožnost asociativnega razmišljanja je v računalništvu že od vsega začetka vodilo in cilj razvoja. Asociacije nam omogočajo logično povezovati stvari med seboj in so osnova procesov učenja. Pot do cilja, verno posnemati z računalnikom zmožnost asociativnega razmišljanja, bo še dolga. V tem trenutku še ne vidimo konca poti, cilj pa imamo vedno pred sabo. Preskočiti nekaj stopenj razvoja se ne da, korakati moramo vztrajno ne glede na ovire tehnologije in stopnje našega umevanja procesov razmišljanja, ki se dogajajo v nas samih.

Množice podatkov, ki smo si jih zapomnili, imamo urejene na nam danes še neznan način. Znano je samo, da si ljudje tekom življenja nezavedno ustvarjamo poti in načine iskanja zapomnjenih podatkov. Ti načini so tako zelo različni, da se sposobnosti pomnenja posameznikov zelo razlikujejo in segajo od slabe pomnilne sposobnosti do skorajda fotografskega spomina. Različna je tudi sposobnost pomnenja glede na predmet, saj si nekateri z lahkoto zapomnijo dogodke, drugi ljudi, tretji matematične izreke in formule.

Predmeti, o katerih razmišljamo, so stvarne ali samo miselne narave. Vsebinsko vsakega od njih tvori cela množica informacij o različnih lastnostih predmeta, ki šele vse skupaj dajo njegovo celotno sliko. Nekatere od lastnosti so enostavne in pripadajo samo temu predmetu, druge pa so take, da vzbujajo naše asociacije in vodijo naše misli tudi na druge predmete. Jakost asociativne povezave misli se s časom spreminja: s ponavljanjem in učenjem jih jačamo, če pa tega ne počnemo, take povezave s časom same od sebe slabijo. Pozabljanje je sproščanje ali informacij o miselnih predmetih ali pa sproščanje asociativnih povezav med njimi.

V računalniku so podatki urejeni na več načinov. Preproste sezname ustvarijo že rutine samega operacijskega sistema, kompleksne strukture pa shranijo in uredijo sistemi za upravljanje s podatkovnimi ba-

zami. Le-ti vsak na svoj način urejajo podatke v podatkovni bazi in vsak ima svoj način, s katerim uporabnik vpisuje podatke oziroma jih išče in pripravlja izpise. Programi s svojimi orodji omogočajo vnos podatkov in preiskovanja datotek na način, ki je pogojen z vsebino obdelave podatkov obravnavanega področja.

Danes uporabljane baze podatkov shranjujejo podatke o stvareh, grupirane v datoteke glede na njihovo vsebino. V okviru ene datoteke so nabrani podatki o stvareh iste vrste in pregled datoteke nam posreduje le seznam npr. komitentov, prometa, osnovnih sredstev ipd. Ko želimo podatke ene datoteke obogatiti z dodatnimi informacijami, moramo pripraviti poseben postopek, s katerim posegamo v druge datoteke po dodatne informacije. Ko želimo pregledovati podatke, ki so shranjeni v drugi datoteki, moramo zaključiti s pregledovanjem trenutne datoteke in prestopiti na opazovanje podatkov druge datoteke. Povezovanje podatkov iz več datotek temelji na postopkih, ki jih je treba za vsak posamezen primer posebej pripraviti.

Ljudje pa ne uvrščamo stvari, o katerih razmišljamo, v različne skupine, temveč kot da so vse stvari le razmeščene v miselnem prostoru, samo z mislimi potujemo od ene stvari ali predmeta k drugi. Pri tem nas ne zanima vzorec lastnosti posamezne stvari ali predmeta, saj ga nevede obravnavamo kot sestavni del predmeta, o katerem premišljujemo.

Zanimivo pa je, da naše misli potujejo usmerjeno, ne naključno. Trenutno prisotna misel je vedno v neki zvezi z prejšnjo mislijo, obe imata najmanj eno lastnost skupno, mi sami pa se tega večinoma ne zavedamo. Presek lastnosti predmetov ali pojmov, med katerimi potujemo z mislimi, je osnova asociativnemu načinu razmišljanja.

Pojavlja se torej potreba po dograditvi mehanizmov za shranjevanje in iskanje informacij v taki smeri, da bi lahko enostavno in hitro prehajali v bazi podatkov od

enega opazovanega predmeta na drugega, ne glede na to, v kateri datoteki baze podatkov je shranjen. Sedanji čas spreminja zahteve po hitrosti, od hitrosti shranjevanja podatkov, ki jo danes uporabljane baze podatkov zagotavljajo, k hitrosti iskanja in prikaza tako enostavnih kot kompleksnih informacij.

Ko ljudje torej opazujemo svet okoli sebe, v nekem trenutku opazujemo neki objekt, istočasno pa vidimo tudi kontekst tega objekta: ko npr. opazujemo človeka v pisarni, vidimo njega, pa tudi pohištvo, knjige, okna, vrata. Ko premaknemo pogled na objekt, ki pripada kontekstu ravnokar opazovanega objekta, postane ta, pogledani objekt nov predmet našega opazovanja, doslej opazovani objekt pa se preseli v njegov kontekst. To je princip, ki ga z današnjimi bazami podatkov težko realiziramo v ustreznih odzivnih časih.

Tako potovanje z enega na drug objekt in nazaj pa zahteva realizacijo koncepta dvo- ali večsmernih povezav med objekti. Današnje povezovalne tehnike (hiperlink) so le enosmerne, saj niso zahtevne za realizacijo (ko v nekem tekstu dodam povezavo – link – na drug tekst, se ta informacija zapiše samo v obravnavanem tekstu, povezani tekst pa ostane nedotaknjen in zato nikoli ne izvemo, kateri vsi teksti se sklicujejo nanj).

V realnem svetu pa so stvari zastavljene drugače: če sedim poleg nekoga, sem njegov sosed, on pa si v nobenem primeru ne more privoščiti ne biti v istem trenutku tudi moj sosed. V realnem svetu so torej stvari najmanj dvosmerne. Za predstavitev realnega sveta torej potrebujemo več kot nam ponujajo posamezni hiperlinki – potrebujemo najmanj usklajene pare hiperlinkov.

Pravo sliko realnega sveta pa dosežemo šele z uvedbo n-terih povezav. Predstavljajmo si, kateri atributi določajo npr. koncert: dvorana, čas in izvajalci. V istem trenutku isti izvajalci ne morejo biti v drugi dvorani. Torej je koncert definiran s tremi delnimi ključi in eden teh delnih ključev je lahko množica. Med dodatnimi atributi koncerta je še spored, ki pa je spet množica skladb različnih avtorjev in pri vsaki od njih lahko nastopa drug solist. Postavlja se vprašanje: kako realizirati vpogled v tako kompleksno strukturo, kot je ravnokar predstavljeni koncert, na eni strani in na drugi strani, kako realizirati ažuriranje take strukture ob istočasnem vzpostavljanju n-smernih povezav med objekti.

## Izvedba Panorame

Panorama je ime sistema za upravljanje s podatkovno bazo, ki vsebuje nov mehanizem shranjevanja in iskanja podatkov. Le-ta pomeni poseben korak v razvoju prijaznih orodij za obvladovanje kompleksnih po-

datkovnih struktur v informacijskih sistemih. Njegova posebnost je v tem, da je usmerjen v hitro poizvedovanje in iskanje kompleksnih informacij na način, ki je podoben načinu človekovega ogledovanja sveta: prikaz neke informacije - zapisa neke datoteke - je dopolnjen še z vsemi podatki o povezanosti tega zapisa z vsemi tistimi zapisi iz celotne podatkovne baze, s katerimi je opazovani zapis povezan.

Izhodišče za izdelavo Panorame je bilo razmišljanje o metodah, ki bi posnemale asociativni način razmišljanja človeka, in uporabo le-teh pri obdelavah velikih baz podatkov, bogatih z veliko različnimi vrstami objektov.

Sistem ima vgrajene mehanizme, ki omogočajo obravnavo podatkov o stvareh na način, ki je podoben človeškemu načinu razmišljanja in ki ni obremenjen s tehniko shranjevanja podatkov. Zato je delo enostavno in dostopno tudi tistim ljudem, ki se na računalništvo in na delo z bazami podatkov ne spoznajo.

Realizacijsko osnovo za mehanizme, ki bi omogočali tako preklapljanje med objekti, smo videli v izdelavi posebne strukture baze podatkov, ki omogoča vzpostavljanje n-terih povezav med objekti.

Objekti, ki jih obravnavamo, shranjujemo in iščemo v bazi, so grupirani v razrede objektov. Za vse objekte enega razreda velja, da imajo enak nabor lastnosti oziroma atributov, razlikujejo pa se v vrednostih teh atributov. Podatki o vseh objektih enega razreda so lahko glede na to, da imajo enak nabor atributov, shranjeni v eni datoteki.

Atributi imajo v večini primerov eno samo vrednost, lahko pa določimo tudi attribute z večkratnimi vrednostmi. Med takimi so posebej zanimivi atributi, ki povezujejo obravnavani objekt z objekti iz njegove okolice, iz njegovega konteksta. To so povezovalni atributi.

Za primer takega atributa vzemimo opis atributov človeka, ki je (so)lastnik več stanovanj. Tako obravnavani človek kot tudi stanovanja so objekti, ki spadajo v dva različna razreda (datoteki), v razred *ljudje* in v razred *stanovanja*. Atribut *lastnik stanovanja* se lahko pojavi pri enem človeku ali nikoli ali enkrat ali pa tudi večkrat, pač glede nato, koliko stanovanj ima človek v lasti. Vrednost vsake pojavnosti takega atributa je drugo stanovanje. Po drugi strani pa tudi stanovanjem priredimo atribut *lastniki*, saj je eno stanovanje lahko v lasti najmanj enega ali pa več ljudi. Ko v Panorami označimo, da je obravnavani človek lastnik določenega stanovanja, se istočasno izpolni tudi ustrezní atribut pri izbranem stanovanju, to je, kdo je njegov (so)lastnik.

Z isto akcijo ažuriramo torej povezovalna atributa na obeh straneh povezave in s tem ustvarimo *vez* med objektoma. Vez je osnova za poizvedovanja z

obeh strani: ko opazujemo podatke človeka, dobimo seznam vseh stanovanj, katerih (so)lastnik je, ko pa opazujemo podatke nekega stanovanja, dobimo podatke o vseh njegovih solastnikih. Seveda se s časom (so)lastniki stanovanja menjajo in to zahteva uvedbo novega razreda, *solastniki stanovanj*, katerih obvezni ključni so na eni strani posamezno stanovanje, na drugi pa posamezen (so)lastnik, dodatni atributi pa govorijo o datumu nakupa, deleža lastništva in morebitnemu datumu prodaje.

Obvladovanje različnih razredov objektov, njihovih atributov ter povezav med njimi zahteva poseben interpretacijski mehanizem, *podatkovni slovar*, ki je integralni del Panorame. V podatkovni slovar vpisujemo definicije razredov in njihovih atributov. Poleg teh podatkov vnesemo tudi informacijo o lokaciji objektov vsakega razreda, saj so lahko podatki, ki jih dosežemo, shranjeni v več fizičnih datotekah.

Ko je razred vpisan v podatkovni slovar, lahko že vpisujemo objekte tega razreda. Če se kasneje izkaže, da potrebujemo nov atribut, ga dodamo v definicijo tega razreda, a s tem ne prizadenemo že vpisanih objektov: ob prvem podrobnem pregledu atributov takega objekta bomo morali dopolniti manjkajoče podatke.

Podatkovni slovar (primer definicije razreda z vidnimi povezavami je prikazan na gornji sliki) omogoča

razpoznavanje in pravo interpretacijo lastnosti opazovanega predmeta. Na preprost način se s pomočjo vezi selimo od opazovanja enega predmeta na opazovanje drugega predmeta in program sproti ugotavlja ali smo pri tej selitvi preskočili v okvir drugega razreda in takoj spremeni seznam atributov tako, da ustreza novo opazovanemu predmetu. Kot primer vzemimo, da najprej opazujemo podatke delavca, zatem pa podatke organizacijske enote, v kateri je zaposlen. Seveda so lastnosti delavca povsem druge kot lastnosti organizacijske enote.

Vezi so eno najmočnejših orodij Panorame. Število vezi med objekti ni omejeno, ne po tipu, ne po številu. Med seboj lahko povezujemo tudi objekte istega razreda in pri tem navedemo pomen odnosa v vsaki smeri vezi: s tem lahko npr. opišemo sorodstvene odnose, ko navedemo, da je eni osebi druga oseba oče, in obratno, da je osebi, ki je oče, prva oseba sin. Z lahkoto opisujemo rodbinska drevesa in razne rodovnike.

Z vezmi ustvarimo lahko tudi sestavnice, s katerimi gradimo nove objekte s pomočjo tistih, ki jih sestavljajo. Vsak od teh objektov lahko spada v drug razred in s tem ima seveda druge attribute. Kot primer vzemimo avtomobilski *motor*, ki ima attribute moči, navora, kubature, število ventilov itd., sestavljajo pa ga objekti *svečke* z atributi toplotne moči in razdalje

razred:  nadrejen:  povezani razred:

polje oz. naslov kolone	tip	dol	obv	polje poveži z razredom
primek in ime	tekst	25	<input checked="" type="checkbox"/>	
<b>klub</b>	ključni podatek (ključ)	25	<input checked="" type="checkbox"/>	klubi
telefon	tekst	22	<input checked="" type="checkbox"/>	
ulica doma	tekst	20	<input checked="" type="checkbox"/>	
ptt in kraj doma	tekst	35	<input checked="" type="checkbox"/>	
email	tekst	25	<input checked="" type="checkbox"/>	
rojen	datum	10	<input checked="" type="checkbox"/>	
zaposlen	tekst	40	<input checked="" type="checkbox"/>	
ulica podjetja	tekst	20	<input checked="" type="checkbox"/>	
ptt in kraj podjetja	tekst	30	<input checked="" type="checkbox"/>	
član od	datum	10	<input checked="" type="checkbox"/>	
funkcije članov	možna povezava	25	<input type="checkbox"/>	funkcije članov
izstopil	datum	10	<input checked="" type="checkbox"/>	
prestopil	možna povezava	25	<input type="checkbox"/>	prestopi
znanstv.naziv	tekst	5	<input checked="" type="checkbox"/>	
strokovni naziv	tekst	18	<input checked="" type="checkbox"/>	

polja
klub
ulica
ptt in kraj
klubski večer
kraj
ustanovljen
Charter Night
številka
sponzorski klub
Guiding Lion
URL
člani
funkcije članov
prihodi iz kluba
odhodi v klub
dobrodelne akcije
donacije
bančni račun

med elektrodami, *gredi* z atributom število ležajev, itd. Po drugi strani je ta motor sestavni del avtomobila. Vsak od elementov sestavnice je torej svoj objekt, ki ga obravnavamo kot samostojno enoto ali kot sestavni del.

Posebnost sistema so tudi *n-terke*, to so povezave *n* objektov v nov objekt. Ti objekti vsebujejo *n* obveznih povezovalnih atributov, ki predstavljajo vse objekte, s katerimi določamo nov objekt. Vsak od povezovalnih atributov ima najmanj eno, lahko pa tudi več vrednosti.

Zamislimo si nekoliko drugačen primer, spremljanje opravljanja izpitov, seminarjev in diplom posameznih študentov. Imamo množice: *študentje*, *predmeti*, *seminarji* in *diplome*. Za spremljanje opravljanja izpitov potrebujemo nov razred *izpiti*, katerega identificirata atributa *študent* in *predmet* (ta dva atributa izpita sta delna ključa objekta *izpit*), potrebujemo pa še ostale attribute kot datum opravljanja izpita ter dobljena ocena. Če pa želimo spremljati vsa opravljanja nekega izpita, tudi neuspešne poskuse, moramo kot delni ključ definirati tudi atribut *datum*. Tudi diploma potrebuje dodatni ključni atribut, *študenta*, ki je diplomu napravil, poleg tega pa še *naslov* diplome ter *datum* opravljanja diplomskega izpita. *Seminar* pa, podobno kot izpit, potrebuje delne ključne *študenta* in *predmet*, poleg tega pa še *naziv* naloge ter *datum* izdelave naloge.

In kaj nam nudi sistem ob vnosu podatkov o študentih, predmetih, izpiti, seminarjih in diplomah? Ko izberemo nekega študenta, dobimo poleg njegovih matičnih podatkov na ekran še podatke o vseh izpiti z datumi in ocenami, podatke o vseh izdelanih seminarskih nalogah ter podatke o diplomah, če je že končal študij. Ko izberemo neki predmet (bodisi ga z miško izberemo iz seznama izpitov opazovanega študenta ali pa iz seznama vseh predmetov), dobimo takoj seznam vseh študentov, ki so napravili izpit iz tega

predmeta, vključno z datumi in ocenami. In enako velja za seminarske naloge.

K vsakemu objektu lahko v *komentar* vpišemo opazke, povezave z dokumenti ali tabelami, slike, skratka vse, kar nudi danes računalniška tehnika. Ko na ekran priključimo objekt, vidimo poleg atributov tega objekta istočasno še vse komentarje, ki dodatno pojasnjujejo opazovani objekt.

S tem se ponujajo izjemne možnosti za opise zares kompleksnih podatkovnih struktur, ki jih ljudje sami z lahkoto obvladamo, za uporabo na računalniku pa jih danes še vedno klestimo in tlačimo v okvire možnosti, ki nam jih nudijo razpoložljiva orodja.

## Način in področja uporabe

Panorama je namenjena širokemu krogu uporabnikov, po eni strani kot osebno orodje, po drugi strani kot dodaten informacijski sistem celotnega podjetja za tisti del potreb, ki jih obstoječi informacijski sistemi ne pokrivajo. V prvi fazi ni mišljena kot orodje za obdelavo podatkov v klasičnem smislu obdelave numeričnih podatkov za potrebe računovodstva ampak kot orodje za obvladovanje informacij, ki so manj formalizirane, so pa zelo kompleksne v svoji strukturi in povezanostih.

Enostavnost je bil eden osnovnih kriterijev zasnove. Poizvedovanja po kompleksnih podatkovnih strukturah zato temeljijo na realizaciji v modernem grafičnem okolju, ki omogoča preprosto delo po prototipnem načelu: *kar vidim, to obstaja in to lahko izberem*.

Za preiskovanje ni potrebno znanje poizvedovalnih jezikov.

Princip prikaza atributov objektov je podoben človeškemu opazovanju sveta okoli sebe: mi vedno vidimo množico predmetov, urejenih v prostoru, in s

razredi: ←	dodaj	spremeni	komentar																																								
<ul style="list-style-type: none"> <li>razredi</li> <li>absolvent</li> <li>države</li> <li>index</li> <li>javni nastopi</li> <li>nagrade</li> <li>repertoar/stud. leto</li> <li>spisek po letnikih</li> <li>status</li> <li>študenti</li> <li>studijsko leto</li> <li>tekmovanja</li> <li>relacije</li> <li>gradniki izbranega razreda: <ul style="list-style-type: none"> <li>I - J</li> <li>K - L</li> <li>M - N</li> <li>O - P</li> </ul> </li> </ul>	<p>naziv Orozco Estrada, A.Fel.</p> <p>prihaja iz države</p> <p>status studiral od do 1</p> <p>redni študent 1997 -</p> <table border="1"> <thead> <tr> <th>studijsko leto</th> <th>sem.</th> <th>glavni predmet</th> <th>ocena</th> <th>Studienabschnitt 7</th> </tr> </thead> <tbody> <tr><td>1997/98</td><td>2</td><td>osnove dr.</td><td>3</td><td>1</td></tr> <tr><td>1997/98</td><td>1</td><td>osnove dr.</td><td>3</td><td>1</td></tr> <tr><td>1998/99</td><td>3</td><td>osnove dr.</td><td>2</td><td>1</td></tr> <tr><td>1998/99</td><td>4</td><td>osnove dr.</td><td>2</td><td>1</td></tr> <tr><td>1999/2000</td><td>1</td><td>dirigranje</td><td>1</td><td>2</td></tr> <tr><td>1999/2000</td><td>2</td><td>dirigranje</td><td>1</td><td>2</td></tr> <tr><td>2000/01</td><td>6</td><td>dirigranje</td><td>1</td><td>2</td></tr> </tbody> </table>		studijsko leto	sem.	glavni predmet	ocena	Studienabschnitt 7	1997/98	2	osnove dr.	3	1	1997/98	1	osnove dr.	3	1	1998/99	3	osnove dr.	2	1	1998/99	4	osnove dr.	2	1	1999/2000	1	dirigranje	1	2	1999/2000	2	dirigranje	1	2	2000/01	6	dirigranje	1	2	<p>polno ime: Orozco Estrada, Andres Felipe. 16.junija je z odliko opravil 1. diplomu. Sodeloval na tekmovanju Mataack v Zagrebu in žal takoj izpadel, čeprav je zaskušil boljše mesto. Silovito se razvija.</p>
studijsko leto	sem.	glavni predmet	ocena	Studienabschnitt 7																																							
1997/98	2	osnove dr.	3	1																																							
1997/98	1	osnove dr.	3	1																																							
1998/99	3	osnove dr.	2	1																																							
1998/99	4	osnove dr.	2	1																																							
1999/2000	1	dirigranje	1	2																																							
1999/2000	2	dirigranje	1	2																																							
2000/01	6	dirigranje	1	2																																							
	<table border="1"> <thead> <tr> <th>javni nastopi</th> <th>repertoar/stud. leto</th> <th>avtor</th> <th>dvorana</th> <th>ansambel</th> <th>vrstni red nastop 1</th> </tr> </thead> <tbody> <tr> <td>2000_01_19</td> <td>koncert</td> <td>1.Orchester:sinfonie D.</td> <td>C.Ph.E.Bach</td> <td>Haydn-Saal</td> <td>Pro Arte</td> <td>0</td> </tr> </tbody> </table>		javni nastopi	repertoar/stud. leto	avtor	dvorana	ansambel	vrstni red nastop 1	2000_01_19	koncert	1.Orchester:sinfonie D.	C.Ph.E.Bach	Haydn-Saal	Pro Arte	0																												
javni nastopi	repertoar/stud. leto	avtor	dvorana	ansambel	vrstni red nastop 1																																						
2000_01_19	koncert	1.Orchester:sinfonie D.	C.Ph.E.Bach	Haydn-Saal	Pro Arte	0																																					
	<table border="1"> <thead> <tr> <th>spisek po letnikih</th> <th>tip st.</th> <th>1.sem.</th> <th>2.sem.</th> <th>studijska smer</th> <th>leto st. 4</th> </tr> </thead> <tbody> <tr><td>1997/98</td><td>0</td><td>3E</td><td>3E</td><td>osnove dirigranja</td><td>1</td></tr> <tr><td>1998/99</td><td>0</td><td>U2</td><td>U2</td><td>osnove dirigranja</td><td>2</td></tr> <tr><td>1999/2000</td><td>0</td><td>S1</td><td>S1</td><td>dirigranje</td><td>3</td></tr> <tr><td>2000/01</td><td>0</td><td>S1</td><td></td><td>dirigranje</td><td>4</td></tr> </tbody> </table>		spisek po letnikih	tip st.	1.sem.	2.sem.	studijska smer	leto st. 4	1997/98	0	3E	3E	osnove dirigranja	1	1998/99	0	U2	U2	osnove dirigranja	2	1999/2000	0	S1	S1	dirigranje	3	2000/01	0	S1		dirigranje	4											
spisek po letnikih	tip st.	1.sem.	2.sem.	studijska smer	leto st. 4																																						
1997/98	0	3E	3E	osnove dirigranja	1																																						
1998/99	0	U2	U2	osnove dirigranja	2																																						
1999/2000	0	S1	S1	dirigranje	3																																						
2000/01	0	S1		dirigranje	4																																						

pogledom se sprehajamo po njih. Opazovani predmet ima svoje lastnosti in vsi predmeti okoli njega so v neki zvezi z njim.

Panorama prikazuje objekte na podoben način. Najprej iz množice objektov razreda izberemo objekt in zatem dobimo na ekran vse njegove atribute. Nekateri od teh atributov so povezovalni atributi, ki predstavljajo povezavo opazovanega objekta z objekti njegove okolice, njegovega konteksta. Na istem ekranu dobimo vpogled v množico datotek, v vsebino zapisov, ki so v povezavi z opazovanim osrednjim objektom. Ko z miško izberemo tak povezovalni atribut (izbrani zapis ene od prikazanih datotek), se preselimo v opazovanje novega objekta, s katerim je prej obravnavani objekt povezan. Takoj dobimo na ekran atribute novega objekta in eden od povezovalnih atributov tega objekta je objekt, ki smo ga opazovali prej.

Na ta način potujemo po lahko zelo kompleksnem informacijskem prostoru, ki ga tvorijo v bazi zapisani objekti, pripadajoči različnim razredom in povezani med seboj s prepredenimi nitmi povezav. Pomembno pri tem potovanju pa je to, da ni potrebe po vpisovanju ukazov, s katerimi določamo smer in cilj potovanja: vedno so na ekranu prikazane naslednje možne postaje našega potovanja.

Področja možne uporabe Panorame so zelo široka in obsegajo predvsem področja, ki so informacijsko

bogata in raznovrstna in zahtevajo kompleksna potovanja po informacijskih strukturah, opazovanja objektov z različnih vidikov: knjižnice, osebni arhivi, arhivi pomembnih dogodkov podjetja, večjezikovni slovarji, katastri, rodbinska drevesa, pravni in upravni informacijski sistemi, področje avtomatizacije pisarniškega poslovanja, informacijski sistem komunalne mreže, bolnišnični informacijski sistem, spremljanje projektov, arhivi umetniških ustanov, ekološki informacijski sistem, gospodarski informacijski sistem ponudb in povpraševanja po izdelkih ipd.

Pomembno je to, da s tem produktom lahko obvladujemo področja, ki pomenijo za običajno računalniško podporo težave in so na podatkovni ravni informacijsko in strukturno bogata in prepletana.

## Reference:

- J.F.Vilbert: Introduction to neural networks, DECUS-Europe, Haag, september 1989
- Iztok Lajovic, Marko Čibej: Panorama: enostavnost kompleksnih informacij, Računalništvo v javni upravi, Brdo 1993
- Leksikon, razvojna dokumentacija, Iskra Delta, Ljubljana, 1985
- VLEX, Iskra Delta, Ljubljana 1985

Iztok Lajovic se je po diplomi leta 1970 na Fakulteti za elektrotehniko ukvarjal z raziskavami umetnih jezikov in računalniških omrežij na Institutu Jožef Stefan. V letih od 1975 do 1980 je vodil računski center podjetja Tekstil v Ljubljani in uvedel enega prvih celovitih on-line sistemov v Sloveniji. V letih 1974 do 1986 je sodeloval je v projektnih skupinah COST 11 pri Evropski gospodarski skupnosti pri razvoju računalniških omrežij. Od leta 1980 do konca 1990 je vodil razvoj programske opreme najprej v Gospodarski banki Ljubljana, zatem pa v LB, Združeni banki. Od leta 1991 je direktor podjetja KreS, Kreativni sistemi, d.o.o