

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

SUSTAV ZA PREPOZNAVANJE UZORAKA PRI VARIJANTNOM KONSTRUIRANJU

MAGISTARSKI RAD

Mentor:
Dr. sc. Dorian Marjanović, izv.prof.

Damir Salopek, dipl. inž.

Zagreb, 2002.

PODACI ZA BIBLIOGRAFSKU KARTICU

UDK: 658.512.2

Ključne riječi:	konstruiranje pomoću računala, rasuđivanje temeljem uzoraka, varijantna konstrukcija, parametarska konstrukcija, konstruiranje energetskog transformatora
Znanstveno područje:	Tehničke znanosti
Znanstveno polje:	Strojarstvo
Institucija u kojoj je rad izrađen:	Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje
Mentor rada:	Dr.sc. Dorian Marjanović, izv.prof.
Broj stranica:	110
Broj slika:	53
Broj tablica:	12
Broj korištenih bibliografskih jedinica:	81
Datum obrane:	
Povjerenstvo:	Dr.sc. Bojan Jerbić, izv.prof. Dr.sc. Dorian Marjanović, izv.prof. Dr.sc. Jože Duhovnik, red.prof Fakulteta za strojništvo, Ljubljana
Institucija u kojoj je rad pohranjen:	Fakultet strojarstva i brodogradnje, Zagreb Nacionalna i sveučilišna knjižnica, Zagreb



Zagreb, 2001-06-05

Zadatak za magistarski rad

Kandidat: *Damir Salopek, dipl. inž.*

Naslov zadatka: **Sustav za prepoznavanje uzoraka pri varijantnom konstruiranju**

Sadržaj zadatka:

Pri rješavanju varijantnih konstrukcijskih problema konstruktori koriste rješenja prijašnjih, sličnih konstrukcijskih zadataka. Potrebno je istražiti mogućnosti razvoja sustava za podršku konstruktoru temeljem uzoraka prijašnjih konstrukcijskih rješenja.

Definirati idejno rješenje te realizirati sustav za podršku konstruktoru pri konstruiranju energetskih transformatora. Temeljem tehničke specifikacije novog proizvoda omogućiti prepoznavanje sličnih rješenja pohranjenih u bazi izvedenih konstrukcija.

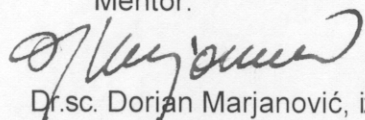
U radu je potrebno:

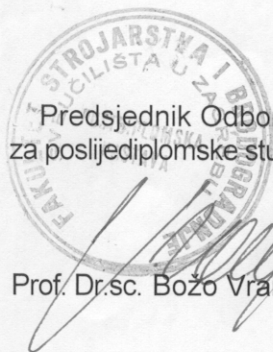
- istražiti metode strukturiranja i pohranjivanja konstrukcijskih parametara varijantnih konstrukcijskih rješenja,
- predložiti strukturu konstrukcijskih parametara energetskih transformatora,
- kreirati bazu konstrukcijskih rješenja energetskih transformatora,
- postaviti kriterije sličnosti konstrukcijskih rješenja,
- razraditi algoritam određivanja sličnih rješenja,
- razraditi metode izbora pogodnih rješenja,
- omogućiti adaptaciju izabranih rješenja unutar postojećeg CAD sustava,
- razraditi metode pohranjivanja verificiranih konstrukcija u bazu konstrukcijskih rješenja na razini sklopova.

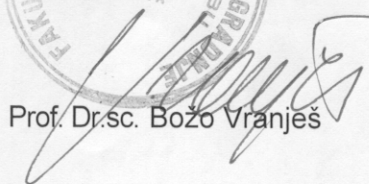
Zadatak zadan: *12. 06. 2001.*

Rad predan:

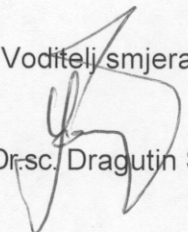
Mentor:


Dr. sc. Dorijan Marjanović, izv. prof.


Predsjednik Odbora
za poslijediplomske studije:


Prof. Dr. sc. Božo Vranješ

Voditelj smjera:


Prof. Dr. sc. Dragutin Ščap

Zahvaljujem mentoru, Dr. sc. Dorianu Marjanoviću, izv. prof. na podršci, savjetima i primjedbama tijekom izrade ovog rada. Članovi povjerenstva, sa svojim primjedbama i savjetima, također su pomogli povećanju kvalitete ovog rada.

Zahvaljujem se Ivanu Milčiću, dipl. inž. koji je, u ime tvrtke Končar Energetski transformatori d.o.o., omogućio da dio radnog vremena odvojim na izradu ovog rada, te na financijskoj podršci.

Hvala kolegama iz Laboratorija za osnove konstruiranja Fakulteta strojarstva i brodogradnje na znanju i iskustvu stečenom na zajedničkom projektu uvođenja CAD sustava u tvrtku Končar Energetski transformatori d.o.o.. Posebno se zahvaljujem kolegi Mr. sc. Mariu Štorgi na korisnim savjetima i raspravama.

Veliko hvala supruzi Nives i kćerima Steli i Izabeli na podršci i strpljenju.

SADRŽAJ

PREDGOVOR.....	V
SAŽETAK.....	VI
SUMMARY	VII
1 UVOD	1-1
1.1 OPIS PROBLEMA I CILJ ISTRAŽIVANJA.....	1-1
1.2 STRUKTURA RADA.....	1-3
2 PROCES KONSTRUIRANJA	2-1
2.1 ZNAČAJKE PROCESA KONSTRUIRANJA	2-1
2.1.1 Razine procesa konstruiranja.....	2-3
2.2 VARIJANTNO KONSTRUIRANJE.....	2-5
2.3 FAZE PROCESA KONSTRUIRANJA	2-7
2.4 RAZVOJ ENERGETSKOG TRANSFORMATORA.....	2-10
2.4.1 Struktura energetskog transformatora	2-13
2.4.2 Osnovni sklop jezgre	2-14
2.4.3 Osnovni sklop konzervatora	2-16
2.4.4 Značajke postojećeg CAE sustava.....	2-17
3 RASUĐIVANJE TEMELJEM UZORAKA	3-1
3.1 OPIS I POVIJESNI RAZVOJ.....	3-1
3.2 CIKLUS RASUĐIVANJA.....	3-4
3.3 ZAPISIVANJE UZORAKA	3-5
3.4 METODE PRONALAŽENJA NAJSLIČNIJIH RJEŠENJA	3-7
3.4.1 Metoda najbliži susjed.....	3-7
3.4.1.1 Težinski faktori.....	3-9
3.4.2 Induktivna metoda.....	3-10
4 ZAPISIVANJE UZORAKA	4-1
4.1 PARAMETAR KONSTRUKCIJE	4-1
4.1.1 Organizacijska struktura parametara	4-2
4.1.2 Značajke čvornih parametara	4-3
4.1.3 Tip vrijednosti parametra	4-4
4.2 OPIS ZADATAKA I UZORKA	4-4
4.3 INFORMACIJSKI MODEL BAZE UZORAKA	4-6
5 PRONALAŽENJE NAJSLIČNIJIH UZORAKA	5-1
5.1 INFORMACIJSKI MODEL	5-1
5.2 PRETRAŽIVANJE POGODNIH UZORAKA	5-2
5.3 RAČUNANJE TEŽINSKIH FAKTORA	5-3
5.3.1 Ocjenjivanje prednosti.....	5-3
5.3.2 Određivanje prednosti	5-5
5.3.3 Primjer izračunavanja težinskih faktora	5-6
5.4 PRIMJER IZRAČUNAVANJA SLIČNOSTI.....	5-7

6	POHRANJIVANJE NOVOG RJEŠENJA.....	6-1
6.1	INFORMACIJSKI MODEL	6-1
6.2	PRIKUPLJANJE VRIJEDNOSTI PARAMETARA	6-2
6.3	PRIKUPLJANJE PODATAKA IZ PARAMETARSKOG MODELA	6-3
6.4	POHRANJIVANJE UZORKA	6-5
6.5	DODAVANJE SKLOPOVA I PARAMETARA	6-6
7	IMPLEMENTACIJSKI MODEL SUSTAVA	7-1
7.1	STRUKTURA SUSTAVA ZA PREPOZNAVANJE UZORAKA.....	7-2
7.1.1	Scenarij pronalaženja najsličnijeg uzorka	7-5
7.1.2	Scenarij pohranjivanja novog rješenja.....	7-6
7.2	ORGANIZACIJA BAZE.....	7-7
7.3	MODULI SUSTAVA.....	7-12
7.3.1	Osnovni modul	7-12
7.3.2	Modul za vezu sa bazom uzoraka.....	7-14
7.3.3	Modul za upravljanje uzorcima	7-16
7.3.4	Modul za administriranje.....	7-18
7.3.5	Sučelje prema CAD sustavu.....	7-19
7.3.6	Korisnička sučelja	7-20
8	PRIKAZ PRIMJENE SUSTAVA	8-1
9	ZAKLJUČAK.....	9-1
9.1	REZULTATI RADA.....	9-1
9.2	MOGUĆNOSTI DALJNJEG ISTRAŽIVANJA	9-2
	LITERATURA	9-3
	KRATKI ŽIVOTOPIS	9-8
	SHORT BIOGRAPHY	9-9

POPIS SLIKA:

Slika 2-1: Rastavljanje ukupne funkcije proizvoda na parcijalne funkcije	2-3
Slika 2-2: Povezanost između razina procesa konstruiranja i rezultata [10]	2-4
Slika 2-3: Primjer tablice familije modela	2-6
Slika 2-4: Familija modela dobivena iz osnovnog modela variranjem značajki	2-6
Slika 2-5: Faze procesa konstruiranja prema proceduralnom modelu VDI 2221 [15]	2-9
Slika 2-6: Pojednostavljeni prikaz razvojnog ciklusa energetskeg transformatora	2-10
Slika 2-7: Konstruiranje energetskeg transformatora – faza izrade ponude	2-11
Slika 2-8: Konstruiranje energetskeg transformatora – faza izrade konstrukcijske dokumentacije	2-12
Slika 2-9: Osnovni sklopovi energetskeg transformatora	2-13
Slika 2-10: Struktura sklopa jezgre	2-14
Slika 2-11: Model jezgre	2-15
Slika 2-12: Struktura sklopa konzervatora	2-16
Slika 2-13: Model konzervatora	2-17
Slika 3-1: Organizacija uzoraka prema Schanku	3-2
Slika 3-2: Shematski prikaz ciklusa rasuđivanja	3-4
Slika 3-3: Prostor problema i prostor rješenja	3-6
Slika 3-4: Metoda nabliži susjed za dva parametra	3-8
Slika 3-5: Povlačenje i guranje uzoraka [45]	3-10
Slika 3-6: Prvi čvor u stablu odlučivanja	3-11
Slika 3-7: Primjer stabla odlučivanja	3-12
Slika 4-1: Struktura parametara u bazi parametara	4-3
Slika 4-2: Opisivanje konstrukcije, novog konstrukcijskog zadatka i uzorka	4-5
Slika 4-3: Model baze uzoraka	4-6
Slika 5-1: Informacijski model pronalaženja	5-1
Slika 5-2: Prikaz pretraživanja pogodnih uzoraka	5-2
Slika 6-1: Informacijski model pohranjivanja novog rješenja	6-1
Slika 6-2: Načini prikupljanja vrijednosti parametara	6-2
Slika 6-3: Model nosača	6-4
Slika 6-4: Relacije i korisnički parametri u modelu nosača	6-5
Slika 6-5: Odabir sklopova za uzorke	6-6
Slika 6-6: Dodavanje novih sklopova u strukturu	6-6
Slika 7-1: Dio ciklusa rasuđivanja koji je realiziran u predloženom sustavu	7-2
Slika 7-2: Shematski prikaz sustava za prepoznavanje uzoraka	7-3
Slika 7-3: Struktura sustava za prepoznavanje uzoraka	7-4
Slika 7-4: Tijek pronalaženja najslabijih uzoraka	7-5
Slika 7-5: Tijek pohranjivanja novog rješenja u bazu uzoraka	7-7
Slika 7-6: Prikaz strukture baze uzoraka i relacije između tablica	7-8
Slika 7-7: Klase koje opisuju module sustava i njihove veze	7-12
Slika 7-8: Prikaz definicije klase SPUApp	7-13
Slika 7-9: Prikaz definicije klase SPUVezaBaza	7-15
Slika 7-10: Prikaz definicije klase SPUUzorak	7-17
Slika 7-11: Prikaz definicije klase SPUAdmin	7-18
Slika 7-12: Dijalog za prijavu korisnika	7-19
Slika 7-13: Prikaz definicije klase SPUCad	7-19
Slika 7-14: Prikaz korisničkog sučelja za pronalaženje uzoraka	7-21
Slika 7-15: Prikaz korisničkog sučelja za pohranjivanje uzoraka u bazu	7-22
Slika 7-16: Prikaz korisničkog sučelja za administriranje baze	7-23
Slika 7-17: Prikaz korisničkog sučelja za pregled tablica u bazi uzoraka	7-24
Slika 8-1: Prikaz dijela tablica u bazi uzoraka	8-1
Slika 8-2: Dijagram vrijednosti nekih parametara promatranih uzoraka	8-9
Slika 8-3: Parametarski model najslabijeg uzorka U173	8-10
Slika 8-4: Parametarski model najrazličitijeg uzorka U218	8-11
Slika 8-5: Parametarski model gotovog rješenja novog konstrukcijskog zadatka	8-12

POPIS TABLICA:

Tablica 1: Primjer baze uzoraka	3-11
Tablica 2: Primjer zadatka.....	3-12
Tablica 3: Ocjenjivanje prednosti parametara.....	5-3
Tablica 4: Ocjena prednosti parametara opisnom ocjenom.....	5-4
Tablica 5: Određivanje prednosti parametara.....	5-5
Tablica 6: Neki parametri sklopa konzervatora energetskog transformatora	5-6
Tablica 7: Tablica prednosti s težinskim faktorima.....	5-6
Tablica 8: Primjer izračunavanja sličnosti	5-7
Tablica 9: Informacije potrebne za prikupljanje vrijednosti parametara unutar modela	6-3
Tablica 10: Vrijednosti parametara za pojedine uzorke i zadatak	8-4
Tablica 11: Tablica prednosti promatranih parametara sa izračunatim težinskim faktorima	8-8
Tablica 12: Izračunate vrijednosti sličnosti.....	8-9

PREDGOVOR

Izrada ovog rada potaknuta je višegodišnjim radom autora na rješavanju konstrukcijskih zadataka pri konstruiranju energetskih transformatora, te na poboljšavanju i automatiziranju procesa konstruiranja energetskog transformatora.

Projekt Uvođenje CAD sustava u tvrtku Končar Energetski transformatori d.o.o. također je pomogao izradi ovog rada. Projekt je rađen u suradnji s djelatnicima Laboratorija za osnove konstruiranja Fakulteta strojarstva i brodogradnje.

Osnovni motiv za izradu ovog rada proizlazi iz činjenice da konstruktori, prilikom rješavanja konstrukcijskih zadataka koriste iskustva stečena pri rješavanju sličnih konstrukcijskih zadataka koristeći prijašnja konstrukcijska rješenja kao podloge.

Korištenjem sustava za prepoznavanje prijašnjih, sličnih konstrukcijskih rješenja očekuje se povećanje efikasnosti primjene CAD sustava, što podrazumijeva bržu izradu konstrukcijske dokumentacije i brzo reagiranje na zahtjeve naručitelja.

SAŽETAK

Zadatak ovog rada je definirati idejno rješenje, te realizirati sustav za prepoznavanje uzoraka pri varijantnom konstruiranju. Predloženi sustav biti će podrška konstruktoru pri konstruiranju energetskih transformatora.

Sustav je zasnovan na metodologiji rasuđivanja temeljem uzoraka (case-based reasoning), gdje uzorak predstavlja prijašnje konstrukcijsko rješenje koje je, na odgovarajući način, pohranjeno u bazu uzoraka. Koristeći prikazani sustav konstruktor, na temelju opisa zadatka, pronalazi najsličnije prethodno konstrukcijsko rješenje i koristi ga kao podlogu za rješavanje novog konstrukcijskog zadatka.

Glavni dio istraživačkog rada odnosi se na razmatranje dviju faza iz tzv. ciklusa rasuđivanja, a to su: pronalaženje najsličnijeg rješenja i pohranjivanje novog uzorka u bazu uzoraka.

Zapisivanje uzoraka u bazu uzoraka realizirano je zapisivanjem vrijednosti parametara konstrukcijskih rješenja u relacijsku bazu podataka.

Pronalaženje najsličnijih uzoraka zasniva se na metodi najbliži susjed gdje se važnost parametara, po kojima se računa sličnost promatranih uzoraka i novog konstrukcijskog zadatka, definira težinskim faktorima.

Konstrukcijska rješenja izvode se pomoću parametarskog trodimenzionalnog CAD sustava. Pohranjivanje uzoraka za buduće korištenje realizirano je na način da se iz parametarskih modela novog konstrukcijskog rješenja prikupljaju vrijednosti parametara i upisuju u bazu uzoraka.

Rješavanje novog konstrukcijskog zadatka realizira se pomoću CAD sustava i nije posebno razmatrano u ovom radu.

Predloženi sustav za prepoznavanje uzoraka implementiran je u CAD sustav Pro/ENGINEER®. Korištenjem Java™ programskog jezika i J-Link® biblioteka realizirane su akcije pronalaženja najsličnijih uzoraka te pohranjivanja novih uzoraka u relacijsku bazu podataka. Relacijska baza podataka realizirana je pomoću SQL Server® baze podataka.

KLJUČNE RIJEČI:

konstruiranje pomoću računala, rasuđivanje temeljem uzoraka, varijantna konstrukcija, parametarska konstrukcija, konstruiranje energetskog transformatora

UDK: 658.512.2

SYSTEM FOR CASE RECOGNITION IN VARIANT DESIGN

SUMMARY

The task of this thesis is to propose a solution and to realize a system for case recognition in variant design. Proposed system will be a support for design of power transformers. System is developed by using the case-based reasoning (CBR) methodology, where case represents previous design solution which is, in appropriate manner, retained in case base. With described system, designer, according to task description, can find the most similar previous design solution and can use it as template for solving new design problem.

The main part of research deals with examining two phases from so called CBR cycle: retrieving the most similar design and retaining new case in case base.

Retaining of cases in case base is realized by recording the values of design parameters in relational database.

Retrieving of most similar cases is based on «nearest neighbour» method where is importance of parameters, by which we count similarity of observed cases and new design task, defined by weighting factors.

Design are created by parametric three-dimensional CAD system. Retaining of cases for future use is realized in a way that we collect parameter values from parametric models of new design. These values are then recorded in case base.

Solving of new design task is realized with CAD system and is not in scope of this thesis.

Proposed system for case recognition is implemented in CAD system Pro/ENGINEER[®]. Actions of retrieving the most similar cases and recording the new cases in relational database are made in Java[™] program language and J-Link[®] libraries. Relational database is developed with SQL Server[®] database.

KEYWORDS:

computer-based design, case-based reasoning, variant design, parametric design, power transformer's design

UDC: 658.512.2

1 UVOD

Današnja civilizacijska i industrijska razvijenost postavlja pred konstruktora vrlo velike zahtjeve. Vijek proizvoda je sve kraći i konstruktor ima sve manje vremena za razvoj proizvoda i izradu konstrukcijske dokumentacije.

Glavni zadatak inženjera je da primjeni svoje znanje u rješavanju tehničkih problema [1] te da optimira rješenje u danim ograničenjima materijala, tehnologije i zahtjeva tržišta. Suvremene zahtjeve tržišta ne bi bilo moguće ispuniti bez razvoja znanosti o konstruiranju i računalnih alata za konstruiranje (CAD¹). Danas se ulažu veliki naponi u razvoj metoda i računalnih alata s ciljem smanjenja troškova izrade konstrukcijske dokumentacije.

Pri rješavanju novih konstrukcijskih zadataka konstruktori obimno koriste prethodna konstrukcijska rješenja [2], [3], [4], kako bi izbjegli greške i ponovno iskoristili provjeren pristup uz odgovarajuće modifikacije. Općenito se može reći da ljudi u svakodnevnom životu rješavaju probleme tako da se prisjećaju kako su takav sličan problem riješili u prošlosti [5]. Ova činjenica koristi se u metodologiji rasuđivanja temeljem uzoraka (Case-Based Reasoning – CBR). U domeni konstruiranja uzorak (case) predstavlja konstrukcijsko rješenje nekog prijašnjeg konstrukcijskog zadatka koje je pohranjeno u bazu uzoraka. Određenim metodama pronalazi se najsličniji zadatak i njemu pripadajuće rješenje koje se koristi kao predložak za rješavanje novog konstrukcijskog zadataka.

1.1 Opis problema i cilj istraživanja

Energetski transformator je proizvod tvrtke Končar Energetski transformatori d.o.o. (Končar Power Transformers Ltd. - KPT) i izrađuje se za poznatog naručitelja (kupca).

U sveukupnim troškovima proizvodnje energetskog transformatora troškovi materijala imaju značajan udio (45% - 60%). Količina bakra koji se koristi za izradu namota transformatora se izračuna električnim proračunom i na osnovu mase i cijene materijala dobiju se troškovi bakra ugrađenog u transformator. Međutim, čelični materijali, izolacijski materijali kao i količina ulja u transformatoru se procjenjuju. Često se događa da procjenjene mase prilično odstupaju od stvarnih masa. O procjeni ugrađenog materijala ovisi i visina ukupnih troškova proizvodnje.

U ranoj fazi razvoja proizvoda (faza izrade ponude) pored ugrađenog materijala procjenjuju se i dimenzije proizvoda (gabaritne mjere, razmak priključaka provodnika, ...). Kod procjene dimenzija, također, se događa da se mjere ne procjene dovoljno točno, što ima utjecaj na povećanje ukupnih troškova razvoja proizvoda. Ukoliko su te mjere predimenzionirane imamo veći utrošak materijala nego što je bilo potrebno, dok s druge

¹ Computer Aided Design

strane, ukoliko su te mjere poddimenzionirane mnogo se vremena troši na razrješavanje konstrukcijskog zadatka, kako u fazi izrade osnutka tako i u kasnijoj konstrukcijskoj razradi sklopova transformatora. Rezultat je mnogo utrošenih radnih sati na izradu konstrukcijske dokumentacije jer se npr. u fazi konstrukcijske razrade ne mogu koristiti uobičajena, standardna rješenja za koja već postoje modeli te prateći crteži.

Energetski transformator po svojim konstrukcijskim osobinama spada u grupu varijantnih konstrukcija. Osnovna funkcija i princip rješenja ostaju isti, a varira se veličina i/ili raspored pojedinih dijelova. Zbog toga se često pri koncipiranju konstrukcijskog rješenja i konstrukcijskoj razradi koriste prijašnja, slična² konstrukcijska rješenja kao predlošci, podloge za rješavanje novih konstrukcijskih zadataka.

U radu će biti predložen sustav koji će temeljem definicije novog konstrukcijskog zadatka omogućiti prepoznavanje sličnih rješenja pohranjenih u bazi izvedenih konstrukcija. Rješenje sličnog konstrukcijskog zadatka može osim konkretnih podataka o masama i mjerama sadržavati i niz dodatnih informacija o problemima nastalim tijekom svih faza nastanka proizvoda npr. problemi nastali u fazi projektiranja i konstruiranja, problemi nastali pri nabavi materijala, zapisnici kod testiranja proizvoda, transporta, montaže itd. U fazi konstruiranja proizvoda sustav za prepoznavanje uzoraka biti će podrška konstruktoru u svim fazama konstruiranja, a s ciljem ubrzanja procesa konstruiranja te smanjenja grešaka.

Pretpostavka je da će predloženi sustav u fazi izrade ponude pomoći korisniku pri procjeni ugrađenog materijala i pri procjeni dimenzija proizvoda čime će se kvalitetnije procijeniti ukupni troškovi proizvodnje.

U fazi konfiguriranja konstrukcije i konstrukcijske razrade korištenjem rješenja prijašnjeg sličnog zadatka olakšava se rješavanje novog konstrukcijskog zadatka i njegova razrada. Uređivanje crteža koji nastaju iz parametarskih modela predstavlja vremenski najzahtjevniji dio konstrukcijske razrade. Ukoliko se predložak koji se koristi kao polazni model ne razlikuje značajno, strukturno i dimenzijski, od konačnog modela, uređivanje crteža biti će vremenski manje zahtjevno. Iz toga se može pretpostaviti da bi uređivanje crteža bilo olakšano ukoliko bi se kao početni model (predložak) koristio model sličan³ konačnom konstrukcijskom rješenju.

Osnovni cilj istraživanja je razviti sustav za prepoznavanje uzoraka koji će pomoći korisniku kod procjene masa i gabaritnih mjera u početnoj fazi razvoja proizvoda, pomoći konstruktoru pri koncipiranju i projektiranju konstrukcijskog rješenja i utjecati na smanjenje vremena izrade konstrukcijske dokumentacije, te na taj način utjecati na smanjenje ukupnih troškova razvoja proizvoda. Razvijeni sustav potrebno je implementirati u postojeći CAE⁴ sustav tvrtke.

² Ovdje se pojam «sličan» odnosi na sjećanje konstruktora da je već u prošlosti rješavao sličan zadatak, a ne na sustavno pretraživanje sličnih rješenja temeljem egzaktnih pokazatelja.

³ Ovdje se pojam «sličan» odnosi na metodološko određivanje sličnosti korištenjem poznatih metoda.

⁴ Computer Aided Engineering

1.2 Struktura rada

U predloženom sustavu za prepoznavanje uzoraka pri varijantnom konstruiranju analizirane su i relizirane glavne zadaće koje sustav treba ispunjavati: organizacija uzoraka, pronalaženje sličnih uzoraka i pohranjivanje novih uzoraka. Rješavanje novog konstrukcijskog zadatka te njegova verifikacija nisu obuhvaćene ovim radom.

Sustav je testiran na realnim konstrukcijskim rješenjima sklopa jezgre energetskog transformatora, a rezultati istraživanja verificirani su prikazom modela konstrukcijskih rješenja sklopa jezgre energetskog transformatora.

U glavi 2 ukratko se daje pregled procesa konstruiranja, njegove glavne značajke te tipovi konstrukcija. Ukratko su opisane karakteristike varijantnih konstrukcija sa osvrtom na proces konstruiranja energetskog transformatora. Navedene su značajke postojećeg CAE sustava u koji treba implementirati sustav za prepoznavanje uzoraka.

Glava 3 daje pregled metodologije rasuđivanja temeljem uzoraka (CBR) kao dio područja primjene umjetne inteligencije u procesu konstruiranja. Daje se pregled glavnih značajki, temeljni problemi i faze unutar tipičnog CBR sustava (tzv. krug zaključivanja). Biti će naveden kratak pregled nekih CBR sustava razvijenih za različite namjene i različita područja primjene (domene).

Glavni dio istraživanja obuhvatit će:

- ◆ razmatranje načina zapisivanja uzoraka u bazu (glava 4),
- ◆ razmatranje primjene CBR sustava kao dijela realnog CAD okruženja za konstruiranje proizvoda gdje je potrebno analizirati sve ključne aktivnosti: konfiguriranje baze uzoraka, pretraživanje baze, povlačenje uzoraka iz baze, određivanje sličnosti uzoraka i odabiranje najpovoljnijeg uzorka te pohranjivanje novog uzorka u bazu (glave 5 i 6),
- ◆ implementacijski model sustava za prepoznavanje uzoraka pri konstruiranju varijantnih konstrukcija (glava 7),
- ◆ ispitivanje predloženog modela na stvarnim konstrukcijama realnog CAD sustava za konstruiranje transformatora (glava 8).

U glavi 9 analizirani su dobiveni rezultati istraživanja i predložene smjernice za daljnja istraživanja.

2 PROCES KONSTRUIRANJA

Zadatak procesa konstruiranja je stvoriti optimalni proizvod u zadanim okolnostima u što kraćem vremenu i sa što manjim troškovima [6]. Pri čemu je uloga konstruktora da primjeni usvojeno znanje u rješavanju tehničkih problema i da optimira to rješenje u danim ograničenjima materijala, tehnologije i ekonomije. Konstruiranje je intelektualno nastojanje da se zadovolje određeni zahtjevi na najbolji mogući način. To je inženjerska aktivnost koja djeluje na gotovo svaku sferu ljudskog života, temelji se na otkrićima i zakonima znanosti i kreira uvjete za primjenu tih zakona na proizvodnju korisnih proizvoda [1].

2.1 Značajke procesa konstruiranja

Mnogi autori opisuju proces konstruiranja na različite načine i sa različitih aspekata. Ovdje su navedene značajke procesa konstruiranja i konstruktorskog rada prema Hubki [7]:

- ◆ Konstrukcijski proces je u prvom redu proces prerade i generiranja informacija, gdje se na temelju ulaznih zahtjeva generira skup informacija koje opisuju proizvod.
- ◆ Konstrukcijski proces je sinteza relativno dobro poznatih elemenata u jednu jedinstvenu, otprije nepoznatu cjelinu sa zahtjevanim određenim svojstvima. Ta sinteza iziskuje kreativan, stvaralački rad. Iz toga proizlazi važna značajka procesa konstruiranja - čovjek mora kontrolirati proces, odnosno imati pretežan udio u obavljanju potrebnih radnji.
- ◆ S filozofskog gledišta proces konstruiranja je također i spoznajni proces: sustav na početku nepoznat - spoznaje se, odnosno postaje poznat. Na temelju toga može se reći da je spoznajna teorija također jedan izvor općih zakonitosti za proces konstruiranja.
- ◆ Konstruiranje se može promatrati i kao proces učenja.
- ◆ Svaki konstrukcijski zadatak može se riješiti na mnogo različitih načina, odnosno može rezultirati sa različitim strojnim sustavima ili sklopovima. Ta karakteristična mnogostrukost mogućih rješenja uvjetovana je količinom svojstava proizvoda koje se trebaju odrediti u postupku konstruiranja.
- ◆ Svaki proces konstruiranja može se razložiti u manje cjeline (faze, dijelove procesa, etape, operacije) koje čine strukturu procesa.
- ◆ Velika kompleksnost međusobno proturječnih zahtjeva dovodi do potrebe za višestrukim ponavljanjem određenih faza nakon početnog apstrahiranja i postavljanja pretpostavki - dok se ne odrede potrebne vrijednosti. Iterativni postupak je jedna od tipičnih značajki konstruiranja.

- ◆ Do sada pretežno samostalna djelatnost (u okviru zadatka), sve više se pretvara u timski rad u kojem se koriste prednosti većeg informacijskog kapaciteta i međusobne razmjene ideja i postupaka.
- ◆ Proces konstruiranja je vrlo zahtjevan kreativni rad, ali ne smije se promatrati kao umjetnost, nego kao znanstveni rad. Određeni misaoni postupci (intuicija, nastajanje ideje) koji se ne mogu racionalno objasniti imaju obilježja umjetničke kreativnosti. Ti postupci ne mogu se formalizirati u svrhu stvaranja cjelovitog teoretskog prikaza konstrukcijskog procesa.

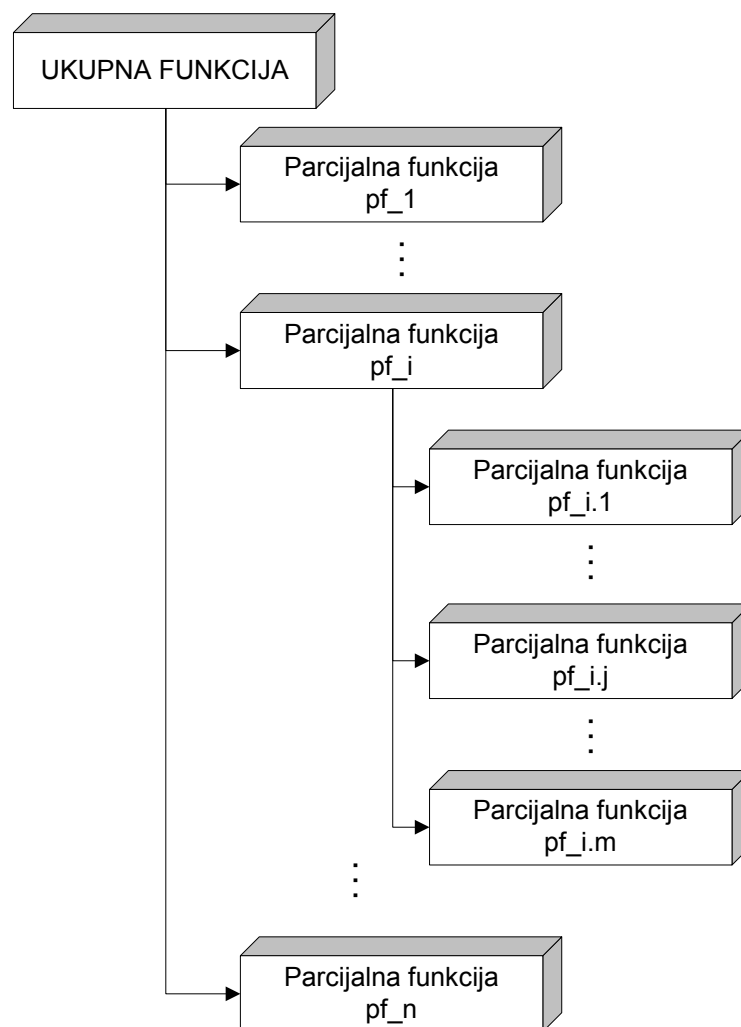
Iz navedenih značajki procesa konstruiranja može se uočiti mnogostrukost upliva na proces konstruiranja, kompleksnost procesa, a i širina potrebnih znanja. Kompleksnost procesa konstruiranja i mnogostrukost upliva okoline čine zadatak modeliranja procesa konstruiranja vrlo zahtjevnim. To se očituje i u velikom broju različitih stavova o konstruiranju i u nizu različitih pristupa modeliranju procesa konstruiranja.

Prema [6] procesom konstruiranja potrebno je definirati i opisati proizvod koji će proizvoditi željene učinke i pri tome zadovoljavati zahtjevana svojstva. Osim postavljenih zahtjeva potrebno je voditi računa i o ostalim aspektima tehničke i ekonomske naravi, kao npr. tehnologičnosti, troškovima, i dr. Često se u praksi javljaju slučajevi kada tehnički zahtjevi na proizvod nisu točno ili u potpunosti definirani što može rezultirati nesporazumima i povećanim troškovima te prekoračenjima rokova.

Svaki proizvod ima svoju tehničku funkciju [1], [6] koja se može rastaviti na parcijalne funkcije koje predstavljaju parcijalne zadatke unutar ukupnog zadatka. Nadalje se može i parcijalna funkcija rastaviti na parcijalne podfunkcije i tako dalje ovisno o proizvodu.

Ukupnu funkciju, strukturu proizvoda možemo prikazati hijerarhijski (Slika 2-1).

Na području znanosti o konstruiranju danas se mnogo napora ulaže na istraživanja kako u nekoj realnoj radnoj okolini izdvajati i čuvati konstrukcijsko znanje. Sve više se smatra da je intelektualno vlasništvo važnije od tehnologija jer znanje nije moguće steći preko noći. Za konkurentnu, uspješnu tvrtku vrlo je važno da zna čuvati i brzo dolaziti do znanja svojih zaposlenika. Nakupljeno iskustvo za postojeći proizvodni program omogućuje brže odazivanje na specifične zahtjeve kupca, odnosno tržišta. Mnogi znanstvenici bave se tim problemom i za očekivati je da će u budućnosti potpora znanjem biti dio svakog konstruktorskog radnog okoliša. Razvijeno je već više prototipova radnih okolina za potrebe konstruktora [8], [9], koji između ostalog omogućavaju čuvanje znanja i pravila o konstruiranju proizvodnog programa. Uglavnom se ostalo na prototipovima jer su radne okoline ostale na uskom području upotrebe i zahtjevaju pažljivi rad pri njihovom razvijanju. Problem je, također, i u ljudskom faktoru jer konstruktori žele ostati kreatori, a ne samo upravljači inteligentnog sustava.



Slika 2-1: Rastavljanje ukupne funkcije proizvoda na parcijalne funkcije

2.1.1 Razine procesa konstruiranja

Svaki konstrukcijski proces započinje definiranjem konstrukcijskog zadatka kojim opisujemo zahtjeve koje proizvod mora ispuniti nakon završetka procesa konstruiranja. Zahtjeve koji opisuju konstrukcijski zadatak možemo razlikovati kao potpuno nove neprepoznate zahtjeve (revolucionarni pristup) ili nadograđivanje prepoznatih funkcija s malobrojnim dodatnim zahtjevima (evolucijski pristup) [10]. U skladu sa zahtjevima i proces konstruiranja se jednom pokreće za potpuno novi proizvod, dok drugi put samo za dopunjavanje funkcije proizvoda. Zbog toga su konstrukcijski zadaci i po opsegu i po zahtjevima dosta različiti.

Razlikujemo četiri vrste procesa konstruiranja koje s obzirom na rezultate koji iz njih proizlaze možemo podijeliti u četiri razine [10] (Slika 2-2):

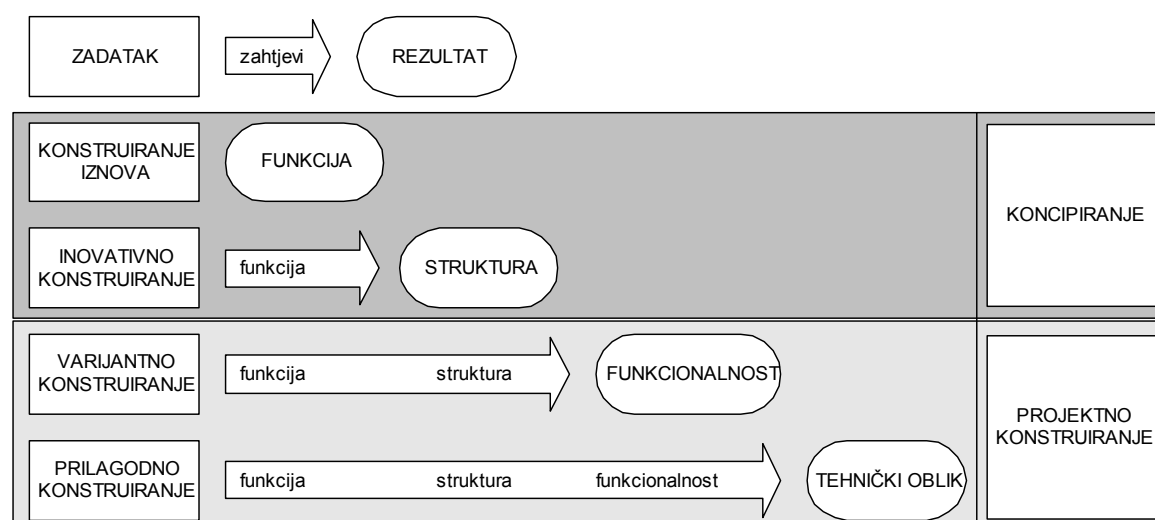
Konstruiranje iznova predstavlja konstruiranje proizvoda koji ima potpuno novi princip rada, koji se temelji na već poznatom fizikalnom zakonu ili na potpuno novom. Princip rada je, u načelu, prikazan sa shemom. Za predstavljanje jednog zakona moguće je

upotrijebiti jedan ili više principa rada. Na taj način nastaju izumi koji dobiju naziv patent s patentnom licencom.

Inovativno konstruiranje predstavlja konstruiranje proizvoda gdje je korištenje određenog zakona definirano i gdje variramo principe rada. Inovacija nastaje uslijed kombiniranja različitih principa rada. Tako razvijeni proizvod postaje izum. Najviše izuma izvedeno je pomoću kombinacije različitih principa rada. Razlog je u tome što su principi rada shematizirani i utjecaj materijala još nije prepoznat u shematiziranom obliku. U trenutku kad uključimo materijal shematizirani oblik značajno se promijeni i funkcije proizvoda dobiju potpuno nova značenja.

Varijantno konstruiranje predstavlja konstruiranje proizvoda čiji je princip rada definiran i određen. Pred nama je proizvod koji ispunjava funkcijske zahtjeve i mora zadovoljiti još samo funkcionalnost. Zbog toga na toj razini konstrukcijskog procesa dimenzioniramo proizvod u cjelosti sa svim detaljima. Zbog različitih zahtjeva ovdje se javljaju različiti oblici proizvoda koje opisujemo sa različitom strukturom i topologijom od one koja je bila predstavljena u početku. U slučaju da su promjene velike obično govorimo o inovacijama. Varijantno konstruiranje nije skup različitih modula sa različitim funkcijama (inovativno konstruiranje) već je to varijacija oblika zbog različitih opterećenja, materijala itd.

Prilagodno konstruiranje predstavlja konstruiranje proizvoda koji se u detaljima prilagođava određenim novim uvjetima, materijalu, tehnologiji, dimenzijskim ograničenjima itd. To je najviše definirana razina konstruiranja i najviše zahtjeva specifično poznavanje proizvodnih tehnologija i proizvodnog prostora. Zbog toga se često prilagodno konstruiranje obavlja neposredno u proizvodnji ili montaži.



Slika 2-2: Povezanost između razina procesa konstruiranja i rezultata [10]

U nastavku će pobliže biti opisano varijantno konstruiranje iz razloga što je predloženi sustav ograničen na varijantne konstrukcije.

2.2 Varijantno konstruiranje

Osnovna karakteristika varijantnih konstrukcija je da su osnovna funkcija i struktura proizvoda definirane i ne mijenjaju se, ali se eventualno mijenjaju principi rada pojedinih parcijalnih funkcija [1]. Variraju se veličine i razmještaj unutar granica odabranog sustava koji se ne mijenja. U literaturi se može pronaći podatak da gotovo 80% svih konstrukcijskih zadataka pripada varijantnom konstruiranju [11]. Neki autori navode da je taj udio i veći i da iznosi 90% [12], [13] i da se pri tome oko 70% informacija iz prijašnjih rješenja ponovno koristi. Ponovno korištenje konstrukcijskih rješenja uglavnom se svodi na razvoj specifičnih i/ili standardnih dijelova, ali isto tako i na iskustvo sadržano u konstrukciji takvih dijelova. Konstruktori mogu ponovno iskoristiti cjelokupnu konstrukciju dijela, mogu iskoristiti oblik dijela za drugu funkciju ili mogu iskoristiti neku značajku⁵ iz konstrukcije.

Pod pojmom varijantno konstruiranje često se podrazumijevaju tehnike adaptacije postojećih konstrukcijskih rješenja s ciljem zadovoljavanja novih zahtjeva i ograničenja [14]. Na taj se način dobiju nova ali slična konstrukcijska rješenja. Međutim, ostaje problem pronalaženja sličnih konstrukcijskih rješenja kako bi se mogle izraditi nove varijante rješenja. Mehanizmi pronalaženja sličnih rješenja dolaze u širokom rasponu složenosti od ručnog pretraživanja do automatske identifikacije.

Suvremeni komercijalni CAD sustavi koriste dvije tehnike koje su značajne za modeliranje varijantnih konstrukcijskih rješenja: modeliranje značajkama i parametarsko modeliranje.

Modeliranje značajkama je tehnika koja omogućava korisniku modeliranje kombiniranjem značajki koje možemo shvatiti kao gradivni dio definicije proizvoda ili geometrijskog poimanja proizvoda (kao npr. rupe, utori, rebra, i dr.) [11]. Pored unaprijed definiranih značajki mnogi sustavi omogućavaju korisniku definiranje vlastitih značajki.

Parametarsko konstruiranje nazivamo karakteristiku suvremenih CAD sustava koja omogućava korisniku generiranje novih izvedbi konstrukcijskog rješenja mijenjajući vrijednosti parametara koji implicitno ili eksplicitno opisuju attribute konstrukcijskog rješenja. Pored mijenjanja parametara korisniku je omogućeno i definiranje relacija među parametrima [14] (npr. dubina utora je pola njegove duljine).

U literaturi se termini "parametarska konstrukcija" i "varijantna konstrukcija" praktički izjednačavaju i koriste se u istom kontekstu [14], [11].

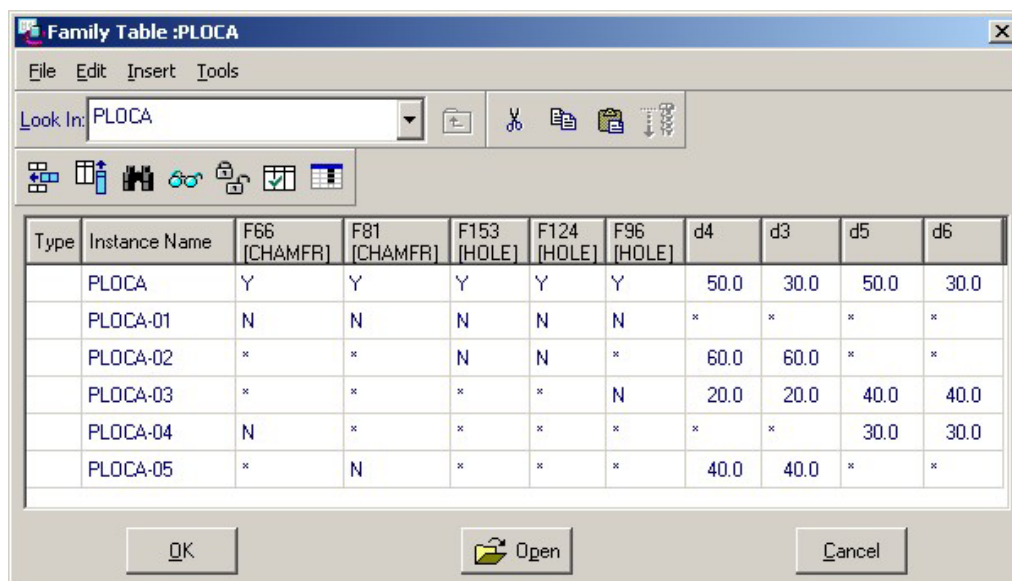
Postoje dva osnovna načina dobivanja varijantne/parametarske konstrukcije [11]:

- ◆ generiranjem novih izvedbi i modificiranjem postojeće (slične) konstrukcije
- ◆ zadavanjem vrijednosti svih parametara koji su ulazni podaci za proceduru koja stvara geometrijski model

Slika 2-3 prikazuje mogućnost generiranja novih izvedbi (instanci) modela iz osnovnog modela (generic) variranjem odgovarajućih parametara unutar tablice koju definira korisnik. Takva tablica opisuje grupu izvedbi nekog modela, tzv. familiju modela, pa se još naziva i tablica familije modela (family table). Kao varijacijski parametar može se koristiti bilo koja značajka (feature) trodimenzionalnog modela: kota, korisnički definirani parametri te dijelovi (part) i sklopovi (assembly). U primjeru tablice familije modela (Slika 2-3) iz osnovnog modela PLOCA.PRT generiraju se izvedbe PLOCA-01, ...,

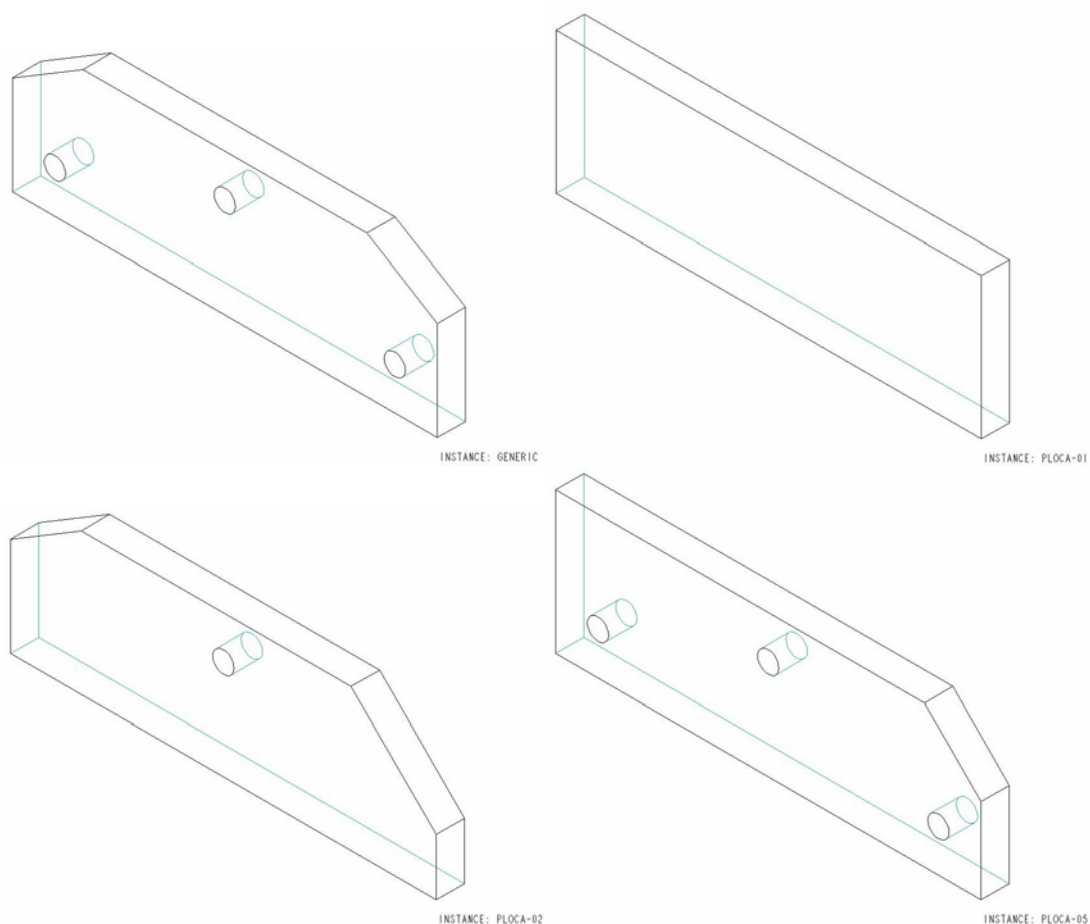
⁵ Pojam značajka odnosi se na konstruiranje značajkama (Feature based design)

PLOCA-05. Stupci sadrže vrijednosti značajke za određenu izvedbu. Ako se radi o geometrijskoj značajki vrijednost može biti Y, N ili *. Zvijezdica (*) označava da vrijednost odgovara osnovnom modelu.



Type	Instance Name	F66 [CHAMFR]	F81 [CHAMFR]	F153 [HOLE]	F124 [HOLE]	F96 [HOLE]	d4	d3	d5	d6
	PLOCA	Y	Y	Y	Y	Y	50.0	30.0	50.0	30.0
	PLOCA-01	N	N	N	N	N	*	*	*	*
	PLOCA-02	*	*	N	N	*	60.0	60.0	*	*
	PLOCA-03	*	*	*	*	N	20.0	20.0	40.0	40.0
	PLOCA-04	N	*	*	*	*	*	*	30.0	30.0
	PLOCA-05	*	N	*	*	*	40.0	40.0	*	*

Slika 2-3: Primjer tablice familije modela



Slika 2-4: Familija modela dobivena iz osnovnog modela variranjem značajki

Slika 2-4 prikazuje familiju modela dobivenu variranjem značajki unutar osnovnog modela prema tablici familije. Na slici je prikazan osnovni model (generic) i izvedbe PLOCA-01, PLOCA-02 i PLOCA-05.

2.3 Faze procesa konstruiranja

Svaki konstrukcijski proces može se raščlaniti na manje cjeline koje predstavljaju faze (korake, operacije, etape) koje čine njegovu strukturu.

Prema [1] svaki konstrukcijski proces može se podijeliti u četiri glavne faze :

1. raščišćavanje zadatka;

Predstavlja početnu fazu konstrukcijskog procesa odnosno tijeka razvoja proizvoda i obuhvaća prikupljanje informacija o zahtjevima koji moraju biti ispunjeni konstrukcijskim rješenjem, ali isto tako i o ograničenjima konstrukcijskog rješenja. Kod izrade proizvoda za poznatog kupca zahtjevi i ograničenja zapisuju se u obliku tehničke specifikacije proizvoda i često proces konstruiranja krene prije nego što su potpuno definirani svi zahtjevi od strane kupca. Kod pojedinačnog načina proizvodnje često se dešava da se zahtjevi kupca mijenjaju tijekom konstrukcijskog procesa što dodatno komplicira fazu raščišćavanja konstrukcijskog zadatka.

2. koncipiranje;

Obuhvaća kreiranje funkcionalne strukture proizvoda, traženje odgovarajućeg tehnološkog principa rješenja i kombiniranje koncepcijskih varijanti rješenja. Koncipiranje rješenja sastoji se od nekoliko koraka. Niti jedan od tih koraka ne bi se smio preskočiti ukoliko se želi postići najpovoljniji koncept rješenja. Prvi korak je uočavanje ključnih problema i na osnovu analize razrada funkcionalne strukture proizvoda. Koncipiranje rješenja je vrlo važna faza konstrukcijskog procesa jer je u kasnijim fazama procesa (izrada osnutka i razrada) vrlo teško, a ponekad i nemoguće ispraviti greške učinjene u konceptu rješenja.

3. izrada osnutka (projektiranje);

Izrada osnutka je dio procesa konstruiranja u kojem se razvija konstrukcijsko rješenje proizvoda u skladu sa tehničkim i ekonomskim kriterijima tako da informacije, koje iz njega slijede prema fazi razrade mogu dovesti do proizvoda. Konstruktor kreće od koncepta rješenja definirajući konstrukcijsko rješenje tako da zadovolji funkciju proizvoda. Često je potrebno izraditi nekoliko varijanti rješenja kako bi se međusobnim uspoređivanjem dobilo više informacija o prednostima i nedostacima pojedine varijante. Za razliku od koncipiranja rješenja izrada osnutka zahtjeva značajni broj korektivnih koraka u kojima se stalno izmjenjuju i nadopunjuju analiza i sinteza što ukazuje na nužnost povezanosti traženja rješenja i razrade sa sposobnošću uočavanja grešaka i optimiziranja. Složenost izrade osnutka očituje se kroz:

- mnogo akcija koje se izvode istovremeno
- neki koraci se moraju ponavljati na višoj razini apstrakcije
- dodaci i promjene u jednom dijelu konstrukcije imaju utjecaja na konstrukciju na nekom drugom dijelu

Konačno konstrukcijsko rješenje podrazumijeva provjeru funkcije, čvrstoće, međusobno podudaranje komponenti i dr. Posebnu pažnju konstruktor mora

obratiti na dodirne točke među konstrukcijskim rješenjima parcijalnih funkcija. Ona moraju biti međusobno usklađena i u slučaju promjene na nekom rješenju mora se znati veza i utjecaj prema ostalim parcijalnim funkcijama.

Korištenjem modernih CAD sustava pri modeliranju varijantnih konstrukcija u ovoj fazi procesa konstruiranja olakšava se međusobno usklađivanje rješenja parcijalnih funkcija proizvoda.

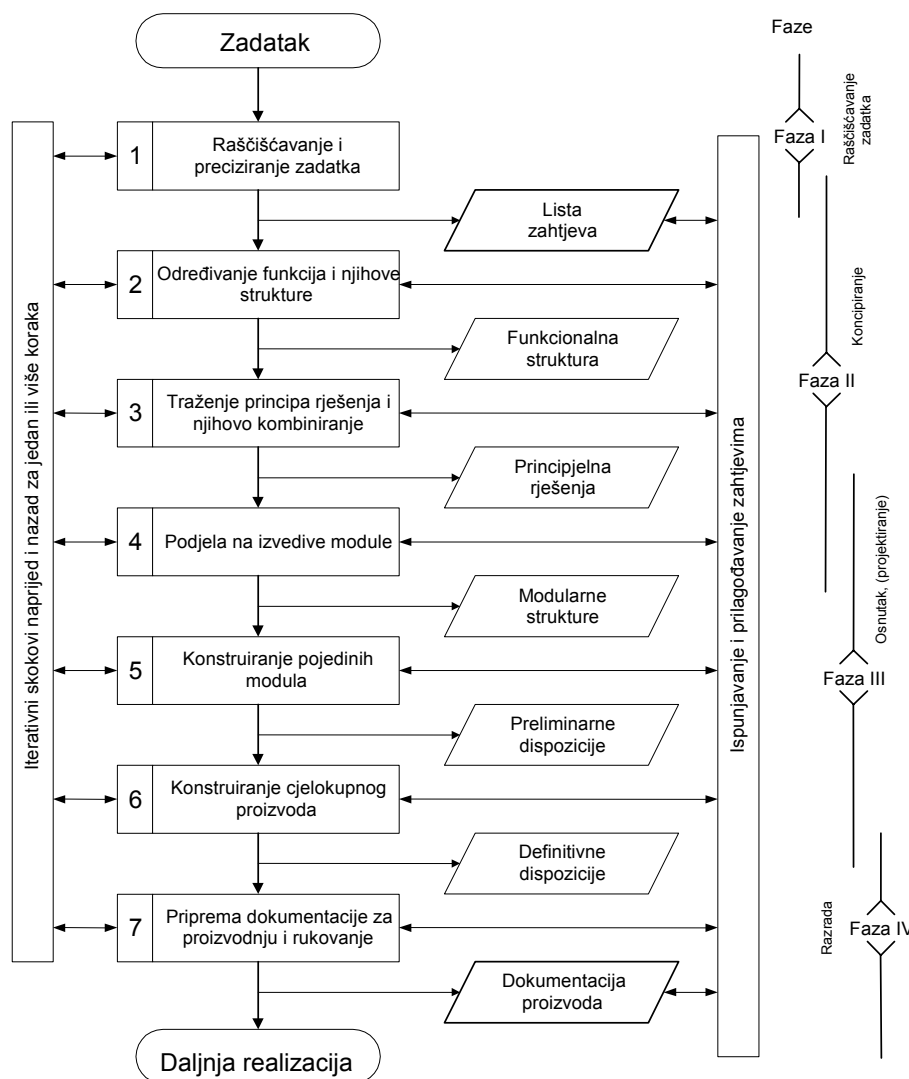
4. razrada;

U posljednoj fazi konstrukcijskog procesa potrebno je odrediti konačni fizički raspored dijelova konstrukcijskog rješenja, definirati oblik, dimenzije i karakteristike površina te definirati materijal. Nadalje je potrebno provjeriti tehničku i ekonomsku izvedivost konstrukcije te izraditi konstrukcijsku dokumentaciju (crteži, sastavnice).

Razrada je često faza procesa konstruiranja koja najduže traje, a uspješnost i brzina uvelike ovise o prethodnim fazama konstrukcijskog procesa posebice koncipiranja rješenja.

Slika 2-5 prikazuje proceduralni model konstruiranja prema VDI 2221 [15]. Na slici je prikazan opći pristup procesu modeliranja odnosno koraci i faze procesa. Ne mogu se postaviti oštre granice između faza, a niti redoslijed faza ne treba kruto promatrati. Proceduralni model procesa konstruiranja razvijen je u obliku preporuka odnosno vodiča. Osim ovog modela postoji još niz teoretskih modela procesa konstruiranja [1], [6].

Proces konstruiranja energetskog transformatora u tvrtki KPT po svojim karakteristikama najviše odgovara proceduralnom modelu konstruiranja.



Slika 2-5: Faze procesa konstruiranja prema proceduralnom modelu VDI 2221 [15]

Prema [11] konstrukcijski proces za konstruiranje varijantnih konstrukcija može se podijeliti u slijedeće faze:

- ◆ izrada tehničke specifikacije,
- ◆ konfiguriranje konstrukcije,
- ◆ konstrukcijska razrada.

Zahtjevi naručitelja (tržišta) se analiziraju i formira se tehnička specifikacija proizvoda koja sadrži zahtjeve koje proizvod mora ispuniti i na osnovu kojih se dalje razrađuje konstrukcijsko rješenje proizvoda.

Na temelju tehničke specifikacije vrši se konfiguriranje konstrukcijskog rješenja što se često naziva i izrada osnutka ili projektiranje. To podrazumijeva definiranje principijelnih rješenja za svaku parcijalnu funkciju proizvoda.

Nakon toga slijedi detaljna konstrukcijska razrada rješenja parcijalnih funkcija proizvoda i izrada konstrukcijske dokumentacije (crteži, sastavnice, ...).

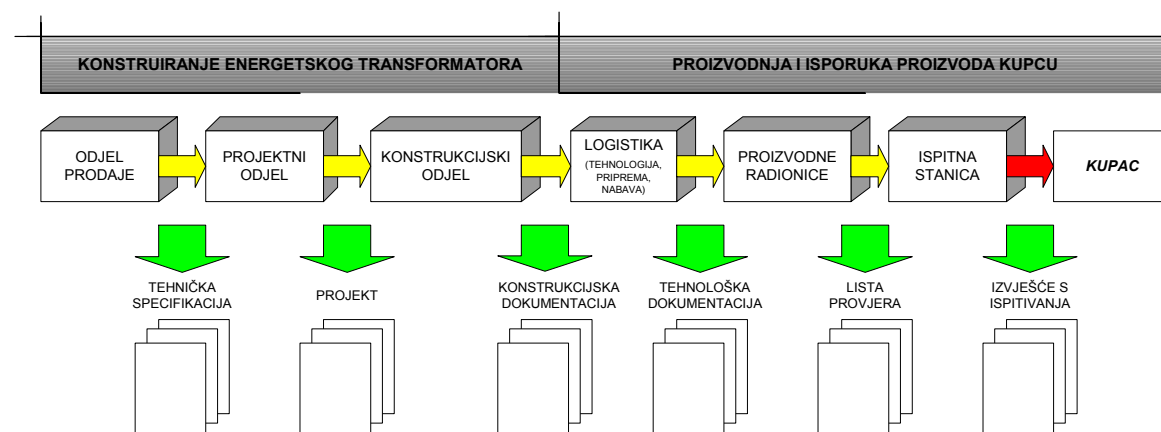
Upravo iz činjenice da je kod varijantnih konstrukcija poznata funkcija i struktura proizvoda uočavamo da nedostaje faza koncipiranja. Drugim riječima, faza koncipiranja je

završena jer se radi o varijantnoj konstrukciji, a ono što nije konceptijski riješeno rješava se kroz konfiguriranje konstrukcije, odnosno kroz izradu osnutka konstrukcijskog rješenja.

2.4 Razvoj energetskeg transformatora

Primjena sustava za prepoznavanje uzoraka pri varijantnom konstruiranju koji je tema ovog rada biti će prikazana na energetskeg transformatoru. Zbog toga se u ovom poglavlju ukratko opisuje razvojni ciklus energetskeg transformatora u tvrtki Končar Energetski transformatori d.o.o. U nastavku je prikazana pojednostavljena struktura energetskeg transformatora i pobliže opisana dva osnovna sklopa energetskeg transformatora: jezgra i konzervator, koji se bitno razlikuju po složenosti strukture.

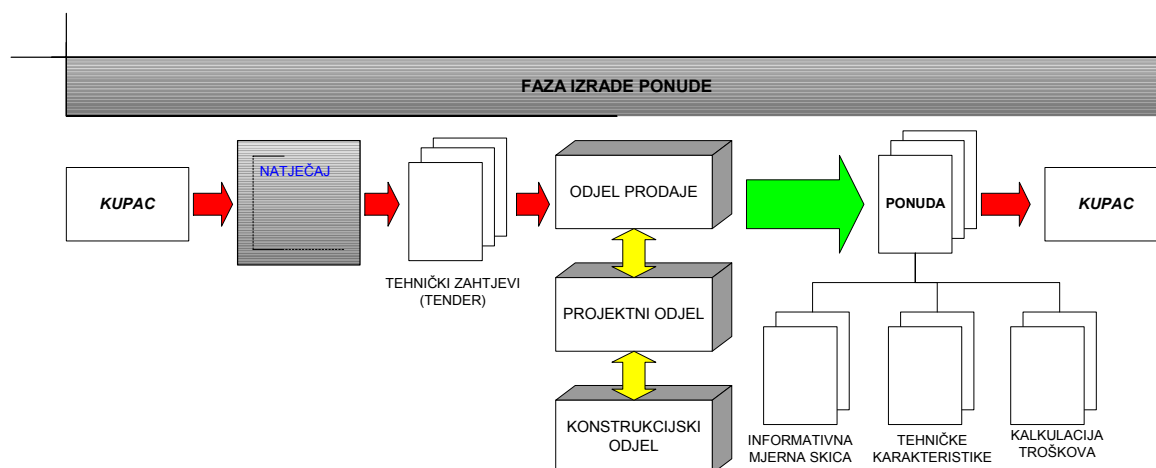
Slika 2-6 prikazuje pojednostavljeni razvojni ciklus energetskeg transformatora kao proizvoda. Razvojni ciklus energetskeg transformatora može se zamisliti kao linearni slijed aktivnosti kroz organizacijsku strukturu tvrtke gdje je rezultat aktivnosti svakog odjela izlazna dokumentacija. Aktivnosti su prikazane sljedno međutim u stvarnosti treba uzeti u obzir i vremenske odnose, odnosno preklapanje pojedinih faza unutar razvojnog ciklusa. To znači da proces izrade konstrukcijske dokumentacije započinje i prije završetka projektne dokumentacije, a izrada nekih komponenti proizvoda započinje i prije nego što je dovršena konstrukcijska dokumentacija.



Slika 2-6: Pojednostavljeni prikaz razvojnog ciklusa energetskeg transformatora

Razvojni ciklus energetskeg transformatora dijelimo na dva glavna dijela: konstruiranje proizvoda i proizvodnja.

Proizvodnja energetskeg transformatora i isporuka naručitelju ovdje neće biti posebno razmatrane iz razloga što se ne uklapaju u temu ovog rada..



Slika 2-7: Konstruiranje energetskog transformatora – faza izrade ponude

Konstruiranje energetskog transformatora dijeli se na dvije glavne faze: faza izrade ponude i faza izrade konstrukcijske dokumentacije.

Slika 2-7 prikazuje fazu izrade ponude koja započinje objavljivanjem natječaja od strane naručitelja. Natječaj sadrži tehničke zahtjeve (tzv. tender) koji opisuju osnovne značajke proizvoda koje naručitelj zahtjeva. Odjel prodaje u suradnji sa projektnim i konstrukcijskim odjelom izrađuje ponudu koja u sebi sadrži informativnu mjernu skicu, tehničke karakteristike proizvoda i kalkulaciju troškova.

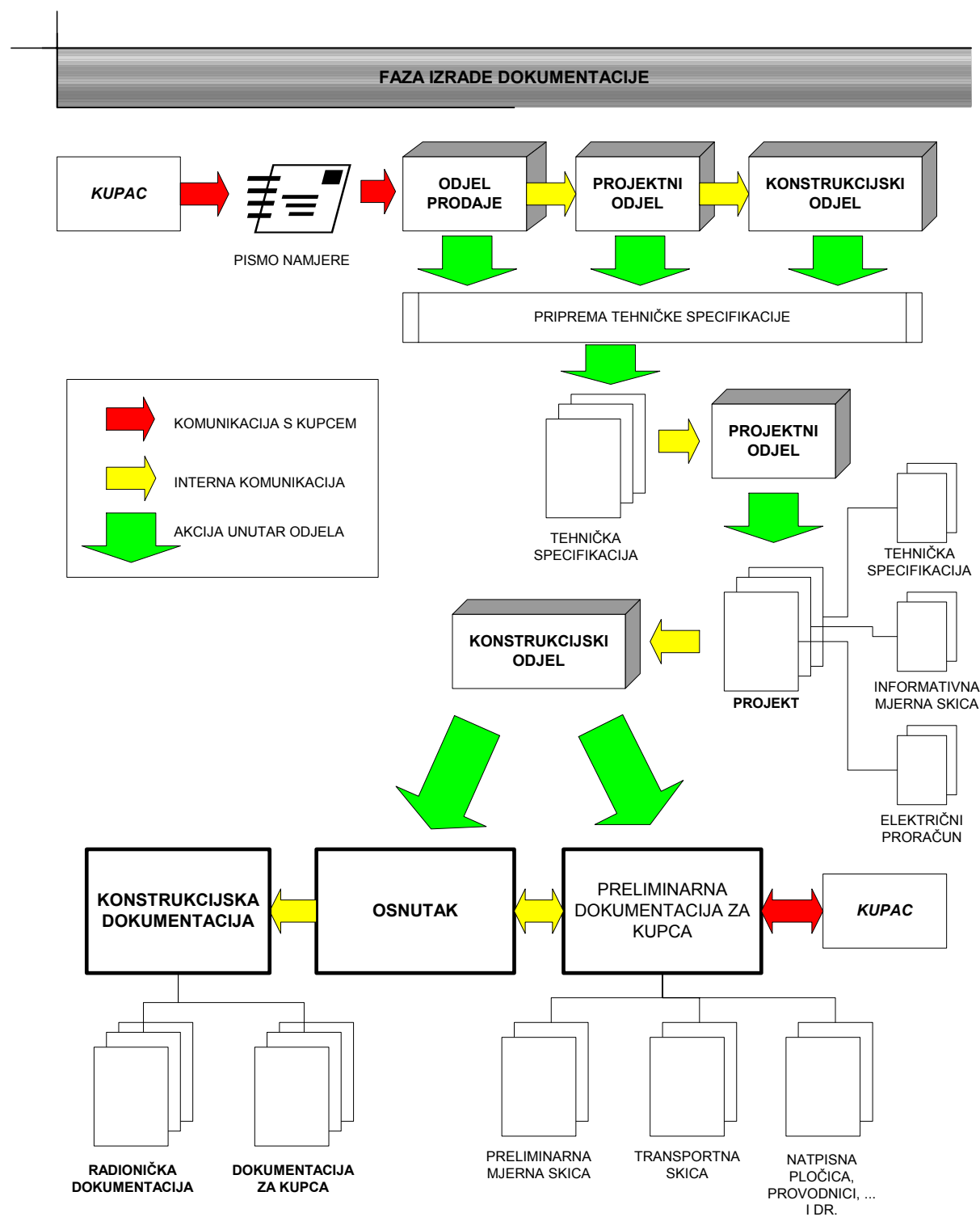
Informativna mjerna skica opisuje osnovnu geometriju transformatora sa kotiranim priključnim i montažnim dimenzijama koje smiju odstupati od konačnog konstrukcijskog rješenja desetak postotaka.

Tehničke karakteristike proizvoda predstavljaju tehničke zahtjeve naručitelja koji su sažeti i opisani u obliku obrazaca.

Kalkulacija troškova obuhvaća troškove rada i materijala na osnovu procenjenog potrebnog rada i procenjenih masa pojedinih dijelova transformatora. Na osnovu kalkulacije troškova formira se cijena proizvoda.

Za izradu navedenih dokumenata i kompletiranje ponude potrebno je izvršiti kompletan električni proračun transformatora koji pored struja i napona obuhvaća i proračun hlađenja, proračun buke i dr.

Ponuda se šalje naručitelju i u slučaju pozitivne odluke kupac odgovara sa pismom namjere koje predstavlja početak druge faze konstruiranja energetskog transformatora fazu izrade konstrukcijske dokumentacije (Slika 2-8).



Slika 2-8: Konstruiranje energetskog transformatora – faza izrade konstrukcijske dokumentacije

Dobivanjem pisma najmere odjel prodaje, projektni odjel i konstrukcijski odjel pripremaju tehničku specifikaciju proizvoda na temelju koje će se izraditi projektna dokumentacija odnosno projekt. Projekt sadrži pored tehničke specifikacije i informativnu mjernu skicu koja je izrađena još u fazi izrade ponude. Glavni dio projekta predstavljaju rezultati električnog proračuna transformatora. U projektu je definiran aktivni dio transformatora

temeljem kojeg se konfigurira konstrukcijsko rješenje proizvoda. Detaljniji opis dijelova transformatora nalazi se u poglavljima 2.4.1, 2.4.2 i 2.4.3.

Nakon izrade projekta energetskog transformatora započinje konfiguriranje konstrukcijskog rješenja (izrada osnutka). Zbog činjenice da je vremenski period od dobivanja pisma namjere od naručitelja pa do isporuke gotovog proizvoda sve kraće konfiguriranje konstrukcijskog rješenja započinje odmah nakon izrade dijela projektne dokumentacije.

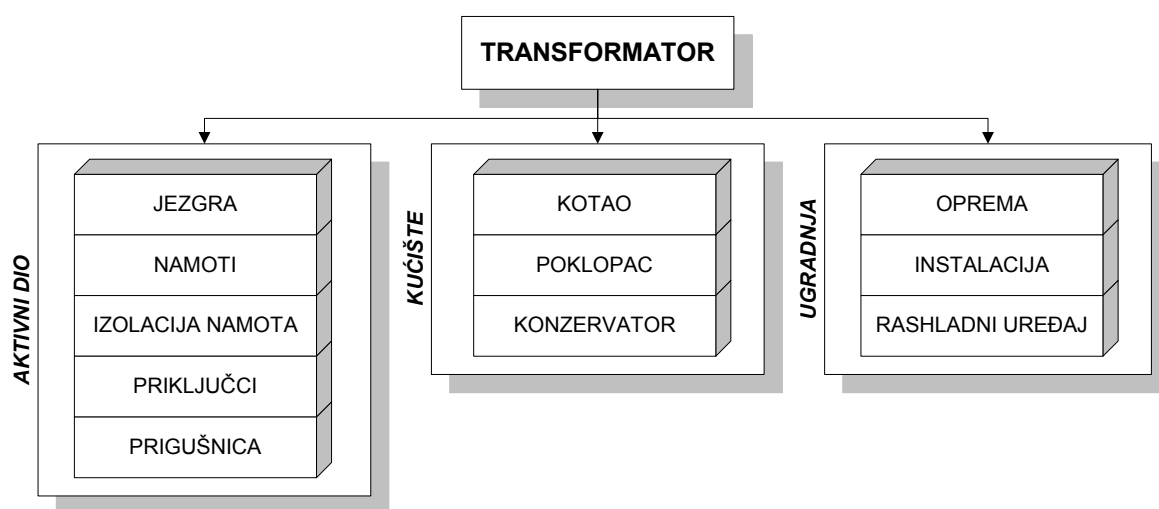
U konstrukcijskom odjelu temeljem projektne dokumentacije izrađuje se osnutak i preliminarna dokumentacija za kupca koja se tijekom izrade usaglašava sa kupcem.

Osnutak predstavlja osnovu za konstrukcijsku razradu svih sklopova transformatora iz koje slijedi radionička dokumentacija. Dokumentacija za kupca obuhvaća: mjernu skicu, transportnu skicu, natpisne pločice, dokumentaciju provodnika, opis i upute i dr. Radionička dokumentacija pored crteža svih sklopova i pozicija obuhvaća i sastavnice sa opisom materijala, standardima i masama.

Radionička i dokumentacija za kupca nazivaju se konstrukcijska dokumentacija.

2.4.1 Struktura energetskog transformatora

U ovom poglavlju će se ukratko opisati fizička struktura konstrukcijskog rješenja energetskog transformatora na razini glavnih sklopova. Sklop energetskog transformatora je varijantna konstrukcija što znači da su poznati ukupna funkcija i struktura proizvoda koja se očituje kroz nekoliko glavnih podsklopova koji se nazivaju osnovni sklopovi transformatora (Slika 2-9).



Slika 2-9: Osnovni sklopovi energetskog transformatora

Osnovne sklopove možemo grupirati u tri grupe: aktivni dio, kućište transformatora i sklopovi za ugradnju na kućište.

Aktivni dio ispunjava osnovnu funkciju proizvoda koja je transformacija vrijednosti ulaznog napona i struje na vrijednost izlaznog napona i struju. Osnovni sklopovi koji čine aktivni dio transformatora su: jezgra, namoti, izolacija namota, priključci i prigušnica (za neke izvedbe transformatora).

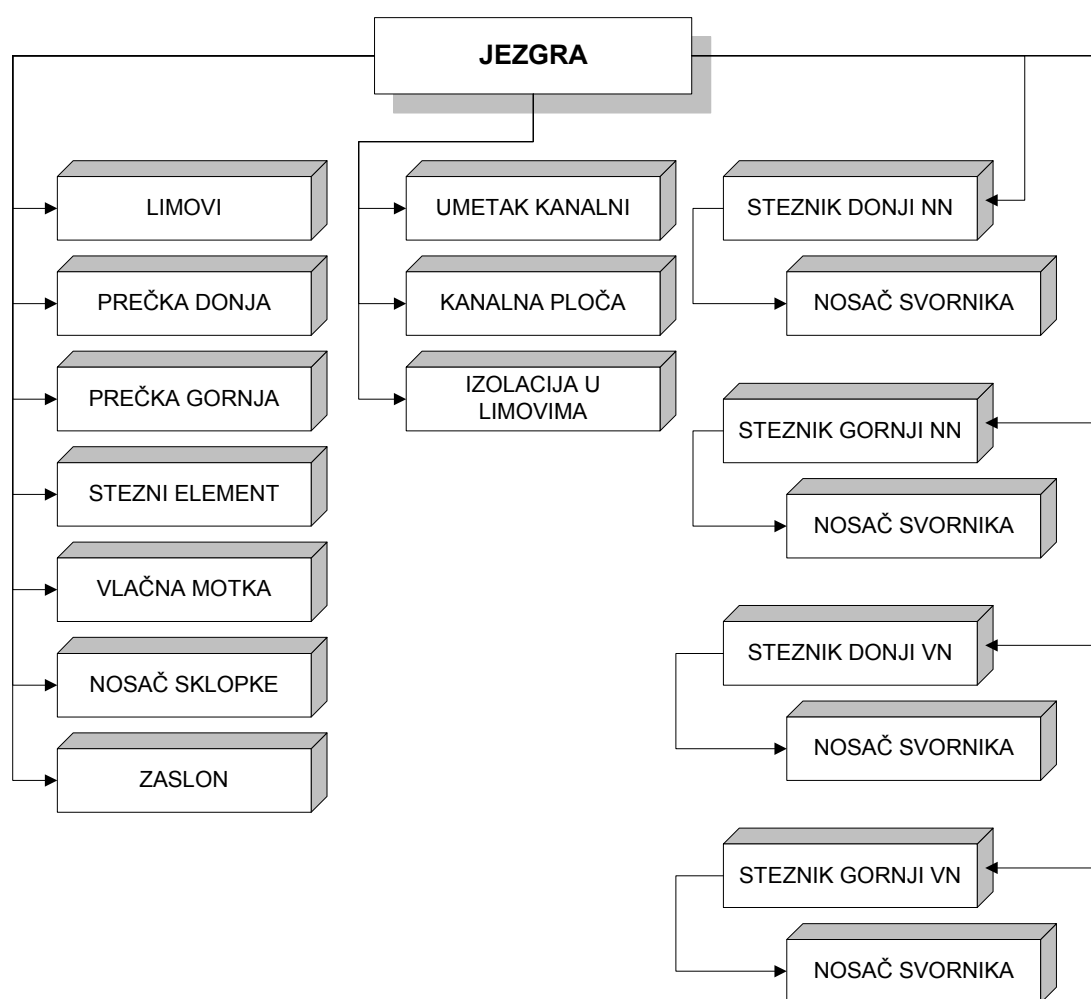
Uloga priključaka je električko povezivanje namota transformatora sa sklopom i sa provodnicima. Kompletan aktivni dio smješta se u donji dio kućišta transformatora koji se

naziva kotao. S gornje strane kotla montira se poklopac koji zajedno sa kotlom predstavlja cjelinu koja se ispunjava transformatorskim uljem i hermetički zatvara. Kotao i poklopac zajedno čine kućište transformatora na koje se postavljaju ostali osnovni sklopovi: konzervator, oprema, rashladni uređaj i instalacija. Konzervator je smješten u grupu kućište iz razloga što se po načinu proizvodnje ne razlikuje od kotla i poklopca iako se ugrađuje na poklopac ili kotao.

U nastavku su pobliže opisana dva osnovna sklopa koja se bitno razlikuju po složenosti strukture: jezgra i konzervator.

Sklop konzervatora koristit će kao jednostavniji primjer za pojašnjavanje izračunavanja težinskih faktora i sličnosti zadatka i uzoraka (poglavlja 5.3.3 i 5.4).

Predloženi sustav za prepoznavanje uzoraka pri varijantnom konstruiranju testirati će se na sklopu jezgre (glava 8).



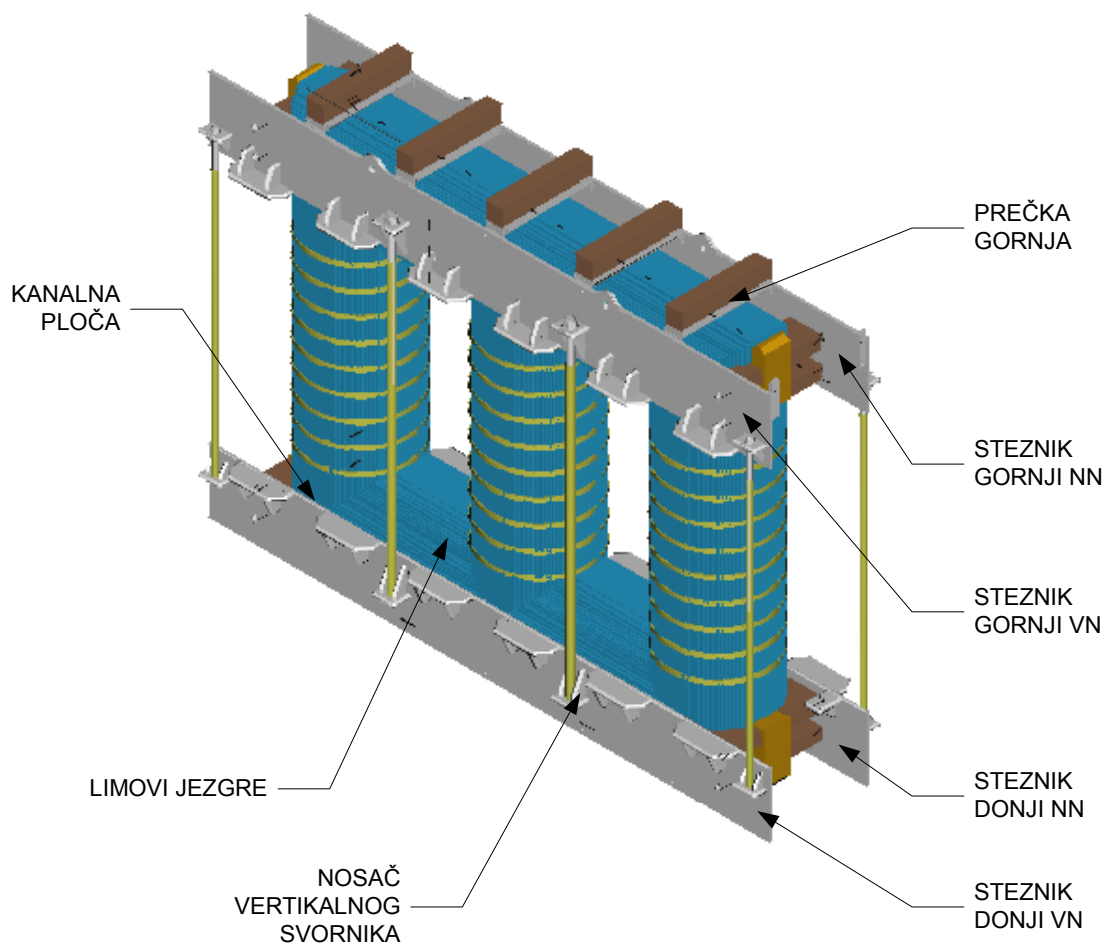
Slika 2-10: Struktura sklopa jezgre

2.4.2 Osnovni sklop jezgre

Funkcija transformatora temelji se na principu elektromagnetske indukcije. U primarnim namotima unutar jezgre transformatora stvara se magnetski tok koji preko limova jezgre inducira napon u sekundarnom namotu. To omogućava transformaciju napona i struje s jednog nivoa na traženi što je i glavna zadaća transformatora.

Jezgra transformatora je sklop koji osigurava princip rada transformatora. Zadaća konstruktora je napraviti jezgru transformatora što manjom, naravno u skladu sa električnim proračunom, kako bi gubici bili manji.

Sukladno tipovima transformatora postoje i razni tipovi jezgri od kojih se najčešće pojavljuju trofazne trostupne i peterostupne jezgre (Slika 2-11 prikazuje model trofazne trostupne jezgre).



Slika 2-11: Model jezgre

Najveći udio u masi jezgre (preko 90%) imaju limovi jezgre sastavljeni od tankih transformatorskih limova debljine do 0.3mm. Limove jezgre potrebno je stlačiti i učvrstiti da bismo dobili kompaktni sklop koji se može transportirati tijekom procesa proizvodnje i da bi jezgra mogla funkcionirati.

Stlačivanje limova je nužno da se onemogući međusobno proklizavanje limova te da se smanji buka koja se javlja tijekom rada transformatora uslijed elektromagnetske indukcije. Stlačivanje limova vrši se pomoću steznog sustava jezgre koji je najčešće izveden od čeličnih ploča na donjem i gornjem jarmu jezgre koje nazivamo steznicima (Slika 2-11). Ploče su međusobno povezane vlačnim motkama ili vertikalnim svornicima (ovisno o izvedbi jezgre) u vertikalnom smjeru odnosno horizontalnim svornicima i prečkama u horizontalnom smjeru. Vlačne motke ili vertikalni svornici preuzimaju aksijalne tlačne sile namota te ukupnu masu jezgre i namota prilikom prenošenja jezgre dizalicom.

Stupovi jezgre učvršćuju se bandažiranjem pomoću traka. Trake mogu biti čelične ili od specijalnih staklenih vlakana koja se za vrijeme montaže polimeriziraju kako bi dobili tražena svojstva.

Pored ovih glavnih dijelova jezgre ugrađuju se još i razni izolacijski dijelovi između prečki i limova, vlačnih motki i limova, steznika i limova te između prečki i poklopca kotla.

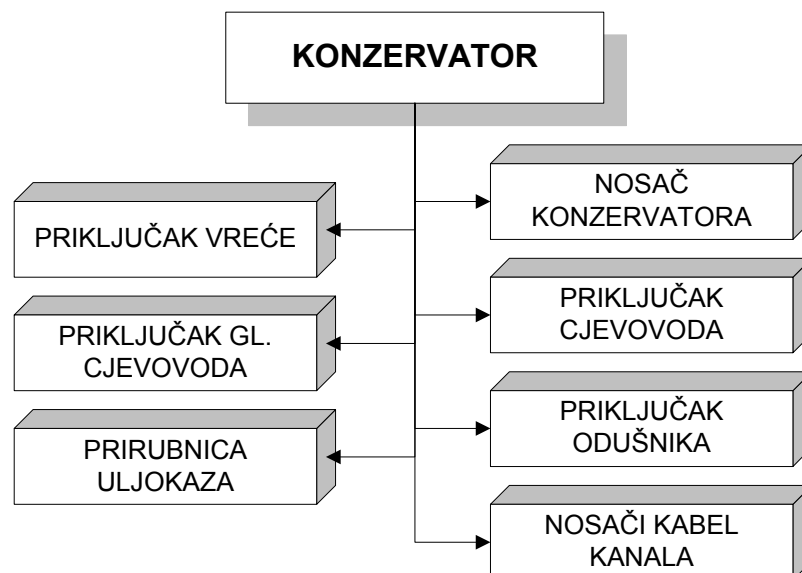
Na steznike jezgre se, također, pričvršćuju nosači namota, nosači vertikalnih svornika te nosači priključaka.

2.4.3 Osnovni sklop konzervatora

Konzervator je dio sustava za zaštitu ulja u transformatoru. Funkcija sustava za zaštitu ulja je sprječavanje direktnog doticaja ulja za zrakom i ovlaživanje ulja vlagom iz zraka, te omogućavanje normalnog toplinskog rastezanja ulja uslijed zagrijavanja transformatora u radu. Sustav za zaštitu ulja čine konzervator opremljen sa svim potrebnim uređajima i opremom.

Konzervator je ekspanzijska posuda koja omogućava toplinsko rastezanje ulja. Razlikujemo dva načina zaštite ulja, a u skladu s njima i dva osnovna tipa konzervatora:

- ◆ bez vreće – ulje je u dodiru sa zrakom
Kod promjene volumena ulja, skupljanje ulja kod hlađenja i širenje kod zagrijavanja, dolazi i do usisavanja i istiskivanja atmosferskog zraka iz konzervatora. Kod ovih izmjena atmosferski zrak prolazi kroz tvar koja izvlači vlagu iz zraka i tako sprječava pogoršanje svojstava ulja uslijed ovlaživanja
- ◆ sa vrećom – ulje je mehanički odvojeno od zraka pomoću gumenog separatora
U ovom slučaju atmosferski zrak je unutar separatora i nema dodira sa transformatorskim uljem. Na taj način se sprječava pogoršavanje svojstava ulja uslijed ovlaživanja.



Slika 2-12: Struktura sklopa konzervatora

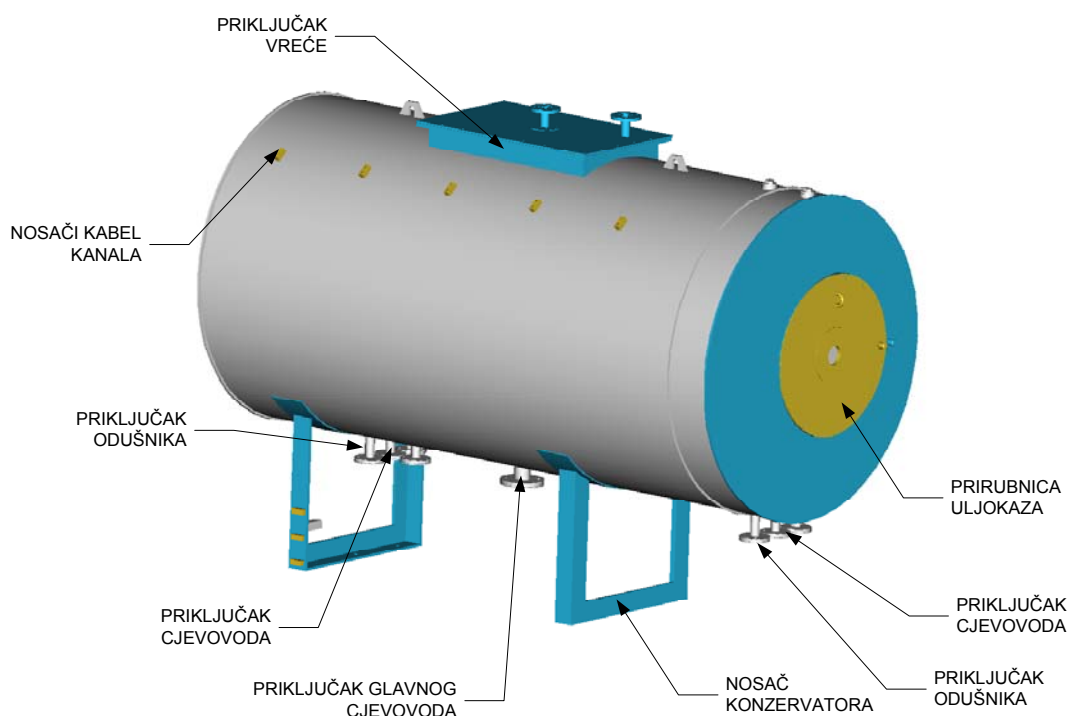
Konzervator se smješta na kućište transformatora montiranjem na nosače konzervatora koji mogu imati različite izvedbe. Na zahtjev kupca konzervator se montira na zid

prostorije u kojoj se nalazi transformator ili na poseban nosač ako je transformator smješten vani.

Slika 2-12 prikazuje strukturu modela sklopa transformatora.

Slika 2-13 prikazuje model konzervatora sa načinom zaštite ulja gdje je ulje mehanički odvojeno od zraka pomoću gumenog separatora (vreće). Na gornjem dijelu konzervatora nalazi se priključak za gumeni separator.

Konzervator na slici ima dodatnu posudu manjeg obujma koja ima istu funkciju kao i veća posuda ali za ulje koje se nalazi u sklopki transformatora.



Slika 2-13: Model konzervatora

2.4.4 Značajke postojećeg CAE sustava

Postojeći CAE sustav tvrtke Končar Energetski transformatori d.o.o. sastoji se od dva CAD sustava: sustav za izradu crteža temeljen na alatu AutoCAD® i trodimenzionalnog parametarskog sustava za modeliranje temeljen na alatu ProENGINEER®.

Korisničke aplikacije u sustavu za izradu crteža razvijaju se programskim jezikom AutoLISP i C++ koristeći biblioteke klasa za pristup crtežima.

U parametarskom sustavu za modeliranje koriste se aplikacije kao pomoć konstruktoru pri modeliranju koje su razvijene programskim jezikom Java® koristeći biblioteke klasa za pristup parametarskim modelima izrađenim pomoću CAD alata u ProENGINEER® [16].

Konstruktorska razrada sklopova kotla, jezgre, poklopca i konzervatora radi se pomoću trodimenzionalnih parametarskih modela i njima pripadajućih crteža. Postojeći modeli-predlošci se prilagođavaju novim konstrukcijskim zahtjevima koji su definirani

konstrukcijskim zadatkom. Kada se rješenje novog konstrukcijskog zadatka bitno razlikuje (dimenzijski i/ili strukturno) od postojećeg modela koji se koristi kao predložak značajno se povećava vrijeme uređivanja postojećeg pripadajućeg crteža. Dijelovi crteža koji opisuju crtež (kote, oznake pozicija, oznake zavara i dr.) ne nalaze se više na odgovarajućem mjestu zbog različitosti modela.

U takvo okruženje potrebno je implementirati i predloženi sustav za prepoznavanje uzoraka.

3 RASUĐIVANJE TEMELJEM UZORAKA

U glavi 3, u prvom dijelu, ukratko se opisuje metodologija [5] rasuđivanja temeljem uzoraka, njezin povijesni razvoj kao i primjeri nekih razvijenih sustava. Detaljniji opis metodologije rasuđivanja temeljem uzoraka može se pronaći u literaturi [17], [18] i [19]. Ukratko će se opisati faze procesa rasuđivanja temeljem uzoraka, te opisati metode za pronalaženje najbližih uzoraka.

3.1 Opis i povijesni razvoj

Pod terminom umjetna inteligencija u procesu konstruiranja (Artificial Intelligence in Design) često se podrazumijevaju niz različitih tehnologija razvijenih u proteklih tri-četiri desetljeća kao što su logičko programiranje, rasuđivanjem temeljem pravila (rule-based reasoning), neuralne mreže, genetski algoritmi, fuzzy logika i dr. [20], [21]. Jedan od novijih pravaca na ovom području, koji u posljednje vrijeme ima sve veću primjenu na raznim područjima, naziva se rasuđivanje temeljem uzoraka⁶ [5], [22], [23].

Općenito, rasuđivanje temeljem uzoraka možemo definirati kao metodologiju koja se koristi za rješavanje problemskih situacija prisjećanjem (pronalaženjem) sličnih prethodnih problemskih situacija koje su riješene u prošlosti i pohranjene na odgovarajući način, te korištenjem informacija i znanja sadržanih u pronađenom sličnom problemu za rješavanje novog problema. Problemske situacije, riješene u prošlosti, u CBR terminologiji nazivamo uzorcima (cases), koji se pohranjuju u memoriju⁷.

U sustavima koji koriste paradigmu rasuđivanja temeljem uzoraka učenje sustava je kontinuirano. Svaki pohranjeni uzorak je jedna problemska situacija riješena u prošlosti koja predstavlja dio ukupnog prikupljenog znanja domene. Korištenjem metodologije rasuđivanja temeljem uzoraka nije nužno da se iskustveno znanje prikupljeno rješavanjem novih problema modificira i prilagođava pravilima pohranjivanja ukupnog znanja unutar neke domene, kao što je to slučaj kod nekih drugih metoda primjene umjetne inteligencije u procesu konstruiranja (npr. kod rasuđivanja temeljem pravila znanje prikupljeno korištenjem sustava potrebno je preoblikovati u pravila kako bi se moglo koristiti za rješavanje budućih problema). Naravno, ovisno o domeni primjene metodologije rasuđivanja temeljem uzoraka, te o problemima koji se rješavaju ponekad neće biti dovoljno samo pohraniti novi uzorak u memoriju već će se morati modificirati prikupljeno iskustvo i nadopuniti opće znanje domene [18].

Ključ razumijevanja metodologije rasuđivanja temeljem uzoraka leži u dva osnovna procesa: pronalaženju najbližeg uzorka pohranjenog u memoriji i adaptaciji pronađenog

⁶ Case-Based Reasoning - CBR

⁷ Pod terminom memorija (case base, case memory) podrazumijeva se neki od načina za pohranu podataka kao što su npr. baze podataka, datoteke, i dr.

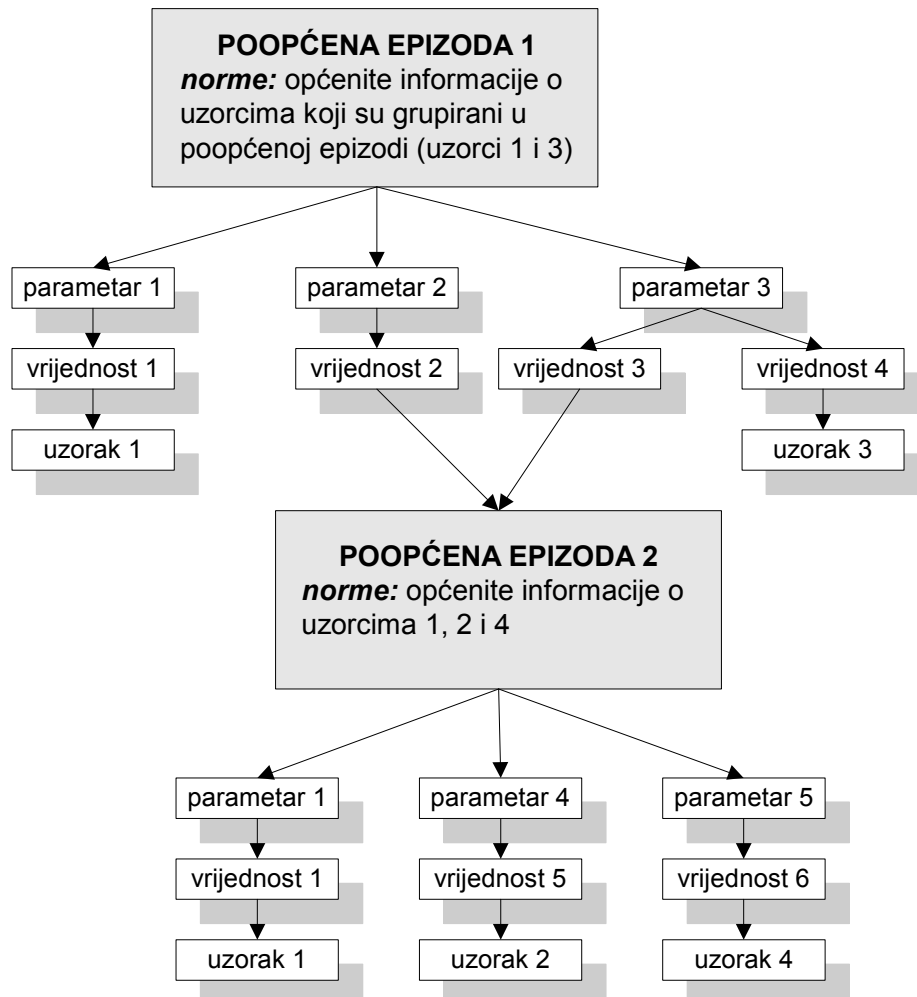
uzorka u skladu sa zahtjevima novog problema koji se rješava [24], [25], [26], [17], [27], [28], [29], [30], [31], [32], [33].

Možemo reći da su osnovne karakteristike metodologije rasuđivanja temeljem uzoraka:

- ♦ rješavanje problema korištenjem rješenja prijašnjih riješenih problema,
- ♦ pronalaženje prijašnjih problema (uzoraka) povlači za sobom određivanje sličnosti problema sa uzorcima u memoriji,
- ♦ riješeni novi problem dodaje se u memoriju kako bi se mogao koristiti za rješavanje novih problema u budućnosti.

Rasuđivanje temeljem uzoraka daje dobre rezultate na mnogim područjima gdje je potrebno rješavati problemske situacije, kao što su planiranje, dijagnosticiranje i konstruiranje.

Početak razvoja CBR paradigme smatraju se radovi Schanka i Abelsona (1977.) [17], [34], [35], [18], [27]. Oni su pretpostavili da je naše znanje o raznim situacijama u kojima se nalazimo sačuvano u obliku svojevrtnih scenarija koji nam omogućuju da postavimo očekivanja i donosimo zaključke. Scenariji imaju određenu memorijsku strukturu koja sadržava informacije o stereotipnim događajima kao što su npr. odlazak liječniku ili u restoran.



Slika 3-1: Organizacija uzoraka prema Schanku

Daljnje istraživanje Schanka dovelo je do razvoja tzv. Schankove teorije dinamičke memorije (dynamic memory) (1982.) [36], [17]. Schank je predložio strukturu dinamičke memorije nazvanu MOPs (memory organization packets) koja ima ulogu spremišta poopćenog znanja i organizacije uzoraka.

Osnovna ideja je organizirati uzorke sa sličnim osobinama u općenitije strukture (pakete) tzv. poopćene epizode. Poopćena epizoda sadrži tri tipa objekata: norme, uzorke i parametre (Slika 3-1).

Norme su zajedničke osobine za sve uzorke unutar neke generalizirane epizode, dok parametri predstavljaju attribute koji definiraju razliku među uzorcima. Vrijednost parametra može upućivati na neku poopćenu epizodu ili direktno na uzorak.

Baza uzoraka⁸ se prikazuje kao mreža čvorova gdje je čvor generalizirana epizoda, naziv parametra, vrijednost parametra ili uzorak.

Janet Kolodner je 1983. razvila prvi CBR sustav nazvan CYRUS koji je sadržavao znanje, u obliku uzoraka, o putovanjima i sastancima bivšeg američkog državnog tajnika Cyrusa Vancea. Sustav je omogućavao korisnicima postavljanje pitanja o tim događajima i temeljio se na Schankovom modelu dinamičke memorije.

Kasnije su razvijeni mnogi sustavi za različite primjene na mnogim domenama temeljeni na raznim varijacijama osnovnog Schankovog modela [17], [34], [19]:

na području planiranja i konstruiranja:

- ◆ JULIA (Hinrichs 1988.)
CBR sustav za pomoć pri izradi jelovnika. Korisnik definira osnovne karakteristike jela na temelju kojih sustav predlaže jelovnik
- ◆ KRITIK (Goel i Chandrasekaran 1989.) [37]
Sustav je razvijen za konstruiranje mehaničkih uređaja i jednostavnih električnih krugova. Korištena je linearna struktura zapisivanja uzoraka u bazu i metoda najbliži susjed za računanje sličnosti među uzorcima
- ◆ ARCHIE (Pearce i dr. 1992.) [38]
Sustav se koristi u području arhitekture, točnije za projektiranje uredskih prostoriya. Sustav je interaktivan i razvijen je pomoću alata za izradu CBR aplikacija REMIND.
- ◆ CADET (Sycara i dr. 1992.) [39]
Koristi se za rješavanje konstrukcijskih problema tako da prikazuje veze između funkcije, strukture i ponašanja uzoraka. Konceptualno konstrukcijsko rješenje proizvoda dobiva se sintezom riješenih konstrukcijskih dijelova koji čine proizvod. Sustav se koristi u širokom području inženjerstva (hidraulika, mehanika, elektrika).
- ◆ DEMEX (Maher i Garza 1996.) [3], [40], [41]
Sustav se koristi za rješavanje problema u području građevinarstva pri projektiranju građevinskih objekata. Sustav koristi dvije skupine podataka u bazi: uzorke koji opisuju određena konstrukcijska rješenja i modele koji opisuju poopćene klase konstrukcijskih rješenja.
- ◆ HOMER (Göker i dr. 1998.) [42]
CBR sustav za pomoć konstruktorima koji se koristi u tvornici automobila DaimlerChrysler kao dio CAD/CAM sustava.

⁸ u nastavku će se koristiti termin baza uzoraka neovisno o realizaciji memorije uzoraka; najčešće se radi o relacijskoj bazi podataka

na području dijagnosticiranja:

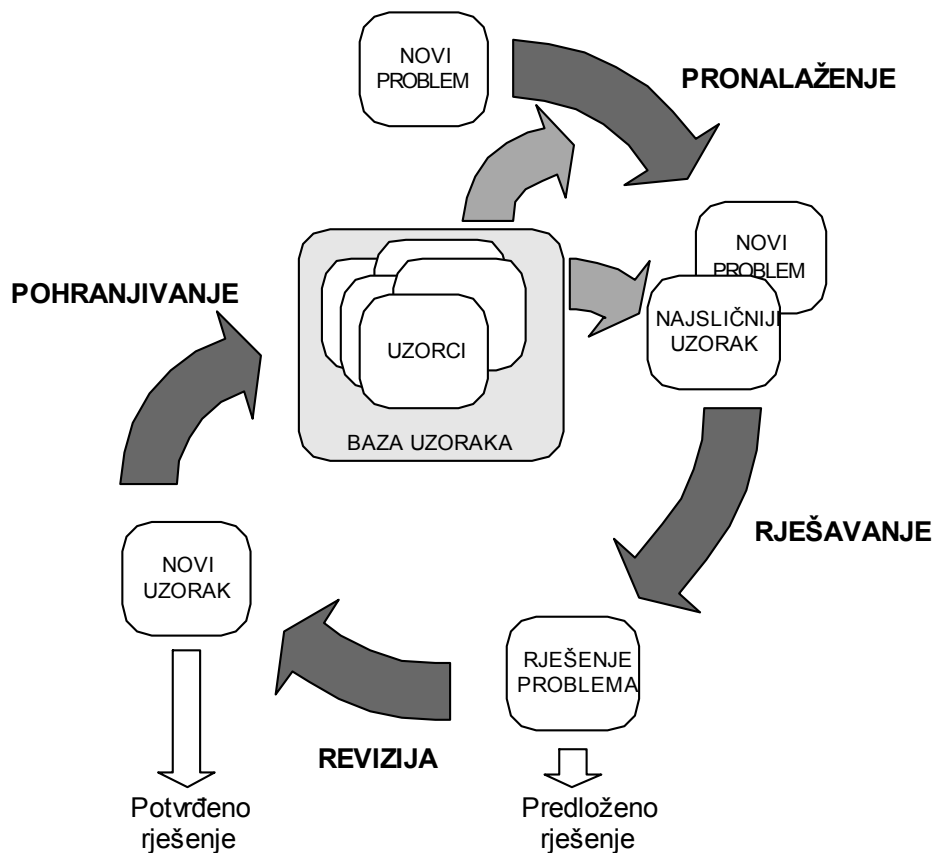
- ♦ CASEY (Koton 1988.) [43]
Sustav je razvijen za pomoć liječniku pri dijagnosticiranju bolesnika sa bolestima srca. Sustav je automatiziran.
- ♦ PROTOS (Bareiss 1988.) [44]
Sustav je razvijen za dijagnosticiranje bolesti kod pacijenata sa slušnim problemima.
- ♦ ISAC (Bonzano 1998.) [45]
CBR sustav za podršku kontrolorima leta pri rješavanju problema u zračnom prometu.

Pored spomenutih razvijeni su i mnogi drugi sustavi temeljeni na CBR metodologiji koji imaju primjenu u raznim domenama.

3.2 Ciklus rasuđivanja

Rasuđivanje temeljem uzoraka možemo, općenito, promatrati kao ciklički proces koji se dijeli u četiri osnovne faze koje zajedničkim imenom zovemo ciklus rasuđivanja (CBR cycle) [18], [19], (Slika 3-2):

1. pronalaženje sličnog uzorka ili više njih,
2. korištenje informacija pronađenog uzorka za rješavanje novog problema,
3. revizija predloženog rješenja,
4. pohranjivanje novog uzorka na prikladan način za buduće korištenje.



Slika 3-2: Shematski prikaz ciklusa rasuđivanja

Općenito, sustav koji koristi metodologiju rasuđivanja temeljem uzoraka za rješavanje problema ima zadaću pronaći najslabije prijašnje rješenje (ili više njih) na temelju opisa novog problemskog zadatka (pronalaženje). Opis novog problema uspoređuje se sa pohranjenim uzorcima i pronalazi se najslabiji uzorak. Pronađeno najslabije rješenje se adaptira u skladu s novim zahtjevima ili se koriste metode kojima se došlo do pronađenog rješenja ili jedno i drugo (rješavanje). Adaptacijom pronađenog slabijeg rješenja, odnosno korištenjem metoda kojima se došlo do slabijeg rješenja dobiva se predloženo rješenje problema. U fazi revizije provjerava se ispravnost rješenja novog problema (npr. kroz realnu primjenu rješenja novog problema). Pogreške koje se eventualno jave u rješenju ispravljaju se ili se ukazuje se na mogućnost njihovog pojavljivanja kako bi se na vrijeme izbjegle pri rješavanju budućih problema. Rezultat revizije je potvrđeno rješenje koje sada postaje novi uzorak koji se pohranjuje u bazu uzoraka za rješavanje nekog budućeg problema (pohranjivanje).

U domeni konstruiranja problem predstavlja novi konstrukcijski zadatak koji se uspoređuje sa prethodnim konstrukcijskim zadacima pohranjenim u bazi uzoraka i pronalazi se jedan ili više najslabijih. Koristeći informacije sadržane u najslabijem uzorku rješava se novi konstrukcijski zadatak. Rješavanjem novog konstrukcijskog zadatka dobijemo novo konstrukcijsko rješenje koje se, ako je potrebno, revidira i nakon toga pohranjuje u bazu kao novi uzorak za buduće korištenje. Treba napomenuti da nije nužno da svako rješenje postane uzorak i da se pohranjuje u bazu. Pri realizaciji sustava mogu se definirati algoritmi i kriteriji kojima se određuje kada se novo konstrukcijsko rješenje pohranjuje u bazu uzoraka kao novi uzorak. Često je to prepušteno korisniku realiziranog sustava.

Korištenjem sustava zasnovanog na rasuđivanju temeljem uzoraka povećava se broj uzoraka pohranjenih u bazi uzoraka, čime sustav postaje efikasniji.

Jedna od glavnih zadataka koju je potrebno riješiti unutar sustava zasnovanog na rasuđivanju temeljem uzoraka je način zapisivanja uzoraka, o čemu je više rečeno u slijedećem poglavlju.

3.3 Zapisivanje uzoraka

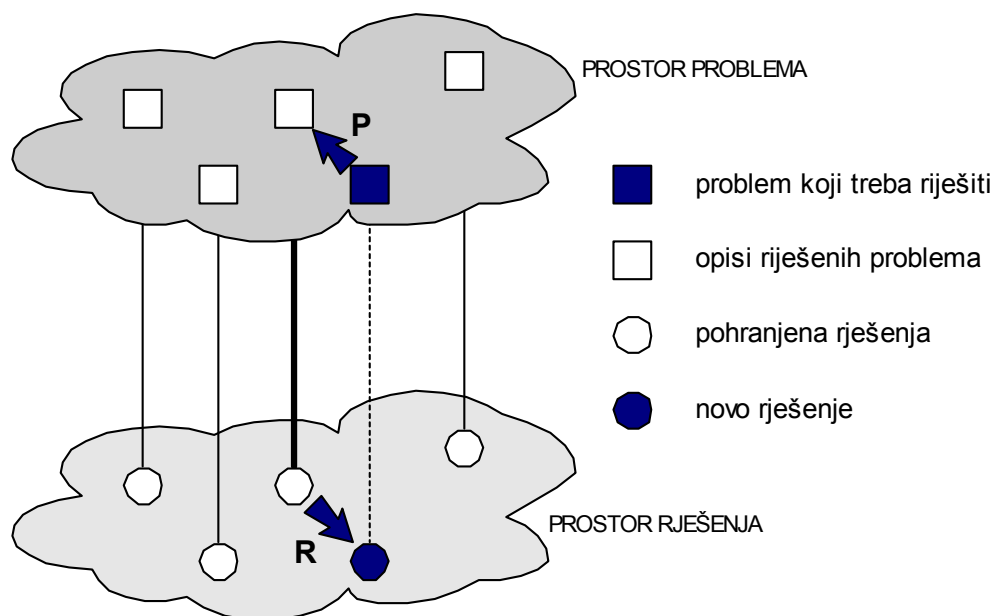
Baza uzoraka predstavlja temelj svakog sustava koji koristi metodologiju rasuđivanja temeljem uzoraka za rješavanje problema. U domeni konstruiranja uzorci predstavljaju prethodna konstrukcijska rješenja koja su na prikladan način pohranjena u bazu uzoraka. Iskustvo stečeno rješavanjem prijašnjih konstrukcijskih zadataka se sakuplja, zapisuje i kasnije koristi za rješavanje novih problemskih situacija. Nova problemska situacija predstavlja novi konstrukcijski zadatak koji se rješava.

Uzorci u bazi mogu sadržavati neko konkretno iskustvo [18] ili mogu biti grupirani kao općeno rješenje. Također, uzorci mogu biti pohranjeni kao zasebne jedinice znanja ili podijeljeni na podjedinice unutar definirane strukture.

Možemo reći da je sustav zasnovan na metodologiji rasuđivanja temeljem uzoraka onoliko dobar koliko je dobra njegova baza uzoraka [17]. Ako ograničimo razmatranje na domenu konstruiranja možemo zaključiti da je dobra baza uzoraka velika baza. Pod pojmom veličine baze misli se na broj pohranjenih uzoraka. Promatrajući bazu uzoraka kao prostor u kojem se nalazi uzorci tada je položaj uzoraka u tom zamišljenom prostoru određen vrijednostima parametara koji opisuju uzorke. Količina uzoraka u bazi proporcionalna je vjerojatnosti da će položaj najslabijeg uzorka biti bliži novom konstrukcijskom zadatku. Pitanje koje se često postavlja je koliko mora biti velika baza da

bi bila efikasna. Najčešće CBR sustavi imaju od nekoliko stotina do nekoliko tisuća uzoraka [19].

Pored veličine baze uzoraka na pronalaženje najslićnijeg uzorka ključni utjecaj imaju težinski faktori pojedinih parametara koji su pobliže opisani u poglavlju 3.4.1.1 .



Slika 3-3: Prostor problema i prostor rješenja

Postoje dva osnovna modela organizacije baze [18], [27], [19]: model dinamičke memorije (Schank, Kolodner) i model uzorka i grupe (Porter, Bareiss). Ova dva modela se još uvijek uvelike koriste u spoznajnoj znanosti međutim niti jedan komercijalni sustav baziran na rasuđivanju temeljem uzoraka ne koristi ove tehnike [19], već se najčešće koristi jednostavna linearna struktura podataka ili uobičajena struktura relacijske baze podataka. U bazi uzoraka zapisane su vrijednosti parametara⁹ koje opisuju svaki uzorak. Uzorci mogu sadržavati različite tipove podataka koji se mogu pohraniti u uobičajenim bazama podataka.

Uzorak možemo smatrati kao neki događaj, priču ili zapis koji se sastoji od problema i rješenja. Bazu uzoraka možemo ilustrirati kao prostor problema i prostor rješenja (Slika 3-3).

Pojedinačni uzorak sačinjavaju dvije komponente: opis problema i pohranjeno rješenje. Opis novog problema koji se rješava nalazi se u prostoru problema. Procesom pronalaženja pronalazimo uzorak sa sličnim opisom problema koji pokazuje na rješenje. Drugim riječima, ako se novi problem nalazi u prostoru problema dolje lijevo od poznatog problema tada će se i rješenje novog problema nalaziti u prostoru rješenja dolje lijevo od poznatog rješenja.

⁹ Često se u literaturi koristi termin indeks; u ovom radu će se koristiti termin parametar kako se ne bi indeks dovodio u vezu sa pojmom indeksiranja u relacijskoj bazi podataka

3.4 Metode pronalaženja najsličnijih rješenja

Pronalaženje uzoraka te određivanje sličnosti svakog pogodnog (poglavlje 5.2) uzorka sa zadatkom koji se riješava predstavlja ključni problem sustava za prepoznavanje uzoraka. Faze rješavanja (adaptacije) i pohranjivanja u bazu biti će uspješne onoliko koliko je bilo uspješno pronalaženje najsličnijeg rješenja. Ovaj zadatak je usko povezan sa načinom organiziranja baze uzoraka.

Danas se najčešće koriste dvije metode za pronalaženje najsličnijeg slučaja [19]: metoda najbliži susjed i induktivna metoda.

3.4.1 Metoda najbliži susjed

Metoda najbliži susjed je najjednostavnija i najčešće korištena metoda za pronalaženje najsličnijeg uzorka [18], [19], [27]. Ukoliko bazu uzoraka zamislimo kao prostor u kojem se nalaze uzorci tada njihov položaj u tom prostoru ovisi o vrijednostima parametara koji ih opisuju. Nakon definiranja novog problema potrebno je pronaći najsličniji odnosno najbliži uzorak u zamišljenom prostoru. Izračunavanjem udaljenosti novog problema i uzoraka u prostoru dobijemo mjeru po kojoj određujemo koji je uzorak najsličniji novom problemu. Metoda najbliži susjed upravo se temelji na izračunavanju udaljenosti između zadatka i svih promatranih uzoraka.

Svaki uzorak u bazi je definiran određenim brojem osobina (parametara, indeksa) koje mogu biti numeričke ili simboličke. Postoje razne definicije i opisi problema pod nazivom "najbliži susjed" [17], [46], [47], [48], [19], [49], [50], [51], [52], [53]. Najčešće korišteni opis zamišlja bazu uzoraka kao skup točaka u euklidskoj geometriji¹⁰.

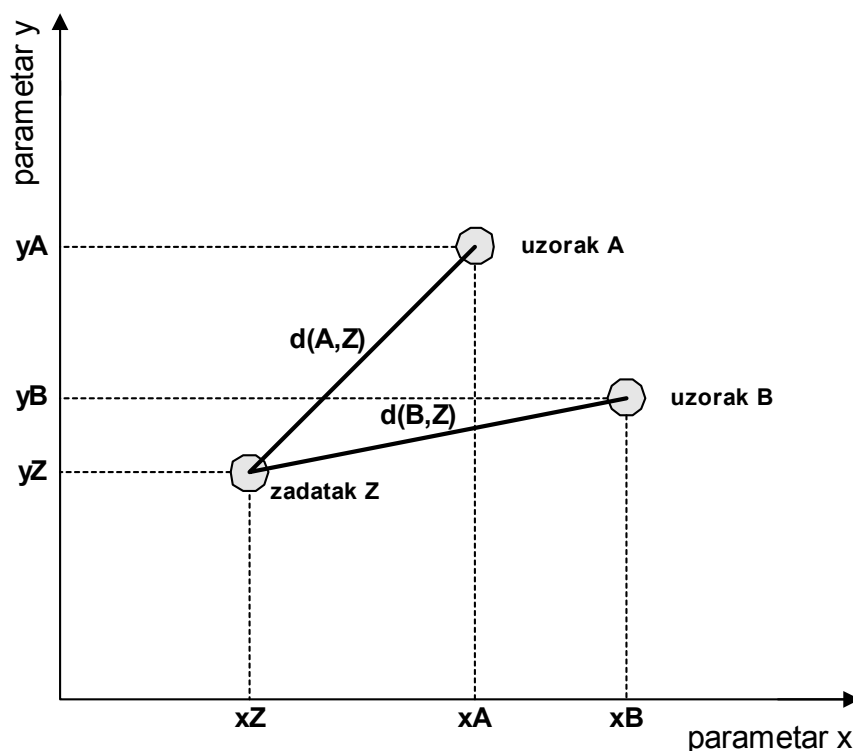
Ako uzorak promatramo kao točku u koordinatnom sustavu opisanu sa dva parametra x i y (Slika 3-4) tada udaljenost između pojedinih uzoraka i novog zadatka računamo prema izrazu:

$$\begin{aligned} [d(A, Z)]^2 &= (X_A - X_Z)^2 + (Y_A - Y_Z)^2 \\ d(A, Z) &= \sqrt{(X_A - X_Z)^2 + (Y_A - Y_Z)^2} \end{aligned} \quad (3-1)$$

Ili općenito ako je uzorak opisan sa n parametara $U = \{u_1, u_2, \dots, u_n\}$ čije koordinate predstavljaju vrijednosti parametara s kojima je uzorak opisan, a zadatak kao točku $Z = \{z_1, z_2, \dots, z_n\}$ čije koordinate predstavljaju vrijednosti parametara kojima je opisan zadatak, tada je udaljenost između tih dviju točaka definirana izrazom:

$$d(U, Z) = \sqrt{(z_1 - u_1)^2 + (z_2 - u_2)^2 + \dots + (z_n - u_n)^2} \quad (3-2)$$

¹⁰ Geometrija zasnovana na zornim predodžbama trodimenzionalnog prostora - po grč. matematičaru Euklidu (3. st. pr. Kr.)



Slika 3-4: Metoda nabliži susjed za dva parametra

Svaki parametar s kojim su opisani uzorak i zadatak nije jednako važan prilikom njihovog međusobnog uspoređivanja. Zbog toga se za svaki parametar zadaje (ili izračunava) i njegov težinski faktor (poglavlje 3.4.1.1). Iz toga proizlazi izraz za udaljenost, j-tog uzorka [48], [50]:

$$d(U_j, Z) = \left[\sqrt{\sum_{i=1}^n w_i \cdot \text{razlika}(u_i, z_i)^2} \right]_j, \quad (3-3)$$

gdje su:

Z – novi konstrukcijski zadatak
U_j – j-ti uzorak iz baze; $j = \{1, m\}$
m – broj uzoraka
n – broj parametara
w_i – težinski faktor parametra *i*; $i = \{1, n\}$
u_i – i-ti parametar uzorka *j*
z_i – i-ti parametar zadatka

$$\text{razlika}(u_i, z_i) = \begin{cases} |u_i - z_i| & \text{– ako je indeks numerički} \\ 0 & \text{– ako je indeks simbolički i } u_i = z_i \\ 1 & \text{– ako je indeks simbolički i } u_i \neq z_i \end{cases}$$

Obično udaljenost normiramo kako bi se vrijednosti kretale između 0 i 1, čime dobijemo izraz za sličnost j -tog uzorka:

$$S_j = 1 - \frac{d(U_j, Z)}{\sum_{j=1}^m d(U_j, Z)}, \quad (3-4)$$

gdje su:

S_j – sličnost j -tog uzorka
 m – broj promatranih uzoraka

Sličnost vrijednosti 0 predstavlja potpunu različitost dok sličnost vrijednosti 1 predstavlja potpuno iste uzorke s obzirom na promatrane parametre.

3.4.1.1 Težinski faktori

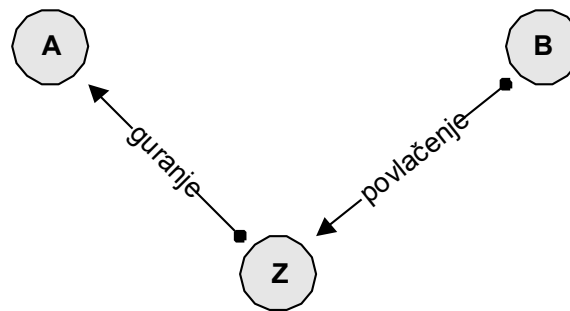
Performanse sustava zasnovanog na rasuđivanju temeljem uzoraka, a koji koristi metodu najbliži susjed za izračunavanje sličnosti između zadatka i uzoraka u bazi, ovise o tome da li su korišteni relevantni parametri pri računanju sličnosti. Dodjeljivanjem težinskih faktora parametrima utječemo na rezultat pronalaženja sličnih uzoraka tako da oni parametri koji su značajniji pri pronalaženju najbližeg uzorka imaju veći težinski faktor. Ne može se točno definirati kako odrediti koji su parametri značajniji. To u prvom redu ovisi o domeni primjene i autorima sustava.

Vrijednost težinskih faktora može se dodijeliti prilikom kreiranja baze parametara pri čemu njihova vrijednost ostaje nepromijenjena tijekom korištenja sustava. Postoje metode kojima se omogućava promjena vrijednosti težinskih faktora tijekom korištenja sustava. Istraživanja su pokazala [48], [54], [55], [45], [56] da promjena vrijednosti težinskih faktora tijekom korištenja sustava daje bolje rezultate pronalaženja najbližih uzoraka. Jedan od načina promjene vrijednosti težinskih faktora tijekom korištenja sustava je tzv. introspektivno poboljšanje težinskih faktora (introspective learning of parameter weights). Ovdje će ukratko biti opisani rezultati dobiveni u sustavu ISAC¹¹. [45].

U sustavu ISAC postoje dva osnovna tipa težinskih faktora: lokalni i globalni težinski faktori. Ukoliko parametar ima globalni težinski faktor vrijednost težinskog faktora biti će ista za sve uzorke u bazi ili drugim riječima njegova važnost je ista za sve uzorke. S druge strane ukoliko se radi o lokalnom težinskom faktoru tada njegova vrijednost ovisi o promatranom uzorku i o vrijednostima drugih parametara, odnosno važnost promatranog parametra razlikuje se od uzorka do uzorka.

Osnovna ideja kod introspektivnog poboljšavanja težinskih faktora je povećati ili smanjiti vrijednost težinskog faktora za promatrani parametar. U sustavu ISAC slučajnim uzorkom odabire se određeni broj uzoraka iz baze na temelju kojih se onda vrši korekcija odnosno poboljšavanje težinskih faktora. Razmatrana su četiri različita načina poboljšavanja koji se temelje na dva osnovna kriterija: kriterij pogreške i kriterij pogotka. U prvom se vrijednost težinskog faktora ažurira u slučaju pogreške dok u drugom u slučaju pogotka. Promjenom vrijednosti težinskih faktora parametara uzorke pomičemo u prostoru uzoraka (Slika 3-5). Cilj nam je uzorke koji vode do sličnijeg rješenja privući bliže zadatku dok nepovoljne uzorke udaljiti od zadatka.

¹¹ ISAC – Intelligent System for Aircraft Conflict Resolution



Slika 3-5: Povlačenje i guranje uzoraka [45]

Četiri načina za poboljšavanje vrijednosti težinskih faktora su:

- ◆ GUM (Good Up Matching): Promatrani uzorak iz baze ima isto rješenje kao zadatak iz promatranog skupa uzoraka. Povećavamo težinske faktore parametrima koji imaju istu vrijednost kao zadatak (povlačimo uzorak prema zadatku).
- ◆ GDU (Good Down Unmatching): Promatrani uzorak iz baze ima isto rješenje kao zadatak iz promatranog skupa uzoraka. Smanjujemo težine parametrima koji imaju različite vrijednosti (guramo uzorak dalje od zadatka).
- ◆ BUU (Bad Up Unmatching): Promatrani uzorak iz baze ima različito rješenje kao zadatak iz promatranog skupa uzoraka. Povećavamo težine parametrima koji imaju različite vrijednosti.
- ◆ BDM (Bad Down Matching): Promatrani uzorak iz baze ima različito rješenje kao zadatak iz promatranog skupa uzoraka. Smanjujemo težine parametrima koji imaju iste vrijednosti.

Kombiniranjem ovih osnovnih metoda dobijemo tehnike koje daju bolje ili slabije rezultate. Ispitivanja u sustavu ISAC pokazala su da za lokalne težinske faktore najbolje rezultate daje kombinacija metoda "Bez GUM" (GDU + BUU + BDM), dok za globalne težinske faktore podjednako dobre rezultate daje kombinacija metoda "Bez GUM" i "Bez BUU" (GUM + GDU + BDM). Detaljniji opis sustava može se pronaći u [45].

3.4.2 Induktivna metoda

Druga metoda pronalaženja sličnih uzoraka koja će ovdje biti spomenuta je metoda indukcije ili induktivna metoda koja koristi tzv. proces indukcije [19], [47] koji se temelji na algoritmu poznatom po imenu ID3 [57]. Pomoću ID3 algoritma kreira se stablo odlučivanja iz baze uzoraka koristeći spoznajnu metodu poznatu pod nazivom informacijska dobit (information gain). Cilj je pronaći parametre koji kroz stablo odlučivanja «najbolje» grupiraju uzorke. Parametar kojeg koristimo za grupiranje uzoraka predstavlja čvor u stablu odlučivanja iz kojeg se stablo dijeli u dvije ili više grana ovisno o vrijednostima koje parametar može poprimiti.

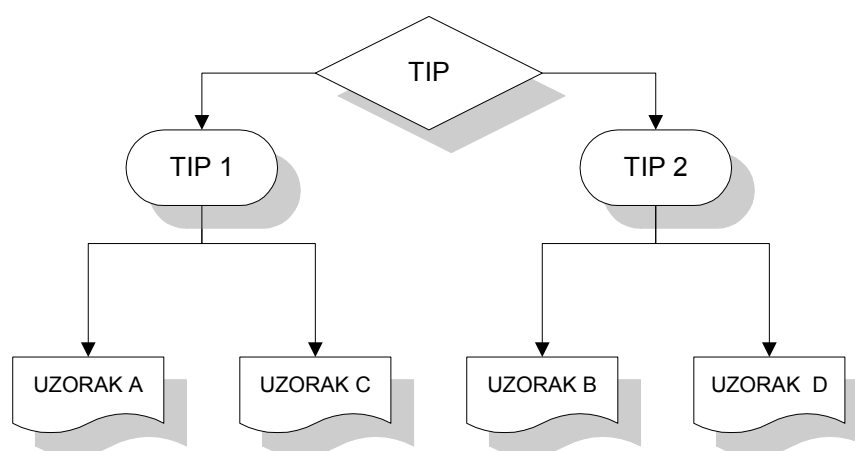
Induktivnu metodu možemo zamisliti kao postupnik kojeg treba zadovoljiti da bi se kroz stablo odlučivanja pronašao najsličniji uzorak u bazi uzoraka. Navođenjem činjenica formira se stablo odlučivanja kojim se dolazi do sličnih uzoraka.

Pretpostavimo da baza uzoraka ima četiri uzorka (Tablica 1) koji su opisani vrijednostima parametara X i Y .

	<i>parametar TIP</i>	<i>parametar X</i>	<i>parametar Y</i>
UZORAK A	<i>TIP 1</i>	1000	1800
UZORAK B	<i>TIP 2</i>	1600	2500
UZORAK C	<i>TIP 1</i>	1250	2100
UZORAK D	<i>TIP 2</i>	800	1500

Tablica 1: Primjer baze uzoraka

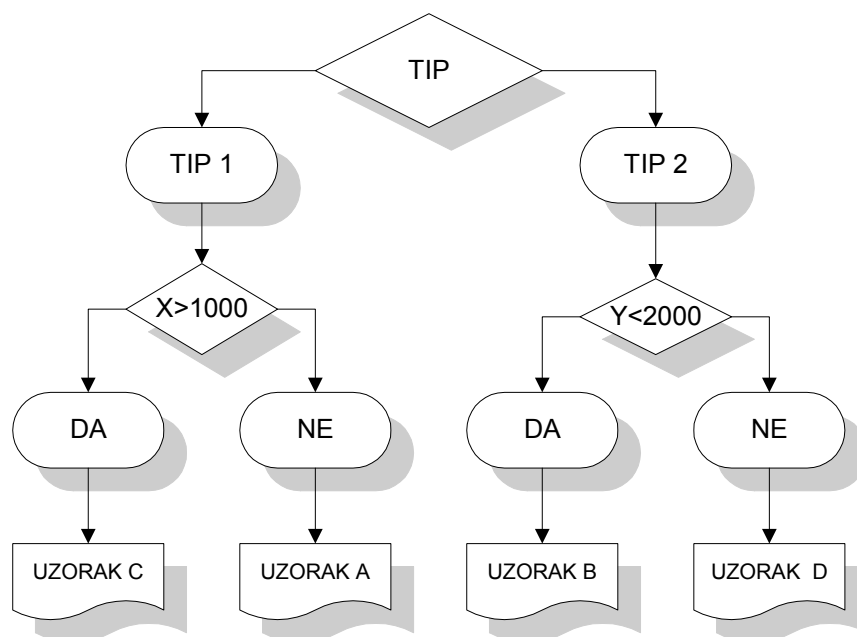
Analizirajući parametre koji opisuju uzorke vidimo da parametar *TIP* može imati samo dvije vrijednosti: *TIP 1* i *TIP 2* što znači da je dobar izbor za prvi čvor u stablu odlučivanja.



Slika 3-6: Prvi čvor u stablu odlučivanja

Dakle, odabirom parametra *TIP* kao prvog čvora u stablu odlučivanja grupirali smo uzorke u dvije grane: u lijevoj grani su uzorci *A* i *C* za koje vrijedi $TIP = TIP\ 1$, dok su u desnoj uzorci *B* i *D* za koje vrijedi $TIP = TIP\ 2$ (Slika 3-6).

Slijedeći čvor u stablu odlučivanja dobijemo pomoću parametara X i Y . Ako u lijevu granu stabla ubacimo čvor kojega određuje parametar X , a u desnu čvor kojega određuje parametar Y dobijemo jedan primjer stabla odlučivanja (Slika 3-7).



Slika 3-7: Primjer stabla odlučivanja

Pretpostavimo da tražimo najslićniji uzorak zadatku koji je opisan tablicom (Tablica 2):

	<i>parametar TIP</i>	<i>parametar X</i>	<i>parametar Y</i>
ZADATAK	<i>TIP 1</i>	<i>800</i>	<i>-</i>

Tablica 2: Primjer zadatka

Stablo odlučivanja započinje s prvim čvorom gdje se provjerava vrijednost parametra **TIP**. Za zadatak vrijedi **TIP=TIP 1**, što nas u stablu vodi na lijevu granu. Dolazimo do čvora gdje se provjerava vrijednost parametra **X**. Pošto je u zadatku zadano **X=800** zaključujemo da je to manje od **1000** što upućuje na **UZORAK A**. Dakle, zaključujemo da je **UZORAK A** najslićniji zadatku

4 ZAPISIVANJE UZORAKA

U prethodnoj glavi pobliže je opisana metodologija rasuđivanja temeljem uzoraka kao ciklički proces. Konstrukcijska rješenja prijašnjih konstrukcijskih zadataka potrebno je na prikladan način zapisati za buduće korištenje.

U glavi 4 razmotriti će se opisivanje varijantne konstrukcije pomoću parametara. Također, će se prikazati opisivanje konstrukcijskog zadatka te uzorka. Predložiti će se model strukture baze uzoraka koja će se realizirati u poglavlju 7.2.

4.1 Parametar konstrukcije

Promatrajući proces konstruiranja kao proces transformacije i generiranja informacija može se smatrati da je podatak temeljna jedinica kojom opisujemo konstrukciju. U računalnom modelu podatak se prikazuje varijablom odnosno, preneseno na proces konstruiranja, parametrom konstrukcije [58], [59]. Dakle, parametar konstrukcije smatrat će se osnovnom jedinicom opisa proizvoda koji je predmet procesa konstruiranja.

Vrijednosti parametara konstrukcije određuju se tijekom procesa konstruiranja, ali isto tako vrijednost parametra može biti poznata već i na početku procesa konstruiranja ukoliko se radi o podacima iz tehničke specifikacije ili o nekom podatku varijantne konstrukcije.

Ako se parametar modelira kao objekt unutar objektnog modela procesa konstruiranja tada svaki parametar konstrukcije ima svoje atribute kao što su:

- ◆ identifikacijska oznaka u bazi parametara,
- ◆ naziv,
- ◆ tip,
- ◆ vrijednost,
- ◆ fizikalna jedinica,
- ◆ hiperveza na opis parametra.

Kompletan skup informacija o proizvodu redovito sadrži velik broj podataka. Činjenica da će svaka složenija konstrukcija biti opisana sa nekoliko tisuća ili desetaka tisuća podataka nameće potrebu da se parametri moraju strukturalno organizirati radi lakšeg manipuliranja [59]. Ukoliko bi se svi podaci koji opisuju konstrukciju modelirali kao parametri konstrukcije to bi značilo i manipuliranje s velikom količinom podataka što bi svakako trebalo izbjeći. Zbog toga bi parametrima trebalo obuhvatiti ključne podatke za povezivanje različitih objekata prikaza konstrukcije, podatke koje dijele informacijski spregnuti zadaci i podatke koji kolaju između raznorodnih programskih alata.

S gledišta metodologije rasuđivanja temeljem uzoraka valja voditi računa da parametri trebaju [19], [27]:

- ◆ biti predskazivi,
- ◆ dobro opisati osobine uzoraka,
- ◆ biti dovoljno apstraktni da omoguće širinu za buduće korištenje baze uzoraka,
- ◆ biti dovoljno konkretni kako bi se u buduće prepoznali.

Parametri koji opisuju uzorke moraju biti tako odabrani da se na temelju njihovih vrijednosti može uočiti razlika među uzorcima, moraju biti predskazivi.

Uzmimo npr. debljinu stijenke priрубnice uljokaza na sklopu konzervatora. Navedeni parametar nije predskaziv iz razloga što je vrijednost spomenute debljine stijenke isti za sve konzervatore. Međutim, promjer konzervatora je predskaziv parametar i dobar je odabir za opis uzorka.

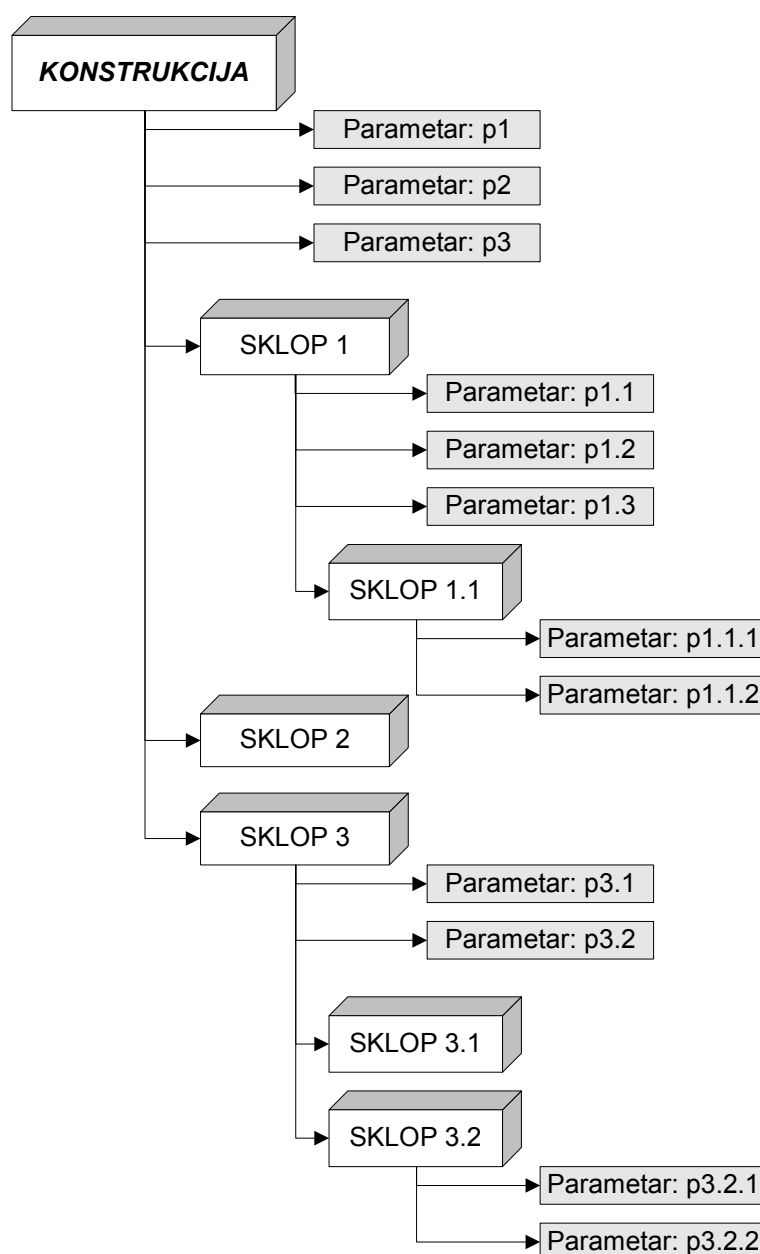
Pored geometrijskih veličina parametri kojima je opisano konstrukcijsko rješenje mogu sadržavati i druge informacije npr.: informacije o konstrukcijskom rješenju proizvoda, konstrukcijskom rješenju nekog dijela ili sklopa proizvoda, bilješke o pogreškama učinjenim tijekom čitavog ili nekog dijela razvojnog procesa proizvoda i dr.

4.1.1 Organizacijska struktura parametara

Parametri se organizacijski strukturiraju u bazu parametara gdje se termin "baza" koristi u nedostatku prikladnijeg, makar baza parametara ne mora imati svojstva baze podataka i nije nužno da se relizira u programskoj okolini baze podataka.

Predlaže se strukturiranje baze parametara (Slika 4-1) u obliku hijerarhijskog stabla. Fizička struktura konstrukcije proizvoda zapisana je u bazu kao hijerarhijsko stablo do razine sklopova i svakom sklopu mogu se pridružiti parametri koji ga opisuju. Sklop u hijerarhijskom stablu ne mora nužno imati pridružene parametre (npr. SKLOP 2 nema pridruženih parametara).

Kod pridruživanja parametara sklopovima izuzetno je važno da svaki parametar ima svoju jedinstvenu adresu [59], ali nije potrebno da ima i jedinstveni naziv. To ne bi bilo praktično jer bi moglo stvoriti probleme pri određivanju imena parametara. Prijedlog je da se imenu parametra doda prefiks koji je određen njegovim položajem unutar fizičke strukture konstrukcije proizvoda (u primjeru energetskog transformatora za jezgru bi prefiks bio J_, za konzervator KN_, itd.).



Slika 4-1: Struktura parametara u bazi parametara

4.1.2 Značajke čvornih parametara

Jedna od metoda za pronalaženje najslbličnijeg uzorka je metoda indukcije (poglavlje 3.4.2) kojom se kreira stablo odlučivanja i na taj način dolazi do najslbličnijeg uzorka. Grananja unutar stabla odvijaju se u tzv. čvorovima. Parametri čija se vrijednost koristi u čvorovima nazivati ćemo čvorni parametri.

Pored kreiranja stabla odlučivanja čvorni parametri mogu se koristiti za grupiranje uzoraka po njihovim karakteristikama na temelju vrijednosti čvornih parametara. Uzorci koji imaju identičnu vrijednost čvornog parametra pripadaju istoj skupini uzoraka.

Ako promotrimo sklop konzervatora energetskeg transformatora tada je dobar čvorni parametar `KN_TIP` koji sadrži podatak o tipu konzervatora upravo zbog toga što je postoji

svega četiri različita tipa konzervatora. S druge strane parametar `KN_PGC` koji sadrži podatak o položaju glavnog cjevovoda nije nikako dobar čvorni parametar jer praktički njegova vrijednost može biti bilo koji cijeli broj.

Uzorci sklopa konzervatora pohranjeni u bazu uzoraka koji imaju vrijednost čvornog parametra `KN_TIP=KON_V` pripadaju grupi uzoraka konzervatora sa vrećom (poglavlje 2.4.3)

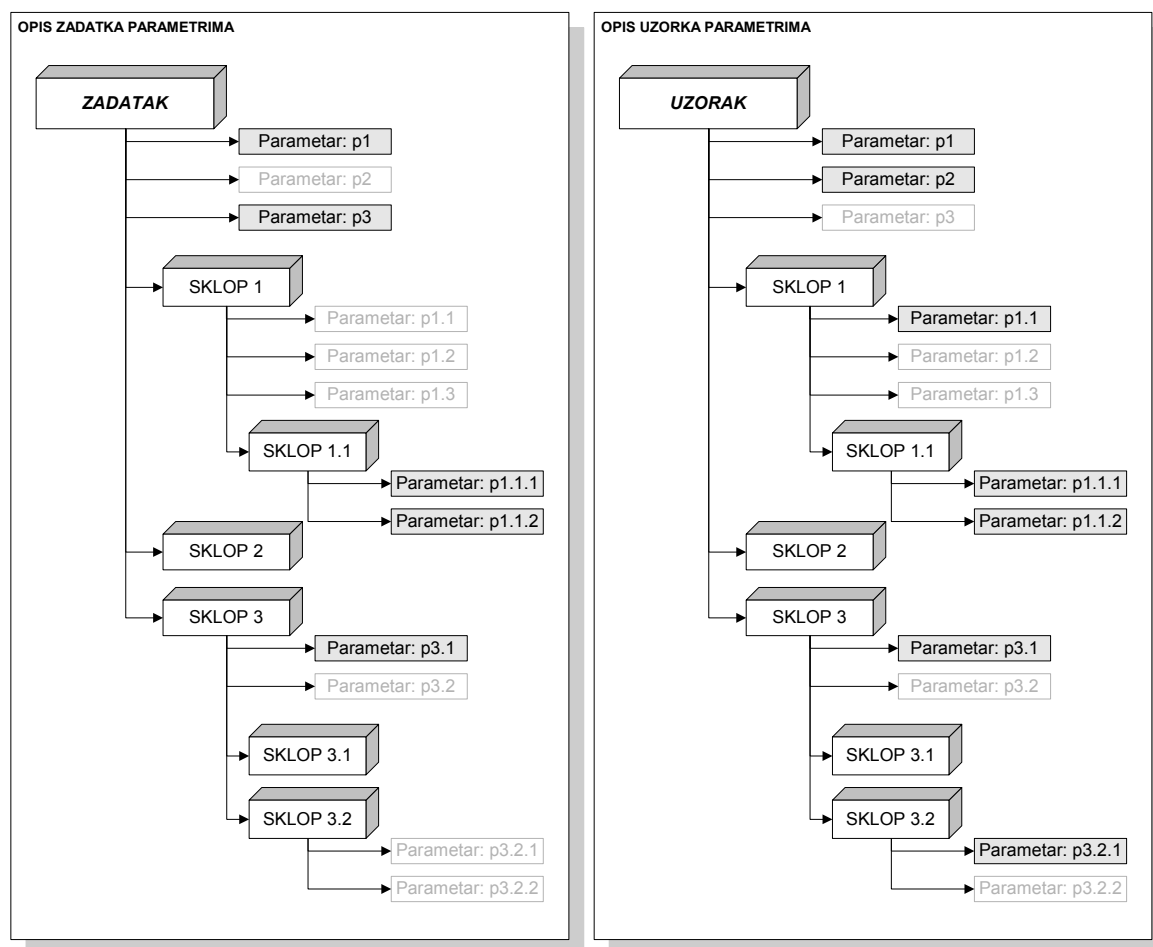
4.1.3 Tip vrijednosti parametra

Parametri mogu poprimiti slijedeće tipove vrijednosti:

- ◆ logička vrijednost
parametar poprima vrijednost logičke jedinice ili nule, npr. dodatni priključak na konzervatoru kotla, ...
- ◆ vrijednost s liste
parametar poprima vrijednost s definirane liste vrijednosti, npr. vrsta transformatora, frekvencija, ...
- ◆ vrijednost iz intervala
parametar poprima neku vrijednost iz zadanog intervala, npr. promjer trostupne, trofazne jezgre koja ima limove jarma iz jednog dijela može biti iz intervala [300,690] npr. 520 mm
- ◆ intervalna vrijednost
parametar poprima vrijednost intervala od ... do, npr. nadmorska visina, temperatura okoline, ...
- ◆ NULL vrijednost
parametar nema definiranu vrijednost, odnosno vrijednost je NULL

4.2 Opis zadatka i uzorka

Novi konstrukcijski zadatak definiran je vrijednostima dijela parametara koji opisuju konstrukciju proizvoda. Broj parametara konstrukcije čija je vrijednost poznata u samom zadatku varira od konstrukcije do konstrukcije. Možemo npr. pretpostaviti da je novi konstrukcijski zadatak (Slika 4-2) definiran skupom vrijednosti parametara $P_z = \{p1, p3, p1.1.1, p1.1.2, p3.1\}$. Vrijednost ostalih parametara nije poznata u zadatku.



Slika 4-2: Opisivanje konstrukcije, novog konstrukcijskog zadatka i uzorka

Uzorak koji je pohranjen u bazu definiran je vlastitim vrijednostima parametara. Općenito, za uzorak ne moraju biti poznate vrijednosti svih parametara konstrukcije. Drugim riječima, za opisivanje uzoraka ne koriste se svi parametri konstrukcije. Možemo na primjer pretpostaviti da je uzorak (Slika 4-2) s kojim uspoređujemo novi zadatak definiran skupom vrijednosti parametara $P_Z = \{p1, p2, p1.1, p1.1.1, p1.1.2, p3.1, p3.2.1\}$

Iz ovog razmatranja vidimo da zadatak i razmatrani uzorak ne moraju biti definirani vrijednostima istih parametara; vrijednost parametra p1.1 poznata je u uzorku dok u zadatku nije, ili obrnuto vrijednost parametra p3 poznata je u zadatku dok u uzorku nije. Skupovi parametara koji opisuju uzorak i parametara koji opisuju zadatak su podskupovi skupa parametara koji opisuju konstrukciju proizvoda, prema tome možemo pisati:

$$\begin{aligned} P_U &\subseteq P_K \\ P_Z &\subseteq P_K' \end{aligned} \quad (4-1)$$

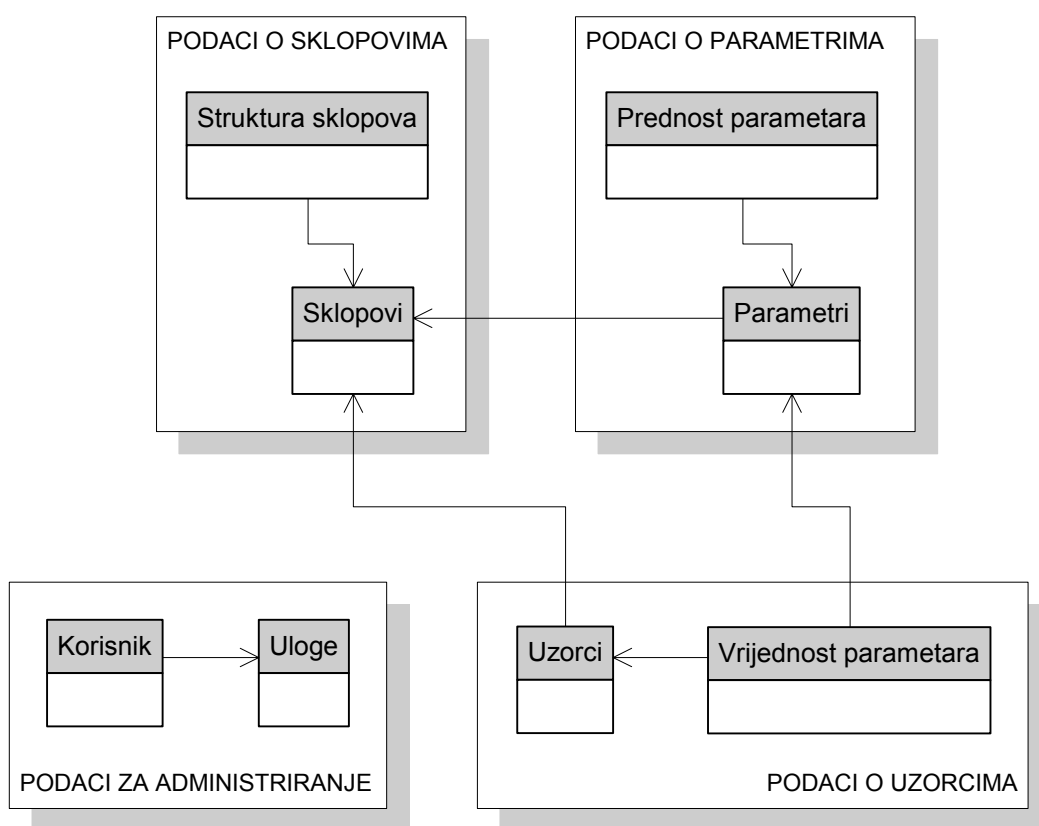
gdje je:

P_U – skup parametara uzorka
 P_Z – skup parametara zadatka
 P_K – skup parametara konstrukcije

Koji će se parametri koristiti za opisivanje uzoraka u bazi definira se prilikom realizacije sustava za prepoznavanje uzoraka. Realizirani sustav mora osigurati mogućnost dodavanja novih parametara za opisivanje uzoraka, odnosno njegovo uklanjanje.

4.3 Informacijski model baze uzoraka

Baza uzoraka koristi se za pohranjivanje uzoraka. U komercijalnim sustavima koji koriste metodologiju rasuđivanja temeljem uzoraka najčešće se koriste dva načina zapisa uzoraka (poglavlje 3.3): jednostavna linearna struktura podataka u obliku tekstualnih datoteka (ASCII) i struktura relacijske baze podataka. Iz prethodnih razmatranja u ovoj glavi slijedi da baza uzoraka mora imati, neovisno o tome koji se model koristi, entitete prikazane na slici (Slika 4-3).



Slika 4-3: Model baze uzoraka

Entiteti su podijeljeni u četiri skupine:

- ◆ Podaci o sklopovima

Sastoje se od entiteta *Sklopovi* koji sadrži podatke o sklopovima za koje se pohranjuju uzorci i entiteta *Struktura sklopa* gdje je zapisana fizička struktura proizvoda. Svakom sklopu zapisanom u entitetu *Sklopovi* pridruženi su njemu podređeni i nadređeni sklopovi zapisani u entitetu *Struktura sklopa*.

◆ Podaci o parametrima

Sastoje se od entiteta `Parametri` i `Prednost parametara`. Entitet `Parametri` sadrži podatke o parametrima koji se koriste za opisivanje uzoraka. Entitet `Prednost parametara` sadrži podatke o međusobnim odnosima među parametrima (poglavlja 5.3.1 i 5.3.2). Parametri zapisani u entitetu `Parametri` pridruženi su sklopovima zapisanim u entitetu `Sklopovi`.

◆ Podaci o uzorcima

Sastoje se od entiteta `Uzorci` koji sadrži podatke o pohranjenim uzorcima i entiteta `Vrijednost parametara` koji sadrži podatke o vrijednostima parametara svakog uzorka pohranjenog u bazu uzoraka. Svakom parametru u entitetu `Parametri` pridružena je vrijednost parametra za svaki uzorak. Svaki uzorak pohranjen u bazu pridružen je nekom sklopu iz entiteta `Sklopovi`.

◆ Podaci za administriranje

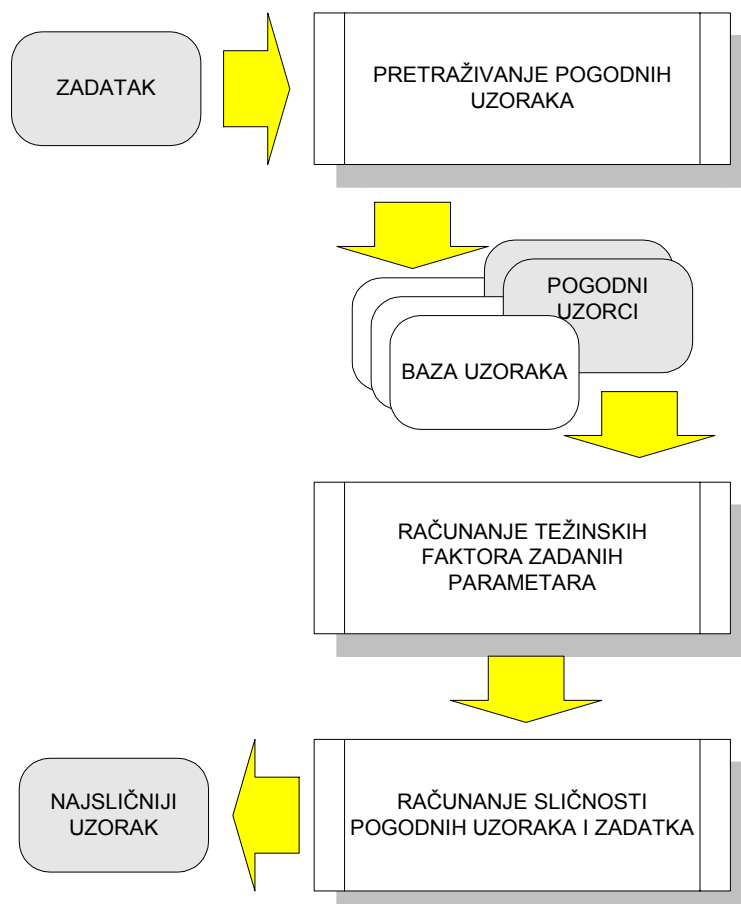
Sastoje se od entiteta `Korisnik` koji sadrži podatke o korisnicima sustava i entiteta `Uloge` koji sadrži definiranu ulogu za svakog korisnika sustava.

5 PRONALAŽENJE NAJSLIĆNIJIIH UZORAKA

U glavi 5 pobliže će se razmotriti faza pronalaženja najslićnijeg uzorka kao dio ciklusa rasuđivanja temeljem uzoraka (poglavlje 3.2). Prvi dio opisuje informacijski model na kojem je temeljena faza pronalaženja najslićnijih uzoraka. U nastavku su opisani koraci faze pronalaženja s primjerima.

5.1 Informacijski model

Da bismo pronašli uzorak najslićniji novom konstrukcijskom zadatku potrebno je izvršiti korake shematski prikazane na slici (Slika 5-1).



Slika 5-1: Informacijski model pronalaženja

U ovom radu se predlaže pronalaženje najbližeg uzorka u tri koraka.

Prvi korak je pretraživanje baze uzoraka temeljem vrijednosti čvornih parametara (glava 5) i dobivanje pogodnih uzoraka. Pogodni uzorci su oni čija vrijednost čvornih parametara odgovara vrijednosti čvornih parametara novog konstrukcijskog zadatka.

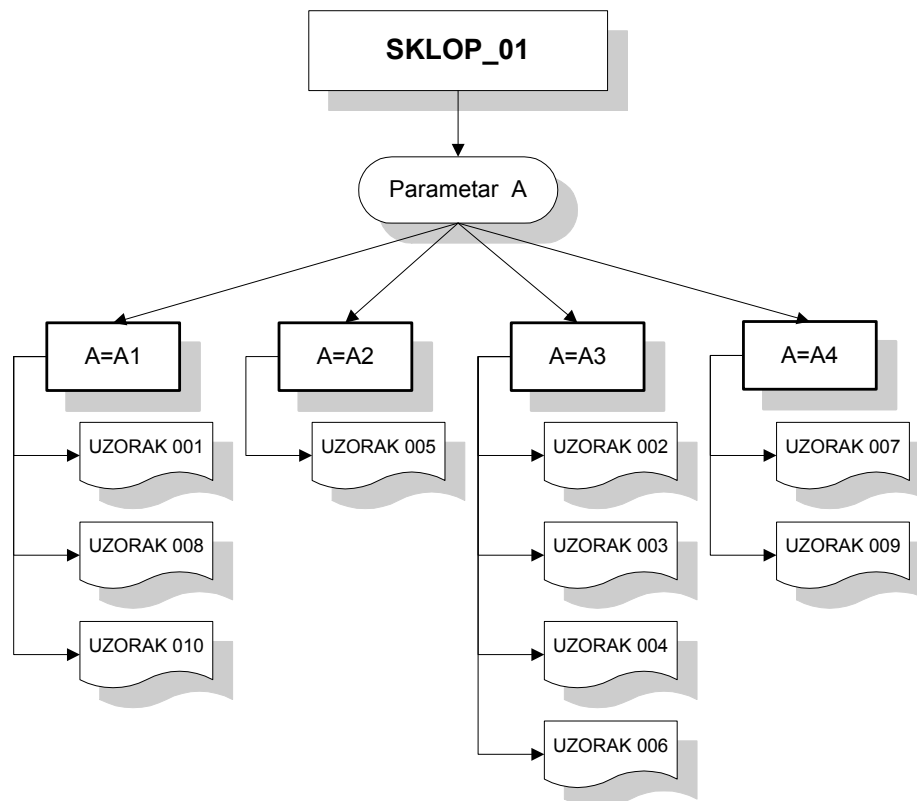
Drugi korak je računanje težinskih faktora onih parametara koji su definirani u novom konstrukcijskom zadatku.

Treći korak je računanje sličnosti pogodnih parametara i novog konstrukcijskog zadatka. Uzorak koji ima vrijednost sličnosti najbližu jedinici predstavlja najbliži uzorak.

U sljedećim poglavljima se pobliže opisuju navedeni koraci.

5.2 Pretraživanje pogodnih uzoraka

Ukoliko je baza uzoraka velika¹² tada bi korištenje metode najbliži susjed zahtjevalo da se svaki parametar zadatka uspoređi sa svakim parametrom svakog uzorka u bazi što bi u slučaju velikog broja uzoraka i parametara zahtjevalo dosta vremena.



Slika 5-2: Prikaz pretraživanja pogodnih uzoraka

Kako bi se spriječilo nepotrebno računanje težinskih faktora i sličnosti svih uzoraka u bazi potrebno je iz baze uzoraka izdvojiti samo pogodne uzorke. Parametri koji opisuju uzorke razlikuju se po svojim vrijednostima koje poprimaju. Ukoliko neki parametar može poprimiti mali broj različitih vrijednosti tada je pogodan da ga proglasimo karakterističnim parametrom ili čvornim parametrom (poglavlje 4.1.2). Uzorci koji imaju istu vrijednost čvornog parametra kao i zadatak nazivamo pogodni uzorci.

¹² Većina CBR sustava u bazi uzoraka sadrži od nekoliko stotina do nekoliko tisuća uzoraka [19]. Vrlo velike baze uzoraka smatraju se one koje sadrže više od 20000 uzoraka

Realizacijom baze uzoraka kao relacijske baze podataka jednostavnim upitom kojim se uspoređuju zadane vrijednosti čvornih parametara zadatka i svih uzoraka u bazi dobiju se pogodni uzorci.

Slika 5-2 prikazuje pretraživanje baze uzoraka i pronalaženje pogodnih uzoraka. Za sklop koji se zove SKLOP_01 definiran je čvorni parametar A koji može poprimiti jednu od četiri moguće vrijednosti A1, A2, A3 ili A4.

Ukoliko novi zadatak za koji tražimo najbliži uzorak ima vrijednost čvornog parametra A=A1 tada uzorci UZORAK 001, UZORAK 008 i UZORAK 010 predstavljaju pogodne uzorke.

5.3 Računanje težinskih faktora

Za računanje sličnosti metodom najbliži susjed potrebno je izračunati težinske faktore promatranih parametara. U poglavlju 3.4.1.1 opisani su razlozi uvođenja težinskih faktora. Postoje dvije vrste težinskih faktora globalni i lokalni [60]. Razlika među njima je što globalni težinski faktor ima istu vrijednost za sve uzorke u bazi dok se vrijednost lokalnog težinskog faktora mijenja od uzorka do uzorka. Ovdje ćemo se ograničiti samo na razmatranje globalnih težinskih faktora.

Kod dodjeljivanja težinskih faktora koristit će se matrica parametara u koju upisujemo međusobne odnose između parametara. U nastavku su opisana dva pristupa određivanju težinskih parametara: ocjenjivanjem prednosti i određivanjem prednosti.

5.3.1 Ocjenjivanje prednosti

Kod pristupa ocjenjivanja prednosti radi se o normiranoj matrici (Tablica 3) gdje se u polja matrice opisuje ocjena prednosti jednog parametra u odnosu na drugi. Npr. vrijednost a_{ij} predstavlja prednost parametra $p[i]$ u odnosu na parametar $p[j]$.

	$p[1]$	$p[2]$	...	$p[j]$	...	$p[n]$
$p[1]$		a_{12}	...	a_{1j}	...	a_{1n}
$p[2]$	a_{21}					
...	...					
$p[i]$	a_{i1}					
...						
$p[n]$						

Tablica 3: Ocjenjivanje prednosti parametara

gdje vrijedi:

$$a_{21} = \frac{1}{a_{12}},$$

(5-1)

ili općenito:

$$a_{ji} = \frac{1}{a_{ij}}. \quad (5-2)$$

Matrica na dijagonali ima jedinice, a simetrični članovi matrice imaju inverzne vrijednosti što predstavlja konzistentnost matrice [61], [62], dakle poznavanjem vrijednosti člana s jedne strane dijagonale određena je vrijednost simetričnog člana.

Vrijednost težinskog faktora parametra $p[I]$ biti će jednaka:

$$w_i = \frac{\sum_{j=1}^n a_{ij}}{\sum_{i=1}^n \sum_{j=1}^n a_{ij}}, \quad i \neq j. \quad (5-3)$$

Računanjem vrijednosti težinskog faktora svakog parametra dobije se vektor

$$w = \{w_1, w_2, \dots, w_i, \dots, w_n\}, \quad (5-4)$$

koji predstavlja težinske faktore parametara u rasponu od 0 do 1 tako da vrijedi:

$$\sum_{i=1}^n w_i = 1, \quad (5-5)$$

što upućuje na normiranost matrice.

Vrijednosti ocjena prednosti mogu se definirati unutar nekog intervala npr. 1...10, gdje 10 predstavlja izrazito visoku ocjenu prednosti. Također se mogu koristiti opisne ocjene kojima se onda pridružuje numerička vrijednost:

<i>Opisna ocjena</i>	<i>Numerička zamjena I način</i>	<i>Numerička zamjena II način</i>
<i>jednaka važnost</i>	<i>1</i>	<i>-2</i>
<i>mala prednost</i>	<i>2</i>	<i>-1</i>
<i>srednja prednost</i>	<i>3</i>	<i>-0</i>
<i>umjerena prednost</i>	<i>4</i>	<i>1</i>
<i>velika prednost</i>	<i>5</i>	<i>2</i>

Tablica 4: Ocjena prednosti parametara opisnom ocjenom

5.3.2 Određivanje prednosti

Određivanje prednosti temelji se na uspoređivanju svakog parametra sa svim ostalim parametrima i dodjeljivanja prednosti promatranom parametru. Ukoliko je prednost na strani parametra kojeg uspoređujemo piše se vrijednost 1, ukoliko je prednost na strani parametra s kojim uspoređujemo piše se vrijednost 2, a ukoliko parametri imaju jednaku važnost piše se 0.

Parametri koji se nalaze u retku matrice su parametri koje uspoređujemo s parametrima u stupcima matrice (Tablica 5):

	$p[1]$	$p[2]$	...	$p[j]$	...	$p[n]$
$p[1]$		1	...	2	...	0
$p[2]$	2			1		
...	...					
$p[i]$	1	2				
...						
$p[n]$	0					

Tablica 5: Određivanje prednosti parametara

gdje vrijedi:

- 1 – u polju $p[i], p[j]$ znači prednost $p[i]$ u odnosu na $p[j]$;
- 2 – u polju $p[i], p[j]$ znači prednost $p[j]$ u odnosu na $p[i]$;
- 0 – znači jednaku važnost parametara

Konzistentnost matrice se postiže zadovoljavanjem izraza:

$$\begin{aligned}
 a_{ij} = 1 &\Rightarrow a_{ji} = 2, \quad i \neq j \\
 a_{ij} = 2 &\Rightarrow a_{ji} = 1, \quad i \neq j. \\
 a_{ij} = a_{ji} &= 0, \quad i = j
 \end{aligned}$$

(5-6)

Težinski faktor parametra $p[i]$ biti će jednak zbroju jedinica u retku dijeljeno sa ukupnim brojem jedinica u matrici:

$$w_i = \frac{\sum_{j=1}^n \{a_{ij} \mid a_{ij} = 1\}}{\sum_{i=1}^n \sum_{j=1}^n \{a_{ij} \mid a_{ij} = 1\}},$$

(5-7)

čime dobijemo normiranu matricu.

U odnosu na ocjenjivanje prednosti može se reći da je određivanje prednosti u neku ruku jednostavnije rješenje zbog toga što je kod određivanja prednosti jednog parametra naspram drugog bitno samo koji je parametar važniji u smislu računanja sličnosti, dok je kod ocjenjivanja prednosti potrebno rangirati prednost ovisno o broju nivoa. Međutim, s

ocjenjivanjem prednosti međusobnu ovisnost parametara podešavamo “finije”, u više nivoa, što u nekim domenama može biti značajna prednost.

5.3.3 Primjer izračunavanja težinskih faktora

Razmotrimo izračunavanje težinskih faktora određivanjem prednosti na primjeru sklopa konzervator energetskog transformatora i to za osam parametara (Tablica 6).

<i>Oznaka</i>	<i>Naziv</i>	<i>Opis</i>
<i>p1</i>	<i>KN_LKON</i>	<i>duljina plašta</i>
<i>p2</i>	<i>KN_DPLASTA</i>	<i>debljina plašta</i>
<i>p3</i>	<i>KN_PGC</i>	<i>položaj glavnog cjevovoda</i>
<i>p4</i>	<i>KN_PPS</i>	<i>položaj priključaka i sušionika</i>
<i>p5</i>	<i>KN_LNRP</i>	<i>udaljenost nosača od ruba plašta</i>
<i>p6</i>	<i>KN_RAZNOS</i>	<i>razmak nosača</i>
<i>p7</i>	<i>KN_LNOSACA</i>	<i>visina nosača</i>
<i>p8</i>	<i>KN_RU</i>	<i>razmak ušica</i>

Tablica 6: Neki parametri sklopa konzervatora energetskog transformatora

U tablici prednosti (Tablica 7) navedeni su međusobni odnosi parametara i izračunate vrijednosti težinskih faktora za svaki parametar.

	<i>p1</i>	<i>p2</i>	<i>p3</i>	<i>p4</i>	<i>p5</i>	<i>p6</i>	<i>p7</i>	<i>p8</i>	ΣI	w_i
<i>p1</i>		1	0	0	1	1	1	1	5	0.20
<i>p2</i>	2		2	2	2	2	2	1	1	0.04
<i>p3</i>	0	1		0	1	1	1	1	5	0.20
<i>p4</i>	0	1	0		1	1	1	1	5	0.20
<i>p5</i>	2	1	2	2		2	2	1	2	0.08
<i>p6</i>	2	1	2	2	1		1	1	4	0.16
<i>p7</i>	2	1	2	2	1	2		1	3	0.12
<i>p8</i>	2	2	2	2	2	2	2		0	0.00
								Σ	25	1.00

Tablica 7: Tablica prednosti s težinskim faktorima

Iz tablice prednosti vidimo da npr. parametar *KN_PGC* (položaj glavnog cjevovoda) oznake *p3* ima jednaku važnost kao i parametri *KN_LKON* (duljina plašta) oznake *p1* i *KN_PPS* (položaj priključaka i sušionika) oznake *p4* dok nad ostalim parametrima ima prednost. Stupac ΣI predstavlja sumu jedinica za parametar koji se nalazi u pojedinom retku matrice. Dijeljenjem broja jedinica za svaki parametar sa ukupnim brojem jedinica u matrici dobijemo vektor težinskih faktora:

$$w = \{0,20 \quad 0,04 \quad 0,20 \quad 0,20 \quad 0,08 \quad 0,16 \quad 0,12 \quad 0,00\} \quad (5-8)$$

5.4 Primjer izračunavanja sličnosti

Za izračunavanje sličnosti ovdje će se koristiti najjednostavnija i najviše korištena metoda najbliži susjed [45], [19], [5], [28], [63].

Kao što je već rečeno u poglavlju 3.4.1 sličnost j-tog uzorka metodom najbliži susjed izračunava se prema izrazu:

$$S_j = 1 - \frac{d(U_j, Z)}{\sum_{j=1}^m d(U_j, Z)} = 1 - \frac{\left[\sqrt{\sum_{i=1}^n w_i \cdot \text{razlika}(u_i, z_i)^2} \right]_j}{\sum_{j=1}^m d(U_j, Z)}.$$

(5-9)

Uzmimo za primjer deset uzoraka sklopa konzervator energetskeg transformatora i izračunajmo sličnost svakog uzorka sa zadatkom koristeći prije izračunate vrijednosti težinskih faktora (Tablica 7). Pretpostavka je da su uzorci pogodni, odnosno da imaju iste vrijednosti čvornih parametara.

	<i>U001</i>	<i>U002</i>	<i>U003</i>	<i>U004</i>	<i>U005</i>	<i>U006</i>	<i>U007</i>	<i>U008</i>	<i>U009</i>	<i>U010</i>	<i>ZADATAK</i>
<i>p1</i>	1900	1500	2200	2100	2600	1800	2900	2000	2100	1900	2400
<i>p2</i>	5	5	6	6	6	6	5	6	6	6	6
<i>p3</i>	540	825	1750	1200	632	446	965	1142	758	800	1050
<i>p4</i>	1250	356	765	458	1795	1250	2562	680	1450	1662	1820
<i>p5</i>	550	425	675	690	690	480	775	600	650	575	625
<i>p6</i>	800	650	850	720	1220	840	1350	800	800	750	1150
<i>p7</i>	255	255	450	460	520	255	335	255	425	400	255
<i>p8</i>	910	750	1200	1000	1500	900	1800	990	1000	920	1400
<i>S_i</i>	0,909	0,831	0,876	0,862	0,952	0,900	0,913	0,882	0,939	0,935	

Tablica 8: Primjer izračunavanja sličnosti

U zadnjem retku tablice izračunate su vrijednosti sličnosti S_j prema izrazu (5-9) pojedinih uzoraka. Što je sličnost bliže jedinici to je uzorak sličniji zadatku.

Iz tablice (Tablica 8) vidimo da uzorak sa oznakom *U005* ima najveću sličnost sa zadatkom. Nakon njega slijede *U009* i *U010*. Najmanje sličan uzorak je uzorak sa oznakom *U002*.

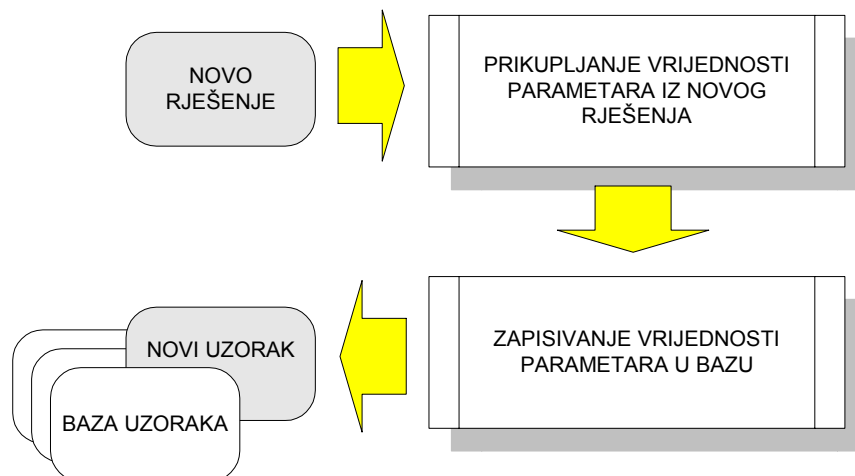
6 POHRANJIVANJE NOVOG RJEŠENJA

Novo konstrukcijsko rješenje koje se dobije rješavanjem konstrukcijskog zadatka, temeljem najbližijeg uzorka kao predloška, potrebno je pohraniti u bazu uzoraka.

U glavi 6 razmotriti ćemo aktivnosti koje je potrebno izvršiti za pohranjivanje novog konstrukcijskog rješenja.

6.1 Informacijski model

Prilikom pohranjivanja novog rješenja u bazu uzoraka prvi korak je prikupljanje vrijednosti parametara novog konstrukcijskog rješenja (Slika 6-1). Pretpostavka je da je novo konstrukcijsko rješenje izrađeno i verificirano, odnosno da se neće više mijenjati. Svako konstrukcijsko rješenje definirano je strukturom sklopova koja je zapisana u bazu uzoraka, a svakom sklopu pridruženi su parametri (glava 4). Pohranjivanjem u bazu uzoraka vrijednosti parametara novog konstrukcijskog rješenja dobiva se novi uzorak u bazi uzoraka. Na taj način prikuplja se iskustvo nastalo rješavanjem novih konstrukcijskih zadataka i povećava se baza uzoraka, odnosno broj pohranjenih uzoraka.



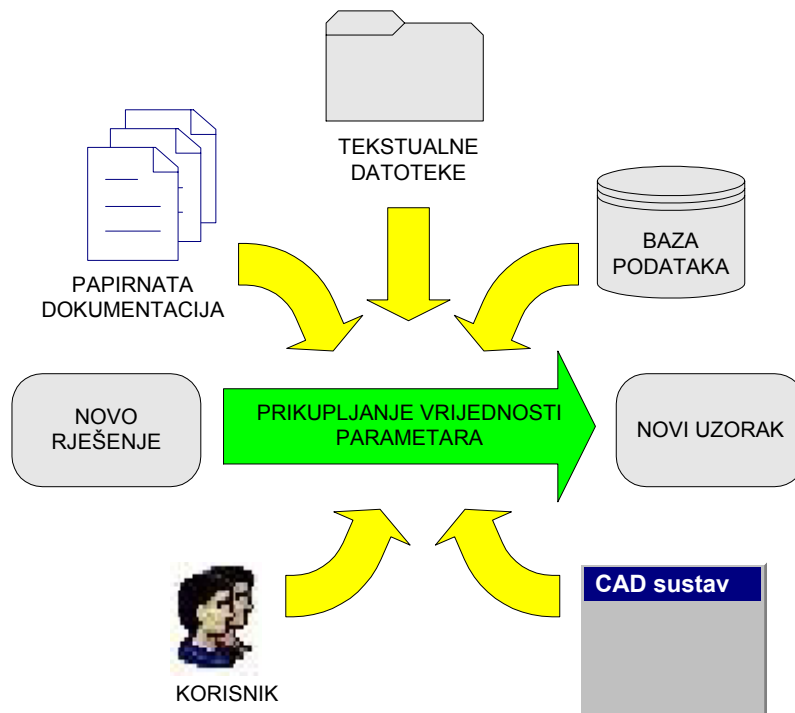
Slika 6-1: Informacijski model pohranjivanja novog rješenja

6.2 Prikupljanje vrijednosti parametara

Prvi korak pri pohranjivanju novog konstrukcijskog rješenja je prikupljanje vrijednosti parametara koji opisuju konstrukcijsko rješenje (Slika 6-2). Nakon prikupljanja vrijednosti parametara novo konstrukcijsko rješenje smatramo novim uzorkom kojeg treba pohraniti u bazu uzoraka.

Vrijednosti parametara novog novog uzorka mogu se prikupljati na različite načine:

- ◆ iz papirnate dokumentacije,
- ◆ čitanjem iz tekstualne datoteke,
- ◆ čitanjem iz baze podataka,
- ◆ čitanjem iz CAD sustava,
- ◆ unosom korisnika u sustav.



Slika 6-2: Načini prikupljanja vrijednosti parametara

U varijantnim konstrukcijama tehnička specifikacija proizvoda sadrži vrijednosti parametara zadatka i djelomično rješenja (poglavlje 4.2). Često je to papirnata dokumentacija iz koje treba prepoznati parametre i njihove vrijednosti zapisati u informatičkom obliku tijekom procesa razvoja proizvoda, proizvodnje ili tijekom pohranjivanja novog uzorka u bazu uzoraka. Pod papirnatom dokumentacijom podrazumijeva se i dokumentacija koja nastaje tijekom procesa razvoja proizvoda, a služi za međufaznu razmjenu informacija: ponuda, projekt, osnutak, proračuni, i dr. (Slika 2-7 i Slika 2-8). Također, pod papirnatom dokumentacijom podrazumijevamo i dokumentaciju nastalu tijekom procesa proizvodnje: tehnološka dokumentacija, lista provjera i izvješća s ispitivanja (Slika 2-6), pod uvjetom da nije zapisana u nekom informatičkom obliku čitljivom iz sustava za prepoznavanje uzoraka.

Tekstualna datoteka, baza podataka i parametarski trodimenzionalni CAD sustav predstavljaju izvore vrijednosti parametara kojima je moguće pristupiti iz sustava za prepoznavanje uzoraka.

Proračuni koji se izrađuju tijekom procesa razvoja proizvoda (npr. električni proračun, proračuni čvrstoće, proračun na potres, i dr.) pomoću korisničkih aplikacija kao rezultate mogu sadržavati vrijednosti nekih parametara modela ili uzorka. Rezultati proračuna zapisuju se u tekstualne datoteke, bazu podataka ili parametarski model.

Osim geometrijskih parametara unutar modela mogu se definirati parametri koji sadrže i neke druge informacije o modelu (npr. vrijeme izrade, napomene, podaci o konstruktoru i dr.).

Crteži izrađeni pomoću CAD sustava za dvodimenzionalnu dokumentaciju mogu se također koristiti kao izvor vrijednosti parametara. U ovom radu neće se posebno razmatrati prikupljanje vrijednosti parametara iz tekstualnih datoteka iz razloga što svi programski jezici koji se koriste za razvoj aplikacija imaju podršku za čitanje tekstualnih datoteka. Jedan od pristupa prikupljanju podataka iz baze podataka prikazan je u glavi 7.

Vrijednosti parametara koji nisu dostupni niti u jednom od navedenih izvora, a nužni su za opisivanje uzorka u bazi korisnik treba unijeti u sustav.

U slijedećem poglavlju поближе će se objasniti način prikupljanja parametara iz parametarskog modela.

6.3 Prikupljanje podataka iz parametarskog modela

Parametarski trodimenzionalni model može osim geometrijskih parametara¹³ imati i korisnički definirane parametre [64] koji sadržavaju podatke pridružene modelu (npr. broj dijela, datum izrade, i dr.). Pored brojčanih vrijednosti (cjelobrojnih i realnih) korisnički definirani parametri mogu poprimiti i tekstualne vrijednosti. Parametri kojima je opisan uzorak mogu se unutar modela definirati kao korisnički parametri (npr. sklop konzervatora ima definiran parametar `DKON` koji odgovara parametru u bazi uzoraka `KN_DKON`).

Slika 6-3 prikazuje primjer parametarskog modela nosača. Može se uočiti da su dimenzije gornje ploče nosača definirane geometrijskim parametrima `d37`, `d38`, `d39`, `d40` i `d41`. Korisnički parametri (ukoliko su brojčani) i parametri modela mogu biti u međusobnoj relaciji. Slika 6-4 prikazuje korisnički definirane parametre za prikazani model te relacije između parametara modela i korisnički definiranih parametara. Korisnički parametri su `DEBLJINA`, `DULJINA`, `SIRINA` i `OZNAKA_MODELA`. Na slici (Slika 6-4) se može uočiti da su parametri `d41`, `d37` i `d38` povezani sa brojčanim korisnički definiranim parametrima relacijom jednakosti.

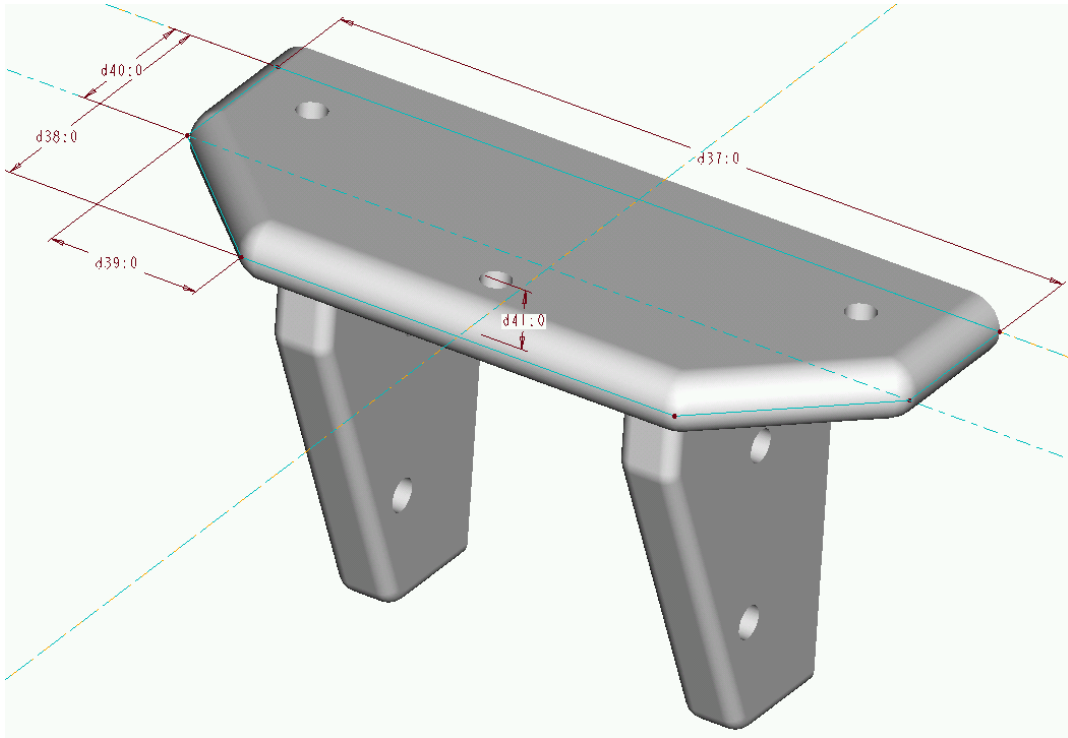
Čitanjem baze modela prikupljamo vrijednosti parametara modela i korisnički definiranih parametara.

Za prikupljanje vrijednosti parametra unutar parametarskog trodimenzionalnog modela potrebno je imati slijedeće informacije (Tablica 9):

<i>informacija</i>	<i>primjer modela nosača</i>	<i>primjer sklopa konzervatora</i>
<i>oznaka parametra u modelu</i>	<code>DEBLJINA</code>	<code>DKON</code>
<i>tip parametra</i>	<code>INTEGER</code>	<code>INTEGER</code>
<i>naziv modela</i>	<code>B10 01 002</code>	<code>KONZERVA SKEL</code>
<i>tip modela</i>	<code>PRT (pozicija)</code>	<code>PRT (pozicija)</code>
<i>putanja do modela</i>	<code>\\edserver1\nosac\</code>	<code>\\edserver1\trafo\T12345\</code>

Tablica 9: Informacije potrebne za prikupljanje vrijednosti parametara unutar modela

¹³ Pod pojmom geometrijski parametar podrazumijeva se parametar koji definira geometriju modela, a kreiran je od strane CAD sustava prilikom modeliranja.



Slika 6-3: Model nosača

- ◆ Oznaka parametra u modelu predstavlja naziv parametra u modelu pod kojim se parametar pohranjuje u bazu modela. U slučaju da je parametar korisnički definiran tada je njegov naziv definirao korisnik prilikom kreiranja parametra. U protivnom, kod geometrijskih parametara nazive automatski dodjeljuje CAD sustav (npr d40:0, d38:0¹⁴, ...).
- ◆ Tip parametra može biti cjelobrojni (integer), realni (real) ili tekstualni (string).
- ◆ Naziv modela je naziv datoteke modela.
- ◆ Tip modela govori da li je model pozicija (part) ili sklop (assembly).
- ◆ Putanja do modela pokazuje mjesto gdje je modela pohranjen nakon rješavanja novog konstrukcijskog zadatka.

Parametre čije se vrijednosti iz bilo kojeg razloga ne mogu pronaći u izvoru koji je definiran u bazi potrebno je predočiti korisniku i dozvoliti mu unos njihovih vrijednosti. U slučaju da korisnik ne unese vrijednosti one poprimaju NULL vrijednost.

¹⁴ CAD sustav dodjeljuje svakoj poziciji (part) unutar sklopa (assembly) identifikacijsku oznaku. U ovom primjeru gornja ploča na modelu nosača ima identifikacijski broj 0.

RELATION -----	PARAMETER -----	NEW VALUE --- --
/*** Relations for B10_01_002:		
D41=debljina	D41	2.000000e+01
D37=duljina	D37	2.500000e+02
D38=sirina	D38	1.000000e+02
Symbolic constant -----	X-refs -----	Current value -----
DEBLJINA	Local	2.000000e+01
DULJINA	Local	2.500000e+02
SIRINA	Local	1.000000e+02
OZNAKA_MODELA	Local	""B50_03_426""

Slika 6-4: Relacije i korisnički parametri u modelu nosača

Informacije o parametrima unutar parametarskog modela koji opisuju uzorke nalaze se u bazi uzoraka (poglavlje 7.2).

6.4 Pohranjivanje uzorka

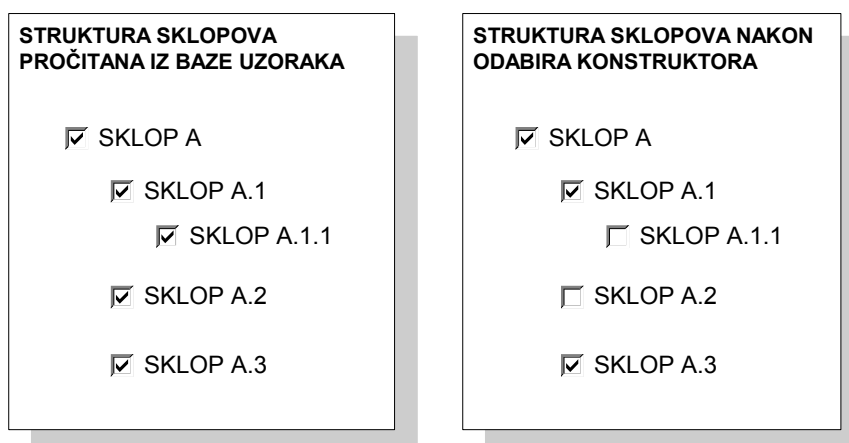
Zapisivanje prikupljenih vrijednosti parametara novog konstrukcijskog rješenja u bazu uzoraka nazivamo pohranjivanje uzorka. Prikupljene vrijednosti parametara zapisuju se u odgovarajući entitet baze uzoraka (poglavlje 4.3).

Za pohranjivanje uzorka u bazu uzoraka potrebno je poznavati:

- ◆ identifikacijsku oznaku,
- ◆ identifikacijske oznake parametara,
- ◆ identifikacijske oznake sklopova,
- ◆ putanju uzorka,
- ◆ vrijednost parametara,
- ◆ opis uzorka.

Identifikacijske oznake su jedinstvene oznake u bazi uzoraka. Putanja uzorka definira mjesto gdje je pohranjen model uzorka sa pripadajućim crtežima i ostalim datotekama (rezultati proračuna, i dr.) nakon pohranjivanja uzorka. Vrijednosti parametara su prikupljene vrijednosti koje treba pohraniti u bazu uzoraka.

Slika 6-5 lijevo prikazuje strukturu nekog sklopa A pročitane iz baze uzoraka. Konstruktor može prihvatiti kompletnu strukturu sklopa sa svim podsklopovima ili može izbaciti sklopove za koje ne želi pohraniti novi uzorak. Slika 6-5 desno prikazuje strukturu sklopa A sa izbačenim sklopovima A.2 i A.1.1.



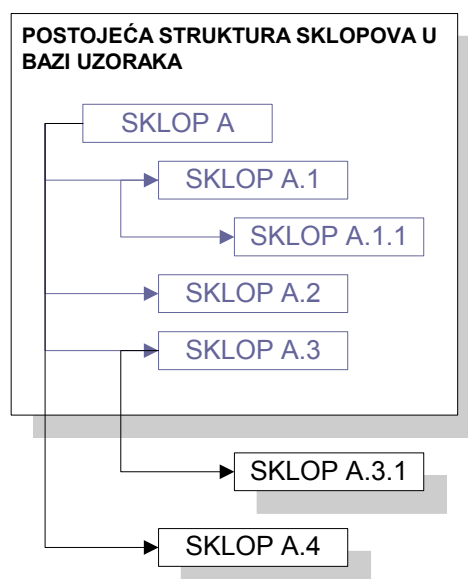
Slika 6-5: Odabir sklopova za uzorke

6.5 Dodavanje sklopova i parametara

U sustavu za prepoznavanje uzoraka potrebno je osigurati dodavanje novih sklopova i parametara u bazu uzoraka.

Dodavanje novog sklopa podrazumijeva ubacivanje sklopa u strukturu čime se omogućava pohranjivanje novih uzoraka za taj sklop. Za ubacivanje novog sklopa potrebno je definirati:

- ◆ identifikacijsku oznaku sklopa,
- ◆ naziv sklopa,
- ◆ nadređene sklopove,
- ◆ podređene sklopove (ukoliko ih ima),
- ◆ opis sklopa.



Slika 6-6: Dodavanje novih sklopova u strukturu

Slika 6-6 prikazuje dodavanje novih sklopova (SKLOP A.3.1, SKLOP A.4) u strukturu sklopa (SKLOP A).

Novi parametar može se dodati u već postojeći sklop ili prilikom dodavanja novog sklopa.

Za dodavanje novog parametra potrebno je poznavati:

- ◆ identifikacijsku oznaku parametra,
- ◆ identifikacijsku oznaku sklopa kojem pripada,
- ◆ naziv parametra,
- ◆ vrstu parametra (čvorni, obični)¹⁵,
- ◆ tip parametra,
- ◆ tip izvora vrijednosti parametra,
- ◆ naziv izvora,
- ◆ oznaku parametra unutar izvora vrijednosti,
- ◆ ocjenu prednosti parametra u odnosu na ostale parametre.

Proširenje sustava, dakle dodavanje novog sklopa i/ili novog parametra zahtjeva od korisnika poznavanje sustava, a naravno i poznavanje konstrukcijskog rješenja proizvoda. Korisnik mora znati značenje pojedinih atributa parametara, ali isto tako, mora dovoljno dobro poznavati problematiku područja kako bi što kvalitetnije ocijenio prednost novog parametra u odnosu na ostale parametre. Iz tih razloga potrebno je implementirati u sustav i identifikaciju korisnika te definirati uloge (poglavlje 4.3).

¹⁵ Termin čvorni parametar opisan je u poglavlju 4.1.2. Parametri koji nisu čvorni nazivaju se obični

7 IMPLEMENTACIJSKI MODEL SUSTAVA

Na osnovu analize osnovnih zadata rasuđivanja temeljem uzoraka, koje CBR sustav mora obavljati, u glavi 7 predložen je implementacijski model sustava za prepoznavanje uzoraka pri konstruiranju energetskog transformatora. Sustav za prepoznavanje uzoraka potrebno je integrirati u postojeći CAE sustav tvrtke.

Model predloženog sustava sastoji se od glavnih modula¹⁶ koji su opisani u nastavku zajedno sa klasama¹⁷ koje ih definiraju. Glavne zadatke koje sustav ispunjava su pronalaženje najslabijih uzoraka i pohranjivanje novih uzoraka u bazu. Predloženi sustav za prepoznavanje uzoraka nije predviđen kao pomoć pri rješavanju novih konstrukcijskih zadataka već se novi konstrukcijski zadatak rješava pomoću postojećeg CAE sustava tvrtke koristeći najslabije prijašnje rješenje, prepoznato pomoću sustava, kao predložak.

Implementacijski model sustava za prepoznavanje uzoraka reliziran je programskim jezikom Java™ [65], [66], [67] koji je već korišten u postojećem CAE sustavu tvrtke (poglavlje 2.4.4). Za pristup postojećem CAD sustavu (ProENGINEER®) korištene su J-Link® biblioteke [68] koje su sastavni dio osnovne varijante CAD sustava.

Baza uzoraka realizirana je kao relacijska baza podataka koristeći programski paket SQL Server® tvrtke Microsoft® koji je već bio implementiran u CAE sustav tvrtke.

Veza predloženog sustava sa bazom podataka realizirana je preko upravljačkog programa tzv. JDBC™¹⁸-ODBC™¹⁹ mosta koji omogućava klasama programskog jezika Java™ pristup bazi podataka realiziranoj programskim paketom SQL Server®.

Na opisani način povezani su predloženi sustav za prepoznavanje uzoraka, baza podataka, CAD sustav za modeliranje i konstruktor.

Dio postojećeg CAE sustava koji se koristi za izradu dvodimenzionalnih crteža (CAD sustav AutoCAD®) u ovom radu nije posebno razmatran.

¹⁶ Pod pojmom modul podrazumijeva se programska cjelina koja izvršava određenu zadataku

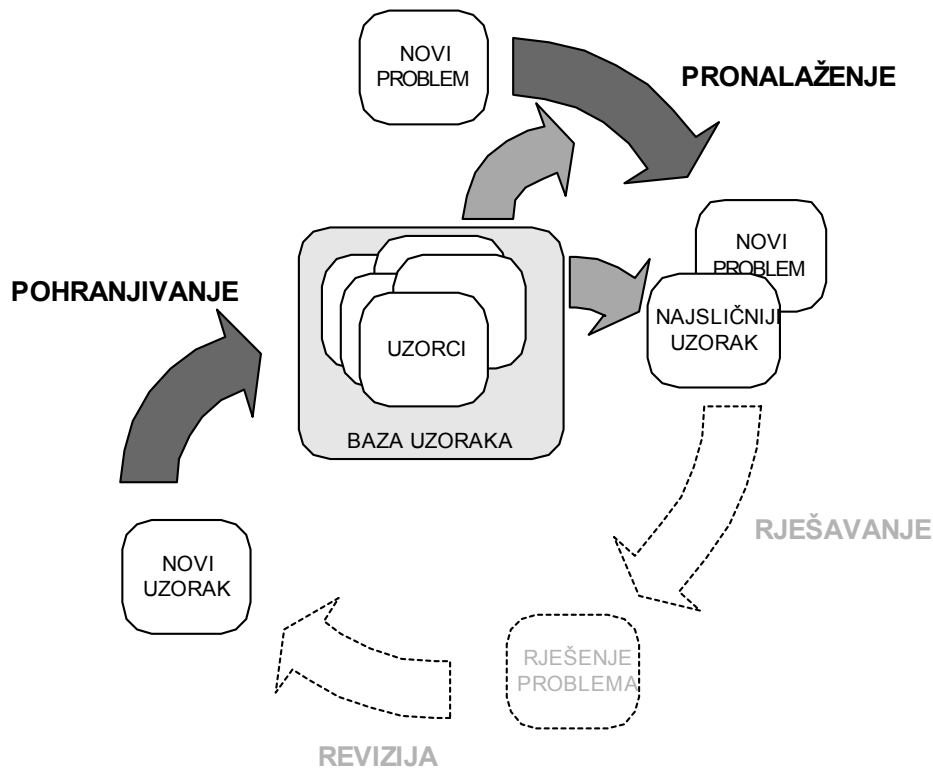
¹⁷ Ovdje se pojam klasa odnosi na objektno orijentiranu paradigmu programiranja (Object Oriented Programming)

¹⁸ JDBC – Java Database Connectivity, programsko sučelje za pristup bazi podataka za programski jezik Java

¹⁹ ODBC – Open Database Connectivity, programsko sučelje za pristup bazi podataka tvrtke Microsoft

7.1 Struktura sustava za prepoznavanje uzoraka

Kao što je navedeno u poglavlju 3.2 proces prepoznavanja sličnih uzoraka uobičajeno se može promatrati kao ciklički proces koji se sastoji iz četiri osnovne faze (Slika 7-1).



Slika 7-1: Dio ciklusa rasuđivanja koji je realiziran u predloženom sustavu

Predloženi sustav za prepoznavanje uzoraka obuhvaća dvije faze ciklusa rasuđivanja: pronalaženje i pohranjivanje; te realizaciju baze uzoraka, što je istaknuto na slici (Slika 7-1). Rješavanje novog konstrukcijskog zadatka te njegova revizija realiziraju se u postojećem CAE sustavu tvrtke (poglavlje 2.4.4).

Na osnovu razmatranja u glavi 4 realizirano je zapisivanje uzoraka u relacijsku bazu podataka. Organizacijska struktura baze uzoraka opisana je u poglavlju 7.2.

Predloženi model sustava za prepoznavanje uzoraka ima dvije osnovne zadaće:

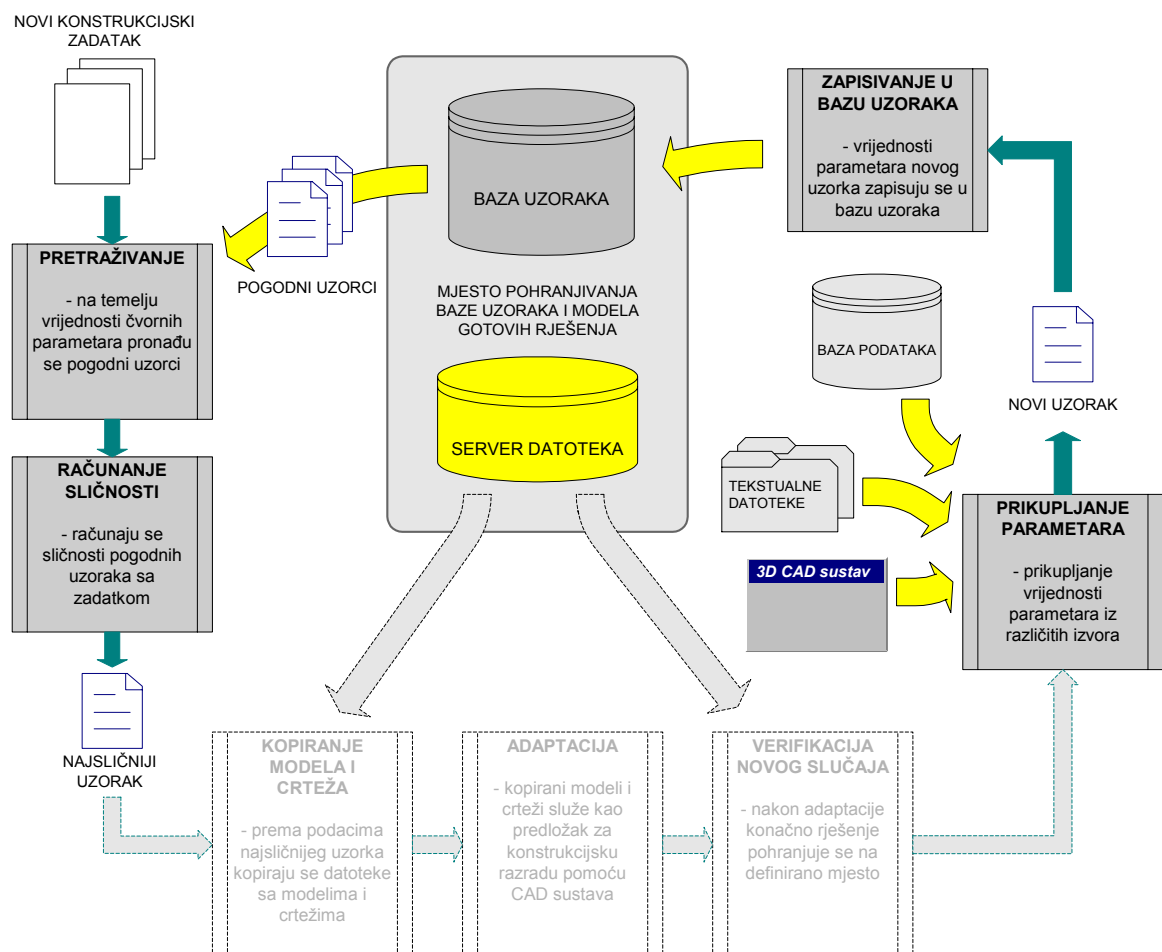
- ◆ pronalaženje sličnih prijašnjih rješenja,
- ◆ pohranjivanje novih uzoraka u bazu uzoraka.

Sukladno sa razmatranjima navedenih zadaća u glavama 5 i 6 potrebno je osigurati izvršavanje aktivnosti:

- ◆ prikupljanje podataka o zadatku,
- ◆ odabiranje čvornih parametara,
- ◆ pronalaženje pogodnih uzoraka na osnovu vrijednosti čvornih parametara,
- ◆ računanje težinskih faktora za parametre čiju vrijednost znamo,

- ♦ računanje sličnosti pogodnih uzoraka,
- ♦ sortiranje uzoraka prema izračunatoj sličnosti,
- ♦ prikupljanje vrijednosti parametara iz trodimenzionalnog modela,
- ♦ pohranjivanje uzorka u bazu zapisivanjem vrijednosti parametara,
- ♦ dodavanje novih parametara,
- ♦ brisanje parametara iz baze,
- ♦ dodavanje novih sklopova,
- ♦ brisanje sklopova.

Slika 7-2 shematski prikazuje predloženi sustav za prepoznavanje uzoraka.



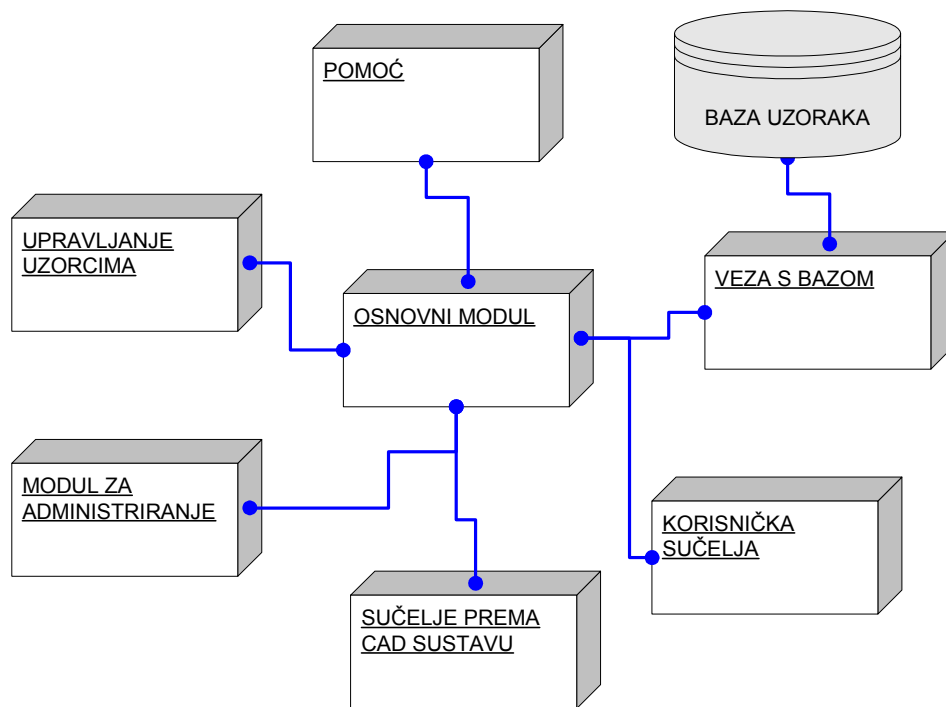
Slika 7-2: Shematski prikaz sustava za prepoznavanje uzoraka

Baza uzoraka sadrži podatke o uzorcima. Definicija novog konstrukcijskog zadatka predstavlja ulazne vrijednosti u sustav. Prvi korak je pretraživanje pogodnih uzoraka u bazi uzoraka temeljem vrijednosti čvornih parametara koje u sustav upisuje korisnik. Za pronađene pogodne uzorke računaju se težinski faktori parametara i sličnosti pogodnih uzoraka sa zadatkom. Pogodni uzorak sa najvećom sličnosti naziva se najbliži uzorak. Dio sustava koji obavlja zadaće rješavanja novog konstrukcijskog rješenja nije podržan predloženim sustavom. Rješavanje novog konstrukcijskog rješenja temeljem najbližijeg uzorka podrazumijeva: kopiranje modela i crteža najbližijeg uzorka iz baze gotovih

rješenja, njegova adaptacija u skladu sa novim konstrukcijskim zadatkom te pohranjivanje novog rješenja u bazu rješenja nakon verifikacije istog.

Prikupljanjem vrijednosti parametara novog, verificiranog konstrukcijskog rješenja dobiva se novi uzorak koji se pohranjuje u bazu uzoraka.

Sukladno shematskom prikazu (Slika 7-2) i dosadašnjim razmatranjima sustav za prepoznavanje uzoraka dijelimo na module (Slika 7-3) koji predstavljaju programsku cjelinu za rješavanje određene zadaće.



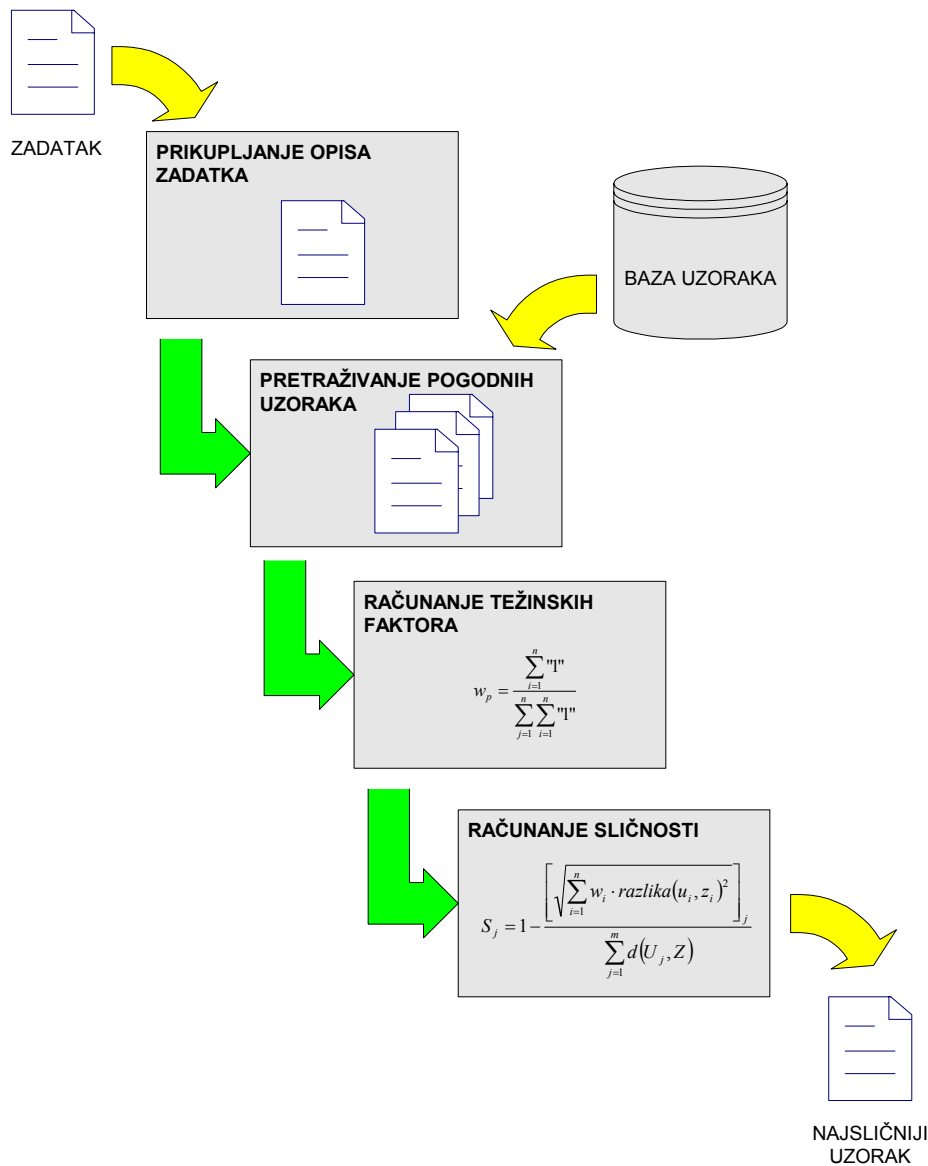
Slika 7-3: Struktura sustava za prepoznavanje uzoraka

- ◆ Osnovni modul – glavni modul i nositelj rada sustava, sadrži metodu `main()` kojom se pokreće sustav,
- ◆ Upravljanje uzorcima – modul unutar kojeg se izvršavaju akcije nad uzorcima pri procesu pronalaženja najbližeg uzorka i pohranjivanja novog uzorka
- ◆ Veza s bazom – modul služi za uspostavljanje veze s bazom, za čitanje podataka iz baze i zapisivanje podataka u bazu,
- ◆ Sučelje prema CAD sustavu – modul definira izgled korisničkih sučelja i sadrži akcije pridružene grafičkim elementima sučelja,
- ◆ Modul za adminstriranje – služi za prijavu korisnika u sustav i promjene u bazi uzoraka,
- ◆ Pomoć – modul sadrži stranice u HTML formatu sa uputama za korištenje sustava.

Opisi i prikazi definicija modula navedeni su u poglavlju 7.3.

7.1.1 Scenarij pronalaženja najslićnijeg uzorka

U skladu s razmatranjima u glavi 5 prikazan je dijagram tijeka pronalaženja najslićnijih uzoraka (Slika 7-4).



Slika 7-4: Tijek pronalaženja najslićnijih uzoraka

Za pronalaženje najslićnijeg uzorka potrebno je:

- ◆ prikupiti opis zadatka,
- ◆ prikazati korisničko sučelje na kojem korisnik odabire sklop za koji želi pronaći najslićnije uzorke,
- ◆ pronaći u bazi uzoraka parametre koji su pridruženi sklopu i ispisati ih na korisničkom sučelju,
- ◆ upisati vrijednosti čvornih parametara i parametara na temelju kojih će se računati slićnost,
- ◆ izračunati težinske faktore parametara čija je vrijednost upisana, a koji nisu čvorni parametri,

- ◆ pronaći pogodne uzorke u bazi uzoraka na temeljem vrijednosti čvornih parametara,
- ◆ izračunati sličnosti pogodnih uzoraka i prikazati korisniku rezultate.

Proces pronalaženja najbližeg uzorka započinje prikupljanjem vrijednosti parametara novog konstrukcijskog zadatka. Pretpostaviti ćemo da se vrijednosti parametara upisuju u sustav pomoću korisničkog sučelja (Slika 7-14). Na temelju odabira sklopa za koji se uzorak pohranjuje sustav pročita, iz baze uzoraka (poglavlje 7.2), strukturu sklopa i parametre koji su pridruženi sklopu te ih prikaže korisniku u korisničkom sučelju (Slika 7-14).

Slijedeći korak pri pronalaženju najbližeg uzorka je pretraživanje baze uzoraka temeljem vrijednosti čvornih parametara s ciljem dobivanja pogodnih uzoraka. Vrijednosti čvornih parametara upisuju se u sustav preko korisničkog sučelja (Slika 7-14).

Sustav iz baze uzoraka pročita međusobne prednosti parametara i na temelju njih izračuna težinske faktore parametara. Težinski faktori se računaju samo za parametre čija je vrijednost upisana u prvom koraku i koji nisu čvorni parametri.

Posljednji korak pri pronalaženju najbližeg uzorka je računanje sličnosti pogodnih uzoraka metodom najbliži susjed (metoda je opisana u poglavlju 3.4.1).

7.1.2 Scenarij pohranjivanja novog rješenja

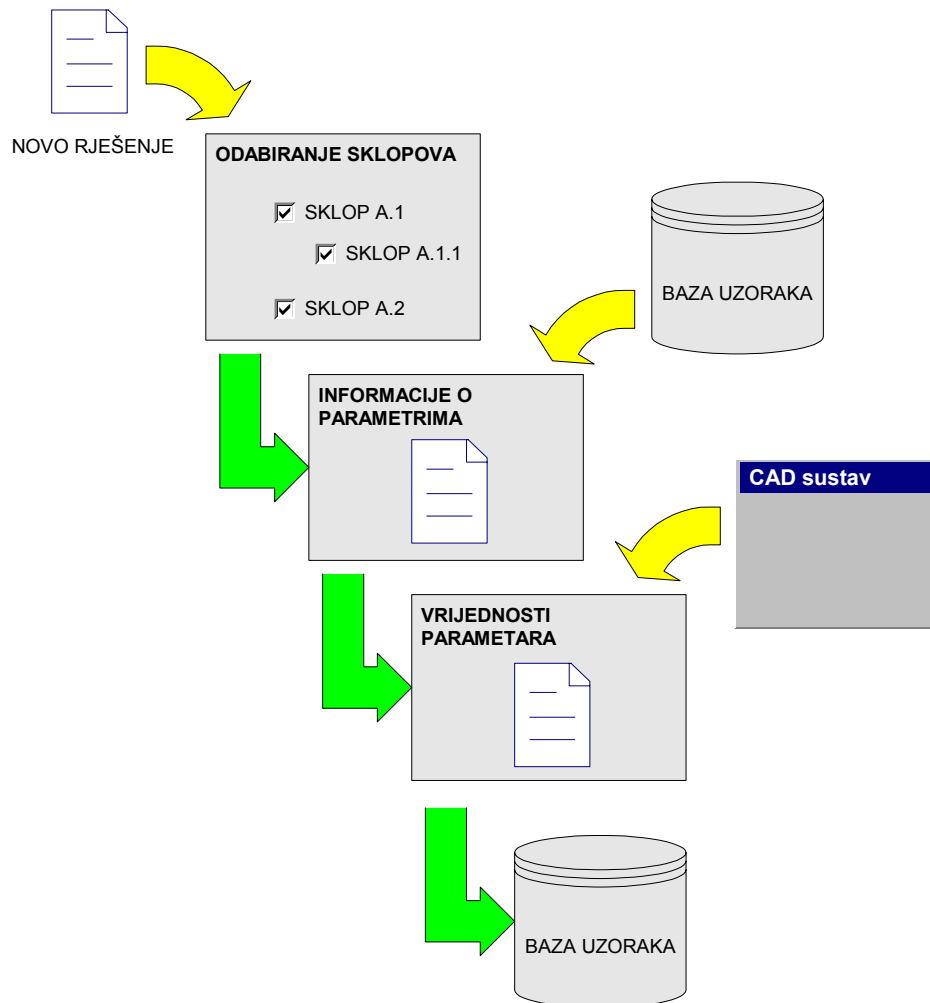
Temeljem razmatranja u glavi 6 pohranjivanje novog rješenja dijelimo u nekoliko koraka (Slika 7-5):

- ◆ odabiranje sklopova koji se žele pohraniti kao novi uzorci,
- ◆ prikupljanje informacija o parametrima,
- ◆ prikupljanje vrijednosti parametara,
- ◆ zapisivanje u bazu pročitanih vrijednosti parametara.

Tijek pohranjivanja novog konstrukcijskog rješenja započinje odabirom sklopa iz strukture proizvoda za koji se pohranjuje uzorak. Struktura proizvoda i svih sklopova koji ga čine zapisana je u bazi uzoraka. Iz prikazane strukture odabranog sklopa sa nazivom i opisom svim podsklopova korisnik odabire podsklopove za koje pohranjuje uzorke. Nakon odabira sklopova prikupljaju se iz baze uzoraka informacije o parametrima koji su pridruženi odabranim sklopovima.

Na primjer, ako konstrukcijsko rješenje predstavlja novo konstrukcijsko rješenje jezgre koje se pohranjuje u bazu uzoraka kao uzorak tada korisnik odabire sklop jezgre. Sustav prikuplja informacije o strukturi sklopa jezgre i parametrima koji su pridruženi sklopu jezgre i svim sklopovima u strukturi sklopa jezgre.

Temeljem informacija o parametrima prikupljaju se vrijednosti parametara iz parametarskih modela koji predstavlja novo konstrukcijsko rješenje.



Slika 7-5: Tijek pohranjivanja novog rješenja u bazu uzoraka

7.2 Organizacija baze

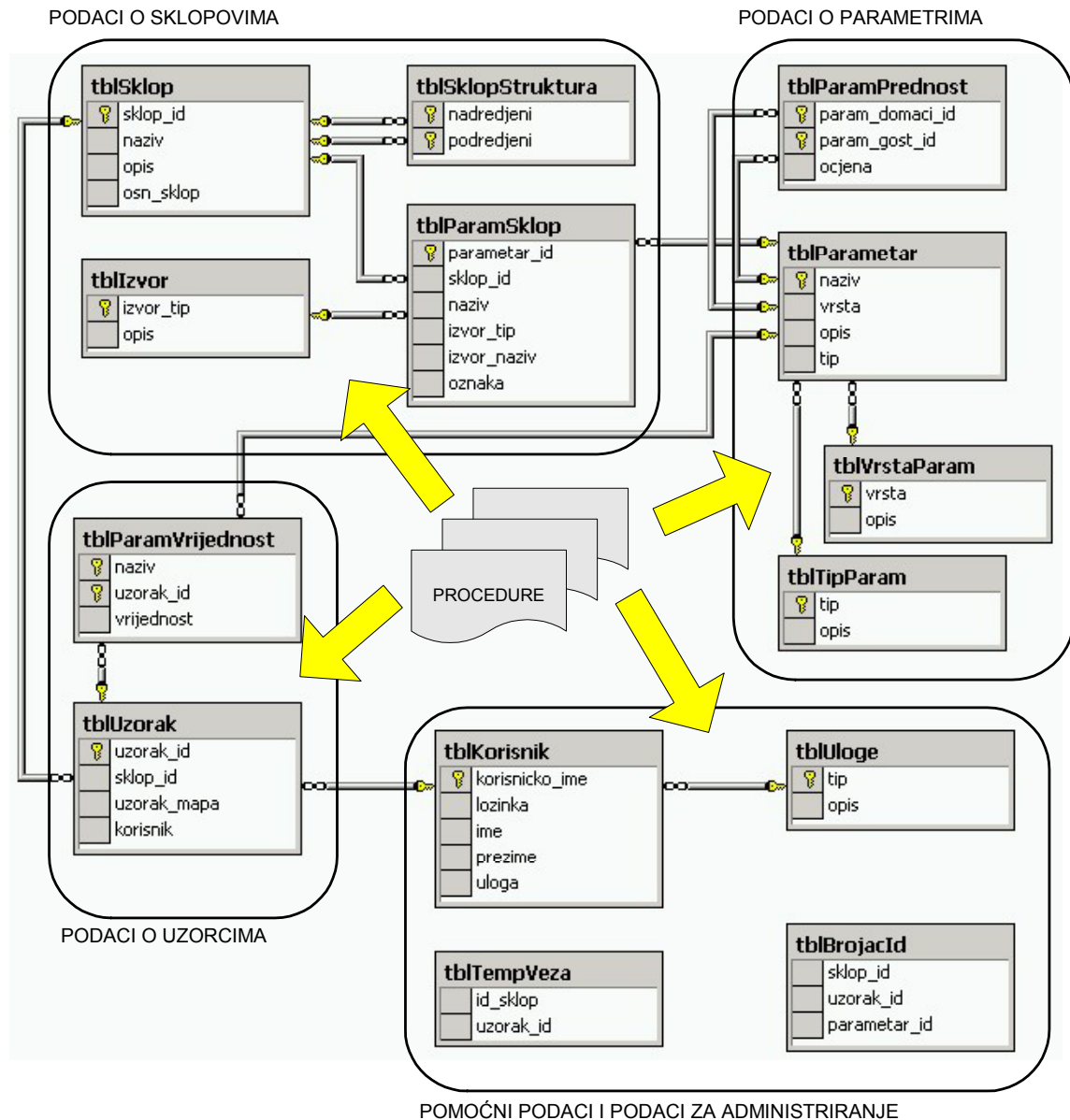
U skladu sa razmatranjima u glavi 4 i opisanog informacijskog modela u poglavlju 4.3 predlaže se organizacija baze uzoraka kao relacijske baze podataka. Strukturu baze čine tablice koje su međusobno povezane relacijama i procedure koje služe za upisivanje podataka u tablice (Slika 7-6). Entiteti baze uzoraka spomenuti u poglavlju 4.3 ovdje su prošireni u skladu sa pravilima razvoja relacijskih baza podataka [69].

Podaci o sklopovima sadrže: osnovne podatke o sklopovima proizvoda (tablica `tblSklop`), podatke o strukturi sklopova (tablica `tblSklopStruktura`) te podatke o parametrima koji su pridruženi pojedinim sklopovima (tablica `tblParamSklop`). U tablici `tblIzvor` sadržani su podaci o tipovima izvora prikupljanja vrijednosti parametara pridruženih sklopu (poglavlje 6.2).

Podaci o parametrima osim korištenih parametara zapisanih u tablici `tblParametar` sadrže i međusobne prednosti parametara u skladu s razmatranjima u poglavlju 5.3.

Podaci o uzorcima zapisani su u tablici `tblUzorak`, a vrijednosti parametara svakog uzorka zapisane su u tablici `tblParamVrijednost`.

Podaci za administriranje sadrže: podatke o korisnicima sustava (tablica `tblKorisnik`), njihova korisnička imena i lozinke s kojima se identificiraju prilikom korištenja sustava.



Slika 7-6: Prikaz strukture baze uzoraka i relacije između tablica

Tablice u bazi podataka koje opisuju podatke o sklopovima su:

- ◆ **tblSklop** – tablica opisuje sklopove proizvoda:
 - **sklop_id** – identifikacijska oznaka sklopa,
 - **naziv** – naziv sklopa,
 - **osn_sklop** – identifikator osnovnog sklopa; ako je sklop osnovni sklop tada je vrijednost 1 u protivnom 0,
 - **opis** – kratki opis sklopa;
- ◆ **tblSklopStruktura** – tablica u kojoj se definira položaj svakog sklopa u strukturi energetskog transformatora:
 - **nadredjeni** – identifikacijska oznaka nadređenog sklopa (roditelj),
 - **podredjeni** – identifikacijska oznaka podređenog sklopa (dijete);

- ◆ `tblParamSklop` – tablica sadrži podatke o svim parametrima koji su pridruženi nekom sklopu i koriste se kod pronalaženja najsličnijeg rješenja:
 - `parametar_id` – identifikacijska oznaka parametra,
 - `sklop_id` – identifikacijska oznaka sklopa kojem pripada parametar,
 - `naziv` – naziv parametra,
 - `vrsta` – vrsta parametra (čvorni ili obični),
 - `izvor_tip` – tip izvora iz kojeg će se prikupiti vrijednost parametra prilikom pohranjivanja novog uzorka,
 - `izvor_naziv` – naziv izvora, datoteke,
 - `oznaka` – oznaka parametra unutar izvora;
- ◆ `tblIzvor` – tablica definira izvore prikupljanja parametara;
 - `izvor_tip` – tip izvora vrijednosti parametra,
 - `opis` – kratki opis.

Podaci o parametrima sadrže tablice:

- ◆ `tblParametar` – tablica sadrži parametre:
 - `naziv` – naziv parametra,
 - `vrsta` – vrsta parametra,
 - `opis` – kratki opis parametra,
 - `tip` – tip parametra;
- ◆ `tblParamPrednost` – tablica sadrži međusobne odnose među parametrima:
 - `param_domaci_id` – identifikacijska oznaka parametra koji se uspoređuje,
 - `param_gost_id` – identifikacijska oznaka parametra s kojim se uspoređuje,
 - `ocjena` – ocjena prednosti;
- ◆ `tblVrstaParam` – tablica opisuje vrste parametara:
 - `vrsta` – vrsta parametra: CV – čvorni, OB – obični,
 - `opis` – kratki opis;
- ◆ `tblTipParam` – tablica opisuje tipove parametara:
 - `tip` – oznaka tipa parametra: num – numerički, sym – simbolički parametar,
 - `opis` – kratki opis.

Podaci o uzorcima sadrže tablice:

- ◆ `tblUzorak` – tablica opisuje uzorke:
 - `uzorak_id` – identifikacijska oznaka uzorka,
 - `sklop_id` – identifikacijska oznaka sklopa kojeg uzorak predstavlja,
 - `uzorak_mapa` – mapa (direktorij) gdje se nalazi trodimenzionalni model uzorka,
 - `korisnik` – korisničko ime korisnika koji je pohranio uzorak u bazu;
- ◆ `tblParamVrijednost` – tablica koja povezuje uzorak sa parametrima i čuva informaciju o vrijednosti parametara za svaki uzorak:
 - `parametar_id` – identifikacijska oznaka parametra,

- `uzorak_id` – identifikacijska oznaka uzorka,
- `vrijednost` – vrijednost parametra.

Pomoćni podaci i podaci za administriranje sadrže tablice:

- ◆ `tblKorisnik` – tablica sadrži osnovne podatke o korisnicima sustava:
 - `korisnicko_ime` – korisničko ime pod kojim se korisnik prijavljuje na sustav,
 - `lozinka` – lozinka korisnika za pristup sustavu,
 - `ime` – ime korisnika,
 - `prezime` – prezime korisnika,
 - `uloga` – uloga korisnika;
- ◆ `tblUloge` – tablica uloga koje se dodjeljuju korisnicima ovisno o prijavi u sustav:
 - `tip` – oznaka uloge,
 - `opis` – kratki opis uloge;
- ◆ `tblBrojacId` – tablica sadrži zadnju broječanu vrijednost identifikacijskih oznaka za sklop, uzorak i parametar i služi za automatsko odbrojanje istih:
 - `sklop_id` – zadnja vrijednost identifikacijske oznake sklopa,
 - `uzorak_id` – zadnja vrijednost identifikacijske oznake uzorka,
 - `parametar_id` – zadnja vrijednost identifikacijske oznake parametra;
- ◆ `tblTempVeza` – tablica služi za privremeno povezivanje sklopa i uzorka pri pohranjivanju uzoraka:
 - `id_sklop` – identifikacijska oznaka sklopa, redni broj sklopa koji se pohranjuje kao uzorak,
 - `uzorak_id` – identifikacijska oznaka uzorka koji pripada sklopu.

Pored opisanih tablica sa podacima bazu uzoraka čine i procedure (stored procedures) pomoću kojih se izvršavaju akcije nad podacima u tablicama. Priprema podataka (ulaznih varijabli u procedure) implementirana je u module sustava. Procedure se pozivaju iz modula koji ostvaruje vezu s bazom (Slika 7-3).

Procedure za dodavanje novog sklopa u bazu uzoraka su:

- ◆ `spDodavanjeSklopa` – procedura služi za dodavanje podataka u tablicu `tblSklop`; ulazne varijable u proceduru su:
 - `@SklopId` – identifikacijska oznaka parametra,
 - `@Naziv` – naziv sklopa,
 - `@Opis` – opis sklopa,
 - `@OsnSklop` – identifikator osnovnog sklopa;
- ◆ `spDodavanjeSklopStruktura` – procedura služi za dodavanje podataka u tablicu `tblSklopStruktura`; ulazne varijable u proceduru su:
 - `@NadredjeniId` – identifikacijska oznaka nadređenog sklopa,
 - `@PodredjeniId` – identifikacijska oznaka podređenog sklopa.

Procedure za dodavanje novog parametra u bazu uzoraka su:

- ◆ `spDodavanjeParamParametra` – procedura služi za dodavanje podataka u tablicu `tblParametar`; ulazne varijable u proceduru su:
 - `@ParametarId` – identifikacijska oznaka parametra,
 - `@Naziv` – naziv parametra,
 - `@Vrsta` – vrsta parametra,
 - `@Opis` – opis parametra,
 - `@Tip` – tip parametra;
- ◆ `spDodavanjePrednosti` – procedura služi za dodavanje podataka u tablicu `tblParamPrednost`; ulazne varijable u proceduru su:
 - `@DomaciId` – identifikacijska oznaka parametra koji se uspoređuje,
 - `@GostId` – identifikacijska oznaka parametra s kojim se uspoređuje,
 - `@Ocjena` – ocjena prednosti.

Procedure za dodavanje novog uzorka u bazu uzoraka su:

- ◆ `spDodavanjeUzorka` – procedura služi za dodavanje novog uzorka u bazu; ulazne varijable u proceduru su:
 - `@SklopID` – identifikacijska oznaka sklopa za koji se zapisuje uzorak,
 - `@UzorakMapa` – putanja do modela uzorka,
 - `@Korisnik` – korisničko ime korisnika koji je pohranio uzorak,
 - `@IDSklop` – odnosi se na `id_sklop` u tablici `tblTempVeza` a predstavlja redni broj sklopa koji se pohranjuje u `tblUzorak` i ima privremeni karakter;
- ◆ `spDodavanjeParamVrijednost` – procedura služi za dodavanje podataka u tablicu `tblParamVrijednost`; ulazne varijable u proceduru su:
 - `@ParametarId` – identifikacijska oznaka parametra,
 - `@UzorakId` – identifikacijska oznaka uzorka,
 - `@Vrijednost` – vrijednost parametra u uzorku.

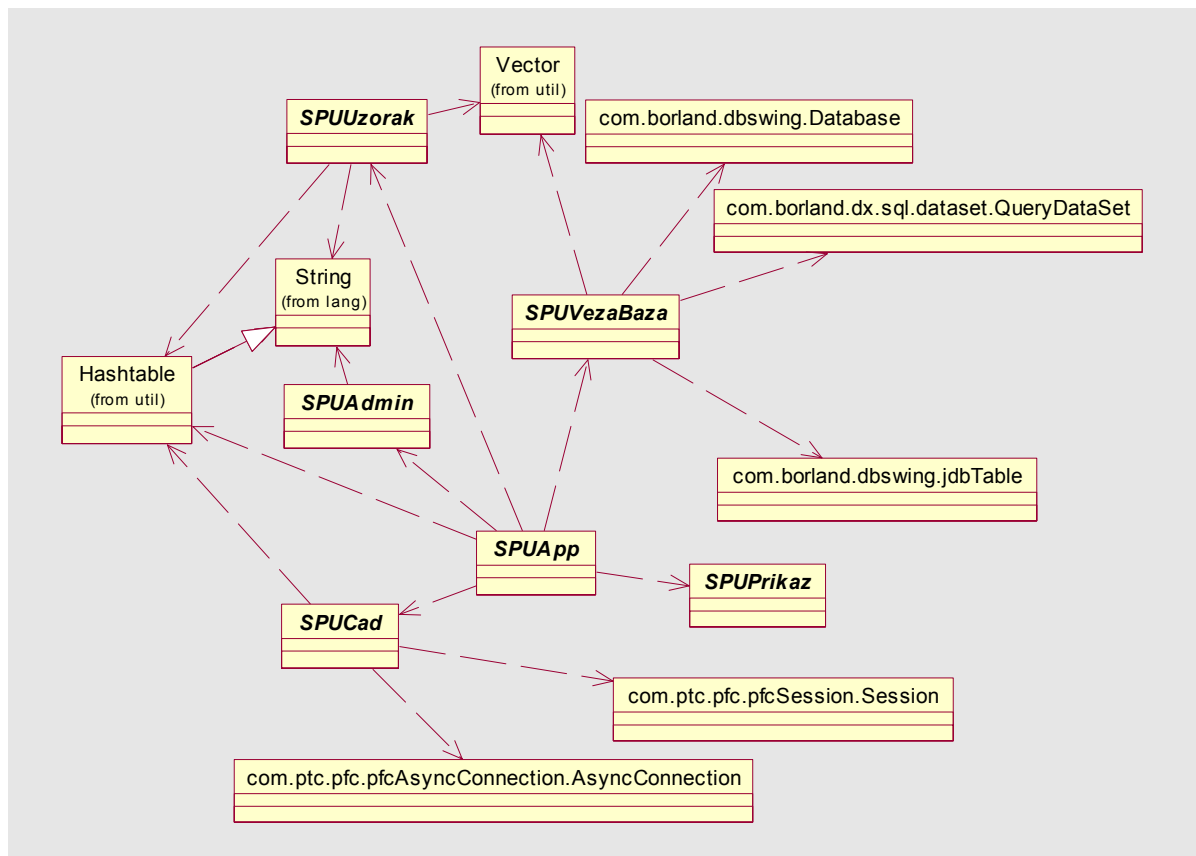
Procedure za brisanje iz baze uzoraka su:

- ◆ `spBrisanjeUzorka` – procedura služi za brisanje uzorka iz baze; ulazna varijabla u proceduru je:
 - `@UzorakId` – identifikacijska oznaka uzorka koji se briše;
- ◆ `spBrisanjeParametra` – procedura služi za brisanje parametra iz baze uzoraka; ulazna varijabla u proceduru je:
 - `@ParametarId` – identifikacijska oznaka parametra koji se briše;
- ◆ `spBrisanjeSklopa` – procedura služi za brisanje sklopa iz baze uzoraka; ulazna varijabla u proceduru je:
 - `@SklopId` – identifikacijska oznaka sklopa koji se briše;
- ◆ `spBrisanjeTblTemp` – procedura služi za brisanje podataka iz privremene tablice `tblTempVeza`.

7.3 Moduli sustava

Moduli sustava relizirani su u skladu sa objektno orijentiranom paradigmom programiranja [65], [70], [66]. Opisani su klasama objekata koji sadrže atribute i metode. Atributi predstavljaju podatke odnosno varijable kojima su opisana svojstva objekata određene klase dok metode predstavljaju ponašanje objekata koji pripadaju klasi. Metodama se definiraju sve akcije koje se vrše nad objektima pripadajuće klase.

Slika 7-7 prikazuje osnovne klase koje definiraju module korištene u sustavu (podebljano). Atributi korišteni u osnovnim klasama instanciraju se iz ostalih prikazanih klasa.



Slika 7-7: Klase koje opisuju module sustava i njihove veze

7.3.1 Osnovni modul

Osnovni modul sustava je središnji dio sustava u kome se nalazi metoda `main()` kojom se pokreće sustav. Modul je izveden kao samostalna aplikacija koja se pokreće izvan CAD sustava i baze uzoraka. Glavne zadaće osnovnog modula su:

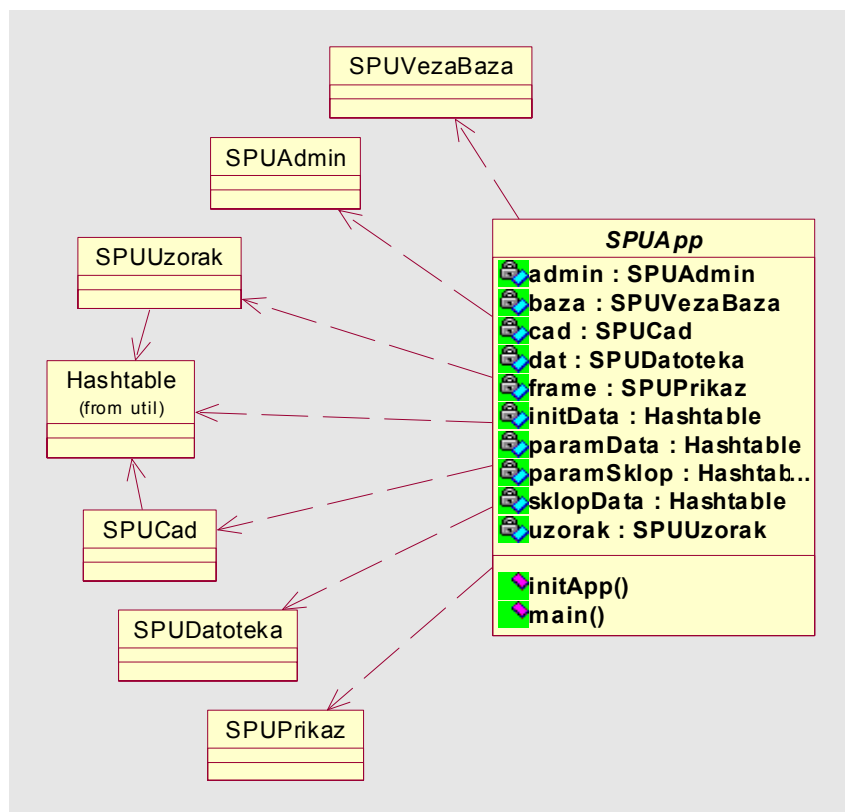
- ◆ upravljanje sa cijelim sustavom,
- ◆ inicijaliziranje početnih postavki sustava,
- ◆ ostvarivanje veze među modulima.

Radnje koje se izvršavaju unutar Osnovnog modula su:

- ◆ instanciranje objekta `frame` tipa **SPUPrikaz**,
- ◆ instanciranje objekta `baza` tipa **SPUVezaBaza**.

- ◆ instanciranje objekta `adminData` tipa ***SPUAdmin***,
- ◆ instanciranje objekta `uzorak` tipa ***SPUUzorak***,
- ◆ instanciranje objekta `initData` tipa ***Hashtable***,
- ◆ instanciranje objekta `paramData` tipa ***Hashtable***,
- ◆ instanciranje objekta `sklopData` tipa ***Hashtable***,
- ◆ inicijalizacija radnog direktorija,
- ◆ inicijalizacija naziva baze,
- ◆ inicijalizacija komande kojom se pokreće CAD sustav,
- ◆ inicijalizacija relativne putanje za svaki osnovni sklop.

Osnovni modul opisan je klasom ***SPUApp*** (Slika 7-8).



Slika 7-8: Prikaz definicije klase ***SPUApp***

Klasa ***SPUApp*** sadrži atribute:

- ◆ `admin` – atribut tipa ***SPUAdmin*** koji predstavlja vezu sa modulom za adinistriranje,
- ◆ `baza` – atribut tipa ***SPUVezaBaza*** koji predstavlja vezu sa modulom za pristup bazi uzoraka,
- ◆ `cad` – atribut tipa ***SPUCad*** koji predstavlja vezu sa modulom za pristup CAD sustavu,
- ◆ `dat` – atribut tipa ***SPUDatoteka*** koji predstavlja vezu sa modulom za upravljanje datotekama,
- ◆ `frame` – atribut tipa ***SPUPrikaz*** koji predstavlja vezu sa korisničkim sučeljima,
- ◆ `initData` – atribut tipa ***Hashtable*** koji sadrži podatke iz inicijalne datoteke,

- ◆ `paramData` – atribut tipa **Hashtable** koji sadrži opise parametara,
- ◆ `paramSklop` – atribut tipa **Hashtable** koji sadrži vezu među parametrima i sklopovima,
- ◆ `sklopData` – atribut tipa **Hashtable** koji sadrži opise sklopova,
- ◆ `radni` – atribut tipa **String** koji sadrži naziv radnog direktorija,
- ◆ `uzorak` – atribut tipa **SPUUzorak** koji predstavlja vezu sa modulom za upravljanje uzorcima,

Metode definirane u klasi **SPUApp** su:

- ◆ `initApp()` – metoda služi za početnu inicijalizaciju objekata,
- ◆ `main()` – glavna metoda kojom se pokreće aplikacija.

7.3.2 Modul za vezu sa bazom uzoraka

Modul za vezu sa bazom uzoraka ostvaruje vezu sa bazom i služi za razmjenu podataka s bazom uzoraka. Veza sustava sa bazom podataka ostvaruje se preko upravljačkog programa JDBC-ODBC mosta preko atributa `CBRdatabase` klase **Database**²⁰.

Atributi instancirani iz klase **QueryDataSet**²¹ sadrže SQL upite koji se prosljeđuju bazi. Rezultati upita zapisuju se u attribute instancirane iz klase **jdbTable**²².

Zadaje modula za vezu sa bazom uzoraka su:

- ◆ uspostavljanje veze s bazom,
- ◆ slanje upita bazi,
- ◆ primanje i obrada primljenih podataka,
- ◆ upisivanje vrijednosti parametara u bazu,
- ◆ pokretanje procedura.

Modul za vezu sa bazom uzoraka definiran je klasom **SPUVezaBaza** koja sadrži metode i attribute za povezivanje s bazom uzoraka i prikupljanje i obradu podataka iz baze.

Atributi definirani u klasi koji sadrže SQL upite:

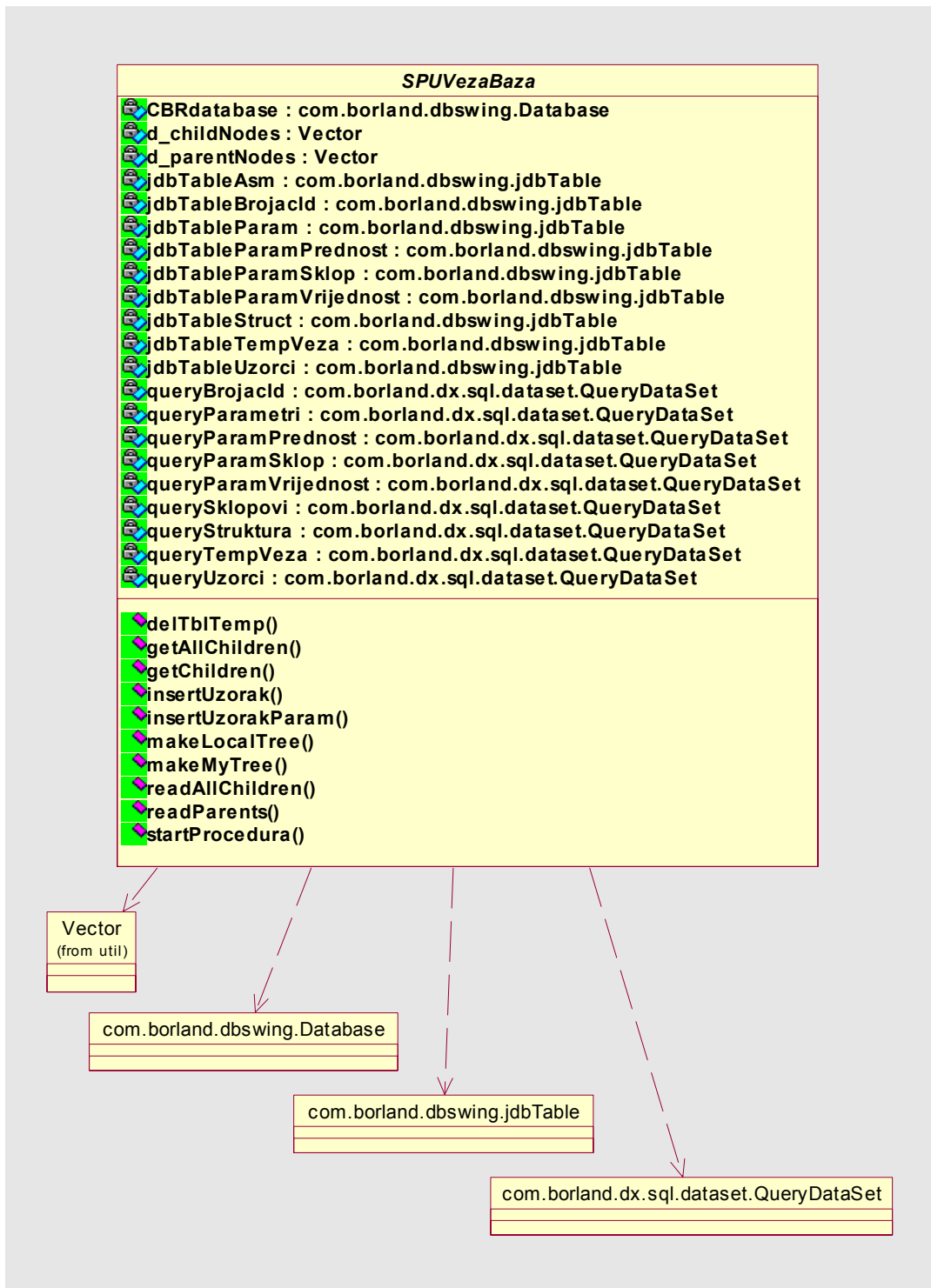
- ◆ `queryBrojacId` – atribut tipa **QueryDataSet** koji sadrži zadnju vrijednost brojača identifikacijskih oznaka,
- ◆ `queryParametri` – atribut tipa **QueryDataSet** koji sadrži podatke o parametrima,
- ◆ `queryParamPrednost` – atribut tipa **QueryDataSet** koji sadrži podatke o prednostima između parametara za računanje težinskih faktora,
- ◆ `queryParamSklop` – atribut tipa **QueryDataSet** koji sadrži podatke o vezama između parametara i sklopova,
- ◆ `queryParamVrijednost` – atribut tipa **QueryDataSet** koji sadrži podatke o vrijednostima parametara u uzorcima,
- ◆ `querySklopovi` – atribut tipa **QueryDataSet** koji sadrži podatke o sklopovima,
- ◆ `queryStruktura` – atribut tipa **QueryDataSet** koji sadrži podatke o vezama između sklopova,

²⁰ Zbog kraćeg pisanja nije ispisan puni naziv klase. Puni naziv vidljiv je u prikazu definicije klase Slika 7-9

²¹ Vidi 20

²² Vidi 20

- ◆ queryTempVeza – atribut tipa **QueryDataSet** koji sadrži podatke iz privremene tablice tblTempVeza,
- ◆ queryUzorci – atribut tipa **QueryDataSet** koji sadrži podatke o uzorcima,



Slika 7-9: Prikaz definicije klase SPUVezaBaza

Atributi definirani u klasi koji sadrže podatke dobivene SQL upitima:

- ◆ `jdbTableAsm` – atribut tipa ***jdbTable*** koji sadrži podatke dobivene SQL upitom opisanim atributom `querySklopovi`,
- ◆ `jdbTableBrojacId` – atribut tipa ***jdbTable*** koji sadrži podatke dobivene SQL upitom opisanim atributom `queryBrojacId`,
- ◆ `jdbTableParam` – atribut tipa ***jdbTable*** koji sadrži podatke dobivene SQL upitom opisanim atributom `queryParametri`,
- ◆ `jdbTableParamPrednost` – atribut tipa ***jdbTable*** koji sadrži podatke dobivene SQL upitom opisanim atributom `queryParamPrednost`,
- ◆ `jdbTableParamSklop` – atribut tipa ***jdbTable*** koji sadrži podatke dobivene SQL upitom opisanim atributom `queryParamSklop`,
- ◆ `jdbTableParamVrijednost` – atribut tipa ***jdbTable*** koji sadrži podatke dobivene SQL upitom opisanim atributom `queryParamVrijednost`,
- ◆ `jdbTableStruct` – atribut tipa ***jdbTable*** koji sadrži podatke SQL upitom opisanim atributom `queryStruktura`,
- ◆ `jdbTableTempVeza` – atribut tipa ***jdbTable*** koji sadrži podatke dobivene SQL upitom opisanim atributom `queryTempVeza`,
- ◆ `jdbTableUzorci` – atribut tipa ***jdbTable*** koji sadrži podatke SQL upitom opisanim atributom `queryUzorci`.

Ostali atributi definirani u klasi:

- ◆ `CBRdatabase` – atribut tipa ***Database*** kojim se uspostavlja veza sa bazom uzoraka,
- ◆ `d_childNodes` – atribut tipa ***Vector*** koji sadrži identifikacijske oznake podređenih sklopova,
- ◆ `d_parentNodes` – atribut tipa ***Vector*** koji sadrži identifikacijske oznake nadređenih sklopova.

Metode definirane u klasi:

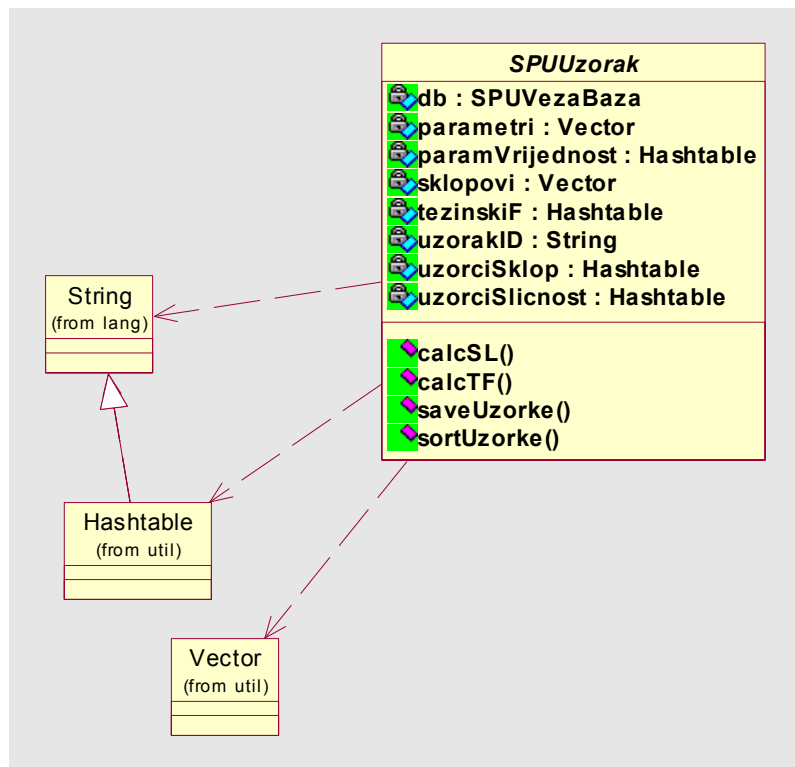
- ◆ `delTblTemp()` – metoda poziva proceduru za brisanje privremene tablice,
- ◆ `getAllChildren()`, `readAllChildren()`, `getChildren()`, `readParents()` – metode služe za kreiranje strukture sklopova,
- ◆ `insertUzorak()` – metoda služi za pripremu podataka i poziva proceduru za ubacivanje uzorka u bazu,
- ◆ `insertUzorakParam()` – metoda služi za pripremu podataka i poziva proceduru za ubacivanje vrijednosti parametra za uzorak koji se pohranjuje,
- ◆ `makeMyTree()` i `makeLocalTree()` – metode služe za predočavanje strukture sklopova u obliku hijerarhijskog stabla,
- ◆ `startProcedura()` – metoda kojom se pokreću procedure definirane u bazi.

7.3.3 Modul za upravljanje uzorcima

Modul za upravljanje uzorcima izvršava akcije potrebne za pronalaženje najbližijeg uzorka i pohranjivanje novog uzorka u bazu koje se tiču samih uzoraka:

- ◆ pretraživanje pogodnih uzoraka na osnovu vrijednosti čvornih parametara,
- ◆ računanje težinskih faktora,
- ◆ računanje sličnosti pogodnih uzoraka,

- ◆ sortiranje uzoraka prema sličnosti.



Slika 7-10: Prikaz definicije klase SPUUzorak

Modul za upravljanje uzorcima definiran je klasom **SPUUzorak** (Slika 7-10). Klasa sadrži atribute:

- ◆ db – atribut tipa **SPUVezaBaza** izjednačen je sa atributom baza iz klase **SPUApp** i preko njega pristupa bazi uzoraka,
- ◆ parametri – atribut tipa **Vector** sadrži parametre pridružene uzorku,
- ◆ paramVrijednost – atribut tipa **Hashtable** sadrži vrijednosti parametara,
- ◆ sklopovi – atribut tipa **Vector** identifikacijske oznake odabranih sklopova za koje se pohranjuju uzorci,
- ◆ tezinskiF – atribut tipa **Hashtable** sadrži izračunate vrijednosti težinskih faktora parametara,
- ◆ uzorakID – atribut tipa **String** sadrži identifikacijsku oznaku uzorka,
- ◆ uzorciSklop – atribut tipa **Hashtable** je pomoćni atribut i koristi se pohranjivanje uzoraka,
- ◆ uzorciSlicnost – atribut tipa **Hashtable** sadrži izračunate vrijednosti sličnosti uzoraka.

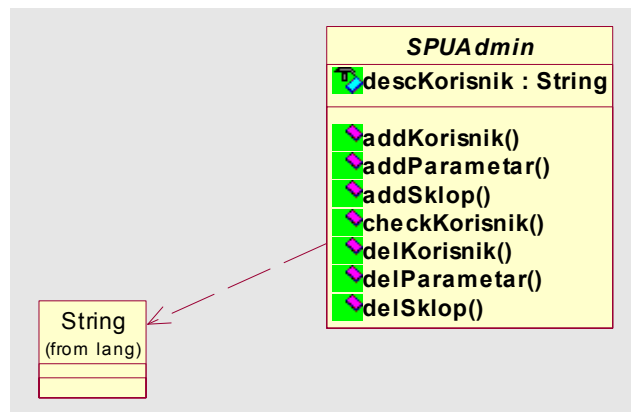
Metode definirane u klasi:

- ◆ calcSL() – metoda računa sličnost između zadatka i pogodnih uzoraka,
- ◆ calcTF() – metoda računa težinske faktore parametara,
- ◆ saveUzorke() – metoda služi za pohranjivanje novih uzoraka u bazu uzoraka,
- ◆ sortUzorke() – metoda služi za pohranjivanje novih uzoraka u bazu uzoraka.

7.3.4 Modul za administriranje

Modul za administriranje provjerava identitet korisnika prilikom pokretanja sustava za prepoznavanje uzoraka. Podaci o korisniku pohranjuju se prilikom inicijalizacije sustava u atribut `descKorisnik`. Modul služi i za ažuriranje baze uzoraka što uključuje: dodavanje/brisanje korisnika, dodavanje/brisanje parametara, dodavanje/brisanje sklopova.

Modul za administriranje definiran je klasom **SPUAdmin**.



Slika 7-11: Prikaz definicije klase SPUAdmin

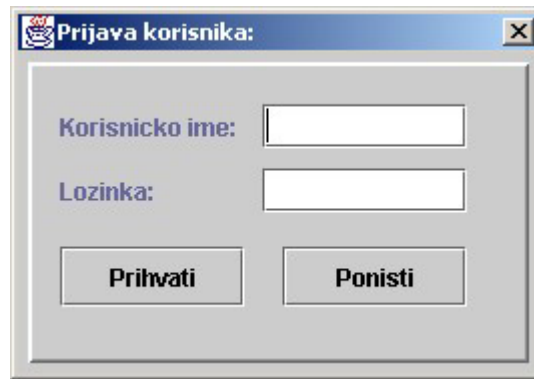
Atribut definiran u klasi:

- ◆ `descKorisnik` – atribut tipa **String** koji sadrži opis korisnika; korisničko ime, lozinku, ime, prezime i ulogu.

Metode definirane u klasi:

- ◆ `addKorisnik()` – metoda služi za dodavanje novog korisnika u bazu uzoraka,
- ◆ `addParametar()` – metoda služi za dodavanje novog parametra u bazu uzoraka,
- ◆ `addSklop()` – metoda služi za dodavanje novog sklopa u bazu uzoraka,
- ◆ `checkKorisnik()` – metoda provjerava identitet korisnika i u skladu s njim dodjeljuje mu ulogu,
- ◆ `delKorisnik` – metoda služi za brisanje korisnika iz baze uzoraka,
- ◆ `delParametar` – metoda služi za brisanje parametra iz baze uzoraka,
- ◆ `delSklop` – metoda služi za brisanje sklopa iz baze uzoraka.

Dio modula za administriranje je i dijalog za prijavu na sustav Slika 7-12. Dijalog za prijavu pokreće se odmah pri pokretanju sustava (metoda `main()`). U prazna polja dijaloga korisnik upisuje svoje korisničko ime i lozinku kojom prijavljuje rad u sustavu. Odabirom dugmeta `Prihvati` sustav provjerava upisane vrijednosti i ukoliko odgovaraju vrijednostima u bazi uzoraka pojavljuje se korisničko sučelje sustava (Slika 7-14) u protivnom se prekida izvođenje aplikacije.



Slika 7-12: Dijalog za prijavu korisnika

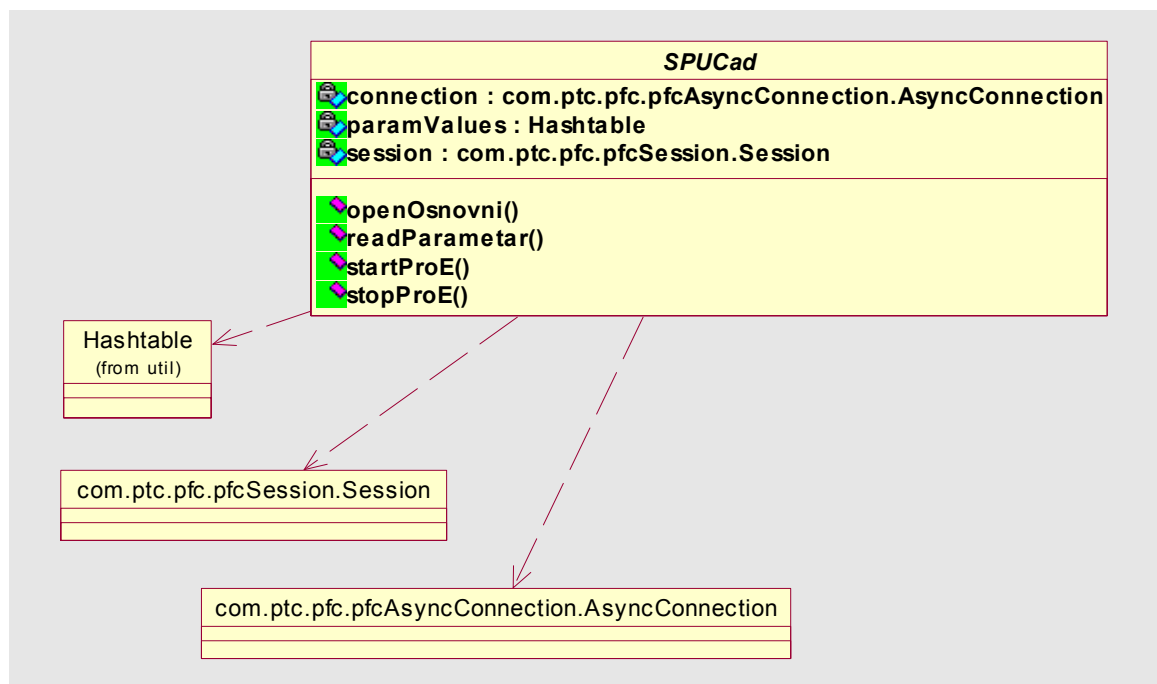
7.3.5 Sučelje prema CAD sustavu

Modul ostvaruje vezu sa CAD sustavom u tzv. asinkronom načinu rada pomoću Java[®] biblioteka kojima se pristupa CAD sustavu ProENGINEER[®] (J-Link[®] modul) [68]. Veza se ostvaruje pokretanjem CAD sustava bez grafičkog sučelja, kao zasebnog procesa operativnog sustava ili spajanjem na već pokrenuti CAD sustav. Pokretanje CAD sustava odvija se iz sustava za prepoznavanje uzoraka prema potrebi, odnosno samo onda kada se prikupljaju vrijednosti parametara modela.

Osnovne aktivnosti aktivnosti podržane u modulu su:

- ◆ pokretanje CAD sustava,
- ◆ otvaranje modela,
- ◆ prikupljanje i obrada podataka iz modela,
- ◆ zatvaranje modela,
- ◆ zaustavljanje CAD sustava.

Sučelje prema CAD sustavu definirano je klasom **SPUCad**.



Slika 7-13: Prikaz definicije klase SPUCad

Atributi definirani u klasi:

- ◆ `connection` – atribut tipa ***AsyncConnection***²³ koji predstavlja vezu prema CAD alatu (Pro/ENGINEER®),
- ◆ `paramValues` – atribut tipa ***Hashtable*** koji sadrži vrijednosti parametara,
- ◆ `session` – atribut tipa ***Session***²⁴ koji predstavlja vezu na sesiju CAD alata.

Metode definirane u klasi:

- ◆ `openOsnovni()` – metoda otvara osnovni sklop,
- ◆ `readParametar()` – metoda vraća vrijednost parametra opisanog u ulaznom atributu `desc` tipa ***Vector***,
- ◆ `startProE()` i `stopProE()` – služe za pokretanje odnosno zaustavljanje Pro/ENGINEER aplikacije.

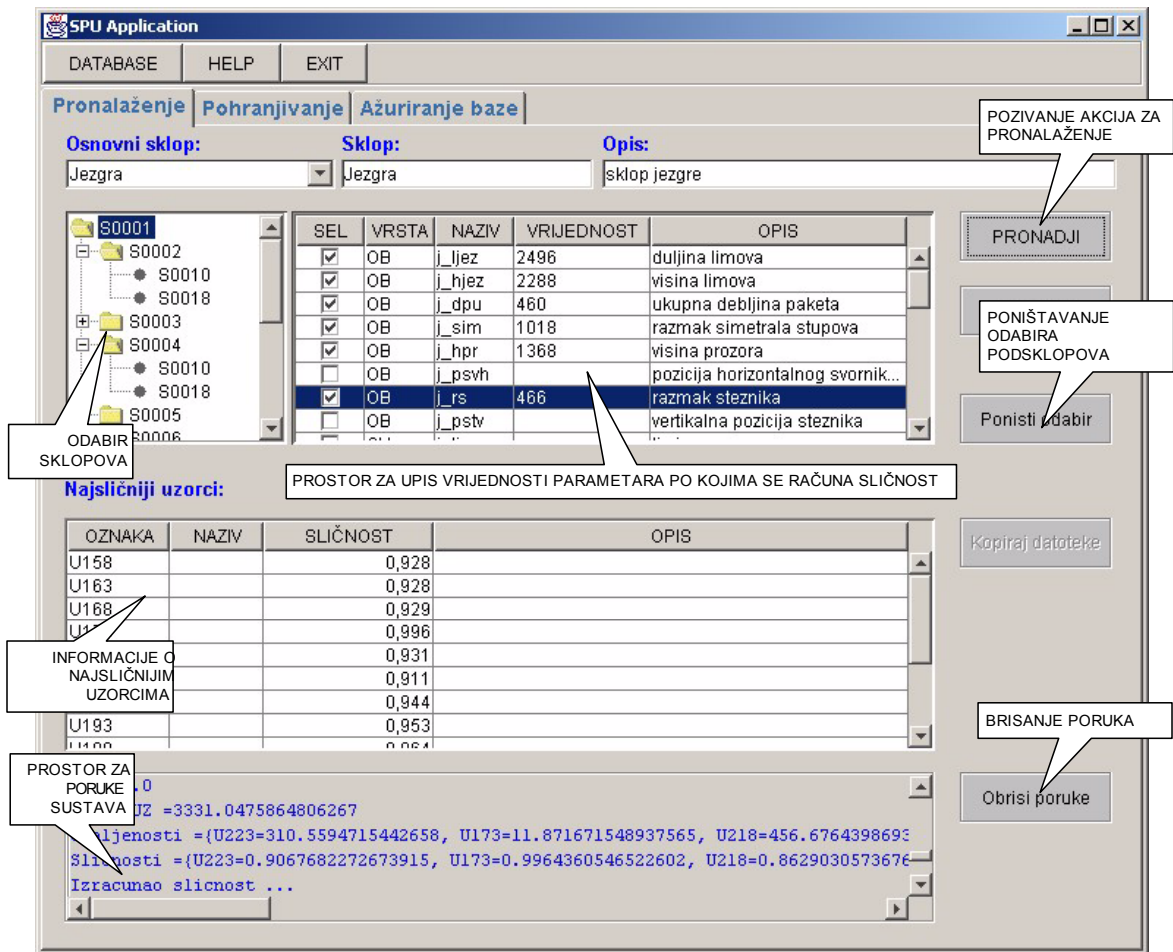
7.3.6 Korisnička sučelja

Komunikacija između korisnika i sustava za prepoznavanje uzoraka obavlja se preko korisničkih sučelja. Korisnička sučelja su grafička, kreirana u skladu sa Java AWT® preporukama i dijelovima Java Swing® biblioteka [67].

Korisnička sučelja definirana su klasom ***SPUPrikaz***. Klasa sadrži attribute koji definiraju korisnička sučelja i metode kojima se pokreću akcije nad grafičkim elementima sučelja. Zbog jednostavnosti nisu navedeni atributi i metode koji definiraju korisnička sučelja već su prikazana sučelja (Slika 7-14, Slika 7-15, Slika 7-16).

²³ Zbog kraćeg pisanja nije ispisan puni naziv klase. Puni naziv vidljiv je u prikazu definicije klase Slika 7-13

²⁴ Vidi 23

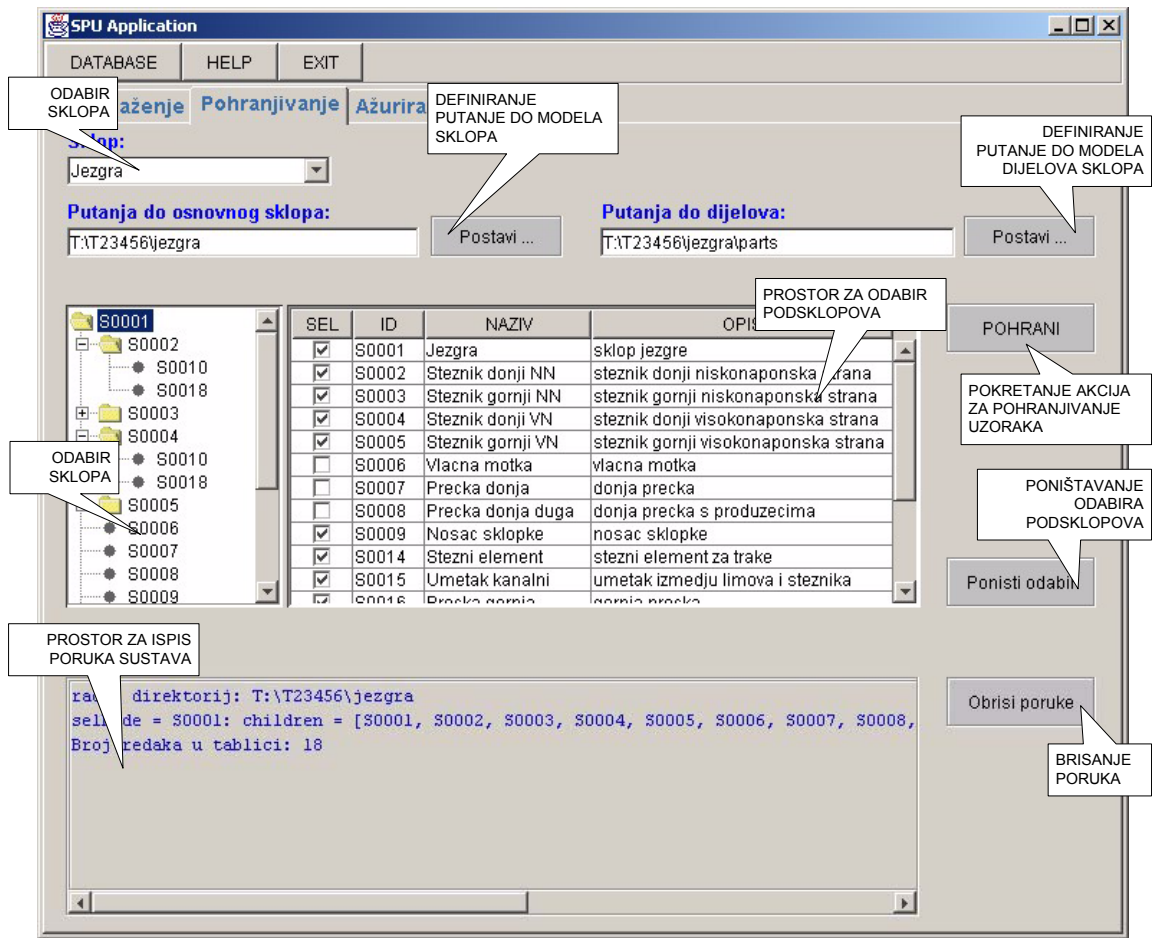


Slika 7-14: Prikaz korisničkog sučelja za pronalaženje uzoraka

Slika 7-14 prikazuje korisničko sučelje faze pronalaženja u sustavu za prepoznavanje uzoraka pri konstruiranju sklopova energetskog transformatora. Iz padajućeg izbornika Osnovni sklop ili iz strukture transformatora odabran je sklop jezgre za koji je iz baze uzoraka pročitana struktura i pripadajući parametri. U prostoru za upis vrijednosti parametara nalazi se lista parametara pridruženih jezgri. U stupac VRIJEDNOST korisnik upisuje vrijednosti parametara na temelju kojih želi pronaći najsličniji uzorak.

Odabirom dugmeta PRONADJI pokreću se akcije za računanje težinskih faktora i sličnosti. Najsličniji uzorci sa izračunatim vrijednostima sličnosti ispisani su u označenom prostoru za ispis.

Dugmad Datoteka i Kopiraj datoteke nisu dostupna i predviđena su za implementaciju čitanja vrijednosti parametara iz tekstualne datoteke i manipulaciju datotekama trodimenzionalnih parametarskih modela i crteža. Navedene akcije nisu obuhvaćene predloženim sustavom.



Slika 7-15: Prikaz korisničkog sučelja za pohranjivanje uzoraka u bazu

Slika 7-15 prikazuje korisničko sučelje za pohranjivanje uzoraka u bazu. Na prikazanom primjeru odabran je sklop jezgre za koji će se pohraniti uzorci. U prostoru za odabir sklopova ispisani su svi podsklopovi koji u skladu sa fizičkom strukturom zapisanom u bazi pripadaju sklopu jezgre. Korisnik može isključiti sklopove za koje ne želi pohraniti uzorke. U dijelu sučelja gdje se definira putanja do modela odabire se mjesto na računalnoj mreži ili na lokalnom računalu gdje su pohranjeni model sklopa jezgre i pripadajući podsklopovi.

Odabirom dugmeta POHRANI sustav čita vrijednosti parametara pridruženih odabranim sklopovima iz trodimenzionalnim parametarskih modela i podatke zapisuje u bazu uzoraka.

Slika 7-16: Prikaz korisničkog sučelja za administriranje baze

Sučelje za administriranje baze (Slika 7-16) služi za dodavanje novih sklopova u fizičku strukturu proizvoda i novih parametara koji opisuju sklopove. Korisnik odabire dodavanje/brisanje sklopova/parametara. Primjer prikazuje dodavanje novog sklopa imenom *Novi sklop* i novog parametra nazivom *j_novi_parametar*. Nakon upisa vrijednosti potrebno je odabrati dugme *Potvrdi*.

Odabirom dugmeta *DATABASE* prikazuje se stanje dijela tablica u bazi uzoraka Slika 7-17. Dugme *HELP* prikazuje pomoć korisniku pri korištenju aplikacije, a dugme *EXIT* završava izvođenje aplikacije.

cbrbase

tblSklop			tblSklopStruktura		tblUzorak		
	sklop_id	naziv					
1	S0001	Jezgra	1	S0001	S0002	1	U158
2	S0002	Steznik donji NN	2	S0001	S0003	2	U159
3	S0003	Steznik gornji NN	3	S0001	S0004	3	U160
4	S0004	Steznik donji VN	4	S0001	S0005	4	U161
5	S0005	Steznik gornji VN	5	S0001	S0006	5	U162
6	S0006	Vlacna motka	6	S0001	S0007	6	U163
7	S0007	Precka donja	7	S0001	S0008	7	U164
8	S0008	Precka donja duga	8	S0001	S0009	8	U165
9	S0009	Nosac sklonke	9	S0001	S0014	9	U166

tblParametar				tblParamSklop			
	naziv	vrsta					
1	j_brt	OB	broj bandaznih tr	1	P0001	S0001	j_ljez
2	j_d11p	OB	sirina izolacijske	2	P0002	S0001	j_hjez
3	j_d12p	OB	duljina izolacijske	3	P0003	S0001	j_dpu
4	j_dfe	OB	promjer jezgre	4	P0004	S0001	j_sim
5	j_dim_A	OB	sirina nosaca svc	5	P0005	S0001	j_hpr
6	j_dn1	OB	debljina nosaca s	6	P0006	S0001	j_psvh
7	i_dp1	OB	sirina vodilice	7	P0007	S0002	j_fipsvh

Zatvori Refresh

Slika 7-17: Prikaz korisničkog sučelja za pregled tablica u bazi uzoraka

8 PRIKAZ PRIMJENE SUSTAVA

Za prikaz primjene predloženog sustava za prepoznavanje uzoraka korištena su realna konstrukcijska rješenja sklopa jezgre energetskog transformatora. U razmatranje je uzeto 15 prijašnjih konstrukcijskih rješenja sklopa jezgre. Jedno od rješenja proglašeno je novim konstrukcijskim zadatkom, a ostala rješenja su pohranjena u bazu uzoraka pomoću realiziranog sustava za prepoznavanje uzoraka.

U bazu uzoraka upisani su sklopovi jezgre energetskog transformatora i njihova struktura. Upisani su parametri koji opisuju sklopove i pridruženi su sklopovima (Slika 8-1).

3>Data in Table 'tblParamSklop' in 'cbrbase' on '01313'

parametar_id	sklop_id	naziv	izvor_tip	izvor_naziv	oznaka
P0001	S0001	l_hjez	prt	jezgra30_sklop	d14
P0002	S0001	l_hjez	prt	jezgra30_sklop	d15
P0003	S0001	l_dpou	prt	jezgra30_sklop	d14
P0004	S0001	l_sim	prt	limovi	d12
P0005	S0001	l_hpr	prt	limovi	d24
P0006	S0001	l_psvh	prt	limovi	d215
P0023	S0001	l_rs	prt	jezgra30_sklop	d19
P0024	S0001	l_pstv	prt	jezgra30_sklop	d17
P0025	S0001	l_tip	asm	jezgra30	tip
P0048	S0001	l_ptuhl			
P0065	S0001	l_vpp			
P0066	S0001	l_dnl			
P0067	S0001	l_vml			
P0068	S0001	l_dp1			
P0069	S0001	l_vpz			
P0070	S0001	l_lpz			
P0071	S0001	l_dfe			
P0072	S0001	l_brt			
P0073	S0001	l_p1t			
		l_rat			
		l_dpou			
		l_vvpot			
		l_vvod			

2>Data in Table 'tblParametar' in 'cbrbase' on '01313'

naziv	vrsta	opis	tip
l_hjez	08	visina limova	num
l_hpr	08	visina prozora	num
l_lpz	08	duljina zadnjeg paketa	num
l_lst	08	duljina steznika	num
l_hjez	08	duljina limova	num
l_p1t	08	pozicija prve trake od dna prozora	num
l_pgjhd	08	pozicija granicnika horizontalno desno	num
l_pgjhl	08	pozicija granicnika horizontalno lijevo	num
l_pipzp	08	preпуст izolacije preko zadnjeg paketa	num
l_pnosl	08	udaljenost nosaca od simetrale jezgre	num
l_pstv	08	vertikalna pozicija steznika	num
l_psvh	08	pozicija horizontalnog svornika od srednjeg stupa	num
l_psvv	08	vertikalna pozicija provrta	num
l_ptuhl	08	pozicija tuljka od simetrale	num
l_pushd	08	pozicija usice horizontalno desno	num
l_pusvl	08	pozicija usice horizontalno lijevo	num
l_pusv	08	vertikalna pozicija usice	num
l_pvnss	08	udaljenost nosaca od simetrale stupa	num
l_rat	08	razmak traka na stupovima	num
l_rs	08	razmak steznika	num
l_rum	08	mali radius na usici	num
l_ruv	08	veliki radius na usici	num
l_sim	08	razmak simetrale stupova	num
l_srd	08	sinak ravnog dijela	num
l_esp	08	debljina stezne ploce	num
l_tip	Cv	tip jezgre	sym
l_vpp	08	visina potpore precke	num
l_vvz	08	visina završnog paketa	num

2>Data in Table 'tblSklop' in 'cbrbase' on '01313'

nadredjeni	podredjeni
S0001	S0002
S0001	S0003
S0001	S0004
S0001	S0005
S0001	S0006
S0001	S0007
S0001	S0008
S0001	S0009
S0001	S0014
S0001	S0015
S0001	S0016
S0001	S0017
S0001	S0019
S0001	S0020
S0001	S0021
S0002	S0010
S0002	S0018
S0003	S0018
S0004	S0010
S0004	S0018
S0005	S0018

Slika 8-1: Prikaz dijela tablica u bazi uzoraka

Tablica 10 prikazuje parametre koji opisuju sklop jezgre. Navedene su vrijednosti parametara za uzorke koji se nalaze u bazi uzoraka i za novi konstrukcijski zadatak. Osjenčani su oni parametri koji su korišteni za pronalaženje najslabijeg uzorka. Navedene su vrijednosti samo za one parametre koji su pridruženi sklopu jezgre. Vrijednosti parametara koji su pridruženi podsklopovima nisu prikazane (vrijednost «-»). Parametri čija vrijednost nije pronađena u modelu imaju vrijednost NULL.

Opis	Oznaka	U158	U163	U168	U173	U178
broj bandaznih traka	j_brt	7	9	11	8	10
sirina izolacijske ploce poz. 84	j_d11p	220,0	220,0	220,0	190,0	210,0
duljina izolacijske ploce poz.84	j_d12p	2546,0	2606,0	2300,0	2200,0	2312,0
promjer jezgre	j_dfe	655	642	490	485	565
sirina nosaca svornika	j_dim_A	NULL	550,0	NULL	NULL	NULL
debljina nosaca svornika	j_dnl	20,0	NULL	NULL	NULL	20,0
sirina vodilice	j_dpl	142,0	168,0	101,0	73,0	168,0
duzina potpora	j_dpot	247,0	227,0	184,0	198,0	187,0
ukupna debljina paketa	j_dpu	636,0	622,0	470,0	470,0	542,0
debljina ravnog dijela	j_drd	5,0	6,0	4,0	4,0	5,0
sirina stezne ploce	j_dsp	100,0	100,0	100,0	80,0	100,0
debljina lima steznika	j_dst	-	-	-	-	-
promjer provrta za horizontalni svornik	j_fipsvh	-	-	-	-	-
promjer provrta za vanjski svornik	j_fipsvv	-	-	-	-	-
promjer usice	j_fiu	-	-	-	-	-
visina limova	j_hjez	2398,0	2554,0	2776,0	2307,0	2791,0
visina prozora	j_hpr	1118,0	1314,0	1816,0	1347,0	1711,0
duljina zadnjeg paketa	j_lpz	2486,0	2546,0	2240,0	2140,0	2252,0
duljina steznika	j_lst	-	-	-	-	-
duljina limova	j_ljez	2986,0	3006,0	2580,0	2500,0	2632,0
pozicija prve trake od dna prozora	j_plt	20,0	20,0	20,0	20,0	20,0
pozicija granicnika horizontalno desno	j_pgihd	-	-	-	-	-
pozicija granicnika horizontalno lijevo	j_pgihl	-	-	-	-	-
prepust izolacije preko zadnjeg paketa	j_pipzp	30,0	30,0	30,0	30,0	30,0
udaljenost nosaca od simetrane jezgre	j_pnosl	-	-	-	-	-
vertikalna pozicija steznika	j_pstv	5,0	5,0	5,0	5,0	5,0
pozicija horizontalnog svornika od	j_psvh	370,0	360,0	290,0	290,0	320,0
vertikalna pozicija provrta	j_psvv	-	-	-	-	-
pozicija tuljka od simetrane	j_ptuhl	NULL	NULL	NULL	NULL	NULL
pozicija usice horizontalno desno	j_pushd	-	-	-	-	-
pozicija usice horizontalno lijevo	j_pushl	-	-	-	-	-
vertikalna pozicija usice	j_pusv	-	-	-	-	-
udaljenost nosaca od simetrane stupa	j_pvnss	-	-	-	-	-
razmak traka na stupovima	j_rat	170,0	150,0	174,0	180,0	180,0
razmak steznika	j_rs	640,0	626,0	474,0	474,0	546,0
mali radijus na usici	j_rum	-	-	-	-	-
veliki radijus na usici	j_ruv	-	-	-	-	-
razmak simetrane stupova	j_sim	1173,0	1193,0	1050,0	1010,0	1046,0
sirina ravnog dijela	j_srd	60,0	70,0	60,0	50,0	80,0
debljina stezne ploce	j_ssp	8,0	8,0	8,0	6,0	8,0
tip jezgre	j_tip	SS30	SS30	SS30	SS30	SS30
visina potpore precke	j_vpp	104,0	109,0	116,0	71,0	80,0
visina zadnjeg paketa	j_vpz	140,0	160,0	140,0	120,0	160,0
visina rebra nosaca	j_vrn1	60,0	NULL	NULL	NULL	60,0
visina steznika	j_vst	-	-	-	-	-
visina vodilice	j_vvod	385,0	365,0	280,0	280,0	315,0
visina vertikalne potpore	j_vvpot	180,0	140,0	100,0	130,0	130,0

Opis	Oznaka	U183	U188	U193	U198	U203
broj bandaznih traka	j_brt	8	10	7	10	6
sirina izolacijske ploce poz. 84	j_d11p	210,0	200,0	190,0	210,0	170,0
duljina izolacijske ploce poz.84	j_d12p	1702,0	2524,0	1888,0	2180,0	1632,0
promjer jezgre	j_dfe	398,0	542,0	445,0	545,0	338,0
sirina nosaca svornika	j_dim_A	300,0	NULL	350,0	450,0	250,0
debljina nosaca svornika	j_dnl	20,0	20,0	20,0	20,0	20,0
sirina vodilice	j_dp1	120,0	154,0	70,0	165,0	110,0
duzina potpora	j_dpot	127,0	185,0	178,0	181,0	105,0
ukupna debljina paketa	j_dpu	373,0	524,0	425,0	525,0	318,0
debljina ravnog dijela	j_drd	6,0	4,0	6,0	5,0	5,0
sirina stezne ploce	j_dsp	100,0	100,0	80,0	100,0	80,0
debljina lima steznika	j_dst	-	-	-	-	-
promjer provrta za horizontalni svornik	j_fipsvh	-	-	-	-	-
promjer provrta za vanjski svornik	j_fipsvv	-	-	-	-	-
promjer usice	j_fiu	-	-	-	-	-
visina limova	j_hjez	1892,0	2588,0	2125,0	2563,0	1534,0
visina prozora	j_hpr	1132,0	1548,0	1245,0	1523,0	894,0
duljina zadnjeg paketa	j_lpz	1642,0	2464,0	1828,0	2120,0	1572,0
duljina steznika	j_lst	-	-	-	-	-
duljina limova	j_ljez	1882,0	2844,0	2148,0	2500,0	1792,0
pozicija prve trake od dna prozora	j_p1t	20,0	20,0	20,0	20,0	20,0
pozicija granicnika horizontalno desno	j_pgihd	-	-	-	-	-
pozicija granicnika horizontalno lijevo	j_pgihl	-	-	-	-	-
prepust izolacije preko zadnjeg paketa	j_pipzp	30,0	30,0	30,0	30,0	30,0
udaljenost nosaca od simetrane jezgre	j_pnosl	-	-	-	-	-
vertikalna pozicija steznika	j_pstv	5,0	5,0	5,0	5,0	5,0
pozicija horizontalnog svornika od	j_psvh	280,0	310,0	270,0	310,0	270,0
vertikalna pozicija provrta	j_psvv	-	-	-	-	-
pozicija tuljka od simetrane	j_ptuhl	NULL	NULL	NULL	NULL	NULL
pozicija usice horizontalno desno	j_pushd	-	-	-	-	-
pozicija usice horizontalno lijevo	j_pushl	-	-	-	-	-
vertikalna pozicija usice	j_pusv	-	-	-	-	-
udaljenost nosaca od simetrane stupa	j_pvnss	-	-	-	-	-
razmak traka na stupovima	j_rat	150,0	160,0	190,0	160,0	160,0
razmak steznika	j_rs	379,0	528,0	431,0	531,0	324,0
mali radijus na usici	j_rum	-	-	-	-	-
veliki radijus na usici	j_ruv	-	-	-	-	-
razmak simetrane stupova	j_sim	751,0	1162,0	854,0	990,0	736,0
sirina ravnog dijela	j_srd	80,0	40,0	50,0	60,0	50,0
debljina stezne ploce	j_ssp	8,0	8,0	8,0	8,0	8,0
tip jezgre	j_tip	SS30	SS30	SS30	SS30	SS30
visina potpore precke	j_vpp	322,0	148,0	95,0	105,0	116,0
visina zadnjeg paketa	j_vpz	140,0	140,0	120,0	140,0	100,0
visina rebra nosaca	j_vrn1	60,0	60,0	60,0	60,0	60,0
visina steznika	j_vst	-	-	-	-	-
visina vodilice	j_vvod	230,0	300,0	260,0	315,0	200,0
visina vertikalne potpore	j_vvpot	70,0	110,0	110,0	130,0	70,0

Opis	Oznaka	U208	U213	U218	U223	Z
broj bandaznih traka	j_brt	8	8	6	6	8
sirina izolacijske ploce poz. 84	j_d11p	190,0	230,0	180,0	190,0	210,0
duljina izolacijske ploce poz.84	j_d12p	1638,0	2408,0	1458,0	1840,0	2236,0
promjer jezgre	j_dfe	333,0	506,0	364,0	424,0	481,0
sirina nosaca svornika	j_dim_A	NULL	NULL	NULL	NULL	400,0
debljina nosaca svornika	j_dnl	20,0	20,0	20,0	20,0	NULL
sirina vodilice	j_dp1	94,0	81,0	57,0	142,0	142,0
duzina potpora	j_dpot	108,0	196,0	139,0	129,0	160,0
ukupna debljina paketa	j_dpu	311,0	473,0	344,0	400,0	460,0
debljina ravnog dijela	j_drd	5,0	8,0	3,0	6,0	5,0
sirina stezne ploce	j_dsp	80,0	160,0	100,0	100,0	100,0
debljina lima steznika	j_dst	-	-	-	-	-
promjer provrta za horizontalni svornik	j_fipsvh	-	-	-	-	-
promjer provrta za vanjski svornik	j_fipsvv	-	-	-	-	-
promjer usice	j_fiu	-	-	-	-	-
visina limova	j_hjez	1816,0	2329,0	1604,0	1732,0	2288,0
visina prozora	j_hpr	1176,0	1329,0	884,0	932,0	1368,0
duljina zadnjeg paketa	j_lpz	1578,0	2348,0	1392,0	1780,0	2176,0
duljina steznika	j_lst	-	-	-	-	-
duljina limova	j_ljez	1778,0	2668,0	1632,0	2040,0	2496,0
pozicija prve trake od dna prozora	j_p1t	20,0	20,0	20,0	20,0	20,0
pozicija granicnika horizontalno desno	j_pgihd	-	-	-	-	-
pozicija granicnika horizontalno lijevo	j_pgihl	-	-	-	-	-
prepust izolacije preko zadnjeg paketa	j_pipzp	30,0	30,0	30,0	30,0	30,0
udaljenost nosaca od simetrane jezgre	j_pnosl	-	-	-	-	-
vertikalna pozicija steznika	j_pstv	5,0	5,0	5,0	5,0	5,0
pozicija hor. svornika od srednjeg stupa	j_psvh	270,0	300,0	403,0	270,0	280,0
vertikalna pozicija provrta	j_psvv	-	-	-	-	-
pozicija tuljka od simetrane	j_ptuhl	NULL	NULL	NULL	NULL	NULL
pozicija usice horizontalno desno	j_pushd	-	-	-	-	-
pozicija usice horizontalno lijevo	j_pushl	-	-	-	-	-
vertikalna pozicija usice	j_pusv	-	-	-	-	-
udaljenost nosaca od simetrane stupa	j_pvnss	-	-	-	-	-
razmak traka na stupovima	j_rat	155,0	175,0	1372,0	170,0	180,0
razmak steznika	j_rs	315,0	477,0	348,0	404,0	466,0
mali radijus na usici	j_rum	-	-	-	-	-
veliki radijus na usici	j_ruv	-	-	-	-	-
razmak simetrane stupova	j_sim	729,0	1084,0	642,0	820,0	1018,0
sirina ravnog dijela	j_srd	40,0	100,0	50,0	80,0	60,0
debljina stezne ploce	j_ssp	8,0	8,0	8,0	8,0	8,0
tip jezgre	j_tip	SS30	SS30	SS30	SS30	SS30
visina potpore precke	j_vpp	170,0	95,0	231,0	104,0	48,0
visina zadnjeg paketa	j_vpz	120,0	180,0	120,0	140,0	140,0
visina rebra nosaca	j_vrn1	60,0	60,0	60,0	60,0	NULL
visina steznika	j_vst	-	-	-	-	-
visina vodilice	j_vvod	200,0	290,0	220,0	240,0	275,0
visina vertikalne potpore	j_vvpot	60,0	90,0	90,0	70,0	90,0

Tablica 10: Vrijednosti parametara za pojedine uzorke i zadatak

Pogodni uzorci pretraženi su na osnovu vrijednosti čvornog parametra tip jezgre (J_{TIP}) i njegova vrijednost je simbolička i iznosi SS30, što znači su u razmatranje uzeti samo oni uzorci u bazi uzoraka koji imaju vrijednost parametra $J_{TIP}=SS30$.

		p01	p02	p03	p04	p05	p06	p07	p08	p09	p10	p11	p12
j_brt	p01		2	0	2	2	0	2	2	2	2	2	2
j_d11p	p02			0	2	1	0	2	2	2	0	0	2
j_d12p	p03				2	1	0	2	2	2	0	0	2
j_dfe	p04					1	1	1	1	0	1	1	1
j_dim_A	p05						2	2	2	2	2	2	2
j_dnl	p06							2	2	2	2	2	2
j_dp1	p07								0	0	1	0	1
j_dpot	p08									0	1	0	1
j_dpu	p09										1	1	1
j_drd	p10											0	2
j_dsp	p11												0
j_dst	p12												
j_fipsvh	p13												
j_fipsvv	p14												
j_fiu	p15												
j_hjez	p16												
j_hpr	p17												
j_lpz	p18												
j_lst	p19												
j_ljez	p20												
j_plt	p21												
j_pgjhd	p22												
j_pgjhl	p23												
j_pipzp	p24												
j_pnosl	p25												
j_pstv	p26												
j_psvh	p27												
j_psvv	p28												
j_ptuhl	p29												
j_pushd	p30												
j_pushl	p31												
j_pusv	p32												
j_pvnss	p33												
j_rat	p34												
j_rs	p35												
j_rum	p36												
j_ruv	p37												
j_sim	p38												
j_srd	p39												
j_ssp	p40												
j_tip	p41												
j_vpp	p42												
j_vpz	p43												
j_vrn1	p44												
j_vst	p45												
j_vvod	p46												
j_vvpot	p47												

	p13	p14	p15	p16	p17	p18	p19	p20	p21	p22	p23	p24	p25	p26
p01	2	2	2	2	2	2	2	2	1	2	2	0	2	1
p02	1	1	2	2	2	2	2	2	0	2	2	1	2	1
p03	1	1	2	2	2	2	2	2	0	2	2	1	2	1
p04	1	1	1	0	0	1	2	0	1	1	1	1	1	1
p05	2	2	2	2	2	2	2	2	0	2	2	0	2	1
p06	2	2	2	2	2	2	2	2	0	2	2	0	2	1
p07	1	1	1	2	2	2	2	2	1	0	0	1	0	1
p08	1	1	1	2	2	0	2	2	1	2	2	1	2	1
p09	1	1	1	0	0	1	0	0	1	1	1	1	1	1
p10	0	0	0	2	2	0	0	2	1	0	0	1	2	1
p11	1	1	1	2	2	0	2	2	1	0	0	1	2	1
p12	1	1	1	2	2	1	2	2	1	0	0	1	2	1
p13	0	0	0	2	2	2	2	2	0	2	2	0	2	0
p14	0	0	0	2	2	2	2	2	0	2	2	0	2	0
p15	0	0	0	2	2	2	2	2	0	2	2	0	2	0
p16	0	0	0	0	0	0	0	0	1	1	1	1	0	1
p17	0	0	0	0	0	0	0	0	1	1	1	1	0	1
p18	0	0	0	0	0	0	0	0	1	1	1	1	0	1
p19	0	0	0	0	0	0	0	0	1	1	1	1	1	1
p20	0	0	0	0	0	0	0	0	1	1	1	1	0	1
p21	0	0	0	0	0	0	0	0	2	2	2	0	2	0
p22	0	0	0	0	0	0	0	0	0	0	0	1	2	1
p23	0	0	0	0	0	0	0	0	0	0	0	1	2	1
p24	0	0	0	0	0	0	0	0	0	0	0	2	2	0
p25	0	0	0	0	0	0	0	0	0	0	0	0	1	1
p26	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p27	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p28	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p29	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p30	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p31	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p32	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p33	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p34	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p35	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p36	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p37	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p38	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p40	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p41	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p42	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p43	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p44	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p45	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p46	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p47	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	p27	p28	p29	p30	p31	p32	p33	p34	p35	p36	p37	p38	p39	p40
p01	2	2	2	2	2	0	2	1	2	2	2	2	2	2
p02	2	2	2	2	2	0	2	1	2	0	0	2	2	2
p03	2	2	2	2	2	0	2	1	2	0	0	2	2	2
p04	1	1	1	1	1	1	1	1	0	1	1	0	1	1
p05	2	2	2	2	2	2	2	0	2	2	2	2	2	2
p06	2	2	2	2	2	2	2	0	2	2	2	2	2	2
p07	1	1	2	0	0	0	2	1	2	1	1	2	1	1
p08	1	1	2	0	0	0	0	1	0	1	1	2	1	1
p09	1	1	1	1	1	1	1	1	0	1	1	0	1	1
p10	0	0	2	2	2	0	2	0	2	0	0	2	0	0
p11	2	1	2	2	2	0	2	1	2	1	1	2	1	0
p12	1	1	0	0	0	0	1	1	2	1	1	0	1	1
p13	0	0	2	2	2	2	2	0	2	0	0	2	0	0
p14	0	0	2	2	2	2	2	0	2	0	0	2	0	0
p15	0	0	2	2	2	2	2	0	2	0	0	2	0	0
p16	1	1	0	0	0	0	0	1	0	1	1	0	1	1
p17	1	1	0	0	0	0	0	1	0	1	1	0	1	1
p18	1	1	0	0	0	0	0	1	0	1	1	0	1	1
p19	1	1	1	1	1	1	2	1	0	1	1	0	1	1
p20	1	1	0	0	0	0	0	1	0	1	1	0	1	1
p21	2	2	2	2	2	2	2	0	2	2	2	2	2	2
p22	1	1	0	0	0	1	2	1	2	1	1	2	1	1
p23	1	1	0	0	0	1	2	1	2	1	1	2	1	1
p24	2	2	2	2	2	2	2	0	2	0	0	2	0	2
p25	0	1	0	0	0	1	0	1	2	1	1	2	0	0
p26	2	0	2	2	2	0	2	0	2	0	0	2	0	0
p27		0	0	0	0	1	0	1	0	1	1	0	1	1
p28			2	2	2	0	2	0	2	0	0	2	0	0
p29				1	1	1	0	1	0	1	1	0	1	1
p30					0	1	1	1	0	1	1	2	1	1
p31						1	1	1	0	1	1	2	1	1
p32							0	1	0	0	0	2	0	0
p33								1	0	1	1	0	1	1
p34									2	0	0	2	2	2
p35										1	1	0	1	1
p36											0	2	0	0
p37												2	0	0
p38													1	1
p39														0
p40														
p41														
p42														
p43														
p44														
p45														
p46														
p47														

	p41	p42	p43	p44	p45	p46	p47	Σ1	w
p01	2	2	2	1	2	2	2		
p02	2	2	2	1	2	2	2		
p03	2	2	2	1	2	2	2		
p04	2	1	0	1	1	1	1	37	0,18
p05	2	2	2	2	2	2	2		
p06	2	2	2	2	2	2	2		
p07	2	2	0	1	2	0	2		
p08	2	2	1	0	2	0	0		
p09	2	1	1	1	2	1	0	34	0,17
p10	2	2	0	0	2	2	2		
p11	2	2	2	1	2	2	2		
p12	2	0	1	0	0	0	0		
p13	2	2	2	2	2	2	2		
p14	2	2	2	2	2	2	2		
p15	2	2	2	2	2	2	2		
p16	2	0	0	0	0	0	1	26	0,13
p17	2	0	0	0	0	0	1	26	0,13
p18	2	0	0	0	0	0	0		
p19	2	1	1	1	0	1	1		
p20	2	2	2	2	2	2	2	25	0,12
p21	2	2	2	2	2	2	2		
p22	2	2	0	0	2	2	2		
p23	2	2	0	0	2	2	2		
p24	2	2	2	2	2	2	2		
p25	2	2	0	0	2	2	2		
p26	2	2	2	2	2	2	2		
p27	2	2	0	2	2	2	2		
p28	2	2	2	2	2	2	2		
p29	2	2	1	0	2	2	2		
p30	2	2	0	0	2	2	2		
p31	2	2	0	0	2	2	2		
p32	2	2	2	2	2	2	2		
p33	2	2	0	2	2	2	2		
p34	2	2	2	2	2	2	2		
p35	2	2	1	0	0	2	0	25	0,12
p36	2	2	2	2	2	2	2		
p37	2	2	2	2	2	2	2		
p38	2	2		1	1	1	1	32	0,16
p39	2	2	0	2	2	2	2		
p40	2	2	0	2	2	2	2		
p41		1	1	1	1	1	1		
p42			1	0	0	0	0		
p43				2	2	2	2		
p44					2	2	2		
p45						0	0		
p46							0		
p47									
Σ								205	1

Tablica 11: Tablica prednosti promatranih parametara sa izračunatim težinskim faktorima

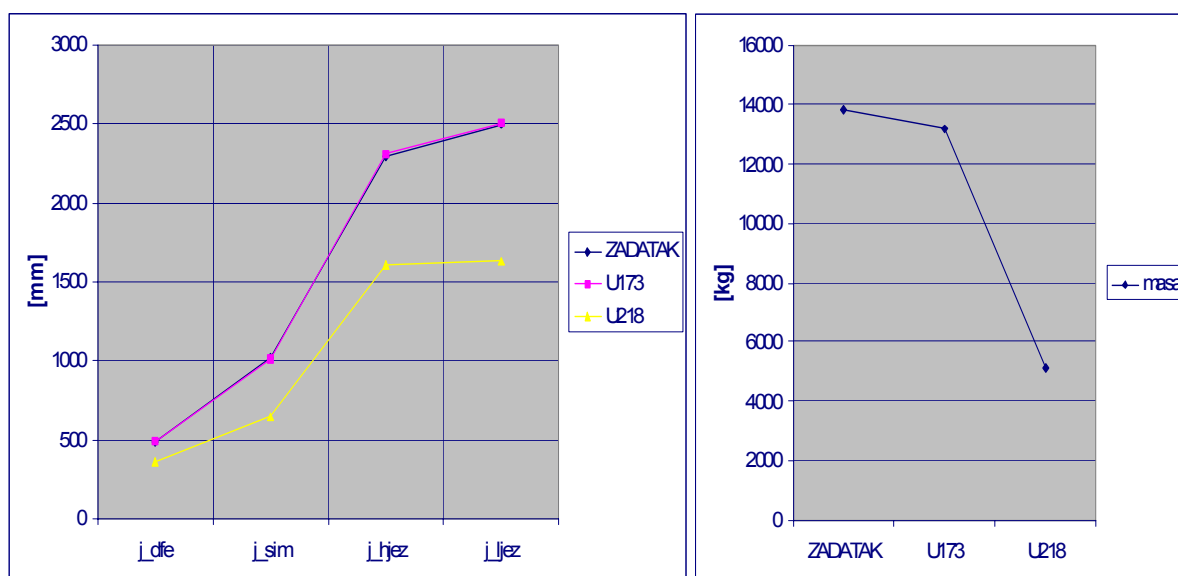
Tablica 11 prikazuje međusobne prednosti svih parametara. Za parametre koji su razmatrani pri računanju sličnosti (osjenčano) izračunate su vrijednosti težinskih faktora.

U158	U163	U168	U173	U178	U183	U188
0,928	0,928	0,929	0,996	0,931	0,911	0,944

U193	U198	U203	U208	U213	U218	U223
0,953	0,964	0,872	0,896	0,979	0,863	0,907

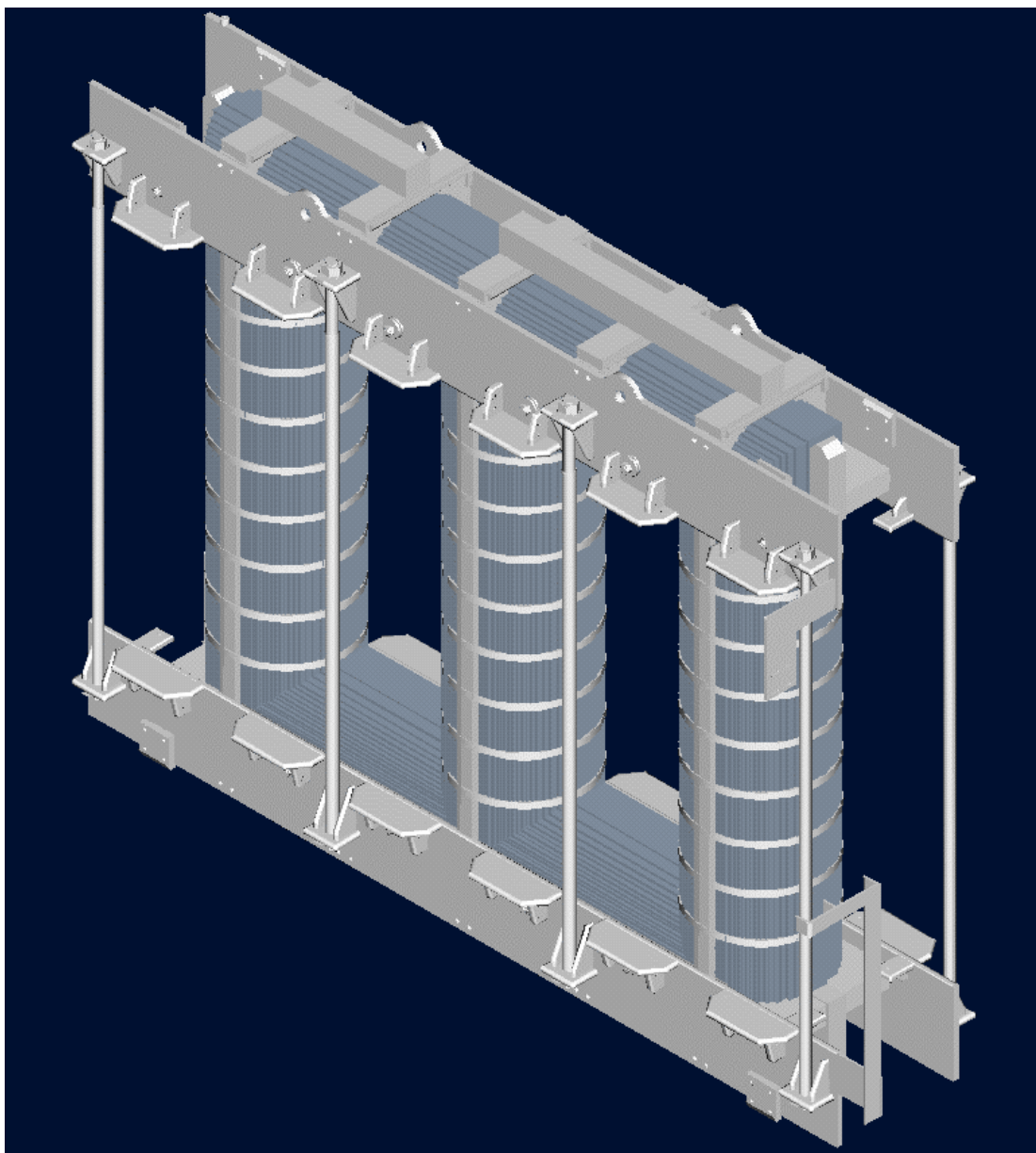
Tablica 12: Izračunate vrijednosti sličnosti

Tablica 12 prikazuje izračunate vrijednosti sličnosti uzoraka u bazi i novog konstrukcijskog zadatka. Može se uočiti da je uzorak s oznakom U173 (Slika 8-3) najbližiji zadatku dok je uzorak s oznakom U218 najmanje sličan zadatku (Slika 8-4). Slika 8-5 prikazuje model sklopa jezgre koji je korišten kao novi konstrukcijski zadatak.



Slika 8-2: Dijagram vrijednosti nekih parametara promatranih uzoraka

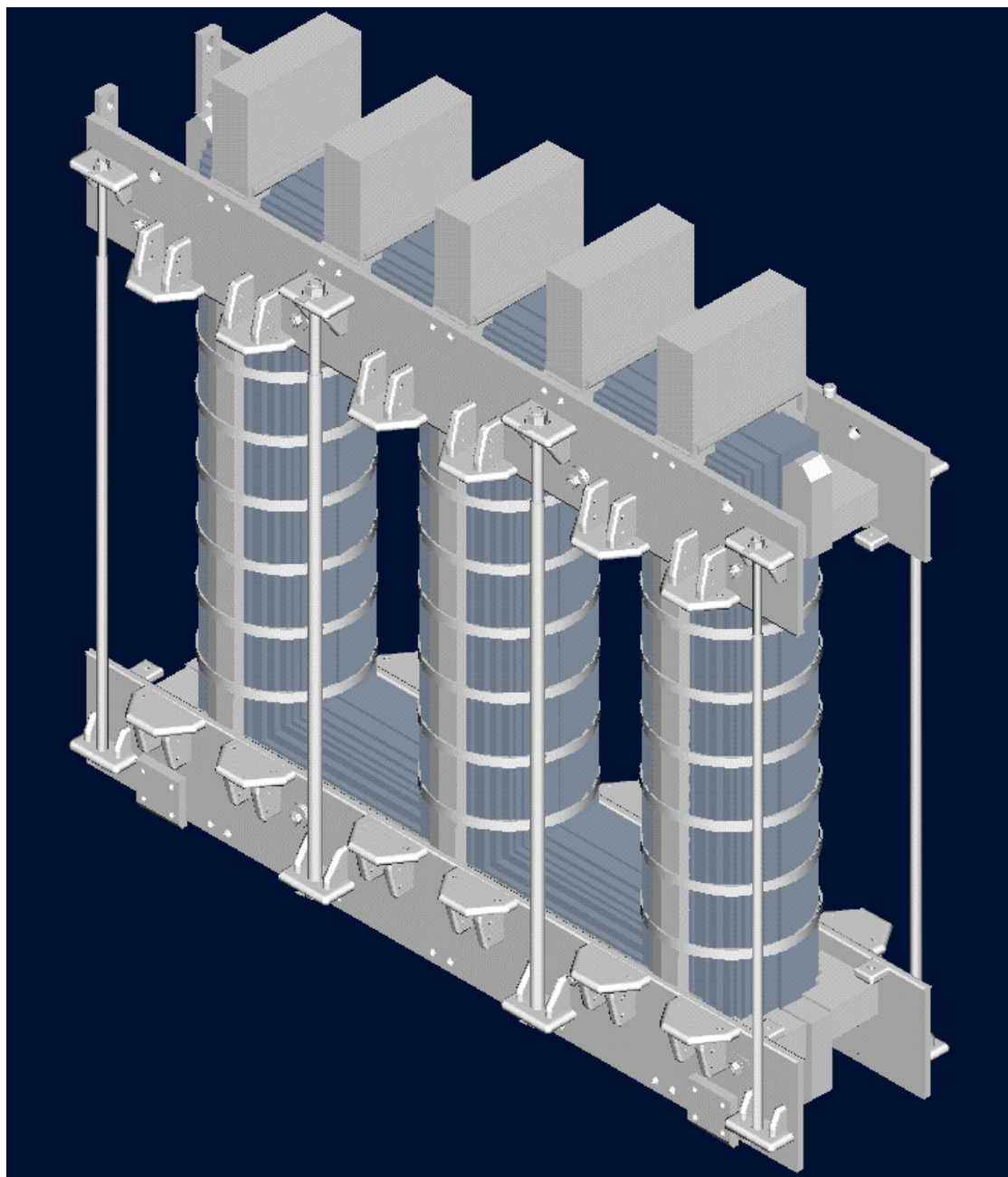
Dijagrami (Slika 8-2) prikazuju usporedne vrijednosti parametara koji opisuju osnovne dimenzije i mase promatranih uzoraka i zadatka. Vidljivo je poklapanje osnovnih dimenzija najbližnjeg uzorka i zadatka i mala razlika u masama.



Slika 8-3: Parametarski model najslićnijeg uzorka U173

Osnovne karakteristike najslićnijeg uzorka prikazanog na slici (Slika 8-3):

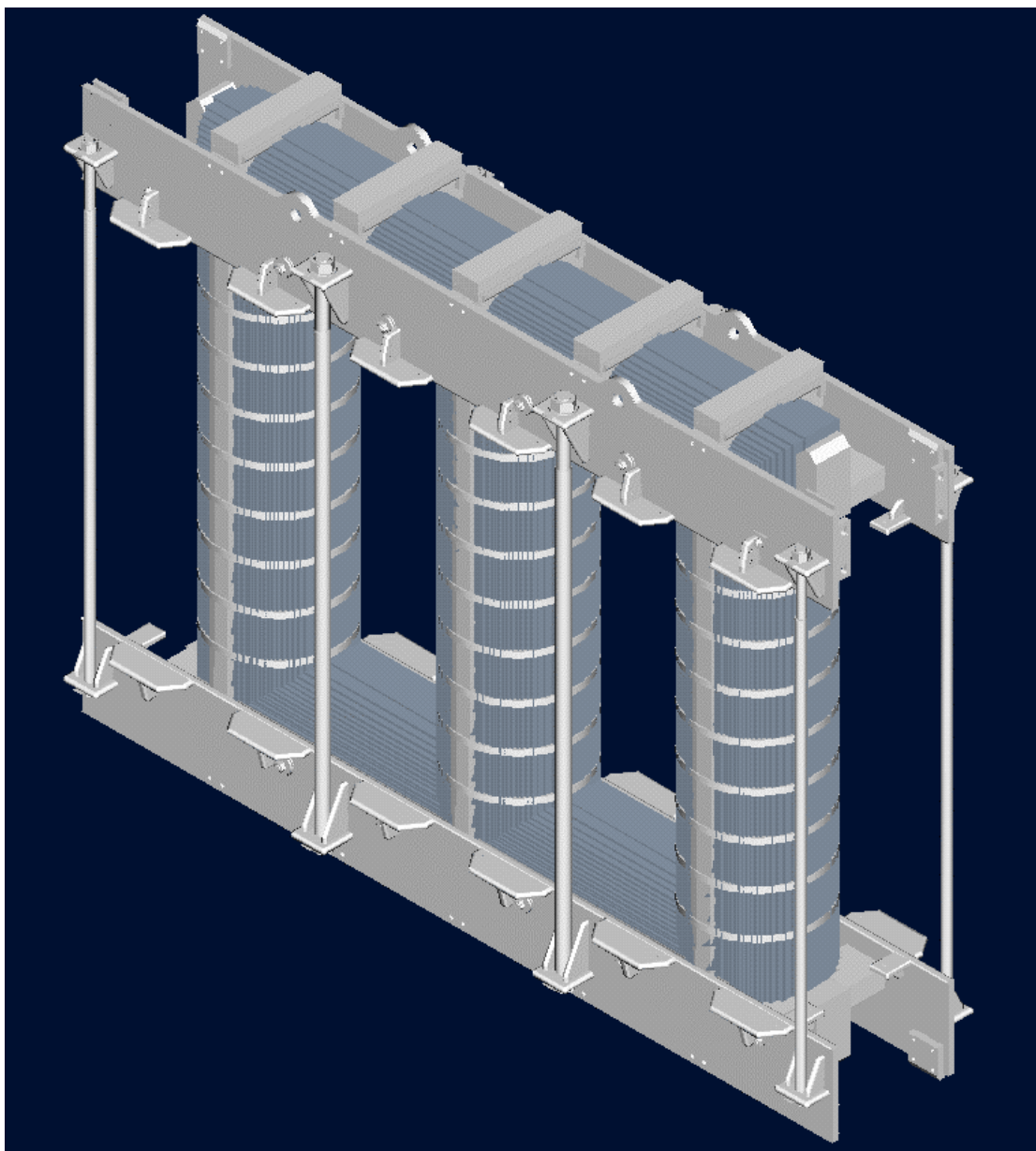
identifikacijska oznaka:	U173
naziv:	Irska 30 MVA
visina limova:	2307 mm
duljina limova:	2500 mm
masa:	13160 kg
sličnost:	0,996



Slika 8-4: Parametarski model najrazličitijeg uzorka U218

Osnovne karakteristike najrazličitijeg uzorka prikazanog na slici (Slika 8-4):

identifikacijska oznaka:	U218
naziv:	Dominikanska Republika 8 MVA
visina limova:	1604 mm
duljina limova:	1632 mm
masa:	5100 kg
sličnost:	0,863



Slika 8-5: Parametarski model gotovog rješenja novog konstrukcijskog zadatka

Osnovne karakteristike zadatka prikazanog na slici (Slika 8-5):

naziv:	Malezija 25 MVA
visina limova:	2288 mm
duljina limova:	2496 mm
masa:	13833 kg

9 ZAKLJUČAK

9.1 Rezultati rada

Prema postavljenom zadatku rada trebalo je istražiti mogućnosti razvoja sustava za prepoznavanje prijašnjih konstrukcijskih rješenja koji će biti podrška konstruktoru u svim fazama procesa konstruiranja proizvoda. Sustav je razvijen primjenom metodologije rasuđivanja temeljem uzoraka koja definira postupak u četiri faze: (1) pronalaženje, (2) rješavanje, (3) revizija i (4) pohranjivanje. Realizirani sustav za prepoznavanje uzoraka pri varijantnom konstruiranju obuhvaća faze: (1) pronalaženje i (4) pohranjivanje.

U glavi 4 analiziran je način zapisivanja gotovih konstrukcijskih rješenja u bazu uzoraka. Predloženi model zapisivanja uzoraka je u obliku hijerarhijskog stabla. Proizvod je fizički strukturiran do razine sklopova i njegova struktura zapisana je u bazu uzoraka. Sklopovima se pridružuju parametri koji ih opisuju. Baza uzoraka realizirana je kao relacijska baza podataka.

Istraživanja u glavama 5 i 6 definiraju informatičke modele pronalaženja najslbližijeg uzorka i pohranjivanja novog uzorka u bazu uzoraka.

Pronalaženje novog uzorka izvedeno je metodom najbliži susjed koja je najjednostavnija i najraširenija metoda. Smanjenje utjecaja nerelevantnih parametara na proces pronalaženja najslbližijeg uzorka postiže se korištenjem težinskih faktora pri računanju sličnosti uzoraka i novog konstrukcijskog zadatka. Težinski faktori računaju se temeljem međusobne usporedbe važnosti parametara koja je zapisana u bazu uzoraka.

Pohranjivanje novog uzorka temelji se na pretpostavci da je model novog konstrukcijskog rješenja konačan (revidiran) i da je modeliran u parametarskom trodimenzionalnom CAD sustavu. U programski kod sustava implementirano je prikupljanje vrijednosti parametara iz parametarskih modela i njihovo zapisivanje u bazu uzoraka. Na taj način je novi uzorak pohranjen u bazu uzoraka.

Prikupljanje informacija o definiciji novog konstrukcijskog zadatka iz tekstualnih datoteka i baza podataka u ovom radu nije posebno razmatrano. Autor je mišljenja da se za primjenu predloženog sustava u realnom okruženju razvoja proizvoda treba razmotriti i implementirati prikupljanje informacija o zadatku iz tekstualnih datoteka i baza podataka. To je posebno je važno ako se sustav namjerava implementirati u okruženje koje ima razvijen PDM²⁵ sustav.

²⁵ Product Data Management

Primjena predloženog sustava prikazana je na realnom sklopu jezgre energetskog transformatora u glavi 8. Dobiveni rezultati prikazuju ispravnost predloženog sustava za prepoznavanje uzoraka. S obzirom na fizičku sličnost pronadenog uzorka i zadatka korištenje pronadenog uzorka kao predložka svakako bi trebao ubrzati izradu konstrukcijske dokumentacije. S druge strane ako usporedimo mase pronadenog uzorka i zadatka možemo uočiti da je masa novog konstrukcijskog rješenja za 5,1% veća od mase najbližnjeg uzorka. Na temelju toga može se pretpostaviti korisnost sustava pri procjenjivanju masa sklopova proizvoda u fazi izrade ponude.

Stvarna dobit sustava pokazati će se korištenjem sustava u realnom okruženju pri rješavanju konkretnih konstrukcijskih problema.

Sustav koji je bio predmetom istraživanja u ovom radu realiziran je programskim jezikom Java. Kao parametarski trodimenzionalni CAD sustav u kojem su izrađeni modeli uzoraka korišten je Pro/ENGINEER, a relacijska baza podataka realizirana je pomoću SQL Server baze podataka. Za pristup CAD sustavu korištene su biblioteke klasa koje su dio CAD sustava (J-Link biblioteka klasa).

9.2 *Mogućnosti daljnjeg istraživanja*

Problem uočen pri realizaciji sustava za prepoznavanje uzoraka je, mogli bismo reći, komplicirano dodjeljivanje prednosti parametara za računanje težinskih faktora pri dodavanju novog parametra ili sklopa odnosno promjena postojećih vrijednosti prednosti parametara koje su upisane u bazu uzoraka.

Jedna od smjernica za daljnje istraživanje svakako je razmatranje jednostavnijeg pristupa pri računanju težinskih faktora. Postoji više razvijenih sustava i znanstvenih radova koji su se bavili ovom problematikom [45], [54], [56]

S obzirom na stupanj razvijenosti i širinu primjene Intranet/Internet tehnologije daljnja istraživanja svakako treba usmjeriti na razmatranje mogućnosti razvoja sustava u mrežnom okruženju. Mnoga istraživanja provode se u tom smjeru [71], [72] [73], [74], [75], [76], [77]. U posljednje vrijeme na području rasuđivanja temeljem uzoraka velika pažnja posvećuje se primjeni XML²⁶ tehnologije kao načinu razmjene informacija između baze uzoraka i korisnika [78], [79], [80], [81]. Pomoću aplikacija koje se izvršavaju na serverskoj strani mrežnog sustava (Java servlet) prikupljaju se podaci iz baze uzoraka i koristeći XML tehnologiju prikazuju korisniku sustava. Korisnik pomoću mrežnog preglednika (web browser) i aplikacija koje se izvršavaju na njegovoj strani mrežnog sustava (Java applet) izračunava sličnosti i komunicira sa sustavom.

²⁶ Extensible Markup Language

LITERATURA

- [1] Pahl B., Beitz W., Engineering Design a systematic approach, The Design Council, London, 1988.
- [2] K. Sycara, D. Navichandra, Retrieval Strategies in a Case-Based Design System, in C. Tong and D. Sriram, editors, Artificial Intelligence in Engineering Design, chapter 4, pages 145-164. Academic Press, New York, NY, 1992. Vol II.
- [3] Maher M. L., Gomez de Silva Garza A., Design by Interactive Exploration Using Memory-Based Techniques, Knowledge-Based Systems, Volume 9, Number 1, 1996.
- [4] Bergmann R., Althoff K.-D. Methodology for Building CBR Applications , Case-Based Reasoning Technology: From Foundations to Applications, M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard, & S. Wess (Eds.).Springer-Verlag Lecture Notes in Computer Science, 1998.
- [5] Watson I., CBR is a methodology not a technology, Research & Development in Expert Systems XV. Miles, R., Moulton, M. & Bramer, M. (Eds.), pp. 213-223. Springer, London, 1998. ISBN 1-85233-086-4, <http://www.cs.auckland.ac.nz/~ian/>
- [6] Hubka V., Eder W. E., Engineering Design – General Procedural Model of Engineering Design, Edition HEURISTA, Zürich, 1992.
- [7] Hubka V., Theorie der Konstruktionsprozesse, Springer, 1976.
- [8] Sriram R. D., Intelligent Systems for Engineering, A Knowledge-based Approach, Springer Verlag, London, 1997.
- [9] Rude S., Wissenbasiertes Konstruieren, Shaker Verlag, Aachen, 1998.
- [10] Duhovnik J., Tavčar J., Elektronsko poslovanje in tehnični informacijski sistemi, PDMS – Products Data Management Systems, LECAD, Univerza v Ljubljani, Fakultet za strojništvo, Ljubljana, 2000.
- [11] Shah J. J., Mäntylä M., Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications, John Wiley & Sons, Inc., New York, 1995.
- [12] Gao Y., Zeid I, Bardasz T., Characteristics of Effective Design Plan to Support Re-use in Case-based Mechanical Design, Knowledge Based Systems, Vol.10, 1998., p337-350
- [13] Smith J. S., Duffy A. H. B., Re-Using Knowledge: Why, What and Where, International Conference on Engineering Design ICED 01, Glasgow, 2001.
- [14] Fowler J. E., Variant Design for Mechanical Artifacts – A State of the Art Survey, National Institute of Standard and Technology, USA, 1994.
- [15] VDI 2221: Methodik zum EntWickeln und Konstruieren technischer Systeme und Produkte, Beuth Verlag, Berlin, 1993.

-
- [16] Žeželj D., Bojčetić N., Marjanović D., Interactive Model for Power Transformer's Core Design, International Conference on Engineering Design ICED 01, Glasgow, 2001.
- [17] Kolodner J., Case-Based Reasoning, Morgan Kaufmann Publishers, Inc., 1993.
- [18] Aamodt A., Plaza E., Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, AI Communications. IOS press, Vol.7: 1, pp.39-59, 1994.
- [19] Watson I., Applying Case-Based Reasoning: Techniques for Enterprise Systems, Morgan Kaufmann Publishers, Inc., San Francisco, California, USA, 1997.
- [20] Tong C., Sriram D., Artificial Intelligence in Engineering Design, Vol. II, Introduction, Academic Press, San Diego, 1992.
- [21] Marjanović D., Implementacija ekspertnih alata u proces konstruiranja, doktorska disertacija, FSB Zagreb, Zagreb, 1995.
- [22] Maher M. L., et al., Case-Based Reasoning in Design, NJ: Lawrence Erlbaum Associates, Northvale, 1995., <http://www.arch.su.edu.au/>
- [23] Detailed description of CBR, <http://www.inreca.org/data/cbr/what/details.html>
- [24] Smyth B., Case-Based Design, PhD Thesis, Trinity College, University of Dublin, 1996.
- [25] Kolodner J., Maintaining organization in dynamic long-term memory, Cognitive Science, Vol.7, s. 243-280, 1983.
- [26] Kolodner J., Reconstructive memory, a computer model, Cognitive Science, Vol.7, s. 281-328, 1983.
- [27] Lopez de Mantaras R., Plaza E., Case-Based Reasoning: An Overview, 1997.
- [28] Bergmann R., et al., Ingredients for Developing a Case-Based Reasoning Methodology. Proceedings of the 5th German Workshop in Case-Based Reasoning (GWCBR '97), LSA-97-01E, University of Kaiserslautern, 1997., pp. 49-58.
- [29] Johnston R., Case-based Reasoning and Institutional Memory, Irish Engineers, Journal, May 1994.
- [30] E. C. Chung, C. A. Vassiliadis, A Case-Based Knowledge System Employed to Trouble Shoot Failures in a Manufacturing Environment, Design and Implementation of Intelligent Manufacturing Systems, Chapter 10, Prentice Hall, 1995.
- [31] Bilgic T., Fox M. S., Constraint-Based Retrieval of Engineering Design Cases, 4th International Conference on Artificial Intelligence in Design, June 1996, Stanford University California, USA
- [32] Goel A. K., Design, Analogy and Creativity, Georgia Institute of Technology, Atlanta, 1997.
- [33] West G., Strachan S., McDonald J., Duffy A., Farrell J., Gwyn B., DEKAS – An Evolutionary Case-Based Reasoning System to Support Protection Scheme Design, International Conference on Engineering Design ICED 01, Glasgow, 2001.
- [34] Jurišica I, Representation and Management Issues for Case-Based Reasoning Systems, Department of Computer Science, University of Toronto, Toronto, 1993.
- [35] Watson I., Marir F., Case-Based Reasoning: A Review, The Knowledge Engineering Review, Vol. 9 No.4, University of Salford, UK, 1994., <http://www.ai-cbr.org>
- [36] Schank R., Dynamic memory; a theory of reminding and learning in computers and people, Cambridge University Press., 1982.

-
- [37] Goel A. K., Bhatta S. R., Stroulia E., KRITIK: An Early Case-based Design System, <http://www-ai-cbr.org/>
- [38] Pearce M., Ashok K. G., Kolodner J. L., Zimring C., Billington R., Case-Based Support – A Case Study in Architectural Design, IEEE Expert, 1992.
- [39] Sycara K., et al., CADET: A case-based synthesis tool for engineering design, International Journal of Expert Systems, Vol. 4, No. 2, 1992.
- [40] Maher M. L., Gomez de Silva Garza A., Developing Case-Based Reasoning for Structural Design, IEEE Expert, Vol. 11, No. 3, June 1996.
- [41] Maher M. L., Gomez de Silva Garza A., The Adaptation of Structural System Designs Using Genetic Algorithms, Glasgow, August 1996
- [42] Göker M., Roth-Berghofer T., et al., The development of HOMER: A case-based CAD/CAM help-desk support tool, Proceedings of the European Workshop on Case-Based Reasoning, 1998.
- [43] Koton P., Using Experience in Learning and Problem Solving, PhD thesis, Laboratory of Computer Science, Massachusetts Institute of Technology, 1989.
- [44] Bareiss R., PROTON; a unified approach to concept representation, classification and learning, Ph.D. Dissertation, University of Texas at Austin, Dep. of Computer Sciences 1988., Technical Report AI88-83.
- [45] Bonzano A., ISAC: a Case-Based Reasoning System for Aircraft Conflict Resolution, PhD thesis, Trinity College, University of Dublin, 1998.
- [46] Cost S., Salzberg S., A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features, 1993.
- [47] Wess S., Althoff K., D., Derwand G., Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning, Topics in Case-Based Reasoning, Proceedings EWCBR-93, pages 167-181, Springer-Verlag, 1994.
- [48] Wettschereck D., Aha D. W., Weighting Features, First International Conference on Case-Based Reasoning, ICCBR-95, 1995.
- [49] Kleinberg J. M., Two Algorithms for Nearest-Neighbor Search in High Dimensions, 1997.
- [50] Aha D. W., Feature Weighting for Lazy Learning Algorithms, Navy Center for Applied Research in Artificial Intelligence, 1998.
- [51] Herthorn P., Rudolph S., Dimensional Analysis in Case-Based Reasoning, Proceedings of International Workshop on Similarity Methods University of Stuttgart, 163-178, 1998.
- [52] Arya S., Mount D. M., Netanyahu N. S., Silverman R., Wu A. Y., An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions, 1998.
- [53] Tsaparas P., Nearest Neighbor Search in Multidimensional Spaces, Department of Computer Science, University of Toronto, 1999.
- [54] Fox S., Leake D. B., Using Introspective Reasoning to Refine Indexing, Proceedings of 14th International Joint Conference on Artificial Intelligence, 1995.
- [55] Muñoz-Avila H., Hüllen J., Feature Weighting by Explaining Case-Based Planning Episodes, Proceedings of the 1996 European Workshop on Case-Based Reasoning, 1996.

-
- [56] Jarmulak J., Craw S., Rowe R., Self-Optimising CBR Retrieval, Proceedings 12th IEEE International Conference on Tools with Artificial Intelligence, pp. 376-383, 2000.
- [57] Quinlan J. R., Induction over large databases, Rep. No. HPP-79-14, Heuristic Programming Project, Computer Science Dept., Stanford University, US, 1979.
- [58] Clarkson P. J., Hamilton J. R., 'Signposting', A Parameter-driven Task-based Model of the Design Process, Research in Engineering Design, Vol. 12, NO. 1, pp. 18-38, 2000.
- [59] Pavković N., Objektno orijentirani pristup modeliranju procesa konstruiranja, doktorska disertacija, FSB Zagreb, Zagreb, 2000.
- [60] Aha D. W., Case-Based Learning Algorithms, DARPA Case-Based Reasoning Workshop, Proceedings, Morgan Kaufmann Publishers, Inc., 1991.
- [61] Obad M., Razvoj sustava za računalnu podršku procesu konstruiranja temeljenu na metodologiji Case-based Reasoning, doktorska disertacija, Strojarski Fakultet Sveučilišta u Mostaru, Mostar, 1999.
- [62] Obad M., Damić V., Dankić B., Development of Integrated CAD System for Support in all Design Phases, DESIGN 2000, Proceedings
- [63] Bergmann R., On the use of Taxonomies for Representing Case Features and Local Similarity Measures, CBR Works application, 6th German Workshop on CBR, 1998.
- [64] Introduction to Pro/ENGINEER, PTC – Documentation, Waltham, 2000
- [65] Šušanj D., Java: Programiranje za Internet i World Wide Web, Znak, Zagreb, 1997.
- [66] The Java Tutorial: A Practical guide for programmers,
- [67] Eckstein R., Loy M., Wood D., Java Swing, O'Reilly, Sebastopol, 1998.
- [68] J.Link User's Guide, PTC – Documentation, Waltham, 2000.
- [69] Microsoft SQL Server Database Developer's Companion, Microsoft Documentation, Microsoft Corporation, 1998.
- [70] Motik B., Šribar J., Demistificirani C++: dobro upoznajte protivnika da biste njime ovladali, Element, Zagreb, 1997.
- [71] Watson I., Oliveira L. Virtual Reality as an Environment for Case-Based Reasoning. In, Advances in Case-Based Reasoning. Cunningham, P. & Smyth, B. (Eds.), pp.448-459. Lecture Notes in Artificial Intelligence # 1488. Springer-Verlag, Berlin, 1998. ISBN 3-540-64990-5, 1998., <http://www.cs.auckland.ac.nz/~ian/>
- [72] Perera S., Watson I. Collaborative case-based estimating & design, Advances in Engineering Software. Vol. 29, No. 10 pp. 801-808. Elsevier Science, 1998. <http://www.cs.auckland.ac.nz/~ian/>
- [73] Doyle M., Ferrario M. A., Hayes C., Cunningham P., Smyth B., CBR Net :- Smart Technology over a Network., Department of Computer Science, Trinity College Dublin, 1998, <http://www.ai-cbr.org/hall.html>
- [74] Salustri A. F., Issues in Internet-Enabled CAE, CSME forum, Toronto, 1998.
- [75] Sun M., N. Bakis, and I. Watson, Intelligent agent based collaborative construction information network, International Journal of Construction Information Technology, Vol.7, No.2, pp35-46, 1999, <http://www.cs.auckland.ac.nz/~ian/>
- [76] Lu S. C-Y., Cai J., STARS: A Socio-Technical Framework for Integrating Design Knowledge over the Internet, IEEE september-october 2000., pp. 54-62.

-
- [77] Sun M., Bakis N., Watson I., Knowledge based collaborative construction information network, Proceedings of International Conference on Construction Information Technology 2000, 17-18 January 2000, Hong Kong, pp575-589, <http://www.cs.auckland.ac.nz/~ian/>
- [78] Gardingen, D., Watson, I., A Web Based Case-Based Reasoning System for HVAC Sales Support, Applications & Innovations in Expert Systems VI. Milne, R., Macintosh, A. & Bramer, M. (Eds.), pp. 11- 23. Springer, London. ISBN 1-85233-087-2, <http://www.cs.auckland.ac.nz/~ian/>
- [79] Hayes C., Cunningham P., Doyle M., Distributed CBR using XML, Department of Computer Science, Trinity College Dublin, 1998.
- [80] Hayes C., Cunningham P., Shaping a CBR view with XML, Department of Computer Science, Trinity College Dublin, 1999.
- [81] Watson I., Gardingen D., A Distributed Case-Based Reasoning Application for Engineering Sales Support, Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99), Vol. 1: pp. 600-605, Morgan Kaufmann Publishers Inc., ISBN 1-55860-613-0, 1999.

KRATKI ŽIVOTOPIS

Damir Salopek rođen je 9. svibnja 1969. godine u Ogulinu gdje je završio osnovnu i prva dva razreda srednje škole. Treći i četvrti razred završava u Zagrebu u tehničkoj školi «Ruder Bošković». Fakultet strojarstva i brodogradnje u Zagrebu upisao je 1987. godine. Kao apsolvent godinu dana radi u Srednjoj tehničkoj školi u Ogulinu kao nastavnik iz predmeta Matematika. Diplomirao je 1995. godine na usmjerenju «Strojarske konstrukcije».

Od 1996. godine radi u tvrtki Končar Energetski transformatori d.o.o. kao konstruktor na poslovima konstrukcije. U periodu od 1999. do 2001. godine aktivno sudjeluje u projektu Uvođenje CAD sustava u tvrtku Končar Energetski transformatori d.o.o. Od 1. siječnja 2002. godine radi u odjelu Razvoj i istraživanje na poslovima razvoja i održavanja CAE sustava tvrtke.

Poslijediplomski studij na Fakultetu strojarstva i brodogradnje upisuje 1997. godine na smjeru «Teorija konstrukcija».

Kao honorarni asistent sudjelovao je u izvođenju vježbi na strojarskom odjelu Tehničkog veleučilišta u Zagrebu iz kolegija Konstruiranje pomoću računala. Izabran je u naslovno zvanje stručnog suradnika na Tehničkom veleučilištu u Zagrebu.

Član je Hrvatskog društva za elemente strojeva i konstrukcije. Služi se engleskim jezikom.

SHORT BIOGRAPHY

Damir Salopek was born on May 9th 1969 in Ogulin, where he completed primary and part of secondary education. The other part of secondary education he completed in technical high school «Ruđer Bošković» in Zagreb. He started study in Faculty of Mechanical Engineering and Naval Architecture of the University of Zagreb in 1987. Before graduation he taught Math in technical high school in Ogulin. He graduated in 1995.

Since 1996 he has worked for Končar Power Transformers Ltd. He had worked as designer in Design Department until January 2002. From 1999 to 2001 he is involved in developing project Introducing CAD System in Končar Power Transformers Ltd. Since January 2002 he has worked in Research and Development Department on developing and maintaining jobs of the CAE system in Končar Power Transformers Ltd..

In 1997, he started postgraduated study on Faculty of Mechanical Engineering and Naval Architecture of the University of Zagreb.

He participated, as free-lance assistant, in teaching subject Computer aided design on Mechanical Engineering Department of the Polytechnic in Zagreb. He has been selected for a title of expert on Polytechnic in Zagreb

He is member of the Croatian society for machine elements and design. He has a good command of English.