

Sustav za konfiguriranje proizvoda modularne arhitekture

Pavlić, Davor

Scientific master's theses / Magistarski rad

2003

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:278008>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-05-15**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

**SUSTAV ZA KONFIGURIRANJE PROIZVODA
MODULARNE ARHITEKTURE**

MAGISTARSKI RAD

Mentor:

Dr. sc. Dorian Marjanović, izv. prof.

Davor Pavlić, dipl. inž.

Zagreb, 2003.

PODACI ZA BIBLIOGRAFSKU KARTICU

UDK: 62.002.2:681.3.06

Ključne riječi: familija proizvoda, platforma proizvoda, arhitektura proizvoda, varijante proizvoda, proces konfiguriranja, konfigurabilni proizvodi, konfiguracijski sustavi, konfiguracijsko znanje, zahtjevi naručitelja, STEP

Znanstveno područje: Tehničke znanosti

Znanstveno polje: Strojarsstvo

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: 49

Broj tablica: 8

Broj korištenih bibliografskih jedinica: 81

Datum obrane:

Povjerenstvo: Dr. sc. Izvor Grubišić, red. prof.
Dr. sc. Dorian Marjanović, izv. prof.
Dr. sc. Anton Jezernik, red. prof. Fakulteta za strojništvo, Maribor

Institucija u kojoj je rad pohranjen: Fakultet strojarstva i brodogradnje, Zagreb
Nacionalna i sveučilišna knjižnica, Zagreb



Zagreb, 24. rujan 2002.

Zadatak za magistarski rad

Kandidat: *Davor Pavlić, dipl. inž.*

Naslov zadatka: **Sustav za konfiguriranje proizvoda modularne arhitekture**

Sadržaj zadatka:

Razvoj konfigurabilnih proizvoda omogućuje prilagođavanje proizvoda individualnim zahtjevima naručitelja. U postojećim uvjetima distribuiranog razvoja proizvoda i zemljopisno razdvojenih naručitelja, evidentan je nedostatak programskih sustava koji će omogućiti prilagođavanje proizvoda zahtjevima naručitelja.

Konfigurabilnost proizvoda može se realizirati modularnom arhitekturom proizvoda. Modularnim pristupom osiguravaju se prednosti serijske proizvodnje i istovremeno omogućuje konstruiranje proizvoda prilagođenih individualnim zahtjevima naručitelja.

U radu je potrebno:

- izložiti pregled pojmova varijabilnosti proizvoda,
- odrediti pojam konfigurabilnog proizvoda i procesa konfiguriranja,
- istražiti teoretske modele za opisivanje strukture proizvoda,
- analizirati arhitekturu konfigurabilnog proizvoda,
- analizirati zahtjeve naručitelja koji utječu na varijabilnost proizvoda,
- istražiti načine zapisa znanja o konfiguriranju proizvoda,
- razraditi i predložiti metodologiju sustava za podršku procesu konfiguriranja,
- realizirati i testirati sustav u mrežnom okruženju.

Zadatak zadan: 09. 10. 2002.

Rad predan: 12. 5. 2003.

Mentor:

Voditelj smjera:

Predsjednik Odbora
za poslijediplomske studije:

Dr. sc. Dorian Marjanović, izv. prof.

Dr. sc. Dragutin Ščap, red. prof.

Dr. sc. Božo Vranješ, red. prof.

Zahvaljujem mentoru dr. sc. Dorianu Marjanoviću, izv. prof. na savjetima i korisnim raspravama tijekom izrade ovog rada.

Članovi povjerenstva, sa svojim primjedbama i savjetima, također su pomogli povećanju kvalitete ovog rada.

Djelatnicima Katedre za osnove konstruiranja zahvaljujem na pomoći i vremenu koje smo zajednički utrošili na brojne analize i rasprave koje su rezultirale većom kvalitetom prikazanog istraživanja.

Zahvaljujem se Draganu Debeliću, dipl. inž. i Edvinu Haviću, dipl. inž., djelatnicima poduzeća "KONČAR - GiM" iz Zagreba, koji su znatni dio svog vremena nesebično odvojili za primjenu teoretskih rezultata istraživanja u industriji.

Na kraju, posebno se zahvaljujem svojoj obitelji i roditeljima na razumijevanju i cjelokupnoj potpori u trenucima izrade ovog rada.

POPIS SLIKA	VIII
POPIS TABLICA.....	IX
PREDGOVOR	X
SAŽETAK RADA	XI
KLJUČNE RIJEČI	XI
SUMMARY	XII
KEYWORDS	XII
1. UVOD	2-1
1.1 POZADINA PROBLEMA.....	2-1
1.2 POLAZIŠTA I CILJEVI ISTRAŽIVANJA	2-2
2. PREGLED POJMOVA U PODRUČJU VARIJABILNOSTI PROIZVODA.....	2-1
2.1 FAMILIJA PROIZVODA	2-1
2.1.1 Pojam familija proizvoda	2-1
2.1.2 Utjecaj vremena promjene proizvoda i raznolikosti proizvoda	2-2
2.1.3 Utjecaj varijabilnosti proizvoda na troškove.....	2-4
2.1.3.1 <i>Princip opisivanja varijanata proizvoda vezanih uz zadovoljavanje tržišnih</i>	
<i>zahtjeva</i>	<i>2-4</i>
2.1.3.2 <i>Princip opisivanja varijanata proizvoda vezanih uz životne faze proizvoda</i>	<i>2-5</i>
2.1.4 Svojstva familije proizvoda.....	2-6
2.2 PLATFORMA PROIZVODA	2-6
2.2.1 Prednosti i nedostaci korištenja platformi proizvoda	2-7
2.3 ARHITEKTURA PROIZVODA.....	2-8
2.3.1 Vrste arhitekture proizvoda	2-9
2.3.1.1 <i>Integralna arhitektura proizvoda.....</i>	<i>2-10</i>
2.3.1.2 <i>Modularna arhitektura proizvoda.....</i>	<i>2-11</i>
2.3.1.2.1 <i>Vrste modularne arhitekture</i>	<i>2-12</i>
2.3.1.2.2 <i>Klasifikacija modula</i>	<i>2-14</i>
3. KONFIGURABILNI PROIZVODI.....	3-1
3.1 ZNAČAJKE KONFIGURABILNIH PROIZVODA	3-2
3.2 PROCES KONFIGURIRANJA	3-3
3.3 KONFIGURACIJSKI SUSTAVI	3-3
3.4 KONFIGURACIJSKO ZNANJE	3-4
3.4.1 Zapis konfiguracijskog znanja pomoću pravila	3-5
3.4.2 Zapis konfiguracijskog znanja pomoću ograničenja.....	3-7
4. PREGLED TEORETSKIH OSNOVA ISTRAŽIVANJA	4-1
4.1 TEORETSKE OSNOVE ISTRAŽIVANJA	4-1
4.2 TEORIJA TEHNIČKIH SUSTAVA.....	4-2
4.3 AKSIOMATSKA TEORIJA KONSTRUIRANJA	4-3
4.4 TEORIJA SVOJSTVA	4-5
4.5 RAZVIJANJE MODULA (ENG. <i>MODULAR FUNCTION DEPLOYMENT</i>)	4-7

5. ZAHTJEVI NARUČITELJA	5-1
5.1 ZAHTJEVI NARUČITELJA	5-1
5.2 UPRAVLJANJE ZAHTJEVIMA.....	5-2
5.3 LISTA ZAHTJEVA	5-3
6. INFORMACIJSKI MODEL.....	6-1
6.1 ISO 10303-STEP STANDARD.....	6-1
6.1.1 Struktura ISO 10303 - STEP standarda.....	6-2
6.1.2 EXPRESS.....	6-4
6.2 STRUKTURA INFORMACIJSKOG MODELA	6-7
6.2.1 Zahtjevi i lista zahtjeva naručitelja.....	6-8
6.2.2 Familija proizvoda i varijante proizvoda	6-11
6.2.3 Vrijednosti zahtjeva i modula	6-12
6.2.4 Vrste modula.....	6-16
6.2.5 Instance modula	6-18
6.2.6 Struktura konfiguracija	6-19
7. RAČUNALNA IMPLEMENTACIJA SUSTAVA.....	7-1
7.1 IZBOR RADNOG OKRUŽENJA	7-1
7.1.1 Klijent - poslužitelj arhitektura.....	7-1
7.1.2 Modeliranje razine poslovne logike	7-3
7.1.3 Modeliranje razine zapisa podataka	7-4
7.2 RAČUNALNI MODEL SUSTAVA.....	7-5
7.2.1 Struktura baze podataka.....	7-5
7.2.1.1 <i>Tablice za opisivanje zahtjeva i lista zahtjeva</i>	<i>7-5</i>
7.2.1.2 <i>Tablice za opisivanje vrijednosti i pridruživanja vrijednosti zahtjevima.....</i>	<i>7-6</i>
7.2.1.3 <i>Tablice za opisivanje varijante proizvoda i familije proizvoda</i>	<i>7-7</i>
7.2.1.4 <i>Tablice za opisivanje modula</i>	<i>7-8</i>
7.2.1.5 <i>Tablice za opisivanje instanci modula</i>	<i>7-9</i>
7.2.1.6 <i>Tablice za opisivanje konfiguracija</i>	<i>7-10</i>
7.2.2 Proces konfiguriranja.....	7-11
7.2.2.1 <i>Odabir instanci temeljnih modula prema zahtjevima.....</i>	<i>7-13</i>
7.2.2.2 <i>Odabir instanci izbornih modula prema zahtjevima.....</i>	<i>7-13</i>
7.2.2.3 <i>Odabir instanci dodatnih modula.....</i>	<i>7-14</i>
7.2.2.4 <i>Izrada konfiguracija od odabranih instanci</i>	<i>7-14</i>
7.2.2.5 <i>Provjera nekompatibilnosti između instanci pojedinih konfiguracija</i>	<i>7-16</i>
7.3 PRIKAZ PRIMJENE SUSTAVA	7-17
7.3.1 Asinkroni strojevi	7-17
7.3.2 Korisnička sučelja	7-22
8. ZAKLJUČAK	8-1
8.1 REZULTATI RADA	8-1
8.2 SMJEROVI DALJNJEG ISTRAŽIVANJA.....	8-3
9. LITERATURA	9-1
KRATKI ŽIVOTOPIS	XIII
SHORT BIOGRAPHY	XIV

Slika 1 - 1: Prikaz međusobnih utjecaja značajki proizvoda na kriterije vrednovanja proizvoda	2-3
Slika 1 - 2: Prikaz međusobnih utjecaja značajki proizvoda na kriterije vrednovanja proizvoda u prikazanom istraživanju	2-4
Slika 2 - 1: Usporedba raznolikosti proizvoda i brzine promjene proizvoda	2-3
Slika 2 - 2: Sličnost u životnim fazama proizvoda.....	2-5
Slika 2 - 3: Stvaranje varijante proizvoda u kasnijoj životnoj fazi proizvoda	2-5
Slika 2 - 4: Svojstva familije proizvoda.....	2-6
Slika 2 - 5: Vrste modularna arhitekture s utorima: a) izmjena komponenata b) dijeljenje komponenata	2-12
Slika 2 - 6: Sabirnička modularna arhitektura	2-13
Slika 2 - 7: Sekcijska modularna arhitektura	2-13
Slika 2 - 8: Povezanost modula i familije proizvoda	2-14
Slika 3 - 1: Serijska prilagodba	3-1
Slika 3 - 2: Zapis konfiguracijskog znanja pomoću pravila	3-6
Slika 4 - 1: Iz realnosti do računalnog modela	4-2
Slika 4 - 2: Općeniti model tehničkog procesa	4-3
Slika 4 - 3: Domene procesa konstruiranja po aksiomatskoj teoriji	4-4
Slika 4 - 4: Klase svojstva prema [18].....	4-6
Slika 4 - 5: Metoda Razvijanje modula.....	4-7
Slika 4 - 6: Matrica identificiranja modula	4-8
Slika 4 - 7: Djelovanje između modula	4-9
Slika 5 - 1: Funkcionalna i konstrukcijska svojstva, prema [61]	5-2
Slika 6 - 1: Struktura STEP standarda	6-3
Slika 6 - 2: Primjer zapisa u EXPRESS-u i EXPRESS-G.....	6-6
Slika 6 - 3: Struktura informacijskog modela.....	6-7
Slika 6 - 4: Dijagram entiteta za opis zahtjeva i liste zahtjeva naručitelja	6-8
Slika 6 - 5: Dijagram entiteta za opisivanje familije proizvoda i općenite varijante proizvoda	6-12
Slika 6 - 6: Dijagram entiteta za opisivanje vrijednosti	6-13
Slika 6 - 7: Dijagram entiteta za opisivanje vrsta modula	6-17
Slika 6 - 8: Dijagram entiteta za opisivanje instanci modula	6-18

Slika 6 - 9: Dijagram entiteta za opisivanje strukture varijante proizvoda	6-20
Slika 7 - 1: Troslojna klijent-poslužitelj arhitektura.....	7-2
Slika 7 - 2: Struktura tablica i relacija za opisivanje zahtjeva.....	7-5
Slika 7 - 3: Struktura tablica i relacija za opisivanje vrijednosti i pridruživanju vrijednosti s zahtjevima	7-7
Slika 7 - 4: Struktura tablica i relacija za opisivanje varijanata proizvoda i familije proizvoda	7-8
Slika 7 - 5: Struktura tablica i relacija za opisivanje modula	7-9
Slika 7 - 6: Struktura tablica i relacija za opisivanje instanci modula.....	7-10
Slika 7 - 7: Struktura tablica i relacija za opisivanje konfiguracija.....	7-11
Slika 7 - 8: Tijek procesa konfiguriranja.....	7-12
Slika 7 - 9: Postupak odabira instanci temeljnih modula prema zahtjevima	7-13
Slika 7 - 10: Postupak odabira instanci dodatnih modula	7-14
Slika 7 - 11: Postupak provjere nekompatibilnosti između instanci.....	7-16
Slika 7 - 12: Trofazni asinkroni klizno-kolutni motor	7-17
Slika 7 - 13: Hijerarhijska struktura osnovnih komponenata kod klizno-kolutnog asinkronog motora	7-18
Slika 7 - 14: Dispozicijski crtež asinkronog klizno-kolutnog motora s trajno spuštenim četkicama	7-19
Slika 7 - 15: Prikaz korisničkog sučelja za odabir predloška	7-22
Slika 7 - 16: Prikaz korisničkog sučelja za popunjavanje vrijednosti zahtjeva za odabranu listu zahtjeva.....	7-23
Slika 7 - 17: Prikaz korisničkog sučelja za odabrane instance temeljnih i izbornih modula	7-24
Slika 7 - 18: Prikaz korisničkog sučelja za odabrane instance temeljnih, izbornih i dodatnih modula	7-25
Slika 7 - 19: Prikaz korisničkog sučelja za dobivene konfiguracije	7-26
Slika 7 - 20: Prikaz korisničkog sučelja za odabranu varijantu proizvoda prilagođenu zadanim zahtjevima	7-27

POPIS TABLICA

Tablica 2 - 1: Odnos arhitekture proizvoda o funkcijama proizvoda	2-9
Tablica 2 - 2: Primjeri proizvoda s integralnom i modularnom arhitekturom	2-10
Tablica 3 - 1: Zapis konfiguracijskog znanja pomoću ograničenja	3-7
Tablica 4 - 1: Aksiomi u konstruiranju	4-5
Tablica 7 - 1: Konfiguracije instanci modula	7-15
Tablica 7 - 2: Broj ponavljanja instanci modula	7-15
Tablica 7 - 3: Podjela na module kod klizno-kolutnog asinkronog motora	7-20
Tablica 7 - 4: Popis vrijednosti instanci modula	7-21

Ovaj rad je dio istraživačkog rada u području unaprjeđivanja računalne podrške procesu konstruiranja varijanata proizvoda prilagođenih zahtjevima naručitelja, te je prikaz iskustva autora u primjeni i razvoju sustava za konfiguriranje proizvoda modularne arhitekture.

U uvjetima globalizacije tržišta, sve je veći broj proizvoda koji su prilagođeni zahtjevima naručitelja. Konstruiranje pojedinačnih proizvoda zahtjeva da se sve faze procesa konstruiranja ponavljaju svaki put kada dođe do promjene u listi zahtjeva. Takav pristup konstruiranju proizvoda više ne osigurava konkurentnost na tržištu. Osim toga, životni se vijek proizvoda neprestano skraćuje, proizvodi su sve kompleksniji te se i ukupno vrijeme potrebno da se proizvod pojavi na tržištu skraćuje. U provedenom istraživanju, proizvod modularne arhitekture korišten je kao osnova za razvoj varijanata proizvoda koje su prilagođene zahtjevima naručitelja.

Rad je dio cjelokupnog istraživanja unutar projekta 0120017 "Modeli i metode unaprjeđenja računalne podrške razvoja proizvoda", financiranog od strane Ministarstva znanosti i tehnologije Republike Hrvatske.

Poduzeća širom svijeta susreću se, zbog povećanja konkurentnosti na tržištu, s potrebom konstruiranja proizvoda prilagođenih zahtjevima individualnih naručitelja u što kraćem vremenskom periodu i uz što manje troškova. Poduzeća su pritom suočena s izazovom osiguravanja što veće varijantnosti proizvoda na tržištu, ali sa što manjom razlikom (u konstrukciji, proizvodnji, održavanju, ...) među samim varijantama. Zbog toga je razvoj varijanata proizvoda potrebno sagledati kao konfiguriranje novih proizvoda, od ranije definiranih modula. Istraživanje predstavljeno u ovom radu usmjereno je k razvoju sustava za konfiguriranje proizvoda modularne arhitekture u mrežnom okruženju.

U radu su objašnjeni pojmovi familije proizvoda, platforme proizvoda i arhitekture proizvoda, te je posebno naglašena modularna arhitektura proizvoda. Opisana su područja koja proces konfiguriranja obuhvaća, a to su konfigurabilni proizvodi, konfiguracijski sustavi, konfiguracijsko znanje i zahtjevi naručitelja. Ukratko su prikazana teoretska područja vezana uz istraživanje: teorija tehničkih sustava, aksiomska teorija konstruiranja te teorija svojstva. Prikazana je također i metodologija razvijanja modula (MDF).

Glavni cilj istraživanja u ovom radu je prijedlog informacijskog modela za podršku konfiguriranju proizvoda modularne arhitekture na temelju zahtjeva naručitelja. Predloženi informacijski model opisan je u skladu sa semantikom ISO 10303 (STEP) standarda i realiziran je korištenjem aplikacijskog protokola ISO 10303-214. Osnovni dijelovi informacijskog modela definirani su kao: zahtjevi i liste zahtjeva naručitelja, familija i varijante proizvoda, moduli i vrste modula, vrijednosti modula i zahtjeva, instance modula i struktura konfiguracija.

Na osnovu predloženog informacijskog modela realizirana je računalna implementacija sustava za konfiguriranje proizvoda modularne arhitekture. Računalna implementacija sustava testirana je i potvrđena na realnom primjeru asinkronog klizno-kolutnog motora s trajno spuštenim četkicama.

Ključne riječi:

familija proizvoda, platforma proizvoda, arhitektura proizvoda, varijante proizvoda, proces konfiguriranja, konfigurabilni proizvodi, konfiguracijski sustavi, konfiguracijsko znanje, zahtjevi naručitelja, STEP

Enterprises all over the world focus on the development of products adapted to the customer's demands in less time and with tight resources available in order to be competitive on the market. In the same time, enterprises confront the challenge of offering a wider range of product variants on the market but with less difference (in design, production, maintenance, etc.) between the variants. Therefore, the configuration of products from the previously defined modules is seen as a method of developing product variants. The subject of this thesis is the development of the configuration system for modular products.

This paper deals with the following issues: product family, product platform and product architecture. The modular product architecture is being discussed into more details. Additionally, the fields comprised into the configuration process are referred to as follows: configurable product, configurable system, configuration knowledge and customer requirements. A concise outline of existing design theories underlying this research is given, too. Also presented herewith is the methodology "Modular Function Deployment".

The main aim of this research is to propose an information model for configuration of modular products upon the customer's requirements. The proposed information model is developed in accordance with the semantics of the product modeling standard ISO 10303 (STEP) and is realized by using the application protocol ISO 10303-214. The basic model structures of the information model are the following: requirement model, product class model, association model, property model, instance model and structure model.

The realization of a computer-aided support is based on the proposed information model and tested on a real example of three-phase high-voltage slip-ring induction motors.

Keywords:

product family, product platform, product architecture, product variant, configuration process, configurable product, configurable system, configuration knowledge, customer requirements, STEP

1

Uvod

1.1 Pozadina problema

Početak 20. stoljeća Henry Ford je rekao: "Možete izabrati bilo koju boju automobila sve dok je ona crna". Od tada se tržište uvelike promijenilo. Niti jedan proizvod danas ne bi mogao opstati na tržištu da ima takvu mogućnost izbora. U današnjim uvjetima tržišne globalizacije evidentno je povećanje potreba za proizvodima koji su prilagođeni individualnim zahtjevima naručitelja. Primjer jednog takvog proizvoda je i *automobil*. Danas se ne kupuje automobil koji se nalazi na skladištu, već se naručuje automobil koji upravo odgovara zahtjevima određenog naručitelja. Zadovoljavanje zahtjeva naručitelja, danas je jedan od ključnih faktora uspjeha na tržištu. S obzirom da postoje potrošači, s različitim zahtjevima i kriterijima izbora, nužna je veća ponuda varijanti proizvoda kako bi se ispunili zahtjevi svih potrošača. Pritom su i poduzeća suočena s izazovom, da osiguraju što veću varijantnost proizvoda na tržištu, ali sa što manjom razlikom (u konstrukciji, proizvodnji, održavanju, ...) među samim varijantama.

Povećanje zahtjeva tržišta uzrokuje i određene posljedice u proizvodnji. Do sada nije dokazano da bilo koja vrsta proizvodnje može učiniti ukupni proizvodni proces učinkovitijim, nego što je to serijska proizvodnja. Zbog toga, može se zaključiti da s povećanjem zahtjeva određenog proizvoda, proizvodni proces poduzeća, koje proizvodi takav proizvod, treba imati značajke serijske proizvodnje. Poduzeća žele, s povećanjem zahtjeva tržišta, postići veći broj ponovnog korištenja, ali ne u smislu ponovnog korištenja proizvoda nakon što je postigao svoju osnovnu primjenu, već ponovnog korištenje komponenata, dokumentacije, procesa proizvodnje itd. Također, ponovno korištenje omogućuje druge pozitivne efekte, kao što su smanjenje količine dokumenata koji se pohranjuju, jednostavnije

administriranje dokumenata itd., što na kraju rezultira i ukupnim smanjenjem troškova.

Može se zaključiti kako postoje razlike među poduzećima, kojima je u interesu što više ponovnog korištenja i potrošačima, koji žele što veću mogućnost izbora. Jedan od načina prevladavanja tih razlika prikazan je u ovom radu.

1.2 Polazišta i ciljevi istraživanja

Zbog povećanja tržišnih zahtjeva, poduzeća širom svijeta susreću se s potrebom konstruiranja većeg broja varijanata proizvoda u što kraćem vremenskom periodu i uz što manje troškova. Zbog toga je potrebno, razvoj varijanata proizvoda sagledati kao konfiguriranje novih proizvoda, od ranije definiranih modula. Svaki modul može biti sastavljen od složene hijerarhijske strukture komponenata, ali u odnosu s drugim modulima mora imati niz jasno definiranih veza, relacija, koje određuju na koji se način modul može povezati s drugim modulima [1]. Proizvod, na temelju kojeg se konfiguriranjem konstruiraju varijante proizvoda, naziva se konfigurabilni proizvod. Konfigurabilni proizvod ima slijedeća svojstva [2]:

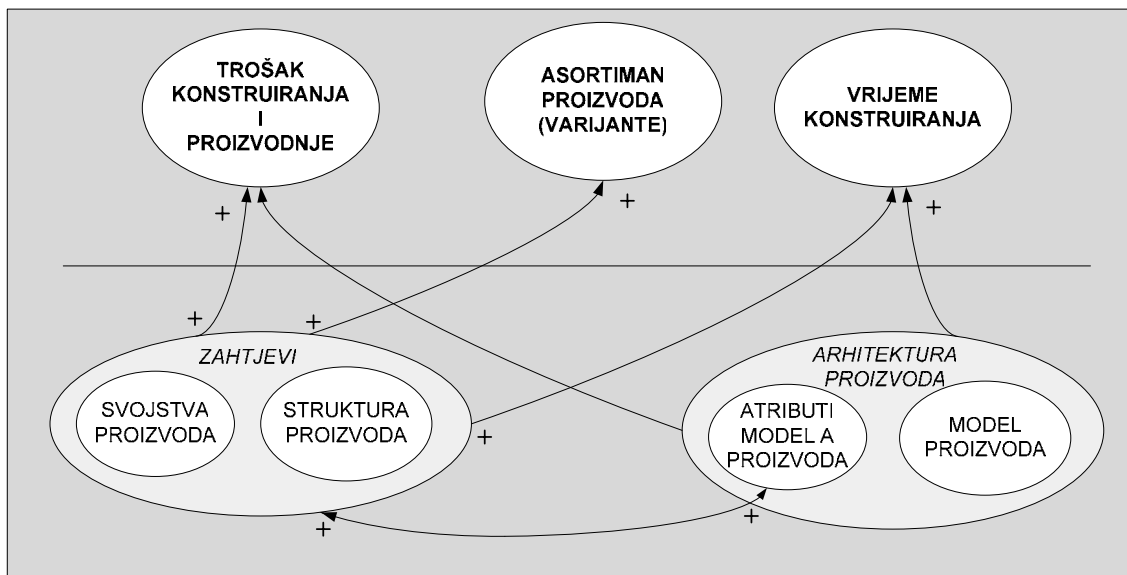
- arhitektura proizvoda je predefinirana, s ciljem zadovoljavanja šireg opsega različitih zahtjeva naručitelja,
- svaka varijanta konstruirana je prema individualnim zahtjevima naručitelja,
- svaka varijanta je određena kombinacijom gotovih modula.

Konfiguriranje proizvoda modularne arhitekture ima za cilj smanjenje:

- troškova razvoja varijante proizvoda (ponovnim korištenjem gotovih modula),
- troškova proizvodnje,
- administrativnih troškova i
- troškova održavanja (kroz standardizaciju).

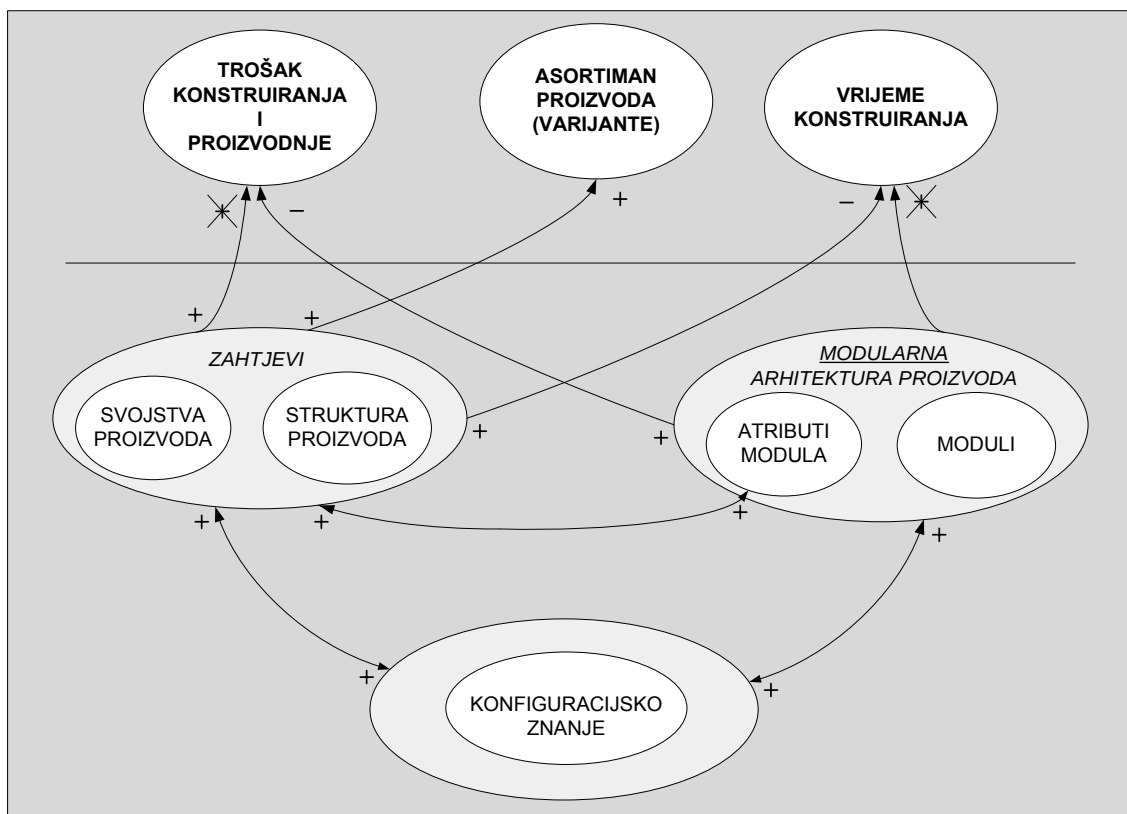
Pri razvoju konfigurabilnih proizvoda, potrebno je voditi računa o različitim značajkama koje utječu na razvoj varijanata proizvoda i na kriterije za vrednovanje tehničkih, ekonomskih ili ukupnih vrijednosti varijanata proizvoda. Primjer utjecaja dviju značajki (zahtjeva i arhitekture proizvoda) na kriterije vrednovanja proizvoda (asortiman proizvoda, vrijeme konstruiranja varijanata proizvoda te trošak konstruiranja i proizvodnje varijanata proizvoda) prikazan je na slici [Slika 1 - 1]. Ukoliko dođe do povećanja količine zahtjeva i razlike između njih, doći će i do povećanja broja atributa modela proizvoda koji opisuje arhitekturu proizvoda. S povećanjem zahtjeva ujedno će doći i do povećanja broja varijanata proizvoda, ali zbog povećanja broja atributa modela, povećati će se i vrijeme konstruiranja proizvoda, a samim time i trošak konstruiranja i proizvodnje. Na taj se način ne

ostvaruje smanjenje troškova, već je moguće i povećanje troškova što je u suprotnosti s očekivanjem s obzirom na povećanje asortimana proizvodnje.



Slika 1 - 1: Prikaz međusobnih utjecaja značajki proizvoda na kriterije vrednovanja proizvoda

U ovom radu, tijekom razvoja konfigurabilnog proizvoda razmatrati će se međusobni utjecaj slijedećih značajki: modularne arhitekture, zahtjeva naručitelja i konfiguracijskog znanja. Na slici [Slika 1 - 2] prikazani je međusobni utjecaj značajki proizvoda na kriterije vrednovanja proizvoda u prikazanom istraživanju. Prilikom povećanja količine i razlike u zahtjevima, povećati će se i broj modula, te s time i broj atributa modula kod modularne arhitekture proizvoda. Veza između zahtjeva i modula ostvarena je i preko konfiguracijskog znanja, pomoću kojeg je definirano kako pojedini zahtjev utječe na izbor modula. Na taj je način moguće, s povećanjem zahtjeva, ostvariti i povećanje broja varijanata pri čemu se vrijeme konstruiranja smanjuje. Vrijeme konstruiranja se smanjuje zbog toga što je omogućeno brže pronalaženje rješenja na postavljene zahtjeve koristeći konfiguracijsko znanje, a ujedno se i smanjuje mogućnost pogreške. Sa smanjenjem vremena konstruiranja, smanjuje se i trošak konstruiranja, dok smanjenje troškova proizvodnje proizlazi iz korištenja gotovih modula tj. serijske proizvodnje tih modula.



Slika 1 - 2: Prikaz međusobnih utjecaja značajki proizvoda na kriterije vrednovanja proizvoda u prikazanom istraživanju

Na temelju navedenog polazišta, definirani su ciljevi ovog istraživanja. Istraživanje predstavljeno u ovom radu usmjereno je na razvoj sustava za konfiguriranje varijanta proizvoda modularne arhitekture na temelju zahtjeva naručitelja. Realizacija takvog sustava zahtjeva definiranje informacijskog modela za podršku konfiguriranju proizvoda na temelju zahtjeva naručitelja, te računalnu implementaciju sustava u mrežnom okruženju.

2

Pregled pojmova u području varijabilnosti proizvoda

2.1 Familija proizvoda

2.1.1 Pojam familija proizvoda

Konstruiranje pojedinačnih proizvoda zahtjeva da se sve faze procesa konstruiranja ponavljaju svaki put kada dođe do promjene u listi zahtjeva. Takav pristup konstruiranju proizvoda više ne osigurava konkurentnost na tržištu. Definiranjem familije proizvoda, proces konstruiranja više nije usmjeren na pojedinačni proizvod, već na razvoj varijanata proizvoda. Varijante proizvoda razlikuju se prema [4]:

- dimenzijama,
- prilagodbi lokalnim tržištima,
- funkcionalnim značajkama i
- estetskoj izvedbi.

Varijante proizvoda koje se razlikuju prema dimenzijama, očituju se u različitim vrijednostima koje mogu poprimiti geometrijske veličine varijanata proizvoda. Osim razlike u vrijednostima geometrijskih veličina, varijante proizvoda razlikuju se i prema vrijednostima funkcionalnih značajki proizvoda, kao što su npr. snaga ili brzina vrtnje. S obzirom da su varijante proizvoda namijenjene prodaji na različitim tržištima, razlike između njih očituju se i u prilagodbi različitim zakonima, standardima ili jeziku tržišta kojemu su varijante proizvoda namijenjene. Također se varijante proizvoda mogu razlikovati prema estetskoj izvedbi tj. prema boji, obliku ili

površinama.

Pojam familije proizvoda nije jednoznačno određen u literaturi. Razlike su vidljive iz slijedećih definicija prema kojima se familija proizvoda opisuje kao:

- grupa srodnih proizvoda. Srodnost članova familije proizvoda tj. varijanata proizvoda ostvarena je strukturom tih proizvoda. Struktura proizvoda opisuje elemente (sklopove, funkcije, ...) od kojih je proizvod sačinjen. Varijante proizvoda imaju zajedničku strukturu do određene razine [5];
- grupa proizvoda koja se sastoji od zajedničkih značajki i funkcija. Značajke se općenito odnose na oblik proizvoda, dok se funkcije odnose na namjeru korištenje proizvoda [6];
- asortiman proizvoda koji su slični, a sličnost je definirana kroz različita gledišta. Na primjer, asortiman proizvoda može se smatrati familijom proizvoda ako ima istu ukupnu funkciju, ako ima slična svojstva ili je sačinjen od istih dijelova [7];
- grupa proizvoda koja je, zbog određene koristi, sačinjena od zajedničkih komponenata s ciljem postizanja asortimana varijanata proizvoda koji pokrivaju određeni tržišni segment. Takve familije proizvoda sačinjene su od određenog broja gradivnih elemenata, jedinica, modula, parametarski određenih dijelova ili samo dijelova [8].

U ovom radu, u skladu s prvom navedenom definicijom, familija proizvoda je grupa srodnih proizvoda pri čemu je srodnost ostvarena strukturom proizvoda a elementi strukture su moduli. Moduli predstavljaju grupu funkcionalno ili strukturalno neovisnih komponenata čije međudjelovanje je pretežno usmjereno unutar svakog modula, a djelovanje između modula svedeno je na minimum.

2.1.2 Odnos brzine promjene proizvoda i raznolikosti proizvoda

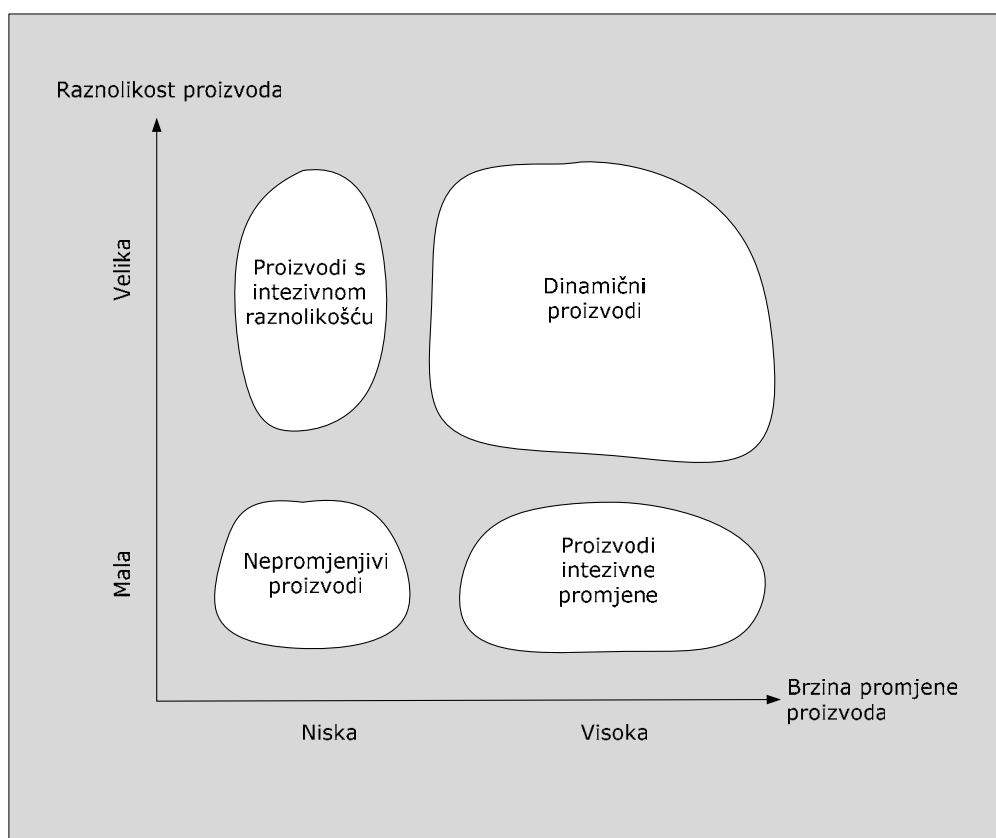
U prethodnom poglavlju, familija proizvoda se razmatrala samo sa stanovišta varijantnosti proizvoda tj. međusobne raznolikosti između varijanata proizvoda. Međutim, prilikom razmatranja ove teme, potrebno je sagledati i utjecaj vremena promjene proizvoda na tržištu [9]. Pod pojmom raznolikosti proizvoda podrazumijeva se broj različitih varijanata iste familije proizvoda koji egzistiraju u isto vrijeme na tržištu, a pod pojmom brzine promjene proizvoda podrazumijeva se učestalost zamijene ili modifikacije proizvoda na tržištu. Ova dva pojma važne su značajke familije proizvoda i sagledavajući ih kao dvije dimenzije u grafičkom prikazu moguće je razlikovati četiri grupe proizvoda [10]. Slika [Slika 2 - 1] prikazuje odnos raznolikosti proizvoda i brzine promjene proizvoda.

Prvu grupu proizvoda, prema [10], čine tzv. *nepromjenjivi proizvodi* tj. proizvodi koji imaju malu raznolikost i nisku brzinu promjene. Stvarni proizvodi s malom raznolikosti i niskom brzinom promjene skoro da danas i ne postoje, premda

postoje proizvodi kod kojih nema velike raznolikosti i dugog vremena promjene, npr. električna energija.

Drugu grupu proizvoda čine *proizvodi s intenzivnom raznolikošću* koji su opisani s velikom raznolikošću i s niskom brzinom promjene. To su najčešće relativno usavršeni proizvodi, kao npr. ručni alati, električne žarulje.

Treću grupu proizvoda čine *proizvodi intenzivne promjene*, a to su familije proizvoda kod kojih dolazi do intenzivnih promjena. Oni su opisani malom raznolikošću i visokom brzinom promjene. Svaki novi proizvod se u vrlo kratkom roku predstavlja tržištu i svaki put efektivno zamjenjuje svog prethodnika, ostavljajući samo jednu varijantu na tržištu u isto vrijeme. Primjer ovakvih proizvoda jesu žileti za brijanje, kod kojih je proizvođač usmjeren na kontinuirano predstavljanje novih žileta tržištu, pri čemu radije zamjenjuje stare varijante novima, nego da istovremeno nudi potrošačima veliku raznolikost proizvoda.



Slika 2 - 1: Odnos raznolikosti proizvoda i brzine promjene proizvoda

Četvrtu grupu proizvoda čine *dinamični proizvodi*. Dinamične familije proizvoda opisani su s velikom raznolikošću i visokom brzinom promjene. Primjere takvih proizvoda nalazimo općenito u tehnici, kao npr. u automobilske industriji, računalnoj industriji itd.

2.1.3 Utjecaj varijabilnosti proizvoda na troškove

Varijabilnost proizvoda omogućuju zadovoljavanje raznolikih zahtjeva tržišta, te se tako osigurava bolja podrška prodajnoj fazi životnog vijeka proizvoda. Međutim, povećavanje broja varijanata uzrokuje i povećanje dodanih troškova u svim ostalim životnim fazama proizvoda [11]. Takve dodatne troškove moguće je utvrditi i izračunati. Neki parametri pomoću kojih je moguće računati dodatne troškove jesu:

- broj dijelova, broj varijanata,
- količina materijala, broj procesa,
- broj odjela u poduzeću, osoba.

Analiziranjem utjecaja dodatnih troškova u cjelokupnom životnom vijeku proizvoda, moguće je izračunati troškove stvaranja novih varijanata. Budući je potrebno postići ravnotežu između stvaranja varijanata i dodatnih troškova, definirana su dva različita principa [11]. Prvi princip opisuje varijante proizvoda vezane uz zadovoljavanje tržišnih zahtjeva, a drugi princip opisuje varijante proizvoda vezane uz životne faze proizvoda.

2.1.3.1 Princip opisivanja varijanata proizvoda vezanih uz zadovoljavanje tržišnih zahtjeva

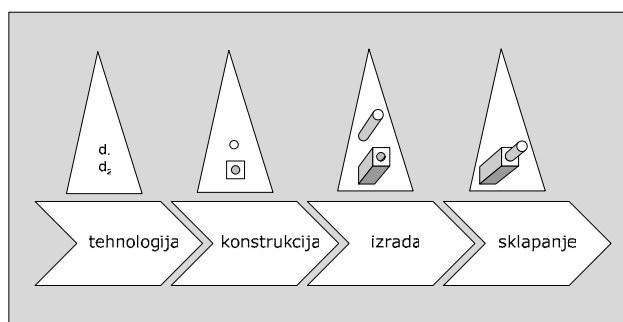
Za što uspješnije pronalaženje varijanata proizvoda koje zadovoljavaju tržišne zahtjeve, potrebno je definirati odgovarajuće smjernice:

- definiranje najšireg potrošačkog područja - različiti tržišni segmenti i pojedini potrošači nemaju iste zahtjeve. Npr., neki korisnici video kamere žele zaštitu video kamere od vode, dok drugi korisnici ne žele imati video kameru s tim svojstvom. Prosjek takvih potreba je video kamera sa zaštitom od prskanja. Ali tko želi takvu kameru? Iz ovog običnog primjera, može se uvidjeti da prosječni korisnik ne postoji. Zbog toga je važno odrediti najšire potrošačko područje i opisati njihove potrebe;
- definiranje razlikovne značajke - potrebno je definirati određenu značajku koja će dati posebno svojstvo proizvodu koje se može iskoristiti prilikom usporedbe s konkurentnim proizvodom. Određeni proizvodi imaju sposobnost da izraze status, drugi sposobnost pripadanja određenoj grupi, itd.;
- određivanje prioriteta - za dobivanje potrebnih varijanata ponekad je potrebno promijeniti samo neku dimenziju, a ponekad i veći dio strukture proizvoda. Informacija o važnosti promjene svake značajke omogućuje konstruktoru da razumije potrebe za kompromisom, optimizacijom, da izbjegne nepotrebne varijante i sl. To je posebno važno, ukoliko je potrebno smanjiti broj varijanata određenog područja.

2.1.3.2 Princip opisivanja varijanata proizvoda vezanih uz životne faze proizvoda

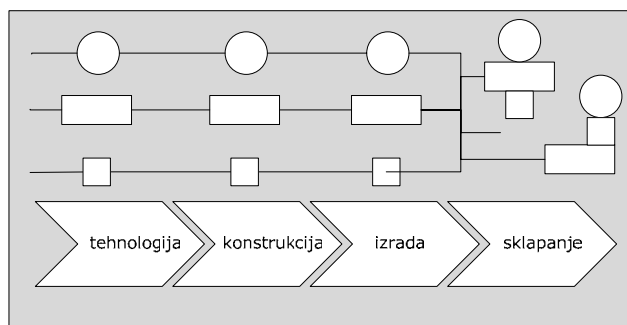
Dodatni troškovi uzrokovani stvaranjem varijanata mogu se smanjiti:

- ponovnim korištenjem znanja, tehnologija, komponenata ili modula, osnovni je princip za postizanje kraćeg vremena dolaska proizvoda na tržište s manjim konstrukcijskim greškama, troškovima i rizikom;
- stvaranjem sličnosti u što više životnih faza proizvoda [Slika 2 - 2]. To znači da će varijante biti slične gledano kroz te životne faze. Npr. zajednički provrti u različitim komponentama mogu učiniti te komponente sličnim u životnim fazama: izradi (bušenje, obrada), sklapanju, itd;



Slika 2 - 2: Sličnost u životnim fazama proizvoda

- stvaranjem varijanata u što kasnijoj životnoj fazi proizvoda - ovim principom smanjenja dodatnih troškova, faza sklapanja proizvoda [Slika 2 - 3] poprima veliko značenje i direktno ima utjecaj na skraćanje faze proizvodnje [12]. Ovaj pristup ima nekoliko prednosti, kao npr.:
 - smanjivanje vremena proizvodnje varijanata,
 - lakše planiranje proizvodnje,
 - smanjivanje cijena dobavljača zbog povećanja broja poluproizvoda i
 - smanjivanje cijene skladištenja poluproizvoda.

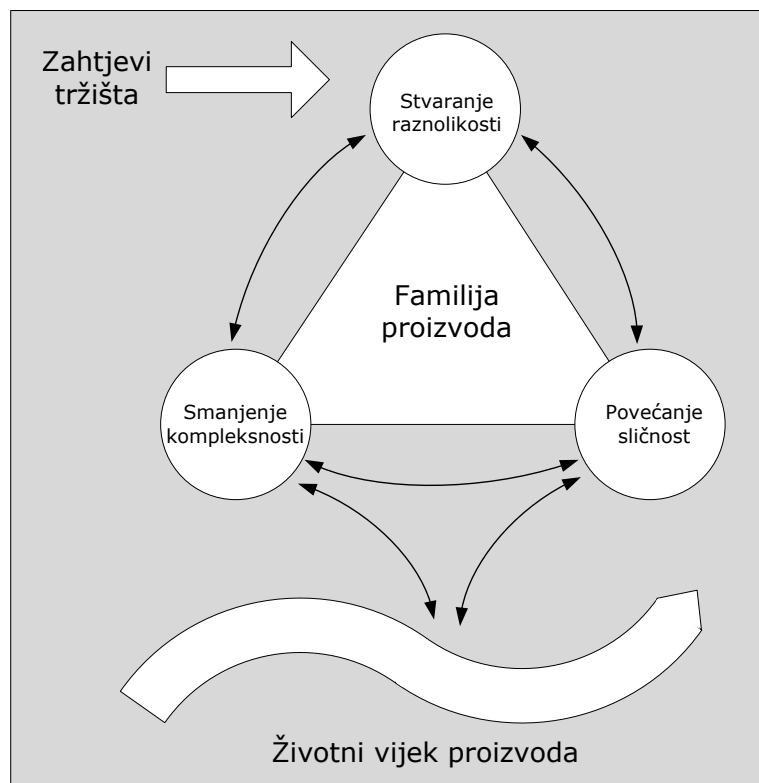


Slika 2 - 3: Stvaranje varijante proizvoda u kasnijoj životnoj fazi proizvoda

2.1.4 Svojstva familije proizvoda

Potaknuti razmatranjem u prethodnom poglavlju nameće se pitanje: "Što potiče stvaranje familije proizvoda?". Razvoj familije proizvoda jedna je od ključnih aktivnosti u procesu razvoja varijanti proizvoda i ima direktni utjecaj na cijenu razvoja varijanti proizvoda. Ritahuhta [22] je opisao tri svojstva familije proizvoda [Slika 2 - 4]:

- stvaranje raznolikosti - familija proizvoda treba osigurati raznolikost varijanata proizvoda zbog zadovoljavanja zahtjeva tržišta;
- smanjenje kompleksnosti - familija proizvoda treba omogućiti smanjenje kompleksnosti u svim aktivnostima koje su povezane s razvojem proizvoda, proizvodnjom i drugim aktivnostima životnog ciklusa proizvoda;
- povećanje sličnosti - familija proizvoda treba omogućiti povećanje stupnja sličnosti vezano uz proizvodni sustav i životni vijek proizvoda.



Slika 2 - 4: Svojstva familije proizvoda

2.2 Platforma proizvoda

Promatrano sa stajališta potrošača, familija proizvoda predstavlja varijante proizvoda koje određeno poduzeće nudi na tržištu zbog zadovoljavanja potreba

potrošača. Promatrano sa stajališta poduzeća, platforma proizvoda predstavlja smjernice koje omogućuju poduzeću da razvija varijante proizvoda kako bi zadovoljilo potrebe potrošača. S ciljem boljeg elaboriranja pojma platforme proizvoda, u postojećoj literaturi pojam platforme proizvoda definira se kao:

- skup pod-sustava i međusobnih veza razvijenih da tvore zajedničku strukturu na temelju koje se efikasno razvijaju i proizvode varijante proizvoda [8], [14],
- skup pravila o elementima i odnosima od kojih se sačinjavaju varijante proizvoda [5],
- skup vrijednosti koje su zajedničke unutar grupe proizvoda [79], [13].
Te vrijednosti grupirane su u četiri kategorije:
 - komponente - dijelovi proizvoda, strojevi potrebni za njihovu izradu te računalni programi,
 - procesi - korišteni za izradu ili sklapanje komponenata u proizvod te pridruženi proizvodni procesi,
 - znanje - konstrukcijsko znanje, tehnologija, matematički modeli i metode za testiranje,
 - osobe i odnosi - timovi, odnosi među članovima timova i odnosi među timovima.

Iz ovih definicija proizlazi da platforma proizvoda ne predstavlja samo proizvod, već predstavlja i skupinu smjernica na temelju kojih se brže i efikasnije razvijaju varijante proizvoda. Korištenjem platformi proizvoda, poduzeće je načelno orijentirano na gledanje u budućnost i temelji se na razvoju ne samo trenutnog proizvoda nego i varijanata proizvoda koji će odgovoriti na zahtjeve tržišta.

Potrebe tržišta se mijenjaju tijekom vremena, a mijenja se i tehnologija koja se koristi pri razvoju proizvoda. Obje promjene zahtijevaju promjene u platformi proizvoda. U dobro definiranoj platformi proizvoda, dinamika promjene platforme proizvoda je manja u odnosu na dinamiku razvoja novih varijanata proizvoda [5].

U ovom radu, platforma proizvoda obuhvaća smjernice za proces konfiguriranja varijanata proizvoda. Te smjernice preuzete su iz poduzeća čiji proizvod je korišten za testiranje razvijenog računalnog sustava.

2.2.1 Prednosti i nedostaci korištenja platformi proizvoda

Razvoj proizvoda na temelju platforme proizvoda motiviran je slijedećim potencijalnim prednostima [16]:

- povećanju fleksibilnosti i smanjenju vremena do izlaska proizvoda na tržište - korištenjem konstrukcijskih smjernica smanjuje se vrijeme potrebno za razvoj i proizvodnju proizvoda, a ujedno se i ostvaruje mogućnost naknadne reakcije na tržišne promjene;

- većoj kvaliteti - više pažnje se posvećuje razvoju komponente koja se više puta koristi, nego razvoju komponente korištene samo jedanput;
- smanjenju rizika novog proizvoda - zamjenom pojedinih dijelova platforme i zasnivanje svake nove generacije proizvoda na kombinaciji dokazanih (korištenih) i novih elemenata, dovodi do smanjenja rizika od stvaranja pogrešaka kako u konstrukciji tako i u proizvodnji samog proizvoda.

Pored navedenih prednosti, prilikom razvoj proizvoda na temelju platforme proizvoda manifestiraju se slijedeći nedostaci [16]:

- ograničenje prema radikalnim promjenama - prednosti platforme nije moguće iskoristiti prilikom razvoja platforme i prve varijante familije proizvoda. Prednost platforme dolazi do izražaja razvojem slijedećih varijanata proizvoda. To može biti uzrok i opiranju radikalnim promjenama koje se moraju napraviti zbog npr. promjene tehnologije;
- smanjenje performansi - potreba da se elementi platforme mogu višestruko koristiti dovodi do smanjenja performanse ili veće cijene pojedinih elemenata proizvoda.

Ovi nedostaci ukazuju na činjenicu da razvoj platforme nije samo razvoj što je moguće više općenite platforme, već više traženje kompromisa između potrebe da se platforma može višestruko koristiti i nedostataka koji takav pristup ima za sami proizvod.

2.3 Arhitektura proizvoda

Često se pojmovi arhitektura proizvoda i struktura proizvoda poistovjećuju. Jedan od razloga je u tome što ovisno o zemljopisnim područjima ovise i interpretacije tih pojmova. Tako se npr. u Americi češće koristi pojam arhitektura proizvoda pod kojim se podrazumijeva struktura proizvoda, dok se u Evropi ta dva pojma razlikuju. U nastavku rada objasniti će se razlika između strukture i arhitekture proizvoda.

Tichem [17] definira dva tipa strukture: (I) tip strukture koji opisuje vezu između nadređenih i podređenih elemenata i naziva se hijerarhijska struktura i (II) tip strukture koji opisuje kako su elementi povezani na istom hijerarhijskom nivou. Pojam "hijerarhijska struktura" dolazi iz Teorije tehničkih sustava [18], te se definira kao značajka koja opisuje ukupnost sustava, odnosno elemente koji čine sustav i relacije između njih. Ovisno o tome koja se domena proizvoda promatra (domena procesa, domena funkcija, domena organa, domena dijelova [19]) možemo razmatrati različite tipove hijerarhijskih struktura, kao npr. hijerarhijska struktura funkcija, hijerarhijska struktura komponenti, itd. Pod pojmom struktura proizvoda često se podrazumijeva hijerarhijska struktura komponenti proizvoda. Može se zaključiti kako je hijerarhijska struktura komponenti jedno od najvažnijih svojstava

proizvoda. Hijerarhijska struktura komponenti je i poveznica većeg broja nositelja informacija koje se razmjenjuju u procesu razvoja proizvoda. To je ključna veza sa slijedećom životnom fazom proizvoda - proizvodnjom [20].

Arhitektura proizvoda podrazumijeva skup elemenata i njihovih veza koje definiraju proizvod [8], [21]. Elemente arhitekture proizvoda čine gradivne jedinice i pravila koja definiraju načine njihovih povezivanja. Osnovna razlika između arhitekture i strukture proizvoda je u tome što, govoreći o arhitekturi proizvoda, govorimo o fiktivnom proizvodu i to u trenutku kada želimo restrukturirati proizvodni program zbog utjecaja koji su proizašli iz okoline (tržišta, poduzeća ...). Rezultat restrukturiranja proizvodnog programa je stvarni proizvod koji je nastao na temelju prihvaćene arhitekture proizvoda, a sastoji se od stvarnih (fizičkih) komponenata. Kada govorimo o međusobnom odnosu između stvarnih komponenata ili od kojih komponenata se sastoji stvarni proizvod, koristi se pojam struktura proizvoda a ne arhitektura proizvoda.

2.3.1 Vrste arhitekture proizvoda

Ovisno o pridruživanju funkcija komponentama, Ulrich [24] razlikuje:

- integralnu i
- modularnu arhitekturu proizvoda.

Kod modularne arhitekture je bitno da su jedna ili više određenih funkcija sadržane u jednom modulu, dok je kod integralne arhitekture jedna funkcija sadržana u više elemenata. Takva podjela prikazana je u tablici [Tablica 2 - 1] [23]. Korištenje modularne ili integralne arhitekture proizvoda ovisi o različitim zahtjevima koji su postavljeni razvoju familije proizvoda. Za ostvarivanje fleksibilnosti i raznolikosti proizvoda upotrebljava se modularna arhitektura, dok se za postizanje stabilnosti i optimizacije proizvoda upotrebljava integralna arhitektura [25].

Tablica 2 - 1: Odnos arhitekture proizvoda o funkcijama proizvoda

$1 : 1$	Jedna funkcija sadržana je u jednom elementu	<i>Modularna arhitektura</i>
$1 : N$	Jedna funkcija sadržana je u više elemenata	<i>Integralna arhitektura</i>
$N : 1$	Nekoliko funkcija sadržano je u jednom elementu	<i>Modularna arhitektura</i>
$N : M$	Nekoliko funkcija sadržano je u više elemenata	<i>Integralna arhitektura</i>

U tablici [Tablica 2 - 2] prikazani su primjeri proizvoda integralne i modularne arhitekture proizvoda.

Tablica 2 - 2: Primjeri proizvoda s integralnom i modularnom arhitekturom

Vrsta	Proizvodi
<i>Integralna arhitektura</i>	
<i>Modularna arhitektura</i>	

2.3.1.1 Integralna arhitektura proizvoda

U prethodnom poglavlju, navedeno je kako je kod integralne arhitekture proizvoda većina funkcija sadržana u više fizičkih elemenata [25]. Nije namjera da se pojedine funkcije izoliraju u pojedine fizičke elemente, već da pojedini elementi dijele funkcije proizvoda. Posljedica ovakvog pristupa je da promjene na jednom elementu utječu na promjene drugih, ako ne i svih fizičkih elemenata. Kao primjer proizvoda s integralnom arhitekturom, u tablici [Tablica 2 - 2] navedeni su sljedeći proizvodi: video i audio kazete, ručni alati i noževi. Karakteristično je za ovakve proizvode da su izrađeni u serijskoj proizvodnji. Zbog toga njihovu cijenu nije potrebno smanjivati na način da više različitih proizvoda sadrži iste komponente kako bi se postigla konkurentnost na tržištu. Zbog serijske proizvodnje takvih proizvoda, nastoje se minimizirati svi troškovi vezani uz proces sklapanja takvih proizvoda, čime se automatski povećava kompleksnost komponenata.

Integralna arhitektura proizvoda sačinjena je od elemenata s [26]:

- nepromjenjivim fizičkim komponentama i
- jednom ili više promjenjivom fizičkom komponentom.

Pomoću integralne arhitekture s promjenjivim fizičkim komponentama dobivaju se varijante proizvoda različitih geometrijskih izvedbi. Takvi proizvodi definirani su [27] kao proizvodi širokog spektra upotrebe koji zadovoljavaju istu funkciju, koji su zasnovani na istom principu, koji su napravljeni u različitim veličinama i proizvedeni su istim proizvodnim procesima. Osnova za promjenjivost geometrijskih veličina je korištenje zakona sličnosti prilikom razvoja varijanata. Za strojske proizvode, ti zakoni sličnosti ne moraju se zasnivati samo na geometrijskoj proporciji, već mogu biti zasnovani na temperaturnoj, elastičnoj i drugim sličnostima [28].

2.3.1.2 *Modularna arhitektura proizvoda*

Iz tablice [Tablica 2 - 1] vidljivo je da modularna arhitektura proizvoda podrazumijeva povezivanje jedne ili više funkcija u funkcijskoj strukturi s jednim elementom u hijerarhijskoj strukturi komponenata proizvoda. Korištenjem modularne arhitekture, proizvod je podijeljen u module koji se mogu mijenjati i kojima je moguće promijeniti geometrijsku veličinu ili funkcije s ciljem dobivanja različitih varijanata proizvoda. Moduli se najčešće opisuju kao grupa funkcionalno ili strukturalno neovisnih komponenata čije međudjelovanje je pretežno usmjereno unutar svakog modula, a djelovanje između modula svedeno je na minimum. Premda funkcija ili struktura pojedinog modula mora biti povezana s funkcijom/strukturom cijelog proizvoda, moduli ne mogu biti potpuno neovisni o drugim modulima i moraju se definirati zajedno sa proizvodom kojemu pripadaju.

Korištenje modularne arhitekture u konstruiranju i proizvodnji proizvoda nosi mnoge prednosti kako za proizvođače tako i za naručitelje, a to su [29]:

- snižavanje cijena varijanti proizvoda - s obzirom da se moduli proizvode u većoj količini nego varijante proizvoda, cijene modula a ujedno i varijanata sačinjenih od modula su manje nego cijene proizvoda koji nisu sačinjeni od modula;
- povećanje mogućnosti zamjene modula - s obzirom da su sučelja modula točno određena, promjena jednog modula moguća je neovisno od ostalih modula;
- povećanju varijantnosti proizvoda - povećanje broja varijanta ostvareno je različitim kombinacijama modula;
- brža isporuka proizvoda - korištenjem gotovih modula u proizvodnji skraćuje se vrijeme potrebno za izradu proizvoda, a time i vrijeme potrebno da se proizvod pojavi na tržištu;
- jednostavnost održavanja i rasklapanja - s obzirom da je proizvod sastavljen od modula, potrebno je zamijeniti samo pojedine module kada je potreban popravak. Iz istog razloga, nadogradnja, održavanje i rasklapanje proizvoda na kraju životnog ciklusa proizvoda, bitno su jednostavniji.

Korištenje standardiziranih sučelja između modula podrazumijeva definiranje funkcionalnih i "susjedskih" veza između modula koja ostaju nepromijenjena u procesu razvoja proizvoda [30]. Glavni smjerovi istraživanja modularne arhitekture proučavaju [29]:

- identificiranje modula,
- konstruiranje modula i
- konstruiranje s modulima.

U ovom radu, istraživanje je usmjereno na područje konstruiranja s modulima

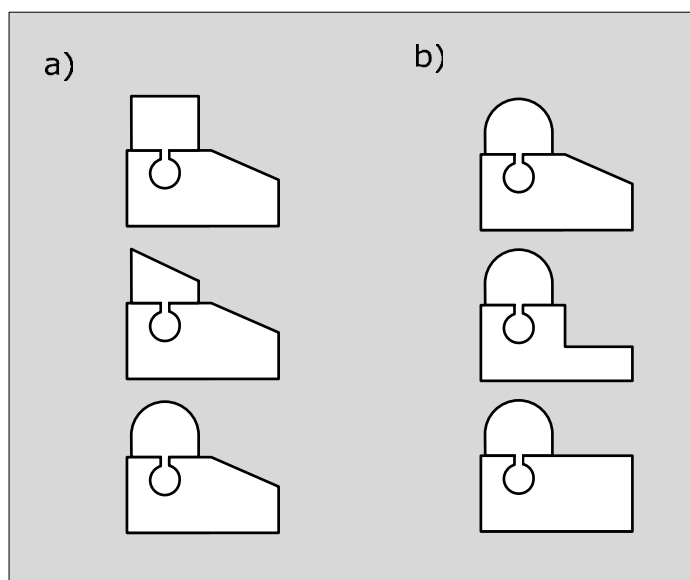
što uključuje odabir kombinacija modula koji zadovoljavaju zadane zahtjeve naručitelja.

2.3.1.2.1 Vrste modularne arhitekture

Mnogi autori opisuju različite vrste modularne arhitekture čime žele razjasniti njeno značenje. U osnovi, podjela modularne arhitekture napravljena je s obzirom na načine međusobne izmjene modula. Ulrich [31] je definirao tri vrste modularne arhitekture:

- 1) arhitektura s utorima (eng. *slot*),
- 2) sabirnička arhitektura (eng. *bus*) i
- 3) sekcijska arhitektura (eng. *sectional*).

Modularna arhitektura s utorima opisana je kao arhitektura kod koje se jedan temeljni modul može povezati s različitim modulima, a s ciljem izvršavanja višestrukih zadataka [Slika 2 - 5].

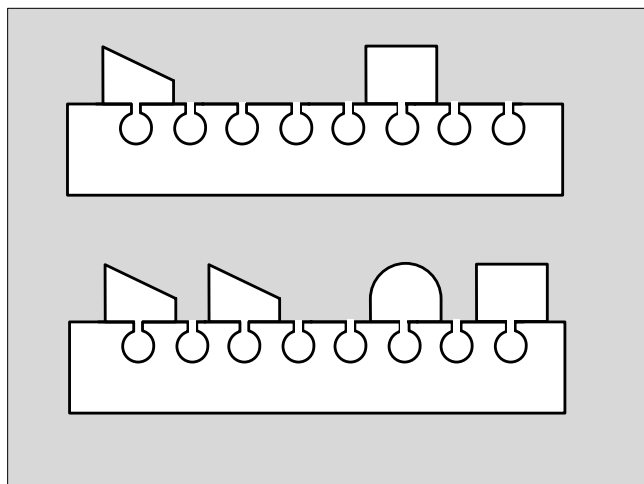


Slika 2 - 5: Vrste modularna arhitekture s utorima:
a) izmjena komponenata b) dijeljenje komponenata

Ulrich i Tung [31] spominju arhitekture *izmjene* i *dijeljenja* komponenata, koje predstavljaju dvije vrste modularne arhitekture s utorima. Arhitektura *izmjene* komponenata opisuje arhitekturu gdje se dva ili više modula mogu povezati preko istih sučelja s istim temeljnim modulom, kreirajući pri tom različite varijante proizvoda koje pripadaju istoj familiji proizvoda. Primjer arhitekture izmjene komponenata koristi se npr. u prijenosnim računalima, kod kojih je često moguć odabir između različitih vrsta baterija, tipkovnica, različitih kapaciteta diskova i sl. Arhitektura *dijeljenja* komponenti opisuje arhitekturu kod koje se isti modul koristi u cijeloj familiji proizvoda ili u različitim familijama proizvoda, gdje je također povezivanje modula ostvareno preko istih sučelja. Primjer modularnosti dijeljenja

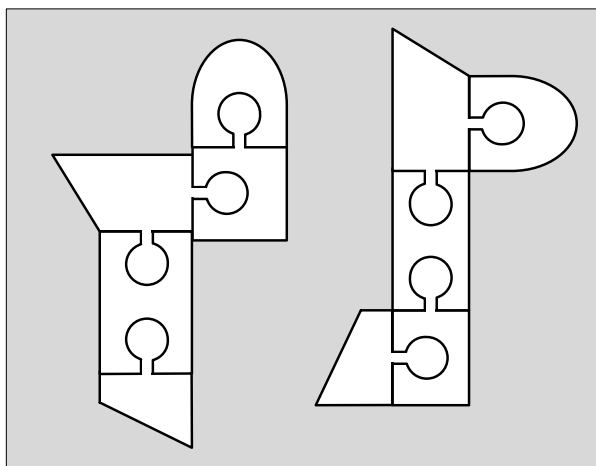
komponente je npr. prilikom korištenja istog punjača baterija za sve mobilne telefone istog proizvođača.

Sabirnička modularna arhitektura opisana je kao arhitektura kod koje se temeljni modul može spajati s više istih ili različitih modula, slika [Slika 2 - 6], čime se osigurava veća fleksibilnost. Važna razlika između sabirničke i arhitekture s utorima je u tome što sabirnička arhitekture dozvoljava mijenjanje pozicije i broja modula. Primjer sabirničke modularne arhitekture jesu memorijski utori na matičnoj ploči računala.



Slika 2 - 6: Sabirnička modularna arhitektura

Sekcijska modularna arhitektura [Slika 2 - 7] opisana je kao arhitektura kod koje je smještaj modula u proizvodu slobodan, što znači da se moduli mogu različito smještati, ali uz uvjet da su međusobno povezani istim sučeljima. Različiti smještaj modula omogućuje i ostvarivanje različitih ukupnih funkcija proizvoda. Za ostvarivanje veće mogućnosti kombiniranja modula, moduli mogu imati više različitih sučelja. Primjeri sekcijske modularne arhitekture su lego kocke ili uredski namještaj.



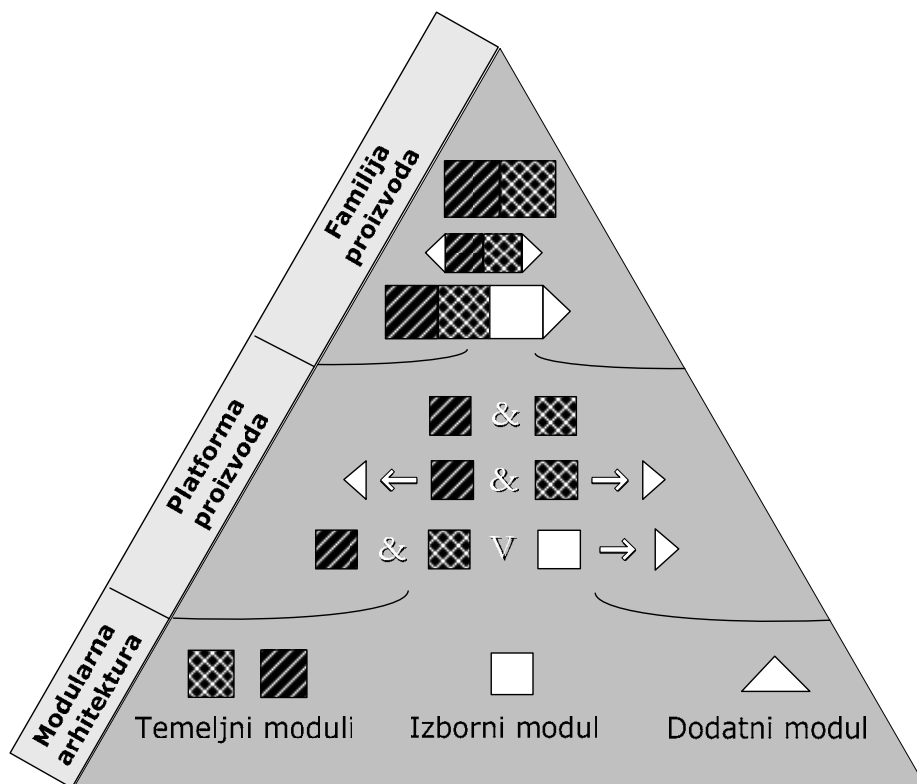
Slika 2 - 7: Sekcijska modularna arhitektura

2.3.1.2.2 Klasifikacija modula

Pojedini moduli mogu biti sastavni dio svih varijanata proizvoda ili se mogu pojavljivati samo u nekim varijantama određene familije proizvoda. Pahl i Beitz [28] vrše klasifikaciju modula, na osnovu njihove uloge u razlikovanju varijanata proizvoda unutar familije proizvoda, na:

- temeljne module - moduli koji su zajednički u svim varijantama proizvoda,
- izborne module - moduli korišteni za stvaranje raznolikosti između varijanata proizvoda i određeni su zahtjevima naručitelja,
- dodatne module - često se nazivaju i moduli za spajanje zbog toga što njihovo postojanje uvjetuju drugi moduli, kako bi u potpunosti mogli ispuniti svoju funkciju,
- specijalne module - prilagođeni su za potrebe individualnog naručitelja.

U ovom istraživanju [Slika 2 - 8], modularna arhitektura određena je s tri vrste modula (temeljni, izborni i dodatni). Struktura svake varijante proizvoda određena je međusobnom ovisnošću modula i zahtjeva, a komponente varijante proizvoda određene su instancom svakog modula.

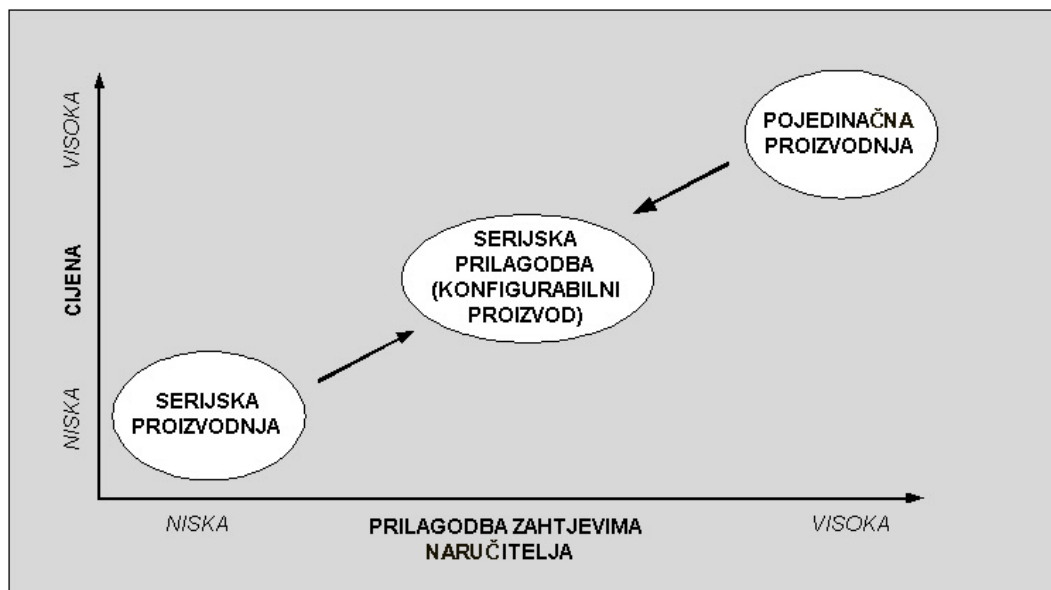


Slika 2 - 8: Povezanost modularne arhitekture, platforme proizvoda i familije proizvoda

3

Konfigurabilni proizvodi

U sadašnjim konkurentnim poduzećima na tržištu, područje razvoja jesu proizvodi koji zadovoljavaju zahtjeve individualnih naručitelja. Takav razvoj proizlazi iz činjenice da je životni vijek proizvoda sve kraći, proizvodi su sve kompleksniji i povećava se broj varijanata proizvoda. Ukupno vrijeme potrebno da se proizvod pojavi na tržištu, sve je kraće. Osim toga, povećan je i pritisak potrošača ali i konkurencije, pa su na tržištu prisutne varijante proizvoda prilagođene pojedinim zahtjevima potrošača.



Slika 3 - 1: Serijska prilagodba

Jedan od glavnih smjerova rješavanja prilagodbe proizvoda zahtjevima naručitelja je

razvoj konfigurabilnih proizvoda. Takvi proizvodi obuhvaćaju spektar različitih, ali usko povezanih varijanata proizvoda koji zadovoljavaju potrebe individualnih naručitelja [32]. Cilj koji se želi postići razvojem konfigurabilnih proizvoda je omogućiti jednostavnije upravljanje i smanjivanje trajanja slijedećih segmenata životnog vijeka proizvoda: narudžbe, prodaje, konstruiranja, proizvodnje, isporuke i održavanja varijanata proizvoda. Razvojem konfigurabilnih proizvoda, počela se razvijati i nova vrsta proizvodnje koja kombinira sve prednosti serijske, ali i pojedinačne proizvodnje, a naziva se serijska prilagodba [33], [34]. Slika [Slika 3 - 1] prikazuje odnos cijene i prilagodbe zahtjeva naručitelja, kao i tendenciju prelaska poduzeća s navedenih proizvodnja na serijsku prilagodbu.

Poduzeća, čiji je proizvodni program obuhvaćao pojedinačnu proizvodnju, prelaze na serijsku prilagodbu kako bi smanjili vrijeme isporuke proizvoda na tržište i kako bi mogli ponovno koristiti znanje stečeno u razvojnem procesu proizvoda. Prilikom prelaska s pojedinačne proizvodnje na serijsku prilagodbu, nailazi se na probleme vezane uz standardizaciju i sistematizaciju proizvoda i proizvodnje. Prelazak s jedne vrste proizvodnje na drugi opravdan je samo ako je broj varijanata proizvoda dovoljno veliki, što direktno ovisi o situaciji na tržištu te o grani industrije u kojoj se proizvod nalazi. Prema istraživanjima provedenima na tržištima [35], potrošači su spremni platiti 10 do 15 % veću cijenu od cijene standardnog proizvoda, s ciljem kupovine proizvoda prilagođenog njihovim zahtjevima. Općenito rečeno, pojedinačna proizvodnja namijenjena je potrošačima koji imaju posebne zahtjeve, ali i koji su spremni platiti te zahtjeve.

U drugom slučaju, poduzeća koja su imala serijsku proizvodnju prelaze na serijsku prilagodbu. Ovo se događa kada postoji potreba povećanja konkurentnosti na tržištu na način da se tržištu ponudi veća mogućnost ponude (odabira) varijanti proizvoda. Pritom se ne smije ugroziti niska cijena na koju su potrošači takvih proizvoda navikli, te veliki broj komada proizvoda s kojim se postiže niža cijena proizvoda.

3.1 Značajke konfigurabilnih proizvoda

Kao arhitektura konfigurabilnih proizvoda najčešće se koristi modularna arhitektura proizvoda, jer se time osiguravaju mogućnosti konfiguriranja i prilagodbe varijanata proizvoda. Konfigurabilni proizvodi imaju slijedeće značajke [32]:

- svaka varijanta konstruirana je prema individualnim zahtjevima naručitelja,
- proizvod je pre-definiran, s ciljem zadovoljavanja šireg opsega različitih zahtjeva naručitelja,
- proizvod ima modularnu arhitekturu,
- svaka varijanta je određena kombinacijom gotovih komponenata (modula).

Konfigurabilni proizvodi imaju slijedeće prednosti [32], [36]:

- mogućnost zadovoljavanja većeg broja zahtjeva naručitelja,
- varijante proizvoda temeljene su na sistematiziranom znanju o konfiguriranju,
- manje vrijeme potrebno je za isporuku varijanata proizvoda na tržište i
- povećana kontrola proizvodnje, npr. veći broj varijanata s manjim brojem komponenata.

3.2 Proces konfiguriranja

Proces konfiguriranja proizvoda je aktivnost kojom se određuje, do određenog nivoa, struktura varijante proizvoda prilagođene zahtjevima naručitelja, unutar ograničenja koja su postavljena arhitekturom proizvoda. Početak procesa konfiguriranja je lista zahtjeva, koja formalno predstavlja zahtjeve naručitelja.

Rezultat procesa konfiguriranja je *konfiguracija* kojom je opisana varijanta proizvoda koja će se proizvoditi za određenu narudžbu. Često rezultat procesa konfiguriranja nije samo jedna konfiguracija, već ih može biti i više. Isto tako, proces konfiguriranja je iterativni postupak kod kojeg rezultat ne mora u potpunosti zadovoljiti zadane zahtjeve. U tom slučaju je potrebno ponoviti i po nekoliko puta proces konfiguriranja, sve dok se ne dobije konfiguracija koja u potpunosti zadovoljava zadane zahtjeve. Ovisno o zahtjevima koji su postavljeni, proces konfiguriranja može ali i ne mora pronaći odgovarajuću konfiguraciju tj. postoji slučaj kada niti jedna konfiguracija ne zadovoljava zadane uvijete.

Zadatak procesa konfiguriranja [38] je, da iz zadane skupine elemenata proizvoda odredi varijantu proizvoda, koristeći poznata pravila i ograničenja između odabranih elemenata koji zadovoljavaju tražene zahtjeve. Zadaće koje proces konfiguriranja treba izvršiti u ovom istraživanju, jesu:

- određivanje povezanosti između modula i zahtjeva,
- određivanje povezanosti između modula,
- odabir modula prema listi zahtjeva,
- određivanje instanci modula prema listi zahtjeva,
- provjera nekompatibilnosti instanci modula i
- provjera potpunosti rezultata konfiguriranja.

3.3 Konfiguracijski sustavi

Konfiguracijski sustavi jesu informacijski sustavi koji se koriste u procesu konfiguriranja varijanata proizvoda na temelju zahtjeva naručitelja. Postoji nekoliko

klasifikacija na temelju kojih se konfiguracijski sustavi dijele. Prva klasifikacija konfiguracijskih sustava odnosi se na [8]:

- prodajne konfiguracijske sustave (front office) i
- konstrukcijske konfiguracijske sustave (back office).

Zadaće i aktivnosti *prodajnih konfiguracijskih sustava* namijenjene su prodavačima i izradi narudžbe naručitelja.

Zadaće i aktivnosti *konstrukcijskih konfiguracijskih sustava* namijenjene su konstruktorima i realizaciji narudžbe naručitelja u konstrukcijskim uredima.

Druga klasifikacija konfiguracijskih sustava je po broju funkcija koje izvršavaju takvi sustavi. Na najnižem nivou, konfiguracijski sustavi samo zapisuju konfiguracijske odluke koje donosi korisnik. Sustav ne provjerava međusobnu kompatibilnost odluka, ili jesu li sve potrebne odluke donesene. Ovakva vrsta konfiguracijskih sustava koristi se kao podrška izradi narudžbi naručitelja i nazivaju se *primitivni konfiguracijski sustavi*. Na srednjem nivou, nalaze se *interaktivni konfiguracijski sustavi*, koji provjeravaju i zapisuju međusobnu kompatibilnost odluka te također vode korisnika s ciljem donošenja svih potrebnih odluka. Kompletna podrška konfiguracijskom procesu ostvarena je tzv. *automatiziranim konfiguracijskim sustavima*. Automatizirani konfiguracijski sustavi sadrže svu funkcionalnost kao i interaktivni konfiguracijski sustavi, te su još u mogućnosti da na osnovu zahtjeva naručitelja automatski izvrše kompletan proces konfiguriranja. Automatizirani konfiguracijski sustavi razlikuju se po mogućnosti donošenja konfiguracijskih odluka na temelju zaključivanja. Neki mogu donositi samo jednostavnije odluke na osnovu izbora korisnika, dok drugi mogu donositi i sve potrebne odluke vezane uz proces konfiguriranja.

U ovom istraživanju, konfiguracijski sustav realiziran je na temelju informacijskog sustava koji se koristi u procesu konfiguriranja proizvoda modularne arhitekture na temelju zahtjeva naručitelja.

Prednosti korištenja konfiguracijskih sustava s modularnom arhitekturom proizvoda jesu:

- smanjenje broja grešaka u procesu konfiguriranja,
- smanjenje vremena konfiguriranja varijanata proizvoda,
- smanjenje troškova proizvodnje varijanata proizvoda i
- povećanje varijantosti (asortimana) proizvoda.

3.4 Konfiguracijsko znanje

Znanje o konfigurabilnim proizvodima tj. konfiguracijsko znanje svrstano je u tri kategorije [2], [81]:

- znanje o konfiguriranju rješenja,

- znanje o konfiguracijskom modelu i
- znanje o zahtjevima.

Znanje o konfiguriranju rješenja odnosi se na načine pronalaženja rješenja tj. sadrži informacije vezane uz način izvođenja procesa konfiguriranja.

Znanje o konfiguracijskom modelu sadrži informacije o entitetima koji se koriste u procesu konfiguriranja, njihova svojstva te pravila i ograničenja na temelju kojih se entiteti konfiguriraju.

Znanje o zahtjevima sadrži informacije o zahtjevima na temelju kojih se izvršava proces konfiguriranja. Znanje o zahtjevima može se opisati po istom principu kao i znanje o konfiguracijskom modelu tj. znanje o konfiguracijskom modelu može se proširiti i na zahtjeve naručitelja. T reba znati da znanje o zahtjevima i znanje o konfiguracijskom modelu opisuju dva različita područja tj. entiteti konfiguracijskog modela nisu isti entiteti i kod zahtjeva te samim time različiti entiteti imaju različite uloge u rješavanju problema.

Konfiguracijsko znanje može se zapisati na dva načina [41]:

- pomoću pravila i
- pomoću ograničenja.

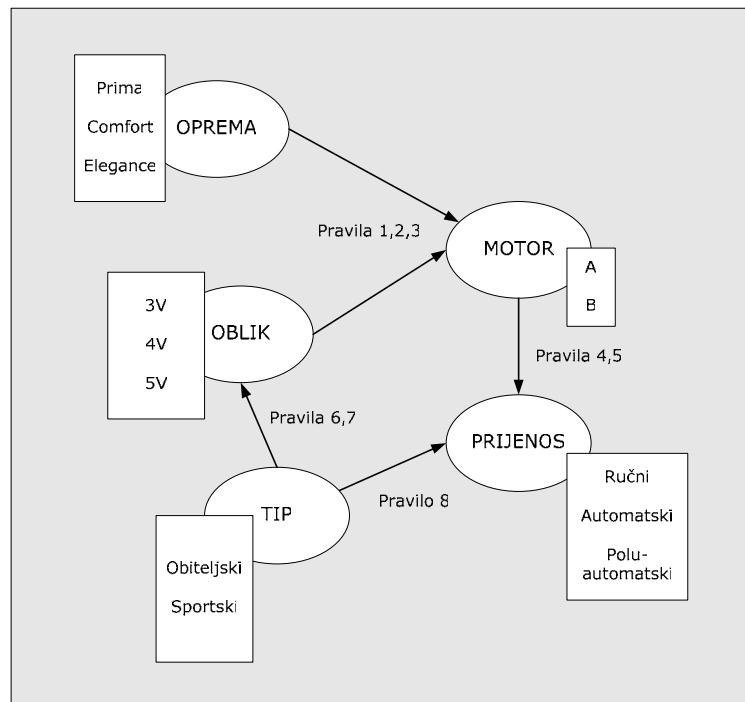
3.4.1 Zapis konfiguracijskog znanja pomoću pravila

Kada je znanje zapisao u obliku skupa pravila, nove činjenice postižu se logičnim zaključivanjem. Za pravilo oblika $A \Rightarrow B$, vrijedi da ukoliko je A poznato, moguće je odmah zaključiti B. Kod zapisa konfiguracijskog znanja pomoću pravila smjer zaključivanja može biti unaprijed ili unazad. Ako je smjer zaključivanja unaprijed, počinje se sa uvjetima, koje znamo da su istiniti i težimo zaključcima koje želimo uspostaviti. Ako je smjer zaključivanja unatrag, počinjemo sa zaključcima, za koje znamo da su istiniti i težimo uvjetima koji moraju biti istiniti. Proces zaključivanja je logični proces i jedini razlog zbog kojeg dolazi do netočnog zaključka leži u netočnosti pravila. To je ujedno i nedostatak ovakvog zapisa, jer u ukupnom skupu pravila ne smije doći do postojanja dva ili više pravila koja su međusobno u suprotnosti. Upravljanje takvim skupom pravila, tj. osiguranje točnosti zapisa kod proizvoda s velikom brzinom promjene, je vrlo teško.

Slika [Slika 3 - 2] grafički prikazuje primjer zapisa konfiguracijskog znanja pomoću pravila. Strelica na kraju linije prikazuje smjer određivanja (zaključivanja) vrijednosti parametra. Krajevi putanja smiju završavati u istom vrhu, jer znanje može potjecati iz različitih smjerova.

Za primjer konfigurabilnog proizvoda odabran je automobil koji se konfigurira na temelju pet parametara: opreme, motora, prijenosa, tipa i oblika. Zahtjevi naručitelja određeni su vrijednostima parametara: tip i oprema. Moguće vrijednosti svakog od tih parametara jesu:

- Oprema: *Prima, Comfort, Elegance*;
- Motor: *A, B* - motor A je određen za sportski tip automobila, a motor B za obiteljski tip automobila;
- Prijenos: *ručni, automatski, poluautomatski* - automatski i poluautomatski prijenos je za obiteljski tip automobila, a ručni prijenos za sportski tip automobila;
- Tip: *Obiteljski, Sportski*;
- Oblik: *3V, 4V, 5V* - oblici s četvero (4V) i petoro vrata (5V) su za obiteljski tip automobila, a s troja vrata (3V) za sportski tip automobila.



Slika 3 - 2: Zapis konfiguracijskog znanja pomoću pravila

Pravila su prikazana u obliku IF - THEN forme, a smjer zaključivanja je unaprijed. Pravila glase:

1. Pravilo: IF Oprema = *Elegance* AND Oblik = *3V* THEN Motor = *A*,
2. Pravilo: IF Oprema = *Elegance* AND Oblik = *5V* THEN Motor = *B*,
3. Pravilo: IF Oprema = *Comfort* AND Oblik = *4V* THEN Motor = *A*,
4. Pravilo: IF Motor = *A* THEN Prijenos = *Ručni*,
5. Pravilo: IF Motor = *B* THEN Prijenos = *Automatski*,
6. Pravilo: IF Tip = *Sportski* THEN Oblik = *3V*,
7. Pravilo: IF Tip = *Obiteljski* THEN Oblik = *4V*,
8. Pravilo: IF Tip = *Sportski* THEN Prijenos = *Ručni*.

Prema zahtjevima naručitelja "*Sportski Elegance*" značajke konfiguracije (varijante konfigurabilnog proizvoda) jesu:

- Tip: *Sportski* (određeno zahtjevima naručitelja),
- Oprema: *Elegance* (određena zahtjevima naručitelja),
- Oblik: 3V (6. Pravilo),
- Motor: A (1. Pravilo) i
- Prijenos: *ručni* (8. Pravilo ili 4. Pravilo);

3.4.2 Zapis konfiguracijskog znanja pomoću ograničenja

Zapisa konfiguracijskog znanja pomoću ograničenja sastoji se od:

- parametra,
- skupa određenih dozvoljenih vrijednosti koje taj parametar može poprimiti i
- ograničenja.

Ograničenja predstavljaju relacije između parametara i dozvoljenih kombinacija vrijednosti parametara koje su u relaciji. Za kombinacije parametara i vrijednosti koje nisu eksplicitno izrečene pretpostavlja se da nisu dozvoljene. Cilj ovakvog zapisa je omogućiti pronalaženje vrijednosti parametara, za koja će vrijediti sva navedena ograničenja. Izgled zapisa znanja pomoću ograničenja ima oblik: Tip Prijenos (*Sportski Ručni*), a znači da vrijednost *Sportski* za parametar Tip te vrijednost *Ručni* za parametar Prijenos jesu međusobno kompatibilne. U tablici [Tablica 3 - 1] su prikazana ograničenja, parametri i dozvoljene kombinacije vrijednosti za primjer naveden u prethodnom poglavlju.

Tablica 3 - 1: Zapis konfiguracijskog znanja pomoću ograničenja

Ograničenja	Parametri	Dozvoljene kombinacije vrijednosti
O1	Oprema Oblik Motor	(<i>Elegance 3V A</i>) (<i>Elegance 4V A</i>) (<i>Elegance 5V B</i>)
O2	Motor Prijenos	(<i>A ručni</i>) (<i>B automatski</i>)
O3	Tip Oblik	(<i>Sportski 3V</i>) (<i>Obiteljski 4V</i>)
O4	Tip Prijenos	(<i>Sportski Ručni</i>) (<i>Obiteljski poluautomatski</i>)

Prema zahtjevima naručitelja "*Sportski Elegance*" konačna konfiguracija se dobije na slijedeći način:

- prema zahtjev *Elegance* za ograničenje O1 odabrane su vrijednosti (*Elegance 3V A*), (*Elegance 4V A*) i (*Elegance 5V B*);
- vrijednosti koje ispunjavaju ograničenje O2 i ograničenje O1 jesu (*Elegance 3V A Ručni*), (*Elegance 4V A Ručni*) i (*Elegance 5V B automatski*);
- prema zahtjev *Sportski* vrijednost koje ispunjavaju ograničenje O3 i ograničenja O1, O2 je: (*Elegance 3V A Ručni Sportski*);
- vrijednost koja ispunjava ograničenje O4 i ograničenja O3, O2, O1 je: (*Elegance 3V A Ručni Sportski*).

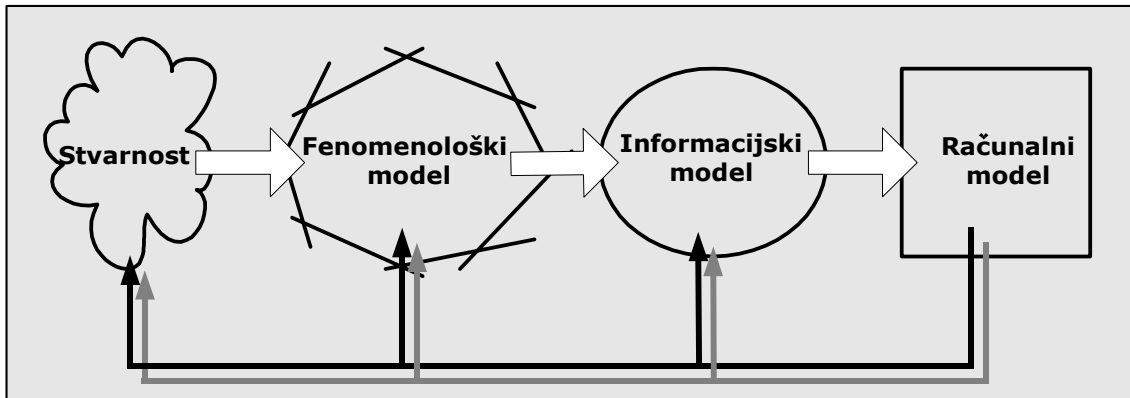
4

Pregled teoretskih osnova istraživanja

4.1 Teoretske osnove istraživanja

Ukupni cilj znanosti o konstruiranju je doprinos poboljšanju razumijevanja fenomena konstruiranja, koji je izražen kroz poboljšane modele proizvoda i procesa konstruiranja [42]. Proces konstruiranja i proizvodi kao njegov fizički rezultat, modeliraju se brojnim, u literaturi opisanim teorijama [28], [43], [44], [45]. Opisane metode modeliranja i modeli procesa konstruiranja na međusobno slične načine, prikazuju odnose stvarnog i apstraktnog svijeta. Prema [43], tijekom preslikavanja stvarnog svijeta u računalne modele, razlikujemo međukorake, odnosno zasebne modele (fenomenološke i informacijske), koji se temelje na različitim teoretskim osnovama [Slika 4 - 1].

Fenomenološki modeli se osnivaju na teorijama konstruiranja, kao što je npr. Teorija tehničkih sustava [44], te definiraju osnovu za modeliranje strukture i ponašanja proizvoda. Fenomenološki modeli temelje se na opažanju i analizi konstruiranja te upotrijebljenih alata u konstruiranju. U slučajevima gdje je to primjenjivo, fenomenološki modeli se detaljnije razvijaju kao informacijski modeli, koji čine osnovu za formalno modeliranje proizvoda i procesa konstruiranja usmjereno k računalnoj implementaciji. Informacijski modeli se osnivaju na informacijskim teorijama, npr. eng. entity-relationship modeliranju ili objektno-orijentiranom modeliranju [46]. Računalni modeli koriste informatičku znanost za uspostavu računalnih modela inženjerskih procesa i proizvoda te se zasnivaju na računalnim tehnologijama i programskim jezicima, kao npr. C++, Java, Visual Basic, SQL, XML,...



Slika 4 - 1: Iz realnosti do računalnog modela

Preslikavanje između navedenih modela s lijeva u desno [Slika 4 - 1], objašnjava tijek istraživanja u kojem se fenomenološki modeli formaliziraju u informacijske i računalne modele. Relacije u suprotnom smjeru predstavljaju potvrđivanje prilikom kojeg se svaka od tri klase modela sučeljava s realnošću koju modelira, te se na taj način provjerava njihova uporabljivost. Čvrsta granica između prikazane tri klase modela znači da se teoretske osnove, koje postoje u pozadini istraživanja, mijenjaju kroz preslikavanje, iz teorija konstruiranja i proizvoda do računalnih teorija.

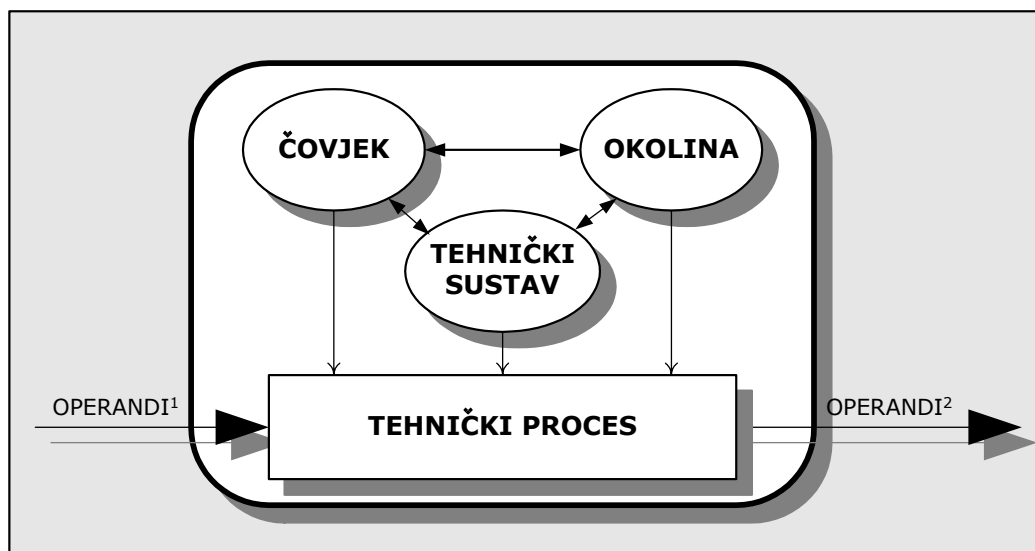
Osnovu rada čine slijedeća teoretska područja koje će biti ukratko objašnjena u idućim poglavljima:

- Teorija tehničkih sustava,
- Aksiomska teorija konstruiranja i
- Teorija svojstva.

4.2 Teorija tehničkih sustava

Primarni cilj Teorije tehničkih sustava je klasificiranje i kategorizacija znanja o tehničkim sustavima u uređeni skup zaključaka vezanih uz njihovu prirodu, reguliranje kontrole, i razvoj [18]. Teorija tehničkih sustava opisuje fenomenološki model koristeći pravila i metode za modeliranje tehničkih sustava, te je kao takva primjenjiva za opisivanje strukture i ponašanja proizvoda.

Hubka i Eder [18] definirali su tehnički sustav (proizvod), čovjeka i okolinu kao elemente potrebne za izvođenje tehničkog procesa u kojem se operandi transformiraju iz ulaznog stanja u izlazno stanje [Slika 4 - 2]. U toj transformaciji operanda mijenjaju se njihovi atributi.



Slika 4 - 2: Općeniti model tehničkog procesa

Da bi se izvršio tehnički proces potrebni su efekti. Tehnički sustav, čovjek i okolina kreiraju te efekte, npr. svjetlost, silu, toplinu. Tehnički proces može se podijeliti u podprocese. Svaki od tih podprocesa također zahtjeva efekte za svoje izvršenje. Potpunost svih podprocesa, odnosno uspješno transformiranje njihovih operanda od ulaznog stanja prema izlaznom, određuje svrhu tehničkog sustava. U strukturi komponenti koje čine tehnički sustav mogu se razlikovati skup konstitucijskih elemenata i skup relacija između njih.

4.3 Aksiomska teorija konstruiranja

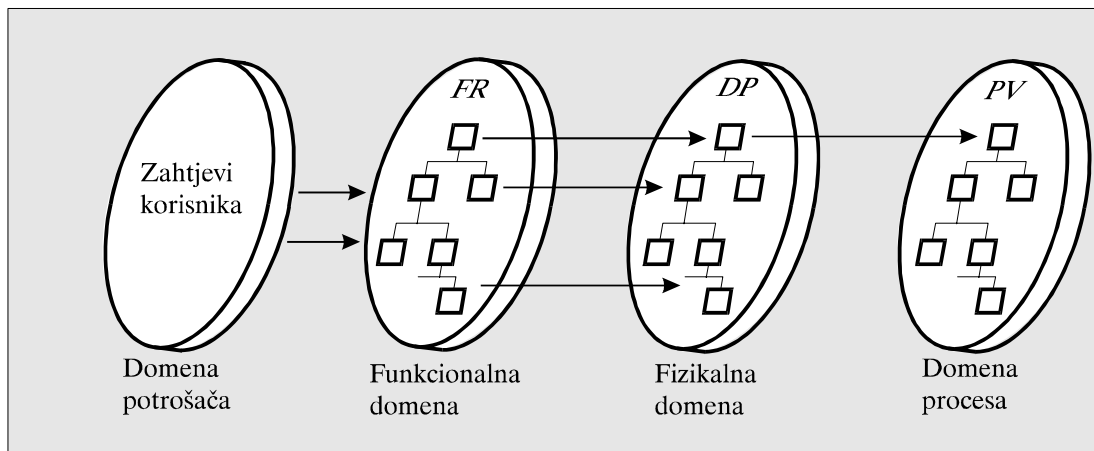
Aksiomska teorija konstruiranja [47] nudi sustavni pristup konstruiranju proizvoda i planiranju proizvodnje. Pri tome Suh [48] naglašava da teorija "osigurava istovremeno formalni opis konstrukcijskog procesa i skup osnovnih principa odlučivanja". Proces konstruiranja podijeljen je na slijedeće korake:

- određivanje ciljeva konstruiranja koji zadovoljavaju zadani skup zahtjeva naručitelja,
- generiranje ideja za kreiranje vjerojatnih rješenja,
- analiza predloženih rješenja,
- izbor rješenja koje najpovoljnije zadovoljava ciljeve i
- primjena (razrada) odabranog rješenja.

Aksiomatski pristup opisanog procesa konstruiranja temelji se na slijedećim konceptima:

- konstruiranje uključuje kontinuiranu obradu informacija između i unutar četiri različite domene [Slika 4 - 3],

- alternativna rješenja kreiraju se pridruživanjem zahtjeva specificiranih u jednoj domeni, skupu značajki druge domene,
- izlaz svake domene prikazuje se u hijerarhijskom obliku od apstraktnog koncepta do potpuno određenih informacija,
- hijerarhijska dekompozicija u jednoj domeni ne može se provesti nezavisno od hijerarhije susjednih domena, odnosno dekompozicija slijedi "cik-cak" preslikavanje unutar morfologija susjednih domena,
- dva aksioma tvore racionalnu bazu za vrednovanje predloženih alternativa rješenja i nakon toga izbor najbolje alternative.



Slika 4 - 3: Domene procesa konstruiranja po aksiomatskoj teoriji

Potrebe korisnika se pojavljuju u domeni korisnika (domena potrošača), a formaliziraju u funkcionalnoj domeni kao skup čiji su članovi funkcionalni zahtjevi (FR) koji u potpunosti opisuju konstrukcijsko rješenje. Pri tome je svaki funkcionalni zahtjev (FR) po definiciji nezavisan od ostalih funkcionalnih zahtjeva. Kako se kompleksnost opisa povećava s brojem funkcionalnih zahtjeva za primjenu teorije je važno da se potrebe opisuju s minimalnim skupom nezavisnih zahtjeva. Po definiciji, funkcionalni zahtjevi su neovisni, premda u svakoj konstrukciji postoje određeni skup zahtjeva koji su međusobno ovisni [49].

U domeni potrošača, nije izvršena detaljnija hijerarhijska dekompozicija zahtjeva, zbog toga što je najvažniji korak u konstruiranju uspostava neovisnih funkcionalnih zahtjeva na temelju zahtjeva potrošača. Kreativna faza konstruiranja ili sinteza uključuje pridruživanje funkcionalnih zahtjeva (FR) parametrima konstrukcije (DP) u fizikalnoj domeni. Parametar konstrukcije (DP) može biti geometrijska značajka komponente, svojstvo materijala, smještaj komponente u sklopu, broj komponenata i dr. Broj mogućih rješenja za svaki dani skup FR ovisi o imaginaciji, i iskustvu konstruktora. Stoga aksiomi služe za određivanje prihvatljivih rješenja.

Tablica 4 - 1: Aksiomi u konstruiranju

Aksiom 1: Aksiom nezavisnosti	Osigurati nezavisnost funkcionalnih zahtjeva.
Aksiom 2: Informacijski aksiom	Minimizirati informacijski sadržaj konstrukcije.

Prvi aksiom [Tablica 4 - 1] određuje prirodu pridruživanja između "onoga što se traži" (FR-a) i "kako se to postiže" parametara konstrukcije (DP-a). Prema Suhu [47] "dobro konstrukcijsko rješenje" osigurava nezavisnost funkcionalnih zahtjeva.

U drugom se, informacijskom aksiomu određuje sadržaj informacija konstrukcijskog rješenja kao mjera za određivanje relativne vrijednosti i usporedbu alternativnih rješenja koja zadovoljavaju prvi aksiom.

4.4 Teorija svojstva

Teorija tehničkih sustava obuhvaća i Teoriju svojstva kod koje se svojstvo definira kao karakteristika objekta kojemu pripada i kojega karakterizira [78] ,[18]. U određenom tehničkom sustavu, postoje različita svojstva s odgovarajućim kvalitativnim i kvantitativnim vrijednostima. Hubka i Eder [80] ,[18] razmatraju podjelu svojstava u nekoliko kategorija:

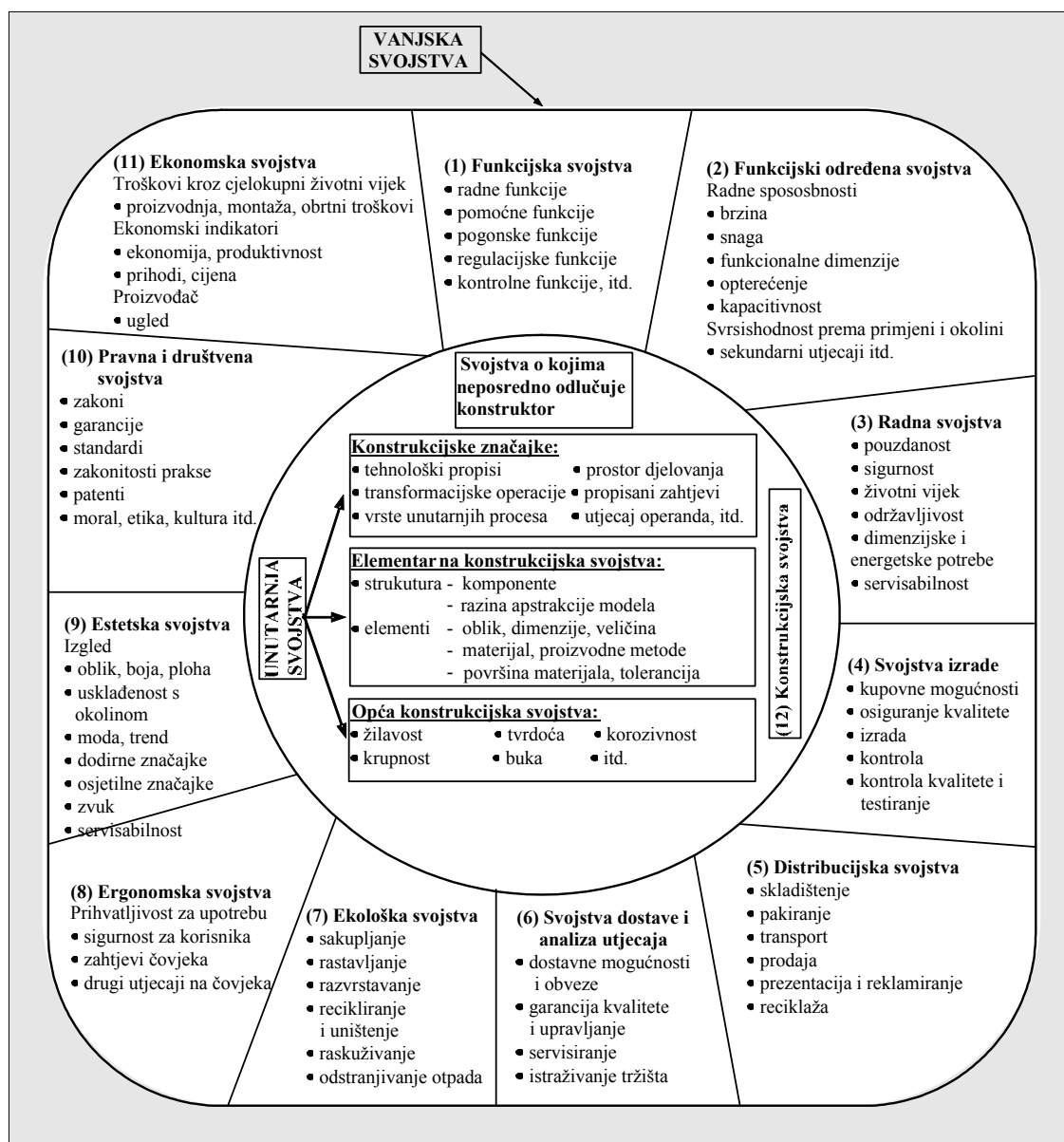
- s obzirom na opažanje (vanjsko/unutarnje),
- uzročnu vezu (uzrok/posljedica),
- funkcionalnu ovisnost (ovisno/neovisno),
- mogućnosti kvantitativnog određivanja (jednostavno/teško/nemoguće),
- važnosti (vrlo važno/važno/manje važno/nevažno),
- tehničko-znanstvenog područja (geometrija/kinematika/mehanika/...),
- potreba za konstruiranjem (vanjska/unutrašnja/konstrukcijska).

Posljednja kategorija je posebno važna za ovo istraživanje. Na slici [Slika 4 - 4] pokazane su tri klase svojstava:

- vanjska svojstva,
- unutrašnja svojstva i
- konstrukcijska svojstva.

Vanjska svojstva su od primarnog interesa korisniku tehničkog sustava. U našem slučaju, vanjska svojstva zahtijevana su i donesena od strane naručitelja. Unutrašnja svojstva su manje orijentirana naručitelju te imaju značajni utjecaj na proizvodnju i konstruktora s obzirom da služe kao pomoć konstruktoru prilikom konstruiranja željenih vanjskih svojstava [18]. Konstrukcijska svojstva predstavljaju

svojstva pomoću kojih konstruktor objedinjuje sva ostala svojstva [18].



Slika 4 - 4: Klase svojstva prema [18]

Prema Hubki i Ederu [50] konstrukcijska svojstva dijele se na:

- strukturu,
- oblik,
- dimenzije,
- veličinu,
- materijal,
- površinu materijala,

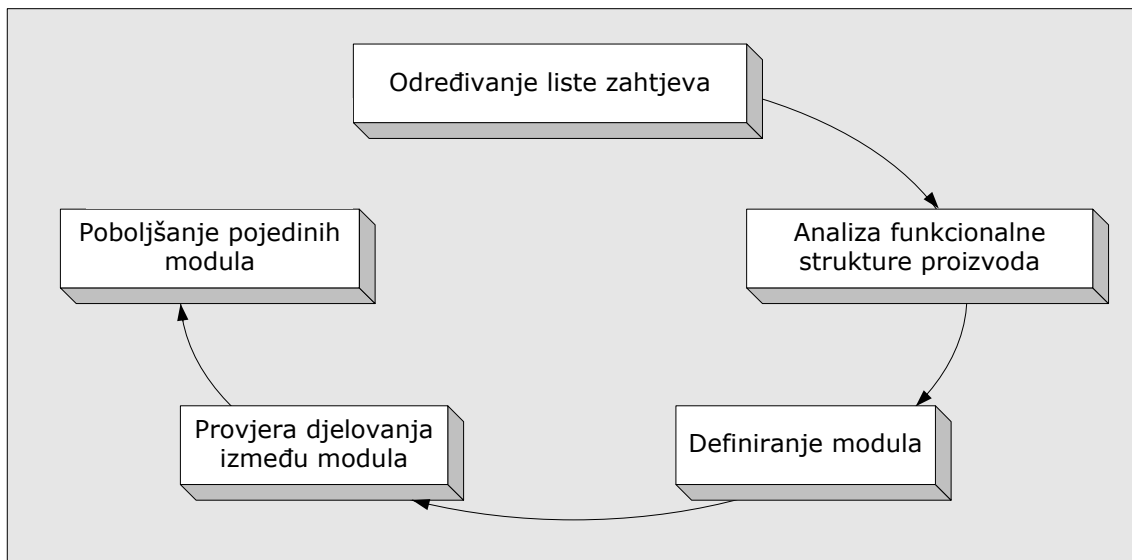
- tolerancije i
- proizvodne metode.

Ova konstrukcijska svojstva presudna su za ostvarivanje svih svojstva koje očekujemo od tehničkog sustava. Kvaliteta tehničkog sustava, koji se razvija ili koji je konstruiran, procjenjuje se uspoređivanjem trenutnih vrijednosti svojstva sustava s prethodno određenim ili dogovorenim vrijednostima svojstva sustava. Prema broju ili kategoriji svojstva koja su korištena kao kriteriji u vrednovanjima određuje se tehnička, ekonomska ili ukupna vrijednost tehničkog sustava.

4.5 Razvijanje modula (eng. *Modular Function Deployment*)

Erixon [52] je razvio metodu Razvijanje modula (eng. Modular Function Deployment), koja opisuje način podjele strukture proizvoda u module korištenjem modulskih smjernica. Metoda se sastoji iz pet glavnih koraka [Slika 4 - 5]:

- određivanje liste zahtjeva za proizvod,
- analize funkcionalne strukture i odabira tehničkih rješenja parcijalnih funkcija,
- definiranja mogućih modula pomoću matrice identificiranja modula,
- provjera djelovanja između modula i
- poboljšanje pojedinih modula.



Slika 4 - 5: Metoda Razvijanje modula

Određivanje liste zahtjeva za proizvod prvi je korak u ovoj metodi. Cilj koji se želi postići je definirati odgovarajuću listu zahtjeva koja sadrži samo konstrukcijske zahtjeve te se u tu svrhu primjenjuje metoda Razvijanje kvalitete (eng. Quality

Function Deployment) [53], [54].

Slijedeći korak metode uključuje sagledavanje proizvoda s tehničkog gledišta. Pritom se proizvod raščlanjuje na funkcije i pod-funkcije koje ispunjavaju zahtjeve tj. vrši se analiza funkcijske strukture proizvoda. Na temelju takve funkcijske strukture odabiru se tehnička rješenja parcijalnih funkcija. Ovakvo raščlanjivanje proizvoda na funkcije i pridruživanje odgovarajućih tehničkih rješenja pojedinim parcijalnim funkcijama prikazuje se u obliku morfološke matrice [55]. Preduvjet za postizanje dobrog modularnog proizvoda je neovisnost između različitih funkcija te njihovih tehničkih rješenja kojima se teži [47].

Identifikacija mogućih modula vrši se pomoću matrice identificiranja modula (eng. Module Indication Matrix). U ovom koraku određuje se koja tehnička rješenja mogu sačinjavati modul i to na način da se izvrši procjena za svako pojedino tehničko rješenje u ovisnosti o modulskim smjernicama. Procjena tehničkih rješenja vrši se uz pomoć ocjena 9, 3 i 1, a ovisi o tome koliko pojedina modulska smjernica utječe na pojedino tehničko rješenje. Na kraju se ocjene pojedinih tehničkih rješenja zbroje i ona tehnička rješenja koja ostvare najveći broj bodova, postaju kandidati za module. Broj modula nekog proizvoda približno je jednak vrijednosti drugog korijena od ukupnog broja dijelova u jednoj varijanti tog proizvoda [56]. Slika [Slika 4 - 6] prikazuje primjer matrice identificiranja modula.

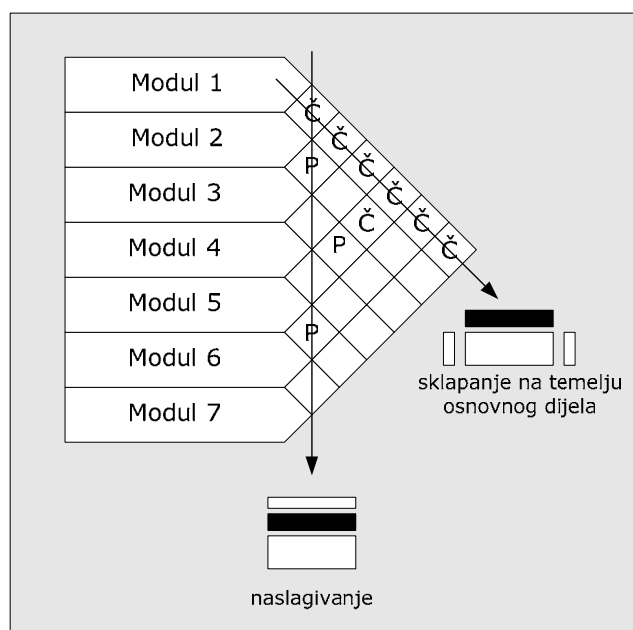
Tehnička rješenja Modulske smjernice								
	TR 1	TR 2	TR 3	TR 4	TR 5	TR 6	TR 7	TR 8
<i>ponovno korištenje</i>	9		9				9	
<i>tehnološki razvoj</i>					9	9		
<i>planirane konstrukcijske promjene u ovisnosti o vremenu,</i>								
<i>tehničku dokumentaciju</i>	1	1	1			1	1	
<i>dizajn</i>							9	9
<i>zajedničke dijelove</i>	3	3	3	9	9	3	3	
<i>ponovno korištenje procesa i organizacije</i>	9		9		9			9
<i>pojedinačno testiranje</i>			9					
<i>nabavu</i>						9	9	
<i>održavanje</i>			3			1	3	
<i>nadogradnja</i>						9		
<i>recikliranje</i>			1					9
SUMA VRIJEDNOSTI SMJERNICA	12	3	25	9	18	22	15	18
KANDIDATI ZA MODULE			X		X	X		X

Slika 4 - 6: Matrica identificiranja modula

Smjernice za određivanje modula tzv. modulske smjernice prisutne su u svim fazama životnog vijeka proizvoda a odnose se na:

- ponovno korištenje,
- tehnološki razvoj,
- planirane konstrukcijske promjene u ovisnosti o vremenu,
- tehničku dokumentaciju,
- dizajn,
- zajedničke dijelove,
- ponovno korištenje procesa i organizacije,
- pojedinačno testiranje,
- nabavu,
- održavanje,
- nadogradnju i
- recikliranje.

Prilikom provjere djelovanja između modula potrebno je obratiti pažnju na sučelja modula (u kontekstu ovog poglavlja, pod pojmom sučelja podrazumijevaju se fizička sučelja između modula). Ona imaju glavnu ulogu u određivanju fleksibilnosti proizvoda. Sučelja mogu biti čvrsta, pokretna ili za prenošenje medija. Čvrsta sučelja povezuju module u proizvode i prenose silu. Pokretna sučelja prenose rotacijsko gibanje dok su tekućine, elektricitet i sl. mediji koji se mogu prenositi. Matrica na slici [Slika 4 - 7] prikazuje način na koji se može prikazati djelovanje između modula. Oznaka (P) označava pokretna sučelja i sučelja za prenašanje medija, dok oznaka (Č) označava čvrsta sučelja.



Slika 4 - 7: Djelovanje između modula

Promatrano sa stajališta sklopa, na slici [Slika 4 - 7] strelicom su označena dva idealna principa sklapanja proizvoda:

- princip sklapanja na temelju osnovnog dijela i
- princip naslagivanja.

Sve oznake koje se nalaze izvan područja strelica, predstavljaju nepoželjna međudjelovanja između sučelja modula, te se kao takva trebaju izbjegavati.

Metoda "Razvijanje modula" ne predstavlja zamjenu za postojeće DfX metodologije, već se na temelju njih vrši daljnje poboljšanje pojedinih modula kako bi se uklonili nedostaci koji su uočeni u prethodnim fazama.

5

Zahtjevi naručitelja

5.1 Zahtjevi naručitelja

Zahtjevi naručitelja opisuju se kao potrebe čije zadovoljavanje potrošač očekuje od proizvoda. Temeljno pravilo koje se koristi prilikom definiranja potreba naručitelja, tj. što se od proizvoda očekuje, je: "Postaviti pravo pitanje pravom potrošaču u pravo vrijeme" [48]. Odgovor na to pitanje, može se sagledati kao [57]:

- svojstvo i
- značajka¹ proizvoda.

Konstruktor direktno određuje značajke koje definiraju proizvod, dok svojstvo proizvoda opisuje ponašanje proizvoda i ne može se direktno odrediti od strane konstruktora [58]. Mortensen i Andreasen [59] dijele svojstva proizvoda na:

- vlastita svojstva i
- povezana svojstva proizvoda.

Povezana svojstva proizvoda odnose se na svojstva proizvoda koja su rezultat povezanosti proizvoda s određenom životnom fazom proizvoda [60].

I drugi autori navode klasifikaciju svojstava. Prvenstveno Hubka i Eder u Teoriji svojstava (poglavlje 4.4) razmatraju podjelu svojstava na:

- vanjska,
- unutrašnja i

¹ Umjesto termina "značajka" proizvoda često se koristi i termin "karakteristika" proizvoda.

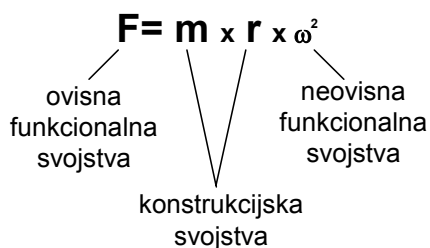
- konstrukcijska svojstva.

Osim njih, Simonek [61] razlikuje dvije vrste svojstva:

- funkcionalna svojstva - opisuju zadaću proizvoda i
- konstrukcijska svojstva - opisuju rješenje zadatke proizvoda.

Daljnja podjela funkcionalnih svojstva je na [Slika 5 - 1]:

- ovisna i
- neovisna.



Slika 5 - 1: Funkcionalna i konstrukcijska svojstva, prema [61]

Birkhofer [62] definira pojmove:

- strukturna svojstva - svojstva koja određuje konstruktor i
- vanjska svojstva - odnose se na ostala svojstva proizvoda.

Birkhofer [62] definira svojstvo kao: *Svojstvo = Značajka + Vrijednost*. Npr. značajka bilo koje grede je njezina dužina, dok je svojstvo određene grede njezina duljina (npr. 100 cm).

U ovom radu, zahtjevi opisuju samo značajku proizvoda, a značajka proizvoda određena je tekstualnim opisom, parametrom i vrijednošću.

5.2 Upravljanje zahtjevima

Upravljanje zahtjevima naručitelja obuhvaća slijedeće aktivnosti: definiranje početnih zahtjeva naručitelja, analiziranje i definiranje liste zahtjeva te vrednovanje (ocjenjivanje) i potvrđivanje zahtjeva. Iako se vrednovanje i potvrđivanje zahtjeva razmatra od početka konstruiranja i proizvodnje prvih proizvoda, analiza i upravljanje zahtjevima nije dokraja istražena te se još uvijek razvija i istražuje. Pohl [63] je među prvima definirao pojam upravljanja zahtjevima kao:

"Sistematizirani proces razvoja zahtjeva putem iterativnog procesa analiziranja problema, dokumentiranja rezultata opažanja u različitim oblicima te provjera ispravnosti postignutih rezultata."

Iz prethodne definicije proizlaze tri koraka upravljanja zahtjevima [63]:

- analiziranje problema - određuje stupanj razumijevanja zahtjeva

naručitelja u određenom vremenu,

- prezentiranje rezultata - prikazuje znanje stečeno analiziranjem problema,
- suglasnost zahtjeva - opisuje stupanj suglasnosti između postignutih rezultata i početnih zahtjeva.

5.3 Lista zahtjeva

Lista zahtjeva predstavlja popis zahtjeva i želja naručitelja koji su nastali analiziranjem potreba potrošača. Zahtjevi i želje naručitelja važni su zbog vrednovanja koje se provodi nakon definiranja liste zahtjeva. Zahtjevi se dalje mogu dijeliti na [64]:

- da/ne i
- tolerirane zahtjeve.

Da/ne zahtjevi predstavljaju zahtjeve naručitelja koje proizvod mora bezuvjetno ispuniti tj. rješenje koje ne ispunjava takve zahtjeve nije prihvatljivo. Zbog toga se takvi zahtjevi isključuju kao kriteriji prilikom vrednovanja.

Tolerirani zahtjevi opisuju zahtjeve naručitelja koji sadrže jasno iskazane potrebne vrijednosti i dopustiva odstupanja.

Osim zahtjeva, potrebe potrošača opisane su i željama naručitelja. *Želje* je potrebno uzeti u obzir kad god je to moguće, ali s uvjetom da je opravdano povećanje cijene proizvoda zbog ispunjavanja želje. Neispunjavanje želja ne smije umanjivati vrijednost rješenja problema, dok ispunjavanje želja povećava vrijednost rješenja. Preporuča se podjela želja na [65]:

- značajnije,
- srednje važne i
- manje važne.

Da bi lista zahtjeva mogla poslužiti kao osnova za kasnije odluke, potrebno ju je što temeljitije sačiniti i što prije kompletirati. Zapravo, kompletiranje neće nikada biti potpuno jer će u toku razrade doći do dopuna i ispravaka, te stoga lista zahtjeva predstavlja popis zahtjeva i želja ali i pregled dopuna i nastalih izmjena zahtjeva i želja [28].

6

Informacijski model

6.1 ISO 10303-STEP standard

ISO 10303 ili STEP (eng. **S**tandard for **E**xchanging **P**roduct **D**ata) je međunarodni standard za računalno-primjenjiv prikaz i razmjenu podataka o proizvodu. Cilj standarda je osigurati neutralni mehanizam za opisivanje i razmjenu podataka o proizvodu tijekom životnog vijeka proizvoda, neovisno o sustavu koji ga koristi. Priroda tog opisa čini STEP pogodnim ne samo za razmjenu podataka u neutralnom formatu nego i kao osnovu za implementiranje baza podataka o proizvodu koje mogu poslužiti za spremanje i dijeljenje istih između različitih korisnika i aplikacija [66]. Početno zamišljeni kontekst korištenja standarda možemo podijeliti na [67]:

- razmjenu podataka o proizvodu – definira način razmjene podataka o proizvodu između različitih aplikacija. Primjena standarda u ovom kontekstu definira oblik podataka koji se razmjenjuje između aplikacija. Svaka aplikacija pritom zadržava kopiju podataka u vlastitom obliku. Razmjenu podataka čini transfer informacija između jednog programskog sustava k drugom, kroz medij koji predstavlja stanje informacije u jednom vremenskom trenutku. Te informacije pohranjuju se digitalno u ASCII ili binarnom zapisu. Glavne značajke korištenja konteksta razmjene podataka o proizvodu su: iniciranje od strane stvaratelja informacije, transformacija podataka u neutralnom formatu, sadržaj koji je određen diskretnom vrijednošću u vremenu i stvara se kopija podataka;
- dijeljenje podataka o proizvodu – definira pristup i način korištenja nad jednom kopijom podataka između više aplikacija simultano. Primjena

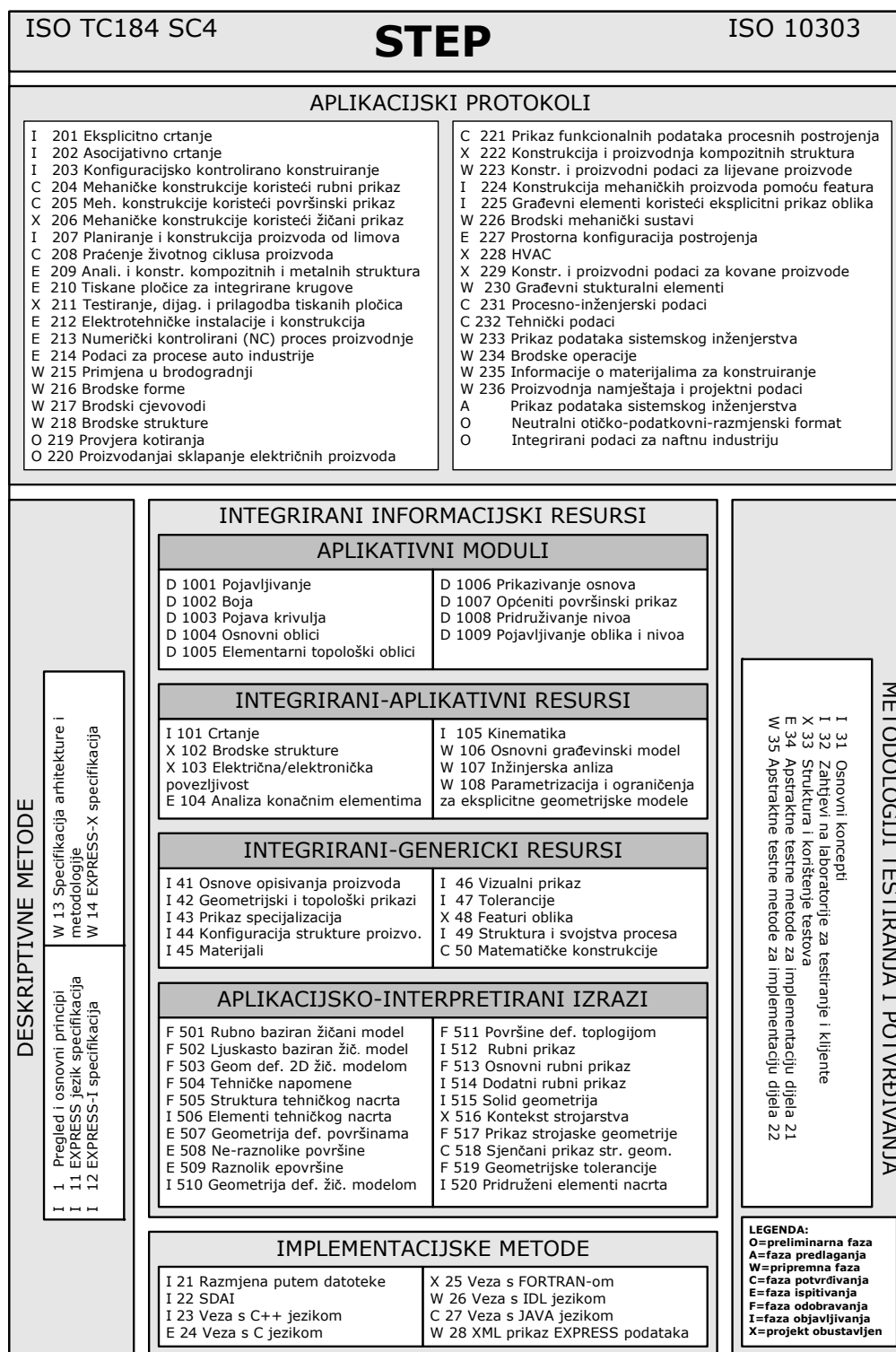
standarda u ovom kontekstu je u ostvarivanju podrške za kreiranje sučelja između pojedine kopije podataka i aplikacija koje ih dijele. Aplikacije ne čuvaju podatke u vlastitom obliku zapisa. Dijeljenje omogućuje jedan logički izvor informacija kojem različiti programski sustavi imaju pristup. Kontrola pristupa, ispravljanje informacija i vlasništvo nad podacima su kontrolirani implementacijom i administracijom izvora informacije. Glavne značajke konteksta dijeljenja podataka o proizvodu su: iniciranje od strane primatelja podatka, podacima se manipulira na zahtjev primatelja, postoje nivoi pristupa podacima i postoji samo jedan izvor podatka;

- arhiviranje podataka o proizvodu – definira načine i procedure dugovječnog spremanja podatka, te je kao takav iskoristiv za podršku procesima arhiviranja podataka.

6.1.1 Struktura ISO 10303 - STEP standarda

Struktura STEP standarda podijeljena je u više dijelova, koji se mogu kategorizirati u nekoliko glavnih grupa [Slika 6 - 1]:

- Deskriptivne metode – predstavljaju mehanizme za definiranje strukture podataka standarda te definiraju jezik (EXPRESS) za specifikaciju formalnog opisa podataka [68];
- Integrirani informacijski resursi –osnovni su semantički elementi koji se koriste za opis proizvoda te osiguravaju seriju općenitih EXPRESS modela od kojih su sačinjeni aplikacijski protokoli. Najveći dio podrške STEP-a razmjeni geometrije definiran je unutar ovih resursa, uključujući podršku za žičane modele, geometriju površina i geometriju punih tijela. Osim toga sadrže podršku za inženjerske analize, proces konstruiranja, materijale te vizualno predočenje. Općenito se može reći da ako neki model mora biti uključen na nekoliko mjesta u STEP-u, biti će definiran unutar integriranih informacijskih resursa;
- Aplikacijski protokoli – jesu informacijski modeli koji opisuju individualnu primjenu standarda za razmjenu i upravljanje s podacima o proizvodu. Aplikacijski protokoli sadrže informacijske modele zapisane u EXPRESS-u koji zadovoljavaju specifične zahtjeve, za opis proizvoda, definirane aplikacijskim kontekstom, te predstavljaju dijelove STEP-a koji se mogu implementirati. Aplikacijski protokoli predstavljaju najveći i najznačajniji dio standarda. Većina dijelova je u razvoju dok je manji dio usvojen. Za razliku od integriranih resursa, koji definiraju načine prikaza geometrije, aplikacijski protokoli definiraju uporabu tih prikaza. Aplikacijski protokoli mogu se implementirati uporabom jedne od implementacijskih metoda;



Slika 6 - 1: Struktura STEP standarda

- Implementacijske metode – predstavljaju standardne tehnike implementacije aplikacijskih protokola. Svaka metoda implementacije određuje način na koji se struktura podataka definirana STEP-om može preslikati u prikaz informacija (npr. tekstualna datoteka) i/ili pristup

informacijama (npr. veza prema programskim jezicima). Implementacijske metode uključuju razmjene podataka upotrebom fizičkih datoteke, standardno sučelje za pristup podacima (SDAI), te sučelja prema programskim jezicima C, C++, FORTRAN, JAVA, XML;

- Metode za testiranje i potvrđivanje - opisuju metodologiju testiranja i potvrđivanja različitih dijelova STEP-a.

Dijelovi standarda, koje čine deskriptivne i implementacijske metode, odvojeni su od dijelova ovisnih od industrijskoj grani (aplikacijskih protokola). Trenutno su aplikacijski protokoli definirani za strojarske i električne proizvode, a u razvoju su aplikacijski protokoli za definiranje kompozitnih materijala, savijanja limova, automobilsku industriju, proizvodne procese, brodograđevnu industriju, itd.

6.1.2 EXPRESS

EXPRESS je razvijen kao jezik za formalno opisivanje informacijskih modela. Počeci razvoja jezika datiraju iz 1982. godine kada se u okviru PDDI (eng. **P**roduct **D**ata **D**efinition **I**nterface) projekta razvio DSL jezik, na osnovu kojega se 1986. godine razvio EXPRESS. Značajke EXPRESS jezika su:

- EXPRESS se može uporabiti za opisivanje ograničenja te strukture i relacija između podataka. Ograničenja predstavljaju eksplicitan oblik zadovoljenja ispravnosti informacijskog modela;
- modeli podataka opisani EXPRESS-om mogu se obrađivati uporabom računala tj. uporabom određenih programskih rješenja. Na ovaj način je izbjegnuta potreba za korisničkom interpretacijom ili transkripcijom zapisa;
- EXPRESS je međunarodno priznati standard za formalno opisivanje modela podataka što predstavlja veliku prednost pri uporabi.

Jezik je do danas prošao kroz nekoliko verzija te se razvijao simultano sa razvojem STEP-a. Tijekom izrade pojedinih verzija EXPRESS je poprimio neke mogućnosti drugih programskih jezika (C, C++, ADA, Algol, Euler, Modula-2, Pascal, PL/I, SQL).

Osnovni element EXPRESS-a je shema. Shema sadrži definiciju modela i služi za određivanje granica definicije modela te kao mehanizam podjele velikih informacijskih modela. U svakoj shemi postoje tri kategorije definicija:

- definicije entiteta - opisuju klase objekata stvarnog svijeta uz pripadajuće osobine. Osobine objekata nazivaju se atributi. Atributi mogu predstavljati jednostavne vrijednosti (naziv, težina, itd.) ili relacije između instanci. Entiteti se mogu organizirati u hijerarhijske strukture te na taj način nasljeđivati attribute od nadređenih entiteta. Mehanizam nasljeđivanja podržava jednostruko i višestruko nasljeđivanje tzv. AND/OR nasljeđivanje;

- definicije tipova - opisuju područja mogućih vrijednosti podataka. Jezik podržava nekoliko ugrađenih tipova podataka, na osnovu kojih se mogu kreirati novi tipovi, uporabom mehanizama generalizacije ili agregacije tipova podataka;
- definicije algoritama - predstavljaju definicije funkcija ili procedura koje služe za definiranje ograničenja.

Atributi se mogu prikazati pomoću jednostavnog tipa podataka (npr. integer, string...) ili preko složenog tipa podataka tj. drugog entiteta. Tipovi podataka određuju područje vrijednosti podataka. Podržani tipovi podataka dijele se na: jednostavne, složene, imenovane i konstrukcijske tipove podataka. Složeni tipovi podataka predstavljaju uređene i neuređene skupine elemenata nekog osnovnog tipa. Imenovani tipovi podataka mogu se dodatno podijeliti na: entitete (predstavljaju objekte koji se koriste u jeziku i pomoću kojih se opisuje domena), svojstva entiteta (opisuju se atributima uz mogućnost da i sam entitet može postati atribut drugog entiteta), i na definirane tipove (koji predstavljaju proširenje skupa standardnih tipova podataka).

Konstrukcijski tipovi podataka predstavlja skupinu podataka koju čine:

- enumeration – enumeracijski tip koji čine uređeni skupovi vrijednosti prikazani imenima i
- select – odabrani tipovi koje čine imenovane skupine drugih tipova.

U EXPRESS-u entiteti su definirani kao klase. Iz definiranih klasa mogu se derivirati druge klase koje nasljeđuju osobine nadređenih klasa. U EXPRESS-u je mehanizam za određivanje klasifikacije ostvaren strukturom nadređeni entiteti/derivirani entiteti. Derivirani entiteti mogu biti međusobno povezani na različite načine ovisno o zadanom operatoru. Dozvoljeni operatori za određivanje odnosa između deriviranih entiteta jesu:

- ONEOF – instance deriviranih entiteta se međusobno isključuju,
- ANDOR – ukoliko se instance deriviranih entiteta međusobno ne isključuju ili uključuju tada se odnos između njih definira kao ANDOR i
- AND – omogućava definiranje višestrukih međusobno uključivih relacija.

Derivirani entiteti nasljeđuju attribute nadređenih te ukoliko derivirani entitet ima više od jednog nadređenog tada nasljeđuje attribute svih nadređenih. No, prilikom višestrukog nasljeđivanja može doći do konflikta između atributa različitih nadređenih entiteta. Navedena situacija se rješava na taj način što svaki atribut može imati prefiks koji sadrži ime nadređenog entiteta iz kojega se atribut nasljeđuje. Informacijski model EXPRESS-a definiran je shemama koje sadrže definiciju entiteta, tipova, funkcija i procedura, te pravila na entitetima i pravila koja definiraju relacije između entiteta ili relacija. U EXPRESS-u definirana pravila se ne mogu izvršavati niti se varijabla mogu pridružiti vrijednosti, ali je uključena potpuna proceduralno-jezična sintaksa za određivanje pravila. EXPRESS je potpuni

proceduralni jezik za definiciju strukturalnih i varijabilnih relacija, omogućuje definiciju apstraktnih tipova podataka, koristeći ograničenja za opis ponašanja objekta. Bitno je naglasiti da je jezik samo specifikacija i nije izvršan. Zbog gore navedenih svojstava EXPRESS je izabran za modeliranje informacijskog modela za konfiguriranje proizvoda modularne arhitekture.

```

SCHEMA primjer;

ENTITY vozilo SUPERTYPE OF (auto AND kamion);
  broj_kotaca: INTEGER;
  naziv_modela: STRING;
  tip_motora: odabir_motora;
END_ENTITY;

ENTITY auto SUBTYPE OF (vozilo);
  broj_putnika: INTEGER;
END_ENTITY;

ENTITY kamion SUBTYPE OF (vozilo);
  nosivost: REAL;
END_ENTITY;

TYPE odabir_motora = ENUMERATION OF
  (dizel, benzinski);
END_TYPE;

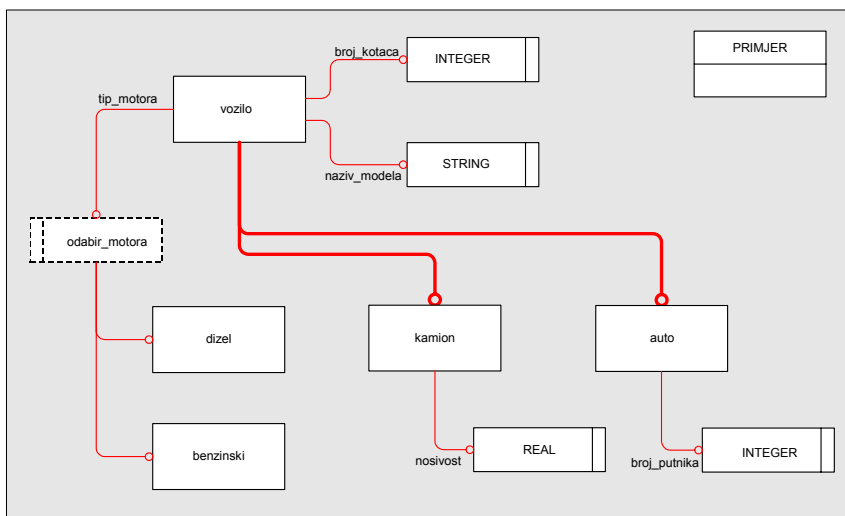
ENTITY dizel;
END_ENTITY;

ENTITY benzinski;
END_ENTITY;

END_SCHEMA;

```

a) EXPRESS

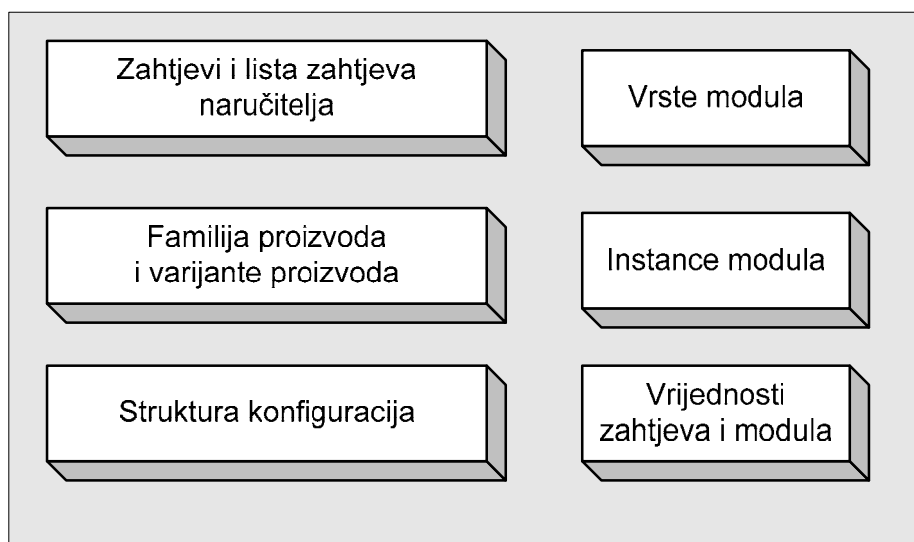


b) EXPRESS-G

Slika 6 - 2: Primjer zapisa u EXPRESS-u i EXPRESS-G

6.2 Struktura informacijskog modela

Informacijski model za podršku konfiguriranju proizvoda modularne arhitekture temeljem zahtjeva naručitelja razmatra se kao tema istraživanja prikazanog u ovom radu. Dijelovi modela, prikazani na slici [Slika 6 - 3], predstavljaju skupinu entiteta uz pomoću kojih su opisana područja vezana uz konfiguriranje proizvoda modularne arhitekture. Navedeni informacijski model opisan je u skladu sa semantikom STEP standarda i realiziran je korištenjem aplikacijskog protokola ISO 10303-214 (STEP AP214) koji opisuje podatke za procese u auto industriji.



Slika 6 - 3: Struktura informacijskog modela

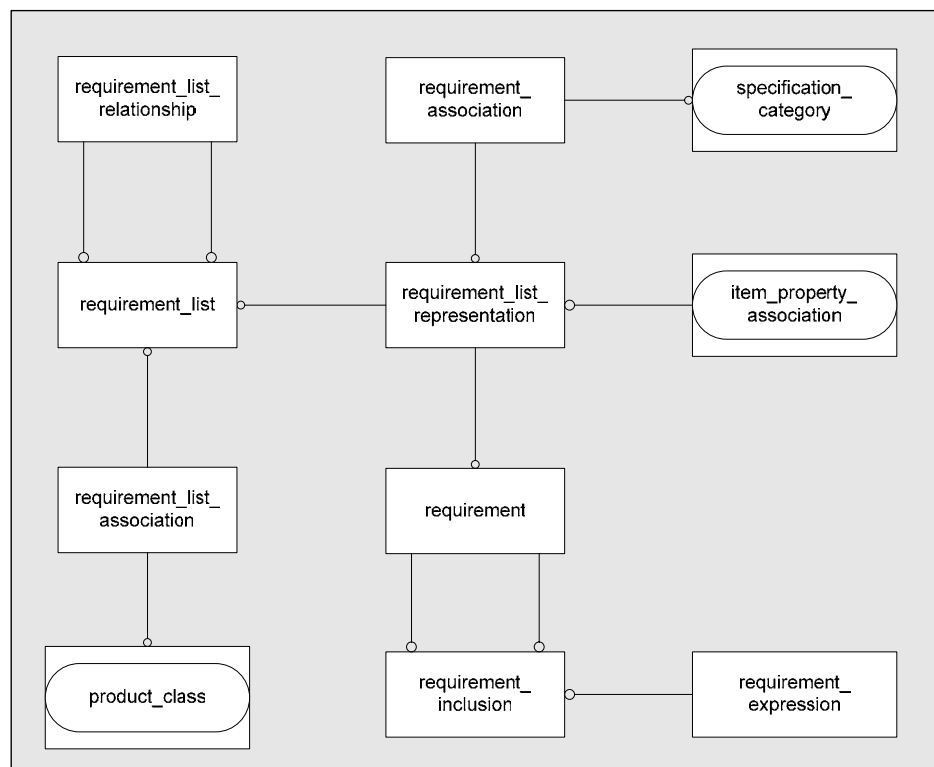
Informacijski model obuhvaća:

- zahtjeve i listu zahtjeva naručitelja - određuje entitete za opisivanje zahtjeva i liste zahtjeva potrebnih za određivanje varijante proizvoda,
- familiju proizvoda i varijantu proizvoda - određuje entitete potrebne za opisivanje familija proizvoda i varijanata proizvoda,
- module i vrste modula - određuje entitete za opisivanje i klasifikaciju modula,
- vrijednosti modula i zahtjeva - određuje entitete za opisivanje vrijednosti modula i zahtjeva,
- instance modula - određuje entitete za opisivanje instanci pojedinih modula, te pravila i ograničenja za njihovo konfiguriranje i
- strukture konfiguracija - određuje entitete za opisivanje strukture konfiguracija.

6.2.1 Zahtjevi i lista zahtjeva naručitelja

Entiteti za opisivanje zahtjeva i liste zahtjeva naručitelje nisu definirani u postojećem aplikacijskom protokolu AP 214, ISO 10303-STEP standardu. Obzirom da početak svakog procesa konfiguriranja započinje određivanjem zahtjeva i definiranjem liste zahtjeva, entiteti opisani u ovom poglavlju predstavljaju proširenje postojećeg ISO 10303 standarda i aplikacijskog protokola AP 214.

Na slici [Slika 6 - 4] prikazani su entiteti za opis zahtjeva i liste zahtjeva naručitelja.



Slika 6 - 4: Dijagram entiteta za opis zahtjeva i liste zahtjeva naručitelja

Entitet *requirement* opisuje zahtjeve definirane za proces konfiguriranja. Ovdje je potrebno naglasiti da entitet *requirement* ne opisuje i vrijednosti koje zahtjev može poprimiti, već samo tekstualni opis zahtjeva. Atributi koji opisuju ovaj entitet su:

- *id* - označava jedinstvenu identifikacijsku oznaku zahtjeva,
- *name* - označava ime zahtjeva i
- *description* - opisuje tekstualnu definiciju zahtjeva.

Prilikom određivanja zahtjeva, čest je slučaj da zahtjevi utječu jedan na drugoga. Primjeri međusobnog utjecaja zahtjeva su kada pojedini zahtjevi isključuju druge ili kada postojanje određenog zahtjeva automatski uključuje i postojanje dodatnih. Zbog toga, entitet *requirement_inclusion* predstavlja zapis o međusobnom

utjecaju zahtjeva. Atributi entitet *requirement_inclusion* jesu:

- *id* - označava jedinstvenu identifikacijsku oznaku zapisa o međusobnom utjecaju zahtjeva,
- *description* - opisuje informacije vezane uz opis međusobnog utjecaja zahtjeva,
- *if_condition* - određuje samo jedan zahtjev koji utječe na zahtjeve definirane atributom *included_requirement*,
- *included_requirement* - određuje dodatne zahtjeve na koje se zahtjev iz atributa *if_condition* odnosi.

Za opisivanje utjecaja jednog zahtjeva na druge zahtjeve tj. za određivanje da li jedan zahtjev uključuje ili isključuje druge zahtjeve, upotrebljava se entitet *requirement_expression*. Atributi koji opisuju entitet *requirement_expression* su:

- *id* - označava jedinstvenu identifikacijsku oznaku međusobnog utjecaja zahtjeva;
- *description* - opisuje da li se zahtjevi međusobno uključuju ili isključuju;
- *operand* - određuje zahtjev na koji se odnosi vrijednost atributa *operation*. Zahtjev koji je određen vrijednošću atributa *operand* mora biti određen i kao vrijednost atributa *if_condition* u entitetu *requirement_inclusion*, jer se vrijednost atributa *operation*, preko atributa *operand*, odnosi na sve zahtjeve koji su određeni atributom *included_requirement* u entitetu *requirement_inclusion*;
- *operation* - određuje operatore za određivanje međusobnih odnosa između zahtjeva. Dozvoljeni operatori jesu:
 - *AND* - omogućava definiranje višestrukih međusobno uključivih zahtjeva,
 - *NOT* - omogućava definiranje zahtjeva koji se isključuju.

Lista zahtjeva sadrži sve zahtjeve vezane uz konfiguriranje jedne varijante proizvoda. Zato je potrebno definirati entitet koji opisuje listu zahtjeva, a zatim i entitet koji opisuje koji zahtjevi pripadaju određenoj listi zahtjeva. Entitet koji opisuje listu zahtjeva, naziva se *requirement_list* i opisan je slijedećim atributima:

- *id* - označava jedinstvenu identifikacijsku oznaku liste zahtjeva;
- *name* - označava ime liste zahtjeva;
- *description* - opisuje informacije vezane uz opis liste zahtjeva;
- *level_type* - određuje razinu kojoj priprada lista zahtjeva. Može postojati dvije razina liste zahtjeva: predložak i lista. Razina "predložak" predstavlja liste zahtjeva koje služe kao predlošci. Razina "lista" predstavlja sve liste zahtjeva koje su nastale na temelju jednog

predložka;

- *version_id* - određuje verziju određene liste zahtjeva.

Međusobni odnos između lista zahtjeva, koje su definirane na istim ili različitim razinama, opisan je entitetom *requirement_list_relationship*. Atributi koji opisuju ovaj entitet su:

- *description* - opisuje informacije vezane uz međusobni odnos između lista zahtjeva;
- *related* - označava drugu od dvije liste zahtjeva koje su povezane entitetom *requirement_list_relationship*. Značenje liste zahtjeva, određene atributom *related*, određeno je vrijednošću atributa *relation_type*;
- *relating* - označava prvu od dvije liste zahtjeva koje su povezane entitetom *requirement_list_relationship*. Značenje liste zahtjeva, određene atributom *relating*, određeno je vrijednošću atributa *relation_type*;
- *relation_type* - označava značenje relacija između dvije liste zahtjeva. Dozvoljene vrijednosti koje ovaj atribut može poprimiti jesu:
 - *derivation* - entitet *requirement_list_relationship* definira relaciju kod koje je lista zahtjeva atributa *related* derivirana iz liste zahtjeva atributa *relating* (na temelju jednog predložka liste zahtjeva derivirane su ostale liste zahtjeva),
 - *version_sequence* - entitet *requirement_list_relationship* definira relaciju kod koje lista zahtjeva atributa *related* označava prethodnu reviziju liste zahtjeva, a listi zahtjeva atributa *relating* označava slijedeću reviziju liste zahtjeva.

Entitet *requirement_list_representation* opisuje pridruženost pojedinog zahtjeva određenoj listi zahtjeva. Atributi koji opisuju ovaj entitet su:

- *description* - opisuje dodatne informacije vezane uz pridruženost zahtjeva i liste zahtjeva,
- *specified_requirement* - označava zahtjev koji je pridružen listi zahtjeva u atributu *specified_requirement_list*,
- *specified_requirement_list* - označava listu zahtjeva kojoj je pridružen zahtjev u atributu *specified_requirement*.

Lista zahtjeva koja je definirana kao predložak, pridružena je određenoj porodici proizvoda. Zato je potrebno osigurati povezanost entiteta koji opisuje listu zahtjeva s entitetom koji opisuje porodicu proizvoda (entitet *product_class* opisuje porodicu proizvoda i objašnjen je u poglavlju 6.2.2). Entitet *requirement_list_association* opisuje navedenu povezanost te se sastoji od slijedećih atributa:

- *description* - opisuje informacije vezane uz međusobnu povezanost liste zahtjeva i familije proizvoda,
- *describing_requirement_list* - određuje listu zahtjeva koja je prodružena familiji proizvoda,
- *described_element* - određuje familiju proizvoda kojoj je pridružena lista zahtjeva.

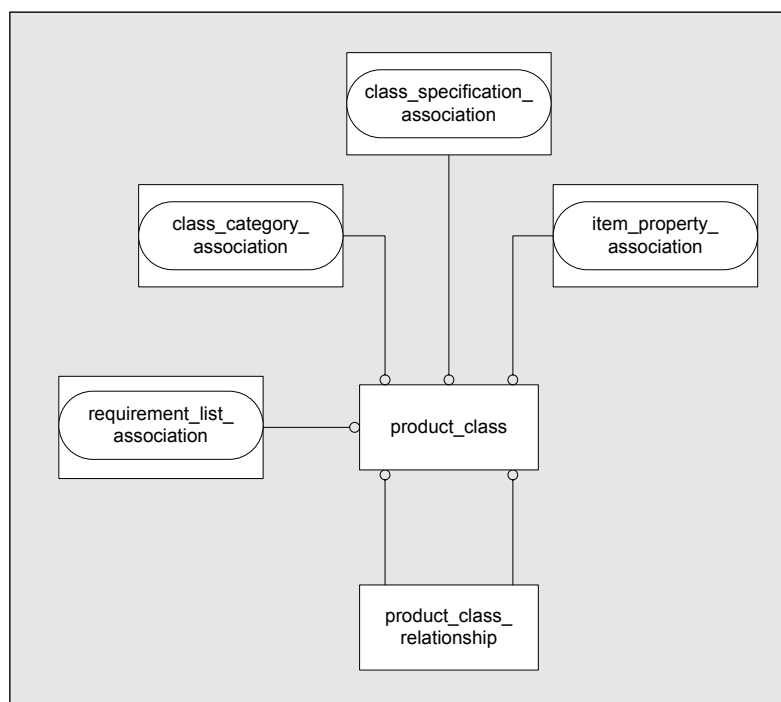
Pojedini zahtjevi utječu na odabir pojedinih modula, te je također potrebno osigurati povezanost između zahtjeva i modula (entitet *specification_category* opisuje modul i objašnjen je u poglavlju 6.2.4). Entitet *requirement_association* opisuje navedenu povezanost te se sastoji od slijedećih atributa:

- *description* - opisuje informacije vezane uz međusobnu povezanost zahtjeva i modula,
- *describing_requirement* - određuje zahtjev koji utječe na odabir modula,
- *described_element* - određuje modul čiji izbor ovisi o zahtjevu navedenim u atributu *describing_requirement*.

6.2.2 Familija proizvoda i varijante proizvoda

Na slici [Slika 6 - 5] prikazani su entiteti za opisivanje familije proizvoda i varijanti proizvoda. Osnovni entitet na slici [Slika 6 - 5] je entitet *product_class* koji ima različita značenje, ovisno o kontekstu u kojem se entitet razmatra. Prilikom razmatranja povezanosti između liste zahtjeva i familije proizvoda, entitet *product_class* opisuje familiju proizvoda. Kada se razmatra od kojih je modula sačinjena varijanta proizvoda, entitet *product_class* opisuje varijantu proizvoda. Atributi koji opisuju entitet *product_class* su:

- *id* - označava jedinstvenu identifikacijsku oznaku familije proizvoda ili varijante proizvoda;
- *name* - predstavlja riječi ili grupu riječi koje opisuju entitet *product_class*;
- *description* - opisuje informacije vezane uz familiju proizvoda i varijantu proizvoda;
- *level_type* - definira značenje u kojem se promatra entitet *product_class*. Ukoliko se entitet *product_class* razmatra kao familija proizvoda, vrijednost atributa *level_type* iznosi "familija". Ukoliko se entitet *product_class* razmatra kao varijanta proizvoda, vrijednost atributa *level_type* iznosi "varijanta";
- *version_id* - određuje verziju entiteta u ovisnosti o značenju entiteta.



Slika 6 - 5: Dijagram entiteta za opisivanje familije proizvoda i općenite varijante proizvoda

S obzirom da se entitet *product_class* razmatra u različitim značenjima, a to su familija proizvoda ili varijanta proizvoda, potrebno je definirati njihovu povezanost. Entitet *product_class_relationship* opisuje vezu između familije proizvoda i pridružene joj varijante proizvoda. Atributi koji opisuju entitet *product_class_relationship* su:

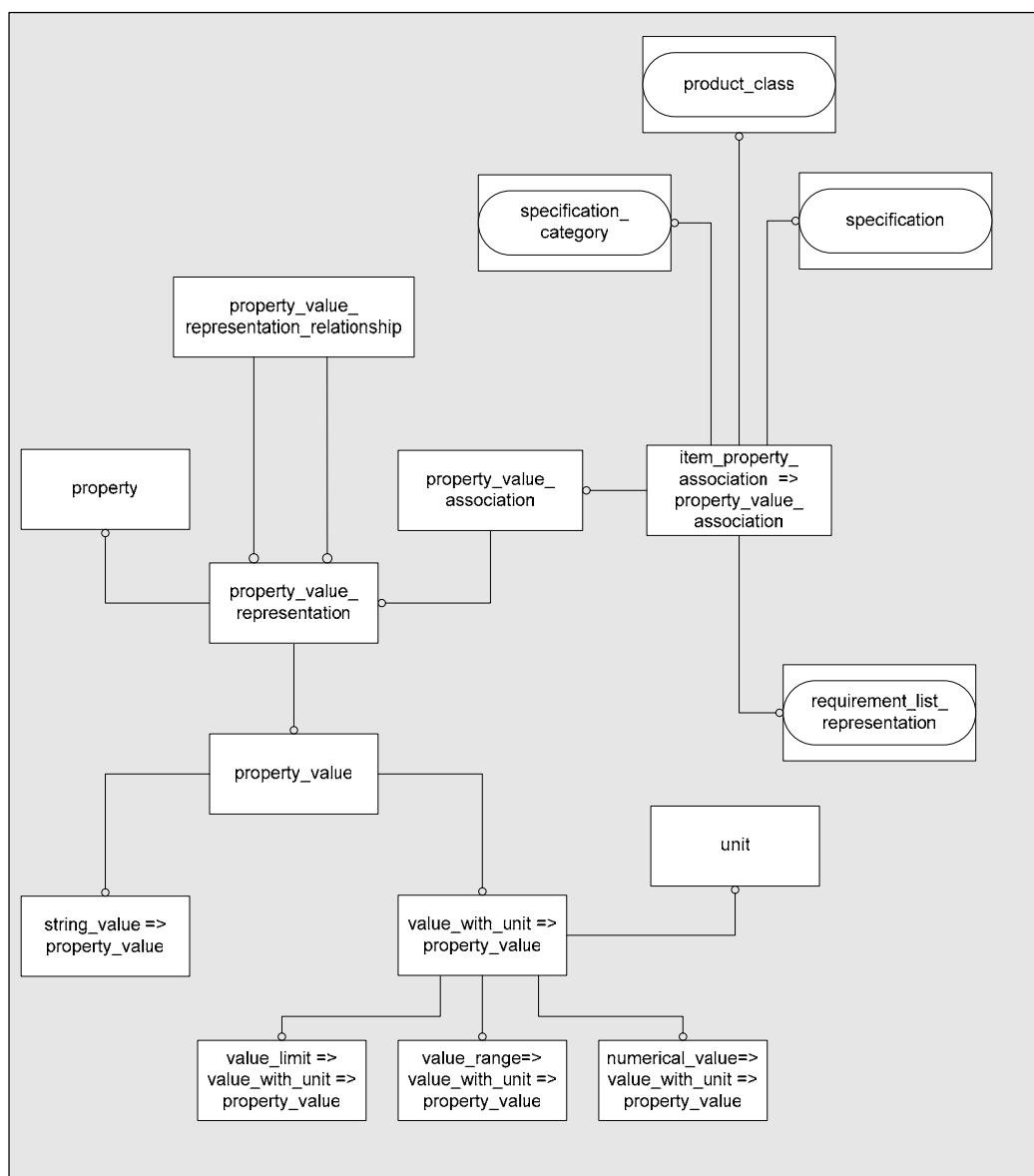
- *description* - opisuje informacije vezane uz familiju proizvoda i varijante proizvoda;
- *related* - označava varijantu proizvoda;
- *relating* - označava familiju proizvoda;
- *relation_type* – označava vrstu povezanosti između vrijednosti definiranih atributima *related* i *relating*. Vrijednost koju ovaj atribut može poprimiti je:
 - *hierarchy* - varijanti proizvoda, određenoj atributom *related*, nadređena je familija proizvoda, određena atributom *relating*.

Ostali entiteti na slici [Slika 6 - 5] objašnjeni su u slijedećim poglavljima, jer ne pripadaju osnovnim entitetima informacijskog modela koji opisuje familiju proizvoda i varijante proizvoda.

6.2.3 Vrijednosti zahtjeva i modula

U prethodna dva poglavlja opisani su entiteti informacijskog modela kojima se

definiraju pojmovi: zahtjevi, liste zahtjeva, familija proizvoda i varijanta proizvoda. Svi entiteti koji opisuju taj dio informacijskog modela, samo identificiraju navedene pojmove i ne razmatraju vrijednosti koje se mogu pridružiti pojedinim entitetima.



Slika 6 - 6: Dijagram entiteta za opisivanje vrijednosti

Tako npr. ukoliko se zahtjeva da cijena proizvoda ne prelazi npr. 1000 kn, tada cijena predstavlja zahtjev koji je postavljen prema proizvodu, a 1000 kn predstavlja vrijednost tog zahtjeva. Na slici [Slika 6 - 6] prikazani su STEP entiteti potrebni za opisivanje vrijednosti svih entiteta koji se koriste za identificiranje pojmova korištenih u ovom informacijskom modelu.

Entitet *property_value* opisuje numeričke ili tekstualne vrijednosti koje su određene entitetima *value_with_unit* i *string_value*. Zbog toga entiteti *value_with_unit* i *string_value* predstavljaju derivaciju (derivacijom entitet preuzima

atribute entiteta iz kojeg je deriviran, te osim njih entitet posjeduje i vlastite attribute) entiteta *property_value*. Entitet *value_with_unit* predstavlja pojedinačnu numeričku vrijednost ili interval numeričkih vrijednosti s gornjom, donjom ili objema granicama, ali ih ne definira, već samo određuje broj decimalnih mjesta numeričke vrijednosti te jedinicu u kojoj je izražena numerička vrijednost. Atributi koji opisuju entitet *value_with_unit* su:

- *significant_digits* - određuje broj decimalnih mjesta potrebnih za zapisivanje vrijednosti i
- *unit_component* - određuje jedinicu u kojoj je vrijednost izražena. Vrijednost atributa *unit_component* određena je *unit* entitetom.

Numeričke vrijednosti dalje su opisane entitetima *numerical_value*, *value_range* i *value_limit* te ti entiteti predstavljaju derivacije entiteta *value_with_unit*. Entitet *numerical_value* opisuje pojedinačnu numeričku vrijednost. Atribut koji opisuje ovaj entitet je:

- *value_component* - određuje količinu numeričke vrijednosti.

Entitet *value_range* opisuje dvije numeričke vrijednosti koje predstavljaju interval unutar kojeg vrijednost mora biti sadržana. Atributi koji opisuju ovaj entitet *value_range* su:

- *lower_limit* - određuje minimalnu prihvatljivu vrijednost,
- *upper_limit* - određuje maksimalnu prihvatljivu vrijednost.

Entitet *value_limit* opisuje donju ili gornju granicu numeričke vrijednosti. Atributi koji opisuju ovaj entitet su:

- *limit* - određuje vrijednost granice,
- *limit_qualifier* - određuje vrstu granice. Dozvoljene granice jesu:
 - *maximum* - određena granica je gornja granica,
 - *minimum* - određena granica je donja granica.

Entiteti *string_value* opisuje tekstualnu vrijednost sastavljenu od jednog ili više znakova. Ovaj entitet može se koristiti kada se npr. želi zapisati simbolička vrijednost (npr. d - za označavanje promjera ili l - za označavanje dužine) ili prilikom zapisa opisnog zahtjeva. Atributi koji opisuju entitet *string_value* je:

- *value_specification* - određuje tekstualnu vrijednost.

Entiteti *unit* opisuje jedinice u kojima će vrijednosti biti izražene. Atribut koji opisuje ovaj entitet ju:

- *unit_name* - određuje oblik u kojem je zapisana jedinica (npr. gram, metar, ...).

Entitet *property* predstavlja značajku koja opisuje proizvod. Kao što je već rečeno, u ovom radu promatraju se samo fizičke značajke koje su mjerljive i koje se

daju definirati od strane korisnika. Entitet *property_value_representation* označava vrijednost fizičke značajke i opisan je slijedećim atributima:

- *definition* - određuje fizičku značajku koja određuje entitet *property_value_representation*,
- *qualifier* - određuje vrstu vrijednosti. Dozvoljene su slijedeće opcije:
 - *nominal* - predstavlja osnovnu ili početnu vrijednost,
 - *specified* - predstavlja određenu (nepromjenjivu) vrijednost,
 - *typical* - predstavlja simboličku vrijednost,
- *specified_value* - označava da li vrijednost brojčana s jedinicom ili tekstualna, a određena je entitetom *property_value*,
- *value_determination* - određuje kako je vrijednost dobivena, a dozvoljene su slijedeće opcije:
 - *calculated* - vrijednost se izračunava,
 - *designed* - vrijednost je određena konstrukcijom,
 - *estimated* - vrijednost je procijenjena.

Entitet *property_value_representation_relationship* opisuje međusobni utjecaj između više vrijednosti. Primjer ovakvog utjecaja je računanje promjera ($D = 2 \cdot r \cdot \pi$). Atributi koji opisuju ovaj entitet su:

- *description* - opisuje informacije vezane uz utjecaj između više vrijednosti;
- *related* - označava vrijednost ili vrijednosti koje utječu na traženu vrijednost;
- *relating* - označava traženu vrijednost;
- *relation_type* - označava vrstu utjecaja između vrijednosti definiranih atributima *related* i *relating*. Dozvoljene opcije koje ovaj atribut može poprimiti jesu:
 - *dependency* - ova opcija definira relaciju kod koje vrijednosti iz atributa *related* utječu na vrijednost iz atributa *relating*,
 - *equivalence* - ova opcija omogućava povezanost dviju vrijednosti koje su jednake (npr. udaljenost od 1 km jednaka je udaljenosti od 1000 m, bez obzira što su različite vrijednosti i jedinice).

Pridruživanje vrijednosti, izražene entitetom *property_value_representation* prema drugim entitetima, započinje pomoću entiteta *property_value_association*. Atributi koji opisuju ovaj entitet su:

- *describing_property_value* - određuje vrijednost koja se pridružuje, a određena je entitetom *property_value_representation*,

- *description* - opisuje dodatne informacije vezane uz pridruživanje i
- *validity_context* - određuje kontekst u kojem je pridruživanje primjenjivo.

Entitet *item_property_association* predstavlja derivaciju entiteta *property_value_association* i označava entitet kojemu se pridružuje vrijednost. Atribut koji opisuje ovaj entitet je:

- *described_element* - određuje entitet kojem se pridružuje vrijednost. Entiteti kojima se treba pridružiti vrijednost su: *product_class*, *requirement_list_relationship*, *specification_category* i *specification*.

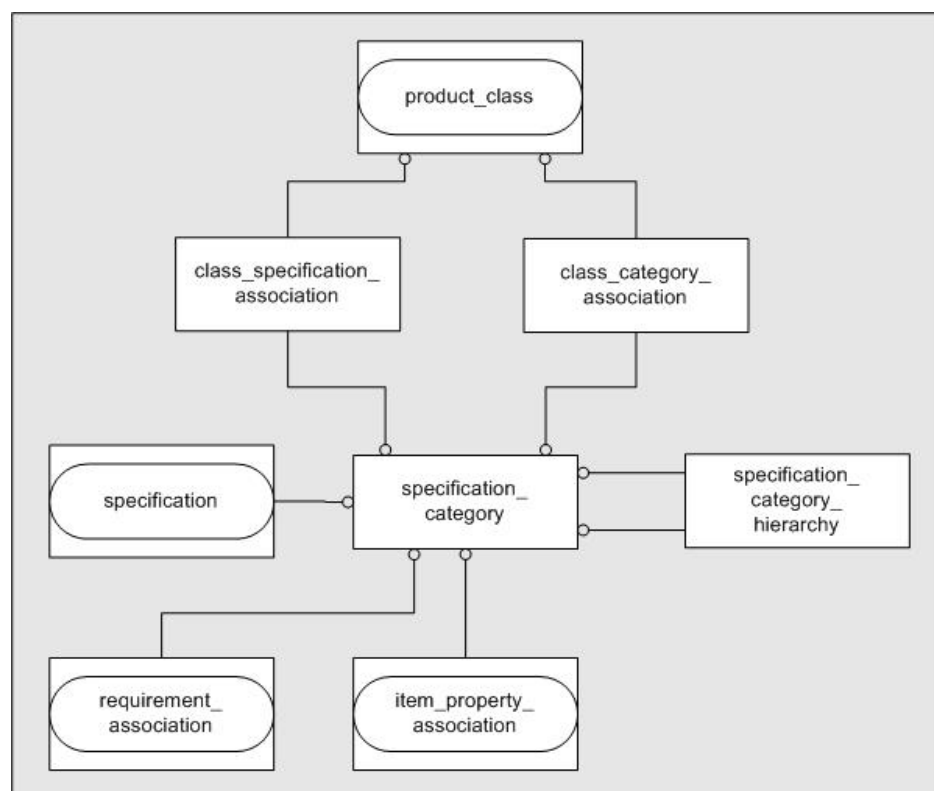
6.2.4 Vrste modula

Prema klasifikaciji modula, usvojenoj u 2.3.1.2.2, proizvod mogu sačinjavati tri grupe modula. To su temeljni, izborni i dodatni. Na slici [Slika 6 - 7] prikazani su STEP entiteti potrebni za opisivanje vrsta modula od kojih je proizvod sačinjen. U informacijskom modelu modul je opisan entitetom *specification_category*. Atributi koji opisuju ovaj entitet su:

- *id* - označava jedinstvenu identifikacijsku oznaku modula,
- *description* - opisuje informacije vezane uz opis modula i
- *implicit_exclusive_condition* - označava da li se instance pojedinog modula međusobno isključuju u varijanti proizvoda. Dozvoljene vrijednosti atributa jesu: *true* i *false* U ovom radu, vrijednost atributa *implicit_exclusive_condition* postavljena je na *true*, zbog toga što je pretpostavka da je svaki modul realiziran samo s jednom instancom tog modula.

Entitet *class_category_association* primjenjuje se za određivanje temeljnih modula u varijantama proizvoda. Atributi koji opisuju ovaj entitet su:

- *associated_category* - označava modul čije je značenje u varijanti proizvoda određeno vrijednošću atributa *mandatory*. Vrijednost atributa *associated_category* određena je entitetom *specification_category*;
- *associated_product_class* - označava varijantu proizvoda koja je određena entitetom *product_class*, a sastoji se od modula određenog atributom *associated_category*;
- *mandatory* - označava značenje modula u varijanti proizvoda. Vrijednost koju ovaj atribut može poprimiti je:
 - *true* – modul određen vrijednošću atributa *associated_category* predstavlja temeljni modul u varijanti proizvoda.



Slika 6 - 7: Dijagram entiteta za opisivanje vrsta modula

Ukoliko lista zahtjeva sadrži zahtjev ili zahtjeve koji nisu realizirani temeljnim modulima, tada se za realizaciju tih zahtjeva koriste izborni moduli. Poneki temeljni i izborni moduli uvjetuju postojanje i dodatnih modula kako bi u potpunosti mogli ispuniti svoju funkciju. Entitet koji definira izborne i dodatne module naziva se *class_specification_association*. Atributi koji opisuju ovaj entitet su:

- *associated_product_class* - označava varijantu proizvoda koja je određena entitetom *product_class*, a sastoji se od modula određenog atributom *associated_specification*;
- *associated_specification* - označava modul čije je značenje u varijanti proizvoda određeno vrijednošću atributa *association_type*. Vrijednost atributa *associated_specification* određena je entitetom *specification_category*;
- *association_type* - označava značenje modula u varijanti proizvoda. Dozvoljene opcije koje ovaj atribut može poprimiti su:
 - *option* - označava izborni modul;
 - *secondary* - označava dodatni modul.

Tijekom konfiguracijskog procesa potrebno je poznavati redoslijed kojim se određuju instance temeljnih modula. Hijerarhijskim zapisom strukture temeljnih modula opisan je redoslijed kojim se određuju instance temeljnih modula. Entitet *specification_category_hierarchy* opisuje hijerarhijski zapis strukture temeljnih

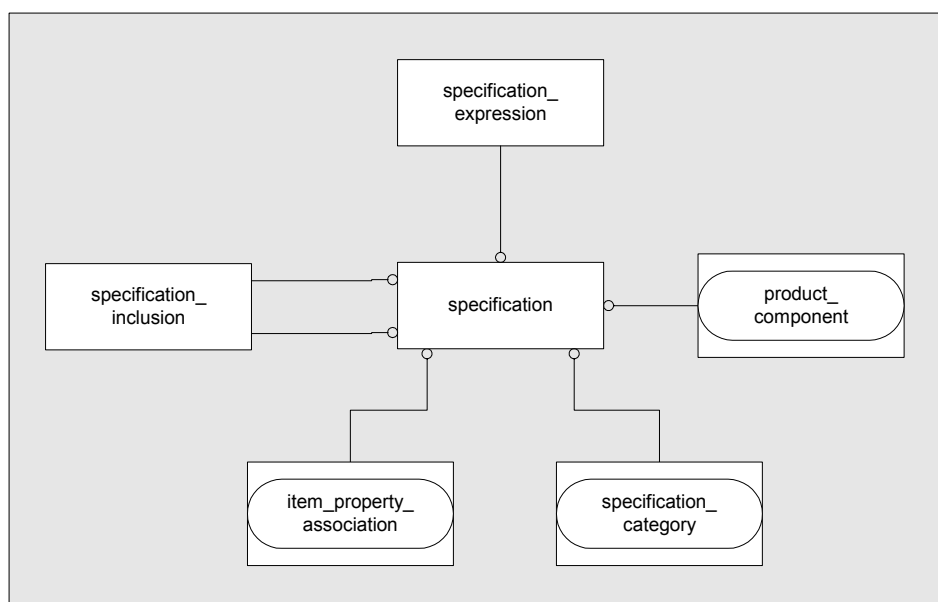
modula. Atributi koji opisuju ovaj entitet su:

- *sub_category* - označava nižu razinu u hijerarhijskoj strukturi temeljnih modula,
- *super_category* - označava višu razinu u hijerarhijskoj strukturi temeljnih modula.

6.2.5 Instance modula

Instanca modula predstavlja modul u varijanti proizvoda. Instance istog modula međusobno se razlikuju u vrijednostima značajki. Na slici [Slika 6 - 8] prikazani su STEP entiteti potrebni za opisivanje instanci modula, a entitet koji opisuje instancu modula, naziva se *specification*. Atributi ovog entiteta su:

- *id* - označava jedinstvenu identifikacijsku oznaku instance modula;
- *name* - predstavlja riječi ili grupu riječi koje opisuju instancu modula;
- *description* - opisuje informacije vezane uz instance modula;
- *category* - određuje modul kojeg instanca predstavlja. Vrijednost atributa *category* određena je entitetom *specification_category*;
- *package* - određuje da li su za potpuno ispunjavanje zahtjeva, osim odabrane instance, potrebne i dodatne instance. Dopuštene vrijednosti koje atribut može poprimiti jesu: *True* ili *False*. Ukoliko je vrijednost atributa "true", tada mora postojati bar jedna instanca koja je pridružena odabranoj instanci, a koja je određena entitetom *specification_inclusion*,
- *version_id* - određuje verziju instance.



Slika 6 - 8: Dijagram entiteta za opisivanje instanci modula

Prilikom određivanja instance modula, čest je slučaj da su pojedine instance jednog modula nekompatibilne s instancama drugih modula ili da postojanje određene instance automatski zahtjeva i postojanje još nekih instanci. Zbog toga, entitet *specification_inclusion* predstavlja zapis o međusobnom utjecaju instanci različitih modula. Atributi entiteta *specification_inclusion* su:

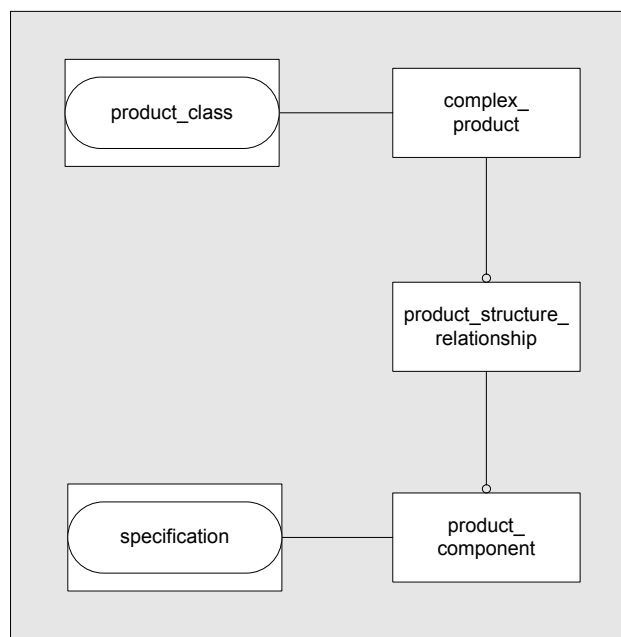
- *id* - označava jedinstvenu identifikacijsku oznaku zapisa o međusobnom utjecaju instanci,
- *description* - opisuje informacije vezane uz opis međusobnog utjecaja instanci,
- *if_condition* - određuje instancu modula koja utječe na instance drugih modula,
- *included_specification* - određuje dodatne instance na koje se instanca iz atributa *if_condition* odnosi.

Za definiranje utjecaja jedne instance na instance drugih modula tj. za određivanje da li jedna instanca uključuje ili isključuje druge instance, koristi se entitet *specification_expression*. Atributi entiteta *specification_expression* su:

- *id* - označava jedinstvenu identifikacijsku oznaku povezanosti instanci;
- *description* - opisuje da li se instance međusobno uključuju ili isključuju;
- *operand* - određuje instancu na koju se odnosi vrijednost atributa *operation*. Instanca koja je određena vrijednošću atributa *operand* mora biti određena i kao vrijednost atributa *if_condition* u entitetu *specification_inclusion*, jer se vrijednost atributa *operation*, preko atributa *operand*, odnosi na sve instance koje su određene atributom *included_requirement* u entitetu *specification_inclusion*;
- *operation* - određuje operatore za određivanje međusobnih odnosa između instanci. Dozvoljeni operatori jesu:
 - *AND* - omogućava definiranje višestrukih međusobno uključivih instanci i
 - *NOT* - omogućava definiranje instanci koje se isključuju.

6.2.6 Struktura konfiguracija

Rezultat procesa konfiguriranja je jedna ili više konfiguracija kojom je opisana određena varijanta proizvoda. Struktura takve konfiguracije opisana je STEP entitetima prikazanima na slici [Slika 6 - 9].



Slika 6 - 9: Dijagram entiteta za opisivanje strukture varijante proizvoda

Entitet *complex_product* predstavlja konfiguraciju i opisan je atributima:

- *id* - označava jedinstvenu identifikacijsku oznaku konfiguracije i
- *version_id* - predstavlja verziju konfiguracije.

Entitet *product_component* predstavlja instance modula koje su odabrane prema listi zahtjeva. Atributi entiteta *product_component* su:

- *name* - predstavlja riječi ili grupu riječi koje opisuju instancu modula,
- *description* - opisuje dodatne informacije vezane uz instancu modula i
- *is_influenced_by* - određuje kojem modulu pripada odabrana instanca.

Rezultat odabira instanci modula prema listi zahtjeva je jedna ili više instanci istih modula. Ukoliko su odabrane dvije ili više instanci jednog modula, broj konfiguracija je veći od jedan. Entitet *product_structure_relationship* opisuje strukturu svih konfiguracija koje su dobivene korištenjem odabranih instanci modula. Atributi koji opisuju ovaj entitet su:

- *description* - opisuje dodatne informacije vezane uz konfiguracije,
- *related* - predstavlja instancu modula u strukturi konfiguracije,
- *relating* - predstavlja konfiguraciju određenu entitetom *complex_product*,
- *relation_type* - označava značenje strukture konfiguracije. Dozvoljena vrijednost koju ovaj atribut može poprimiti je:
 - *decomposition* - predstavlja strukturu konfiguracije koja opisuje od kojih instanci modula je sastavljena konfiguracija.

7

Računalna implementacija sustava

7.1 Izbor radnog okruženja

U današnjim uvjetima tržišne globalizacije, mnogi proizvodi su prerasli u globalne proizvode tj. u proizvode koji zadovoljavaju zahtjeve različitih tržišta. Izazov, ali i potreba koja se postavlja na današnje konstruktore je kako u što kraćem vremenu prihvatili zahtjeve naručitelja, te tim istim potrošačima ponuditi proizvod koji je upravo prilagođen njihovim potrebama. Prilikom izbora radnog okruženja potrebno je voditi računa da korištenje radnog okruženje omogućuje lakše prilagođavanje proizvoda zahtjevima naručitelja.

Korištenje modularne arhitekture predstavlja sadašnjost, ali i budućnost u razvoju proizvoda prilagođenim potrebama naručitelja. Konfiguriranje proizvoda modularne arhitekture samo je jedan od načina pomoću kojeg se ostvaruje prilagođavanje proizvoda potrebama naručitelja. Neke od prednosti interneta, kao jednog od najpoznatijih mrežnih informacijskih servisa, u odnosu na druge tehnologije, je u omogućivanju jednakosti korisničkih sučelja neovisno o računalnom operativnom sustavu, brz prijenos informacija između zemljopisno razdvojenih mjesta te u širokoj upotrebi i lakoći pristupa. Zbog toga je za izbor radnog okruženja računalne implementacije sustava za konfiguriranje modularnih proizvoda odabrani mrežni informacijski servis - internet.

7.1.1 Klijent - poslužitelj arhitektura

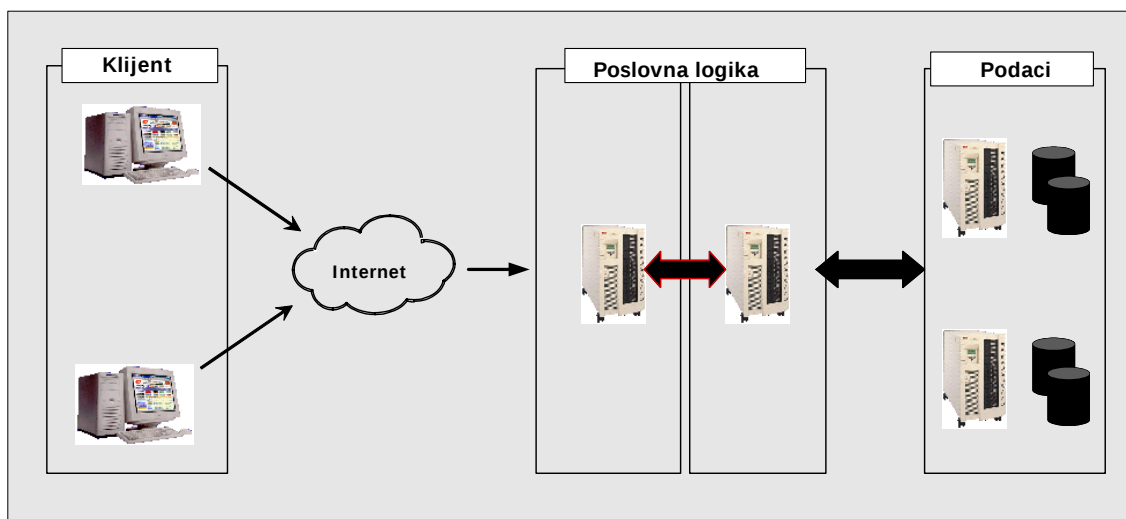
Jedno od važnih svojstava današnjih web programa jest arhitektura u kojoj se odvojeno promatraju procesi koji obavljaju ukupnu komunikaciju s korisnikom (*eng. front-end process; client-side*) i procesi koji upravljaju s podacima (*eng. back-end process; server side*). Takva arhitektura naziva se klijent - poslužitelj arhitektura, a

sastoji se od jednog ili više programa na strani klijenta (u daljnjem tekstu ti će se programi nazivati klijentima) koji šalju zahtjeve drugim programima koji se nalaze na strani poslužitelja (u daljnjem tekstu ti će se programi nazivati poslužitelji). Postoje različiti oblici klijent - poslužitelj arhitektura [69] (dvoslojne, troslojne ili n-slojne klijent - poslužitelj arhitekture).

Troslojna klijent-poslužitelj arhitekture sastoji se (kao što i samo ime kaže) od tri sloja ili razine. To su:

- klijentska razina - omogućava korisnicima web aplikacije slanje zahtjeva za podacima i primanje rezultata,
- razina poslovne logike - obuhvaća skup pravila i ograničenja pomoću kojih se vrši transformacija podataka vezanih uz zahtjeve klijenata te logiku i mehanizme za pristup bazama podataka i
- razina zapisa podataka - uključuje jednu ili više baza podataka za spremanje podataka.

Kod ove arhitekture poslužitelj je podijeljen na dva zasebna dijela: razinu poslovne logike i razinu zapisa podataka. Da bi se ostvarila stvarna troslojna arhitektura, razina poslovne logike mora biti neovisna o klijentskoj razini i razini zapisa podataka. U ovom radu, razina poslovne logike realizirana je izradom serverskih stranica - skripta (više o njima u slijedećem poglavlju) te upita pomoću kojih se vršila transformacija podataka između klijentske razine i razine zapisa podataka. Osim troslojne arhitekture, moguće je razvijati web aplikacije s n-slojnom arhitekturom, a jedina razlika je u tome što se razina poslovne logike dijeli na više neovisnih razina.



Slika 7 - 1: Troslojna klijent-poslužitelj arhitektura

Za arhitekturu računalne implementacije sustava za konfiguriranje modularnog proizvoda izabrana je troslojna arhitektura, jer po mišljenju autora ovog rada, predložena arhitektura najbolje opisuje arhitekturu potrebnu za realizaciju

sustava za konfiguriranje te omogućuje ispunjavanje zahtjeva koji su postavljeni za zadani sustav. Stoga će u nastavku biti opisani glavni dijelovi od kojih se sastoji sustav za konfiguriranje modularnih proizvoda.

7.1.2 Modeliranje razina poslovne logike

Realizacija razine poslovne logike omogućena je korištenjem PHP interpretera (**H**ypertext **P**re-**P**rocesor), PHP programskog jezika i SQL (**S**tructured **Q**uery **L**anguage) jezika. PHP interpreter počeo je razvijati Rasmus Lerdorf 1994. godine. Prva verzija PHP interpreter bila je dostupna za upotrebu 1995. godine, te je tada PHP označavao skraćenicu od **P**ersonal **H**ome **P**age [70]. PHP interpreter sadržavao je vrlo jednostavan analizator za PHP programski jezik, te je analizator ubrzo prerađen i nastao je PHP/FI verzija 2. 1997. godine Zeev Suraski i Andi Gutmans su kompletno preradili analizator te je stvorena PHP verzija 3 interpretera. Trenutačno aktualna verzija PHP interpretera je 4.

PHP interpreter namijenjen je izradi serverskih stranica (skripta) koje omogućuju kreiranje dinamičkih web stranica uz pomoć web poslužitelja. Odabrani PHP interpreter je verzije 4.1.2, a za web poslužitelja odabrani je Apache poslužitelj verzije 1.3.24. PHP interpreter i PHP programski jezik odabrani su za realizaciju razine poslovne logike zbog hardversko/software-ske nezavisnosti programskih alata potrebnih za njezinu realizaciju (PHP interpreter može se instalirati gotovo na svim operativnim sustavima (Windows, Linux, Unix, ...), PHP skripta može se izvršavati na većini web poslužitelja (Apache, Microsoft IIS, ...), a PHP programski jezik ima mogućnost povezivanja sa svim SQL bazama podataka, itd.).

Osnovna prednost PHP programskog jezika je u tome što se kôd (programske linije kojima se modelira poslovna logika) izvršava na strani web poslužitelja, a ne na strani klijenta kao što je to slučaj prilikom korištenja npr. programskih jezika JavaScript ili VBScript.

PHP programski jezik koristi se unutar HTML dokumenta. PHP kôd započinje i završava posebnim znakovima unutar HTML dokumenta ("`<?php`" - početni znak, "`?>`" - završni znak). Svaki HTML dokument koji sadrži PHP kôd naziva se skripta. Web poslužitelj prima zahtjev u obliku HTML dokumenta, kojega analizira. Prilikom analize HTML dokumenta, kôd se izvršava unutar PHP interpretera. PHP interpreter vraća rezultate poslovne logike web poslužitelju i to prije nego što je web poslužitelj generirao novu HTML stranicu, koju će vratiti klijentu koji je poslao zahtjev. Rezultati poslovne logike mogu se proslijediti web poslužitelju u obliku HTML-a ili u nekom drugom tekstualnom obliku, kao npr. PDF ili XML obliku. Također se rezultati poslovne logike ne moraju proslijediti web poslužitelju, već se mogu pospremiti na tvrdi disk u obliku datoteke te na taj način formirati spremišta rezultata poslovne logike kojima se može pristupiti i preko drugih alata, a ne samo preko web preglednika [71].

7.1.3 Modeliranje razine zapisa podataka

Na osnovi predloženog informacijskog modela za konfiguriranje proizvoda modularne arhitekture, za potrebe računalne implementacije sustava definirana je razina zapisa podataka. Informacijski model opisan u prethodnom poglavlju realizirati će se korištenjem relacijske baze podataka. Baza podataka je skup međusobno povezanih podataka, pohranjenih zajedno bez nepotrebne redundancije, s ciljem da ih koriste različiti programi. Podaci su pohranjeni u obliku neovisnom od programa koji ih koriste [72]. Uporaba relacijske baze podataka za upravljanje podacima omogućuje slijedeće prednosti [73]:

- mogućnost pohrane i pristupa podacima neovisno o njihovoj uporabi,
- mogućnost prikaza strukture podataka,
- provjera redundancije podataka i
- upravljanje integritetom podataka.

Teoretske osnove na kojima se temelji relacijski sistem baza podataka jesu [74]:

- svaka jedinka (*eng. Entity*) posjeduje skup atributa koji je opisuju,
- jedan red u tablici odgovara skupu atributa jedne konkretne jedinke,
- pojedini atributi jednoznačno označavaju svaku konkretnu jedinku. Te attribute ili skupove atributa nazivamo primarnim ključem (*eng. Primary key*),
- atribut koji sačinjava primarni ključ mora posjedovati vrijednost,
- strani ključ (*eng. Foreign key*) jedinke je atribut čija vrijednost mora postojati kao vrijednost u primarnom ključu druge jedinke,
- strani ključ pokazuje kako su jedinke međusobno povezane,
- poredak slogova u tablici je nevažan,
- poredak atributa unutar jedinke je nevažan.

Relacijska baza je u prikazanom istraživanju izabrana za pohranu podataka, zbog općenite jednostavnosti korištenja relacijskih baza i upravljanja pohranjenim podacima te mogućnosti provođenja kompleksnih upita. Također, izborom relacijske baze podataka u potpunosti je moguće ostvariti kompletno upravljanje podacima vezanim uz sustav za konfiguriranje. Izabrani alat za modeliranje relacijske baze podataka je Microsoft Access 2002.

7.2 Računalni model sustava

Računalni model sustava za konfiguriranje proizvoda modularne arhitekture prikazan je opisivanjem:

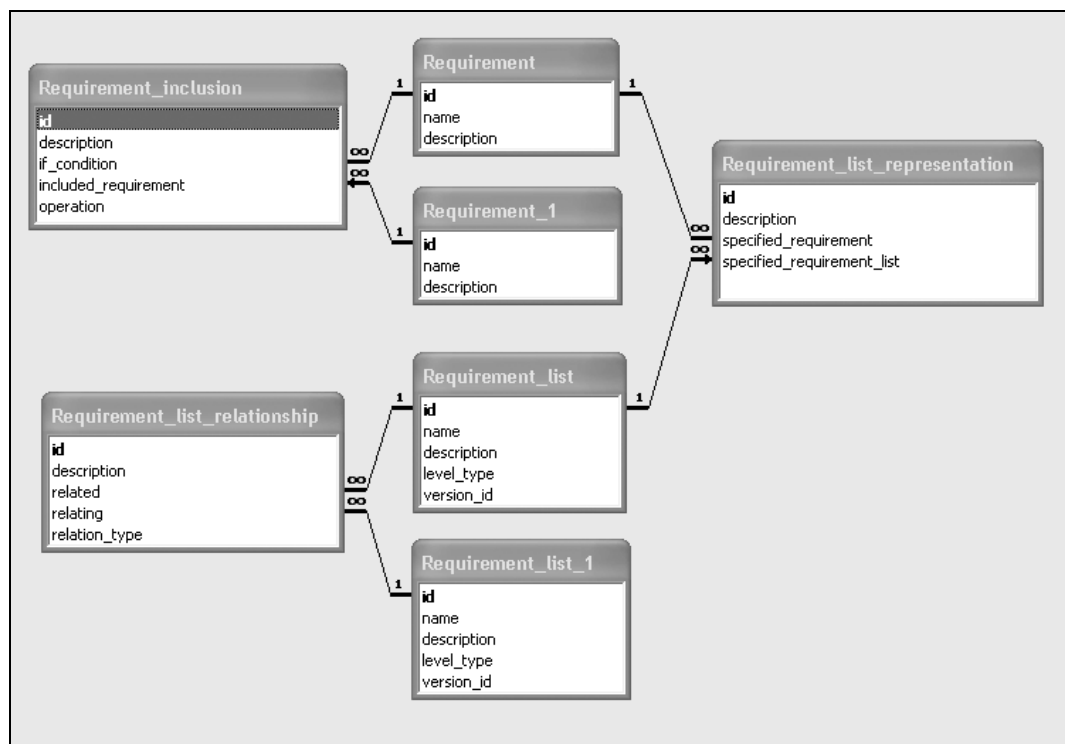
- strukture baze podataka i
- procesa konfiguriranja.

7.2.1 Struktura baze podataka

Na osnovi predložene STEP strukture (poglavlje 6.2) razvijena je struktura tablica i relacija u bazi podataka. Svaki entitet STEP modela može imati više instanci, te će se atributi svake pojedine instance zapisivati u relacijsku bazu kao jedan redak u tablicama, pri čemu će svaka kolona tablice predstavljati pojedini atribut entiteta. Prilikom kreiranja tablica i relacija u bazi podataka ostvareno je u potpunosti preslikavanje strukture STEP modela u bazu podataka. S obzirom da su entiteti na temelju kojih su kreirane tablice opisani u poglavlju 6.2, struktura tablica biti će prikazana slikom koja će sadržavati attribute tablica. Detaljnije će biti opisan redoslijed unosa podataka u tablice. Relacije koje su definirane između tablica su relacije asocijativnosti, a govore o vezama između atributa entiteta.

7.2.1.1 Tablice za opisivanje zahtjeva i lista zahtjeva

Na slici [Slika 7 - 2] prikazani je dio tablica za opisivanje zahtjeva prema dijelu informacijskog modela koji je opisan u poglavlju 6.2.1.

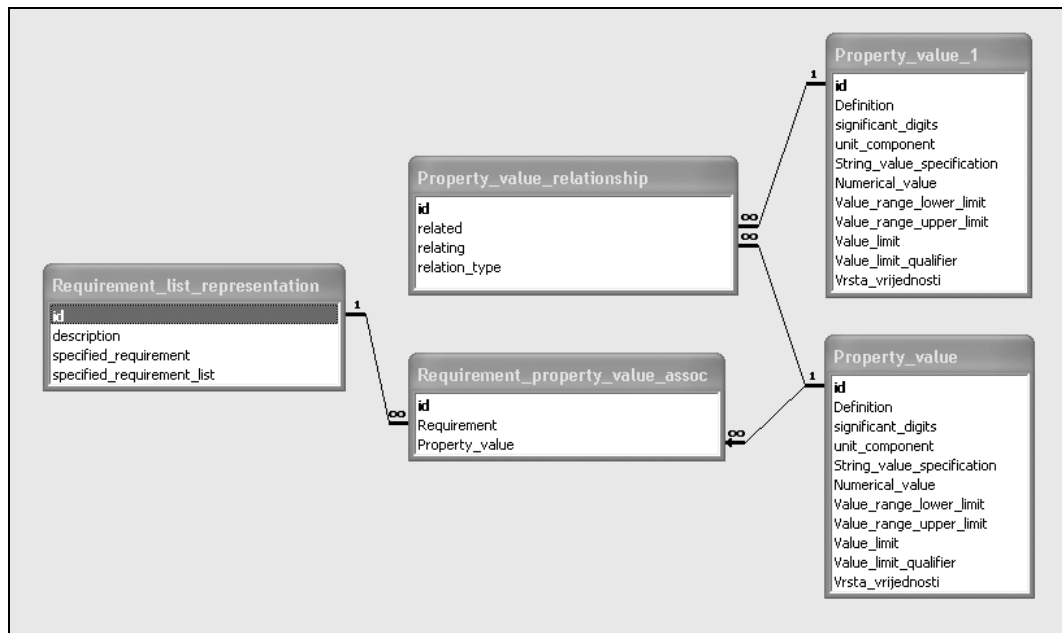


Slika 7 - 2: Struktura tablica i relacija za opisivanje zahtjeva

Proces konfiguriranja započinje s definiranjem liste zahtjeva. U tablici *Requirement_list* upisani su podaci o listama zahtjeva na temelju kojih se vrši proces konfiguriranja. Svaka nova lista zahtjeva nastaje na temelju predloška liste zahtjeva za određeni proizvod. Zbog toga je prvo potrebno unijeti podatke o predlošku listi zahtjeva. Sve liste zahtjeva kod kojih atribut *level_type* poprimi vrijednost "predložak" označavaju predloške lista zahtjeva za određeni proizvod. Vrijednost atributa *level_type* za liste zahtjeva koje nastaju na temelju predloška poprimaju vrijednost "lista". U tablici *Requirement_list_relationship* zapisane su sve liste zahtjeva koje su nastale na temelju jednog predloška, a uspostavljena relacija između lista zahtjeva je jedan-naprema-beskonačno. Nakon upisivanja podataka o listi zahtjeva, potrebno je upisati podatke o zahtjevima u tablicu *Requirement*. STEP entiteti *requirement_inclusion* i *requirement_expression* realizirani su jednom tablicom *requirement_inclusion* kod koje je, osim atributa opisanih entitetom *requirement_inclusion*, pridodan i atribut *operation* koji je definiran entitetom *requirement_expression*. Relacije između zahtjeva koji se međusobno uključuju ili isključuju, zapisane su u tablici *requirement_inclusion*, a ostvarene su zapisom primarnih ključeva pojedinih zahtjeva. Tablica *Requirement_list_representation* sadrži podatke o svim zahtjevima koji su pridruženi određenoj listi zahtjeva.

7.2.1.2 Tablice za opisivanje vrijednosti i pridruživanja vrijednosti zahtjevima

U relacijskoj bazi podataka ne mogu se modelirati derivirani entiteti na način kako su prikazani u dijelu informacijskog modela koji opisuje vrijednosti zahtjeva i modula (poglavlje 6.2.3). Zbog toga su svi derivirani elementi (*string_value*, *value_with_unit*, *value_limit*, *value_range*, *numerical_value*) sadržani u jednoj tablici *Property_value* (Slika 7 - 3), koja je sačinjena od atributa svih deriviranih elemenata. Osim tih atributa, bilo je potrebno pridodati još jedan atribut (*Vrsta_vrijednosti*) koji određuje kojem entitetu informacijskog modela pripada upisana vrijednost tj. koji atribut tablice sadrži upisanu vrijednost. U tablicu *Property_value* upisuju se sve vrijednosti koje su pridružene zahtjevima.



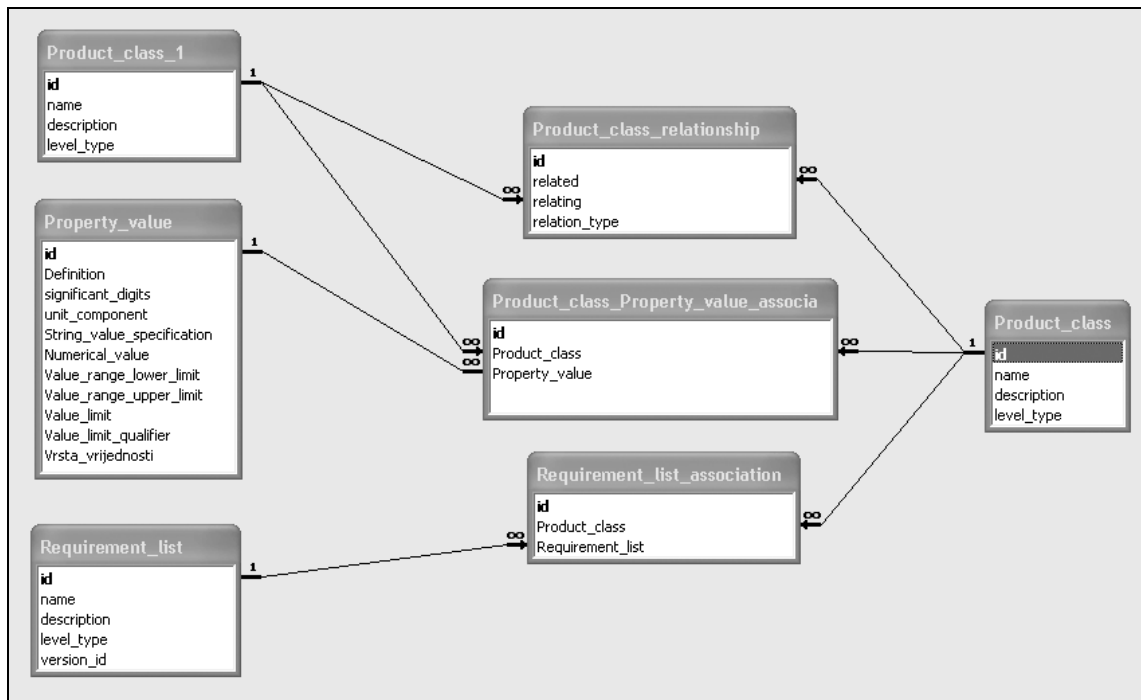
Slika 7 - 3: Struktura tablica i relacija za opisivanje vrijednosti i pridruživanju vrijednosti s zahtjevima

Podatak o pridruživanju određene vrijednosti i zahtjeva određene liste zahtjeva, zapisan je u tablici *Requirement_property_value_assoc*. Tablica *Requirement_property_value_assoc* predstavlja entitet *item_property_association* u informacijskom modelu (poglavlje 6.2.3). S obzirom da je potrebno, osim pridruživanja vrijednosti i zahtjeva, ostvariti i pridruživanje vrijednosti i s ostalim entitetima (familijom proizvoda, varijantama, modulima i instancama) potrebno je za svako pridruživanje kreirati posebne tablice. Sve tablice koje predstavljaju pridruživanje između vrijednosti i ostalih entiteta imaju naziv složen od riječi koja predstavlja entitet i *_property_value_assoc*. To su tablice *Requirement_property_value_assoc*, *Product_class_property_value_assoc*, *Spec_cat_Property_value_association* i *Specification_property_value_assoc*. Ukoliko je više vrijednosti međusobno povezano, njihova povezanost je zapisana u tablici *Property_value_relationship*.

7.2.1.3 Tablice za opisivanje varijante proizvoda i familije proizvoda

Na slici [Slika 7 - 4] prikazana je struktura tablica za opisivanje varijante proizvoda i familije proizvoda prema dijelu informacijskog modela koji je opisan u poglavlju 6.2.2. Nakon upisivanja podataka o zahtjevima i vrijednostima koje ti zahtjevi poprimaju, potrebno je upisati podatke o familiji proizvoda i varijantama proizvoda koje čine familiju proizvoda. Ti su podaci sadržani u istoj tablici *Product_class*. Vrijednošću atributa *level_type* određuje se da li podatak u tablici opisuje familiju proizvoda ili varijantu proizvoda. Ukoliko vrijednost atributa iznosi "familija" tada podaci u jednom redu opisuju familiju proizvoda, a ukoliko vrijednost atributa iznosi "varijanta" tada podaci u jednom redu opisuju varijantu proizvoda. Kojoj familiji proizvoda pripada određena varijanta proizvoda, zapisano je u tablici

Product_class_relationship. Relacija između tablica *Product_class_relationship* i *Product_class* ostvarena je definiranjem dva ista strana ključa u tablici *Product_class_relationship*, koja ne mogu sadržavati identične podatke iz tablice *Product_class*. Prilikom uspostave ovakve relacije, alat za modeliranje baze podataka Microsoft Access, prikazuje relaciju koristeći dvije identične tablice *Product_class* i *Product_class_1*. Važno je napomenuti da su podaci zapisani samo u jednoj tablici, a da je samo prikaz relacija načinjen na opisani način.



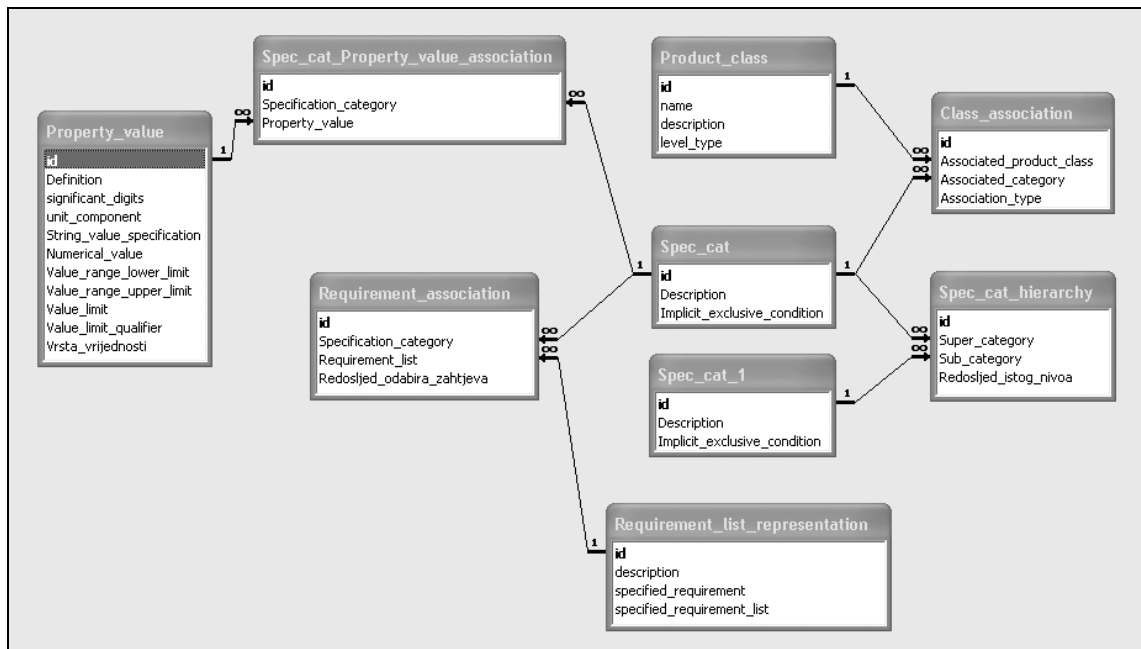
Slika 7 - 4: Struktura tablica i relacija za opisivanje varijanata proizvoda i familije proizvoda

Osim osnovnih podataka koji su spremljeni u tablici *Product_class*, svakoj varijanti ili familiji proizvoda moguće je pridružiti i ostale vrijednosti koje se nalaze u tablici *Property_value*. Pridruživanje između varijante ili familije proizvoda s pripadajućim vrijednostima zapisano je u tablici *Product_class_property_value_assoc*, koja predstavlja entitet *item_property_association* u informacijskom modelu (poglavlje 6.2.2). Osim ovog pridruživanja, u tablicu *Requirement_list_association* zapisani su podaci o pridruživanju između varijante proizvoda i liste zahtjeva koja je definirana za tu varijantu.

7.2.1.4 Tablice za opisivanje modula

Na slici [Slika 7 - 5] prikazana je struktura tablica za opisivanje modula prema dijelu informacijskog modela koji je opisan u poglavlju 6.2.4. Svaka varijanta proizvoda sačinjena je od modula, a moduli su definirani u tablici *Spec_cat* (ime tablice *Spec_cat* je skraćeni oblik od imena entiteta *specification_category*, kojim je definiran modul u poglavlju 6.2.4, a koji se morao skratiti zbog ograničenja prilikom korištenja alata za modeliranje baze podataka). Vrijednosti koje su pridružene

modulu zapisane su u tablici *Spec_cat_Property_value_association*. Entiteti informacijskog modela *class_specification_assocition* i *class_category_assocition* jesu, zbog lakše pretraživanja prilikom modeliranja poslovne logike, realizirani pomoću jedne tablice *class_assocition*. Atribut *Association_type* u tablici *class_assocition* definira vrstu modula, a vrijednosti koje može poprimiti jesu: *mandatory*; *option*; *secondary* (vrste modula opisane su u poglavlju 6.2.4). Prilikom modeliranja poslovne logike, koja se odnosi na odabir modula u procesu konfiguriranja, važno je poznavati redoslijed odabira modula. Hijerarhijska struktura modula u varijanti proizvoda, odgovara redoslijedu odabira modula u procesu konfiguriranja, te su podaci o hijerarhijskoj strukturi zapisani u tablici *Spec_cat_hierarchy*. Ukoliko postoji više modula koji se nalaze na istoj razini u hijerarhijskoj strukturi, pomoću atributa *Redoslijed_istog_nivoa* određeni je redoslijed odabira modula. Vrijednosti koje ovaj atribut može poprimiti jesu od 1 do *n*, gdje vrijednost 1 označava prvi modul koji se odabire od svih modula na toj razini, itd.



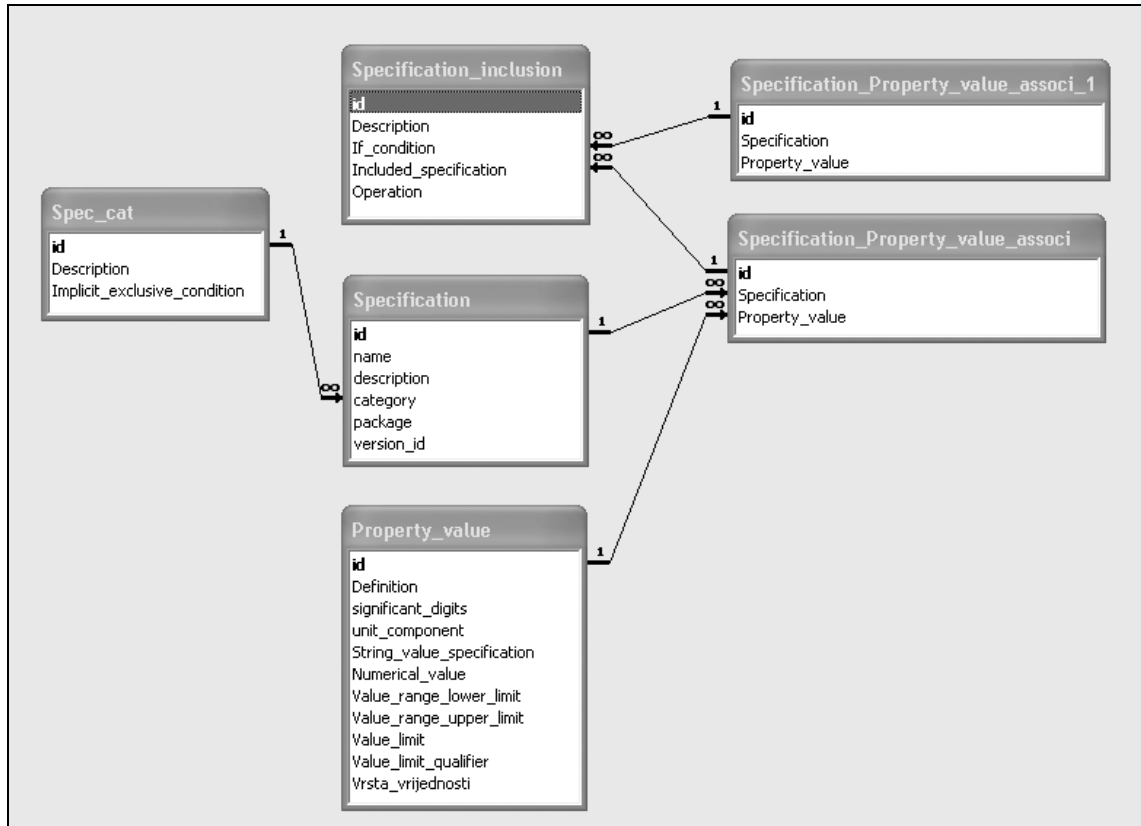
Slika 7 - 5: Struktura tablica i relacija za opisivanje modula

Međusobna povezanost između temeljnog modula i odgovarajućeg zahtjeva u listi zahtjeva (tablica *Requirement_list_representation*) zapisana je u tablici *Requirement_association*. Osim atributa definiranih entitetom *requirement_association* (poglavlje 6.2.4) u tablici *Requirement_association* pridodani je i atribut *Redoslijed_odabira_zahjtjeva* kojim je određeni redoslijed odabira zahtjeva za određeni temeljni modul.

7.2.1.5 Tablice za opisivanje instanci modula

Slika [Slika 7 - 6] prikazuje strukturu tablica za opisivanje instanci modula prema dijelu informacijskog modela koji je opisan u poglavlju 6.2.5. Varijante

proizvoda sačinjene su od instanci modula, koje su zapisane u tablici *Specification*. Vrijednost atributa u tablici *Specification* određuje modul kojega određena instanca predstavlja. Vrijednosti koje su pridružene instanci određenog modula zapisane su u tablici *Specification_Property_value_associ*.



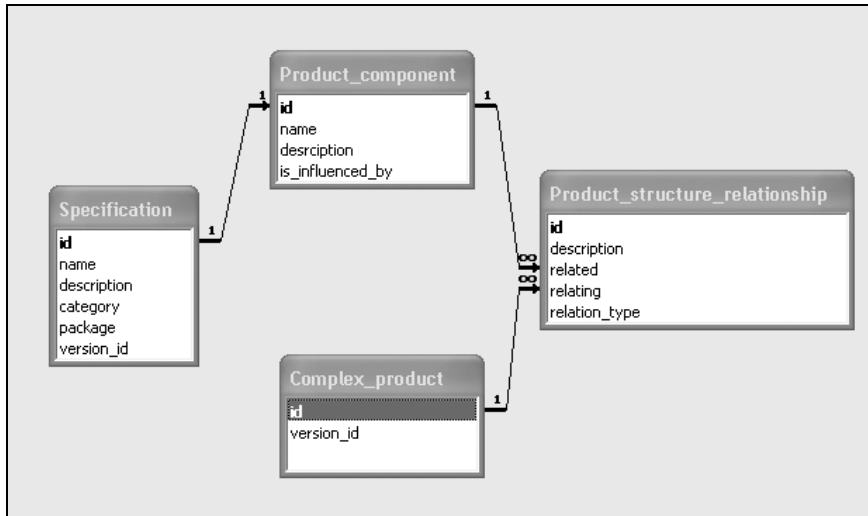
Slika 7 - 6: Struktura tablica i relacija za opisivanje instanci modula

STEP entiteti *specification_inclusion* i *specification_expression*, opisani u poglavlju 6.2.5, realizirani su jednom tablicom *Specification_inclusion* kod koje je, osim atributa opisanih entitetom *specification_inclusion*, pridodan i atribut *operation* koji je definiran entitetom *specification_expression*. U tablici *Specification_inclusion* zapisani je međusobna ovisnost između dviju instanci različitih modula. Ukoliko vrijednost atributa *operation*, kod instance definirane atributom *If_condition*, iznosi *and* tada instanca, definirana atributom *Included_specification*, mora postojati u varijanti proizvoda. Ukoliko vrijednost atributa *operation* iznosi *not*, specificirane instance su međusobno nekompatibilne, te varijantu koja se sastoji od takvih instanci treba odbaciti kao nevaljanu jer nisu zadovoljeni uvjeti kompatibilnosti.

7.2.1.6 Tablice za opisivanje konfiguracija

Struktura tablica za opisivanje konfiguracija, kao rezultata procesa konfiguriranja, prikazana je na slici [Slika 7 - 7], a modelirana je prema dijelu informacijskog modela koji je opisan u poglavlju 6.2.6. Instance modula, koje zadovoljavaju postavljene zahtjeve iz liste zahtjeva za odabrani proizvod, zapisuju se

u tablicu *Product_component*. Sve instance modula koje su zapisane u tablici *Product_component*, moraju biti sadržane i u tablici *Specification*. Time, tablica *Product_component* predstavlja podskup tablice *Specification*. Na temelju tih instanci, vrši se konfiguriranje proizvoda, što rezultira jednom ili više konfiguracijom.



Slika 7 - 7: Struktura tablica i relacija za opisivanje konfiguracija

Konfiguracija proizvoda zapisuje se u tablici *Product_structure_relationship* na način da se u tablicu zapisuje instanca i broj konfiguracije kojoj ta instanca pripada. Ukupan broj konfiguracija, kao i oznake pojedinih konfiguracija, sadržane su u tablici *Complex_product*. Vrijednost koje označavaju brojeve konfiguracija u predloženom sustavu su: 1 (prva konfiguracija), 2 (druga konfiguracija), itd.

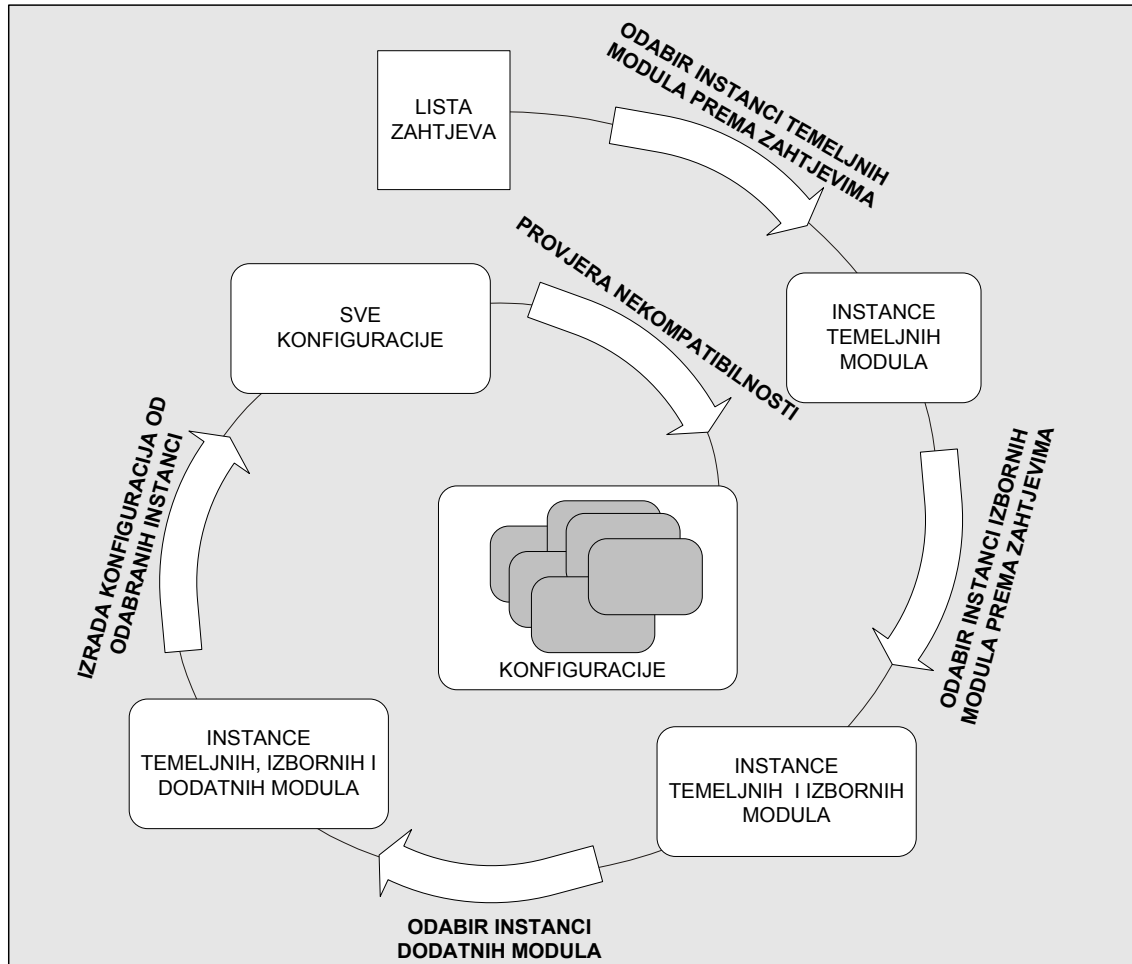
7.2.2 Proces konfiguriranja

Tijek procesa konfiguriranja, koji omogućuje predloženi sustav za konfiguriranje proizvoda modularne arhitekture, prikazan je na slici [Slika 7 - 8], a sastoji se od sljedećih koraka:

- odabir instanci temeljnih modula prema zahtjevima iz liste zahtjeva za određeni proizvod,
- odabir instanci izbornih modula prema zahtjevima iz liste zahtjeva za određeni proizvod,
- odabir instanci dodatnih modula,
- izrada konfiguracija od odabranih instanci i
- provjera nekompatibilnosti između instanci pojedinih konfiguracija.

Proces konfiguriranja proizvoda modularne arhitekture započinje definiranjem liste zahtjeva za određeni proizvod. S obzirom da se s predloženim sustavom mogu konfigurirati različiti proizvodi, potrebno je prvo odabrati vrstu proizvoda te zatim unijeti vrijednosti zahtjeva za tu vrstu proizvoda. Na temelju unesenih vrijednosti

zahtjeva, sustav prvo odabire instance svih temeljnih modula, a zatim i sve instance izbornih modula. Ukoliko neka odabrana instanca zahtjeva i odabir dodatnih instanci za njezino ispravno korištenje, sustav odabire sve dodatne instance na osnovu odabranih instanci.

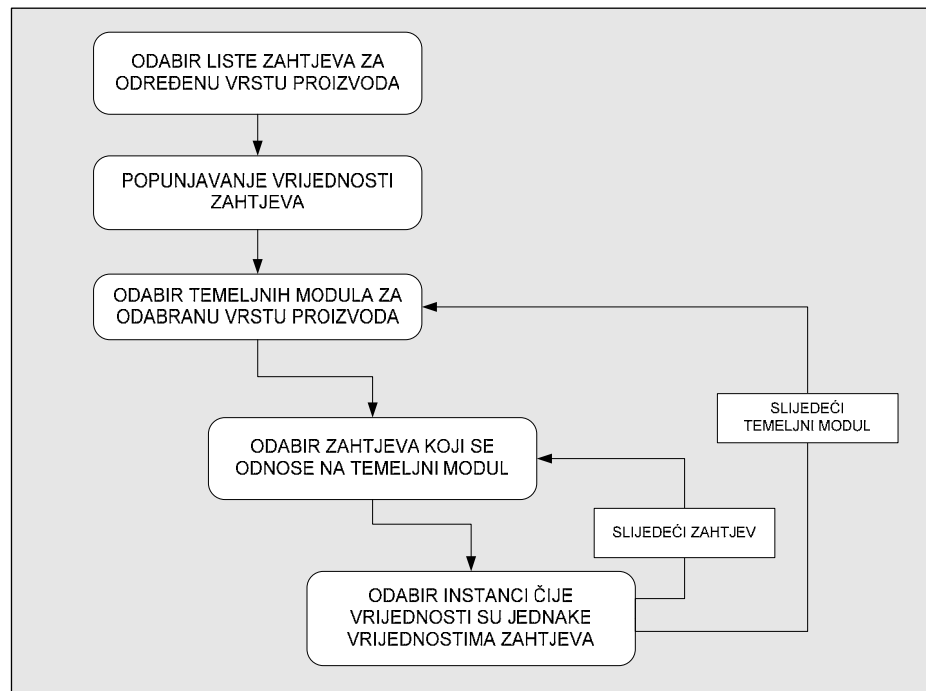


Slika 7 - 8: Tijek procesa konfiguriranja

Nakon što su odabrane sve temeljne, izborne i dodatne instance sustav generira sve moguće konfiguracije na temelju odabranih instanci. Konfiguracija se sastoji samo od jedne instance svakog modula. Za tako odabrane konfiguracije, provjerava se međusobna nekompatibilnost instanci modula unutar jedne konfiguracije. Konfiguracije kod kojih postoji nekompatibilnost između pojedinih instanci, označavaju se kao konfiguracije koje ne zadovoljavaju postavljene zahtjeve. Sve ostale konfiguracije predstavljaju rješenja procesa konfiguriranja od kojih je potrebno odabrati jedno rješenje kao konačno rješenje. Konačno rješenje predstavlja varijantu proizvoda za postavljene zahtjeve. U nastavku će se pobliže objasniti pojedini koraci procesa konfiguriranja.

7.2.2.1 Odabir instanci temeljnih modula prema zahtjevima

Postupak odabira instanci temeljnih modula prikazan je na slici [Slika 7 - 9]. Nakon odabira liste zahtjeva i popunjavanja vrijednosti za zahtjeve, sustav odabire sve temeljne module za odabranu listu zahtjeva. Za prvi odabrani temeljni modul (redoslijed odabira temeljnih modula objašnjen je u poglavlju 7.2.1.4), odabiru se svi zahtjevi koji se odnose na taj temeljni modul. Potom se odabire prvi odabrani zahtjev (redoslijed odabira zahtjeva objašnjen je u poglavlju 7.2.1.4) i pretražuju se sve instance temeljnog modula koje imaju istu vrijednost kao i odabrani zahtjev. Nakon odabira odgovarajućih instanci, instance temeljnog modula zapisuju se u tablicu *Product_component* (objašnjena u poglavlju 7.2.1.6), a sustav odabire slijedeći zahtjev za odabrani temeljni modul. Postupak se ponavlja, ali s tom razlikom da se svi slijedeći zahtjevi, istog temeljnog modula, odnose samo na instance temeljnog modula koje su odabrane u prethodnim zahtjevima.



Slika 7 - 9: Postupak odabira instanci temeljnih modula prema zahtjevima

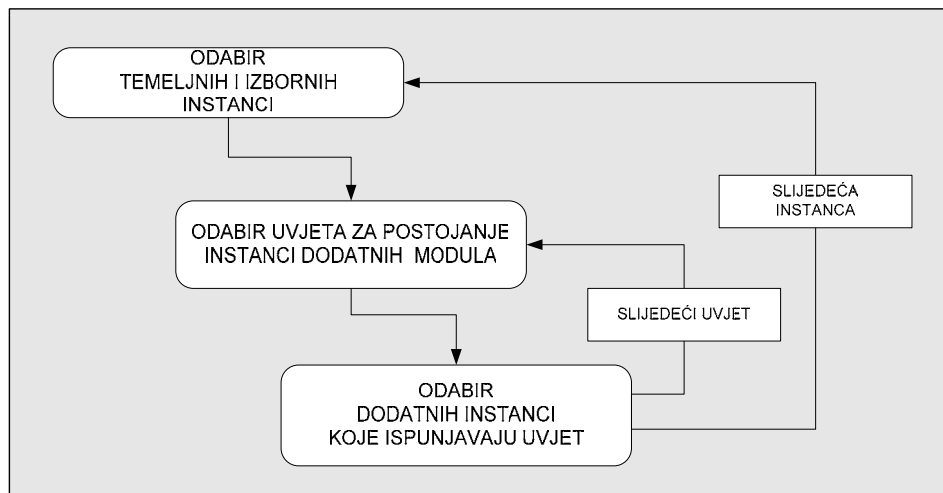
Nakon što su odabrane sve instance prema zahtjevima za jednim temeljnim modulom, sustav prelazi na slijedeći temeljni modul i postupak se ponavlja.

7.2.2.2 Odabir instanci izbornih modula prema zahtjevima

Postupak odabira instanci izbornih modula izvodi se na isti način kao i postupak odabira instanci temeljnih modula, s tom razlikom da se umjesto odabira temeljnih modula odabiru izborni moduli. Odabir izbornih modula vrši se na način da se odabiru samo moduli čija vrijednost atributa *Association_type* u tablici *Class_association* iznosi *option* (poglavlje 7.2.1.4) te da postoje vrijednosti zahtjeva za te module.

7.2.2.3 Odabir instanci dodatnih modula

Instance dodatnih modula odabiru se na temelju uvjeta koji su postavljeni za odabrane instance temeljnih i izbornih modula, a zapisani su u tablici *Specification_inclusion* (poglavlje 7.2.1.5). Postavljeni uvjet označava da je za ispravno korištenje temeljnih ili izbornih instanci modula potrebno i postojanje odgovarajućih dodatnih modula. Nakon odabira temeljne ili izborne instance modula i uvjeta koji se odnosi na odabranu instancu, sustav pretražuje sve instance dodatnih modula koje zadovoljavaju traženi uvjet. Instance koje zadovoljavaju traženi uvjet zapisuju se u tablicu *Product_component* (objašnjena u poglavlju 7.2.1.6), a sustav odabire slijedeći uvjet za odabranu instancu modula.



Slika 7 - 10: Postupak odabira instanci dodatnih modula

Odabirom svih instanci dodatnih modula za odabranu instancu, sustav prelazi na odabir novih instanci dodatnih modula za slijedeću instancu temeljnog ili izbornog modula. Ovim postupkom završava odabir svih instanci prema odabranoj listi zahtjeva i sustav prelazi na izradu konfiguracija od svih odabranih instanci.

7.2.2.4 Izrada konfiguracija od odabranih instanci

Rezultat procesa konfiguriranja je jedna ili više konfiguracija, a konfiguracija predstavlja proizvod koji je sastavljen od instanci modula. S obzirom da je svaka konfiguracija sastavljena od samo jedne instance pojedinog modula, ukupni broj konfiguracija koji se može dobiti od odabranih instanci jednak je kartezijevom produktu instanci svih modula. Kartezijev produkt [75] skupova X i Y je skup svih uređenih parova (x, y) , gdje je $x \in X$ i $y \in Y$, odnosno

$$X \times Y = \{(x, y) : x \in X \wedge y \in Y\}.$$

Kartezijev produkt instanci modula dobiven je na način da su moduli zapisani u obliku vektora, gdje pojedini elementi vektora čine instance modula. Primjer ovakvog zapisa je: $x \in \{1, 2\}$, $y \in \{3, 4, 5\}$ i $z \in \{6, 7\}$. U tablici [Tablica 7 - 1] prikazani je ukupni broj konfiguracija i raspored instanci po konfiguracijama za zadani primjer.

Tablica 7 - 1: Konfiguracije instanci modula

	Modul x	Modul y	Modul z
Konfiguracija 1	1	3	6
Konfiguracija 2	1	3	7
Konfiguracija 3	1	4	6
Konfiguracija 4	1	4	7
Konfiguracija 5	1	5	6
Konfiguracija 6	1	5	7
Konfiguracija 7	2	3	6
Konfiguracija 8	2	3	7
Konfiguracija 9	2	4	6
Konfiguracija 10	2	4	7
Konfiguracija 11	2	5	6
Konfiguracija 12	2	5	7

Ukupni broj konfiguracija jednak je umnošku svih instanci modula i iznosi $n_{konfig} = 2 \cdot 3 \cdot 2 = 12$. Broj ponavljanja instanci pojedinog modula jednak je količniku ukupnog broja konfiguracija s brojem instanci pojedinog modula i svih prethodnih modula. Primjer broja ponavljanja instanci pojedinih modula prikazan je u [Tablica 7 - 2].

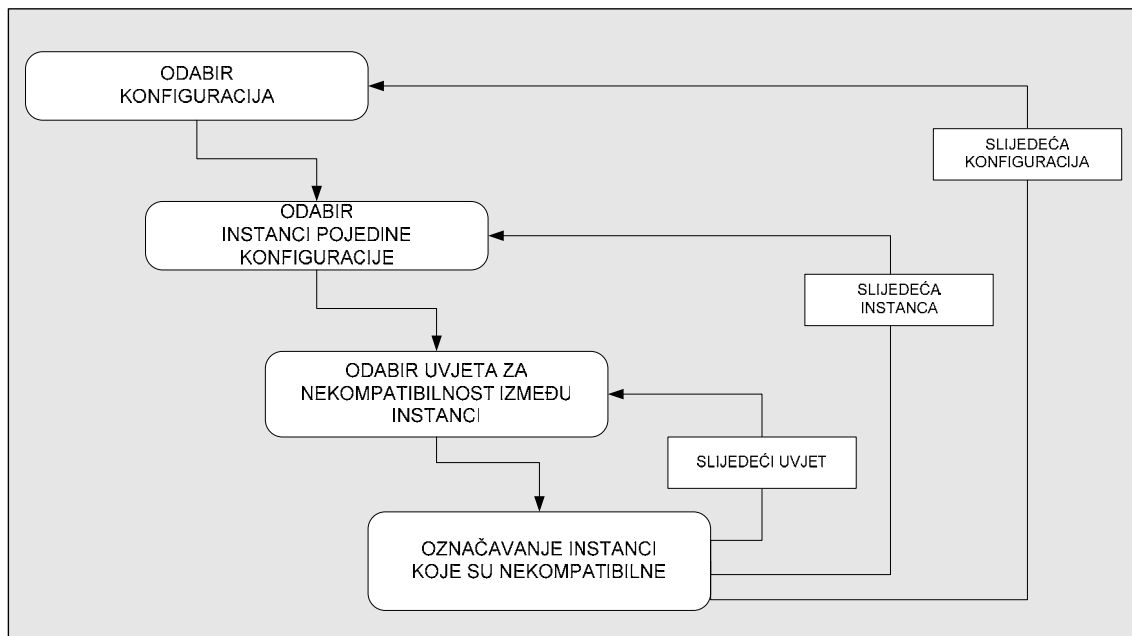
Tablica 7 - 2: Broj ponavljanja instanci modula

	Modul x	Modul y	Modul z
Broj ponavljanja instanci	$n_{x_pon} = \frac{12}{2} = 6$	$n_{y_pon} = \frac{12}{2 \cdot 3} = 2$	$n_{z_pon} = \frac{12}{2 \cdot 3 \cdot 2} = 1$

Poznavajući ukupni broj konfiguracija i broj ponavljanja instanci pojedinih modula, raspored instanci po konfiguracijama dobije se upisivanjem u stupce matrice instance pojedinih modula.

7.2.2.5 Provjera nekompatibilnosti između instanci pojedinih konfiguracija

Slika [Slika 7 - 11] prikazuje zadnji korak u procesu konfiguriranja, a odnosi se na provjeru nekompatibilnosti između instanci pojedinih konfiguracija. Nakon izračunavanja svih konfiguracija odabranih instanci modula, sustav odabire pojedine konfiguracije [Slika 7 - 11] te za svaku konfiguraciju odabire njezine instance. Prilikom odabira pojedine instance odabire se i uvjet nekompatibilnosti za tu instancu. Ukoliko uvjet nekompatibilnosti postoji za odabranu instancu, zapisan je u tablici *Specification_inclusion* (poglavlje 7.2.1.5). Na temelju tog uvjeta provjerava se da li je navedena instanca modula međusobno nekompatibilna s ostalim instancama u odabranoj konfiguraciji. Ukoliko je uvjet ispunjen (postoje najmanje dvije instance koje su međusobno nekompatibilne u odabranoj konfiguraciji), nekompatibilne instance se označuju. Za pojedinu instancu može postojati i više uvjeta nekompatibilnosti, te po završetku jednog uvjeta sustav prelazi na odabir drugih uvjeta. Nakon što su provjereni svi uvjeti za pojedinu instancu, sustav prelazi na odabir sljedeće instance odabrane konfiguracije. Postupak se ponavlja sve dok se ne provjere svi uvjeti nekompatibilnosti za sve konfiguracije.



Slika 7 - 11: Postupak provjere nekompatibilnosti između instanci

Konfiguracije koje u sebi sadrže međusobno nekompatibilne instance, moraju se odbaciti kao nevaljane konfiguracije. Na temelju preostalih konfiguracija (konfiguracije koje ne sadrže nekompatibilne instance), konstruktor odabire jednu konfiguraciju kao konačno rješenje te tada završava proces konfiguriranja.

7.3 Prikaz primjene sustava

Odabrani primjer, na kojemu će se testirati razvijeni sustav za konfiguriranje proizvoda modularne arhitekture, je asinkroni klizno-kolutni motor s trajno spuštenim četkicama. Poduzeće koje proizvodi ovakve motore i uz čiju suradnju je omogućeno prikupljanje podataka potrebnih za testiranje prikazanog sustava je "KONČAR - GiM" iz Zagreba. Asinkroni motori se zbog svoje strukture mogu smatrati modularnim proizvodom, iako proces modularizacije nije proveden prije testiranja prikazanog sustava. S obzirom da sami proces modularizacije nije tema ovog rada, navedeni proces nije realiziran. Zbog toga je preuzeta postojeća hijerarhijska struktura komponenata, a moduli su sačinjeni od osnovnih komponenata proizvoda.

7.3.1 Asinkroni strojevi

Asinkronim strojem nazivamo takav stroj izmjenične struje, kojem se brzina vrtnje rotora n , pri zadanoj frekvenciji struje mreže, mijenja u ovisnosti o opterećenju [76], a dobio je ime po tome što brzina rotacionog magnetskog toka i brzina rotora nije ista. Asinkroni stroj se najčešće primjenjuje kao motor, te se dijeli u dvije grupe:

- 1) klizno-kolutni motor (motor s faznim rotorom) i
- 2) kavezni motor (motor s kratkospojenim rotorskim namotom).

Klizno-kolutni motori dalje se dijele na:

- 1) motore s trajno spuštenim četkicama i
- 2) motore s podizačem četkica.

Primjer asinkronog klizno-kolutnog motora s trajno spuštenim četkicama prikazan je na slici [Slika 7 - 12].

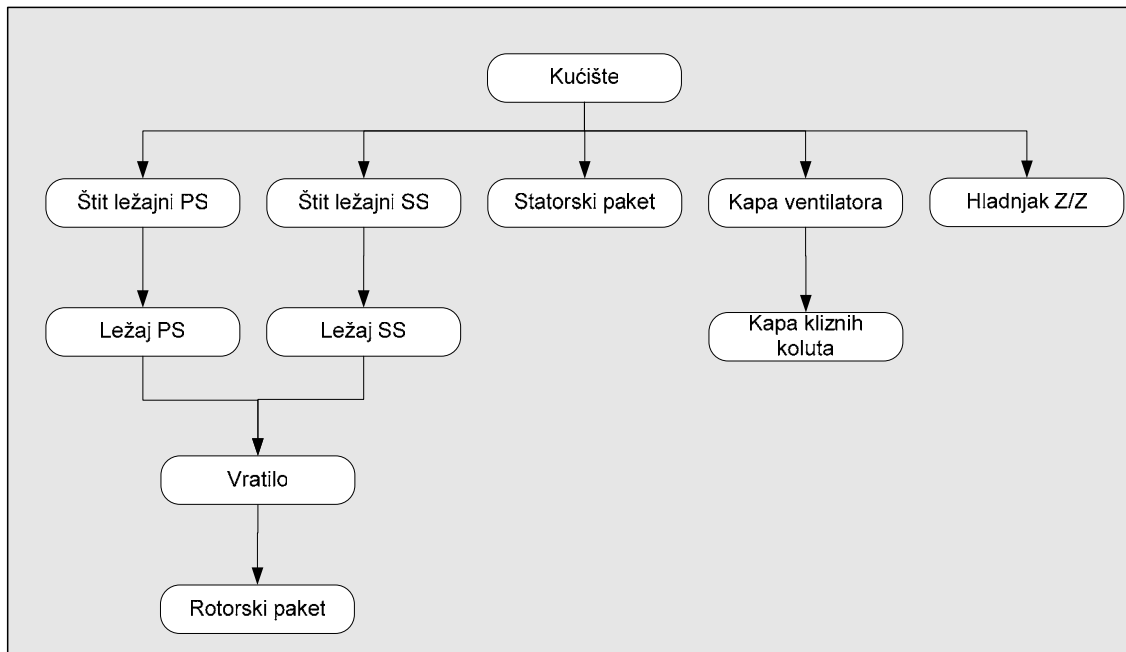


Slika 7 - 12: Trofazni asinkroni klizno-kolutni motor

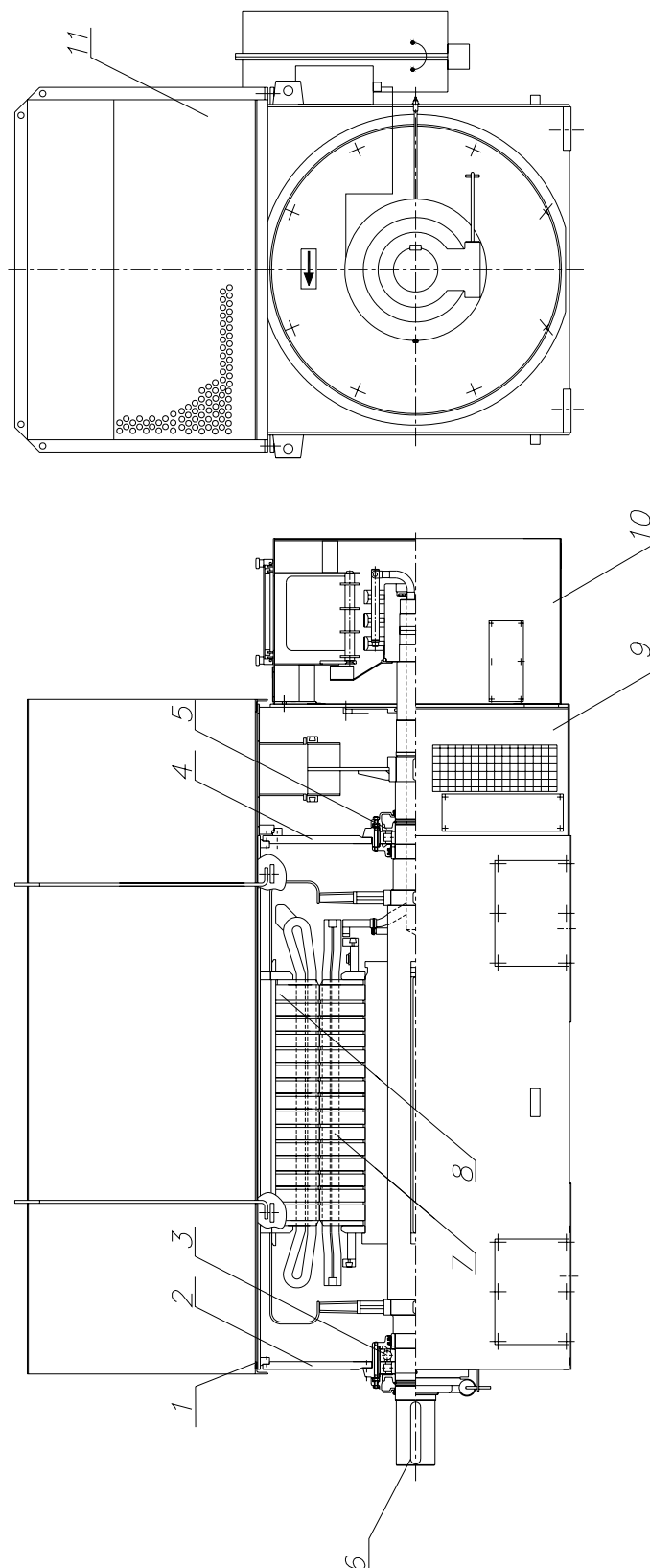
Radi boljeg razumijevanja korištenog primjera u sustavu za konfiguriranje

proizvoda, na slici [Slika 7 - 14] prikazan je dispozicijski crtež asinkronog klizno-kolutnog motora s trajno spuštenim četkicama u horizontalnoj izvedbi. Prilikom razmatranja trofaznog asinkronog klizno-kolutnog motora, odabrane su slijedeće osnovne komponente [Slika 7 - 13]:

- kućište - 1,
- štit ležajni PS (pogonska strana) - 2,
- ležaj PS (pogonska strana) - 3,
- štit ležajni SS (slobodna strana) - 4,
- ležaj SS (slobodna strana) - 5,
- vratilo - 6,
- rotorski paket - 7,
- statorski paket - 8,
- kapa ventilatora - 9,
- kapa kliznih koluta - 10 i
- hladnjak Z/Z (zrak/zrak) - 11.



Slika 7 - 13: Hijerarhijska struktura osnovnih komponenata kod klizno-kolutnog asinkronog motora



Slika 7 - 14: Dispozicijski crtež asinkronog klizno-kolutnog motora s trajno spuštenim četkicama

Osnovni preduvjet za korištenje sustava za konfiguriranje proizvoda modularne arhitekture, je da se proizvod sastoji od modula tj. da je proces modularizacije proveden. Proces modularizacije potrebno je učiniti na razini cijelog asinkronog motora, a ne samo za jednu vrstu klizno-kolutnog motora. Razlog je tome što se različite vrste asinkronih motora razlikuju s obzirom na postavljene zahtjeve, kao npr. postoje li podizači četkica?; Je li motor prisilno hlađen izmjenjivačem Zrak/Zrak?, itd. Promatrajući asinkroni motor na taj način, predložena podjela asinkronog klizno-kolutnog motora na module prikazana je u tablici [Tablica 7 - 3].

Tablica 7 - 3: Podjela na module kod klizno-kolutnog asinkronog motora

Osnovne komponente	Temeljni moduli	Izborni moduli	Dodatni moduli
<i>Kućište</i>	X		
<i>Štit ležajni SS</i>			X
<i>Ležaj SS</i>	X		
<i>Štit ležajni PS</i>			X
<i>Ležaj PS</i>	X		
<i>Vratilo</i>	X		
<i>Rotorski paket</i>	X		
<i>Statorski paket</i>	X		
<i>Kapa ventilatora</i>	X		
<i>Kapa kliznih koluta</i>		X	
<i>Hladnjak Z/Z</i>		X	

Osnovne komponente u tablici [Tablica 7 - 3] koje su odabrane kao temeljni moduli predstavljaju komponente od kojih je svaka varijanta asinkronog motora sastavljena. Komponente *Štit ležajni SS* i *Štit ležajni PS* svrstane su u dodatne module jer ovise o izboru kućišta. U *Štit ležajni SS i PS* umeću se ležajevi SS i PS koji se nalaze na vratilu. Funkcija *Štita ležajnog SS i PS* je osiguravanje uležištenja ležaja SS i PS te zatvaranje prostora između ležaja i kućišta. Zbog toga, vanjski promjer ležaja mora biti jednak unutarnjem promjeru štita, a vanjski promjer štita mora biti jednak unutarnjem promjeru kućišta. Kapa kliznih koluta odabrana je kao izborni modul, jer kod kaveznog asinkronog motora ta komponenta ne postoji. Također je i komponenta *Hladnjak Z/Z* odabrana kao izborni modul jer egzistira u varijanti samo ako postoji takav zahtjev.

U dogovoru s poduzećem "KONČAR - GiM", stvarna imena instanci modula tj. brojevi crteža, nisu korištena u realiziranom primjeru zbog tajnosti podataka.

Tablica 7 - 4: Popis vrijednosti instanci modula

Instance modula	Opis vrijednosti	Vrijed.	Jed.
<i>Hladnjak_ZZ_1</i>	Zrak	<i>DA</i>	
<i>Kapa_1</i>	KAPA_ventilatora	<i>DA</i>	
<i>Kapa_kliznih_koluta_1</i>	KAPA_KK_spustene_cetkice	<i>DA</i>	
<i>Kuciste_1</i>	KUCISTE_Promjer_otvora	<i>940</i>	<i>mm</i>
	KUCISTE_Oznaka	<i>M</i>	
<i>Lezaj_PS_1</i>	LEZAJ_PS1_unutarnji_promjer	<i>130</i>	<i>mm</i>
	LEZAJ_PS1_vanjski_promjer	<i>280</i>	<i>mm</i>
<i>Lezaj_PS_2</i>	LEZAJ_PS2_unutarnji_promjer	<i>150</i>	<i>mm</i>
	LEZAJ_PS2_vanjski_promjer	<i>300</i>	<i>mm</i>
<i>Lezaj_SS_1</i>	LEZAJ_SS1_unutarnji_promjer	<i>130</i>	<i>mm</i>
	LEZAJ_SS1_vanjski_promjer	<i>280</i>	<i>mm</i>
<i>Lezaj_SS_2</i>	LEZAJ_SS2_unutarnji_promjer	<i>150</i>	<i>mm</i>
	LEZAJ_SS2_vanjski_promjer	<i>300</i>	<i>mm</i>
<i>Vratilo_1</i>	VRATILO_Promjer_na_lezaju	<i>130</i>	<i>mm</i>
	VRATILO_Vanjski_promjer_rebra	<i>325</i>	<i>mm</i>
	VRATILO_Visina	<i>500</i>	<i>mm</i>
	VRATILO_Promjer_na_rotoru	<i>180</i>	<i>mm</i>
<i>Vratilo_2</i>	VRATILO_Promjer_na_lezaju	<i>150</i>	<i>mm</i>
	VRATILO_Vanjski_promjer_rebra	<i>325</i>	<i>mm</i>
	VRATILO_Visina	<i>500</i>	<i>mm</i>
	VRATILO_Promjer_na_rotoru	<i>180</i>	<i>mm</i>
<i>Rotorski_paket_1</i>	ROTOR_Vanjska_mjera	<i>616</i>	<i>mm</i>
	ROTOR_Unutarnji_provrt	<i>325</i>	<i>mm</i>
	ROTOR_Broj_kanala	<i>14</i>	<i>kom</i>
	ROTOR_Ukupna_duzina	<i>760</i>	<i>mm</i>
<i>Statorski_paket_1</i>	STATOR_Vanjska_mjera	<i>900</i>	<i>mm</i>
	STATOR_Broj_kanala	<i>14</i>	<i>kom</i>
	STATOR_Unutarnji_provrt	<i>620</i>	<i>mm</i>
	ROTOR_Ukupna_duzina	<i>760</i>	<i>mm</i>
<i>Stit_lezajni_PS_1</i>	STIT_PS1_Vanjski_promjer	<i>940</i>	<i>mm</i>
	STIT_PS1_Unutarnji_promjer	<i>280</i>	<i>mm</i>
<i>Stit_lezajni_SS_1</i>	STIT_SS1_Unutarnji_promjer	<i>280</i>	<i>mm</i>
	STIT_SS1_Vanjski_promjer	<i>940</i>	<i>mm</i>

Imena instanci modula sačinjena su od imena modula i broja koji označava broj instance. Npr. *Vratilo_1* je ime za instancu modula *Vratilo* i ujedno označava prvu instancu tog modula. U tablici [Tablica 7 - 4] prikazane su vrijednosti koje su pridružene instancama modula, a koje su korištene u realiziranom sustavu i zapisane u tablicama baze podataka. Svaki od modula *Vratilo*, *Ležaj_SS* i *Ležaj_PS* sadrži po dvije instance, dok ostali moduli sadrže samo po jednu instancu. Ovakvom podjelom instanci modula omogućeno je da rezultat procesa konfiguriranja nije samo jedna konfiguracija već nekoliko konfiguracija.

7.3.2 Korisnička sučelja

Komunikacija između korisnika i sustava za konfiguriranje proizvoda modularne arhitekture odvija se preko korisničkih sučelja, koja su izrađena korištenjem programa Dreamweaver MX. Na slici [Slika 7 - 15] je prikazano korisničko sučelje za odabir predloška asinkronog kaveznog motora. Odabrani je 6AKZ predložak ili predložak asinkronog kaveznog motora s trajno spuštenim četkicama.



Slika 7 - 15: Prikaz korisničkog sučelja za odabir predloška

Nakon odabira predloška proizvoda, potrebno je unijeti vrijednosti za zahtjeve koji su određeni odabranim predloškom [Slika 7 - 16].

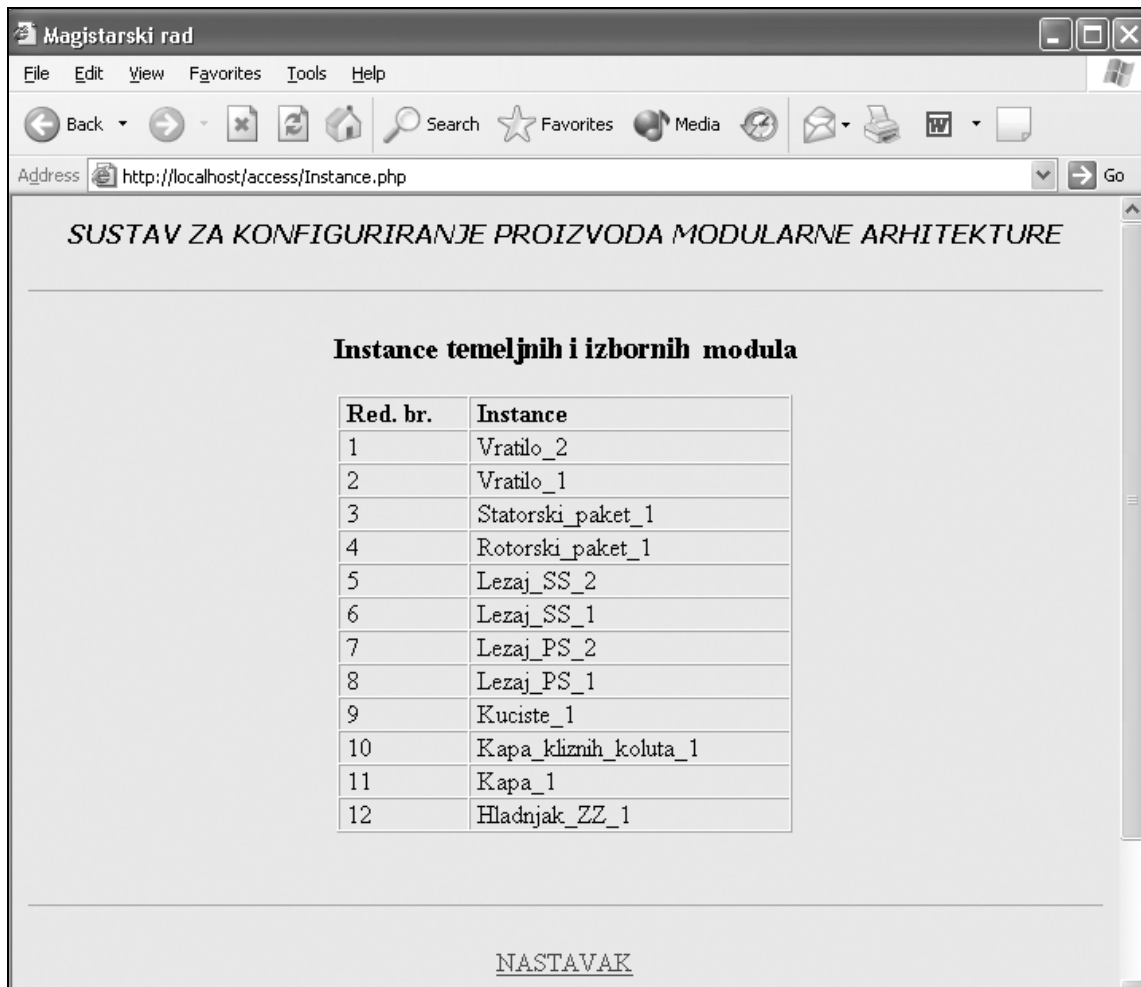
ID	Tekst zahtjeva	Vrijednost zahtjeva	Jedinica
01	Napon iznosi	5500	V
02	Snaga iznosi	1000	kW
03	Broj okretaja iznosi	990	1/min
04	Zrakom hlađeni motor	DA	
05	Visina osovine iznosi	500	mm
06	Oznaka kućišta iznosi	M	
07	Ukupna dužina statora iznosi	760	mm
08	Broj kanala za hlađenje statora iznosi	14	kom
09	Unutarnji provrt statora iznosi	620	mm
10	Vanjska mjera statora iznosi	900	mm
11	Ukupna dužina rotora iznosi	760	mm
12	Broj kanala rotora iznosi	14	kom
13	Unutarnji provrt rotora iznosi	325	mm
14	Vanjska mjera rotora iznosi	616	mm
15	Promjer otvora kućišta iznosi	940	mm
16	Promjer vratila na ležaju iznosi		mm
17	Promjer vratila na rotoru iznosi	180	mm
18	Vanjski promjer rebra osovine iznosi	325	mm
19	Vanjski promjer štita iznosi	940	mm
20	Unutarnji promjer štita iznosi	280	mm

NASTAVAK

Slika 7 - 16: Prikaz korisničkog sučelja za popunjavanje vrijednosti zahtjeva za odabranu listu zahtjeva

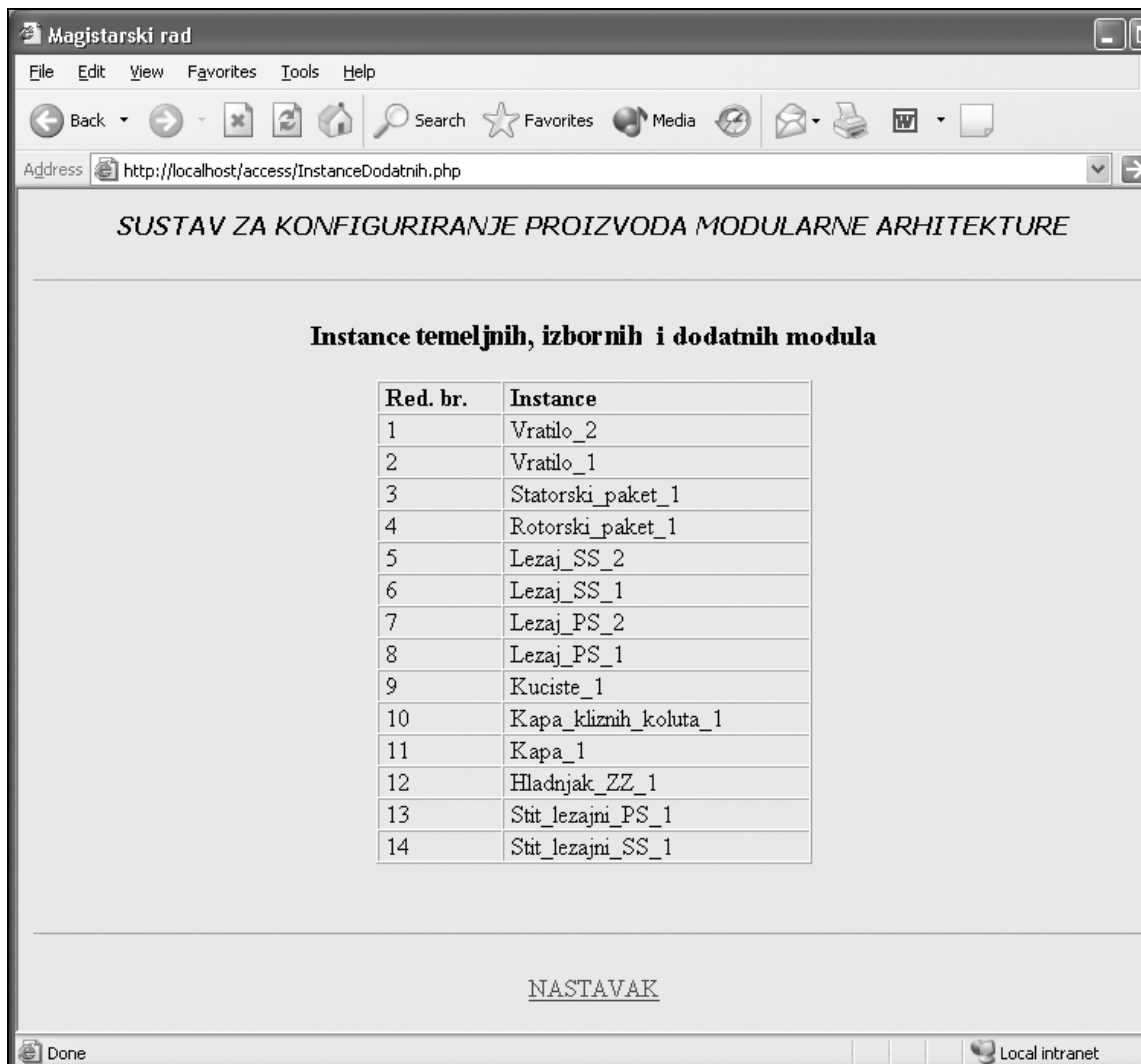
Za zahtjev broj 16 (Promjer vratila na ležaju) na slici [Slika 7 - 16] nije unijeta nikakva vrijednost, jer se time htjelo postići odabiranje svih instanci modula *Vratila* čiji promjer vratila na rotoru (zahtjev broj 17) iznosi 180 mm, a bez obzira na vrijednost promjera vratila na ležaju. U tablici [Tablica 7 - 4] opisane su dvije

instance modula *Vratilo*, *Vratilo_1* i *Vratilo_2* koje ispunjavaju zadani zahtjev. Zahtjev broj 16 utječe i na odabir instanci modula *Ležaj_PS* i *Ležaj_SS*, te s obzirom da nije unijeta vrijednost tog zahtjeva odabrane su instance *Lezaj_PS_1* i *Lezaj_PS_2* za modul *Lezaj_PS* te instance *Lezaj_SS_1* i *Lezaj_SS_2* za modul *Lezaj_SS*. Za ostale temeljne i izborne module postoji samo po jedna instanca pojedinih modula, čije vrijednosti odgovaraju zadanim zahtjevima. Odabrane instance temeljnih i izbornih modula prikazane su na slici [Slika 7 - 17] .



Slika 7 - 17: Prikaz korisničkog sučelja za odabrane instance temeljnih i izbornih modula

Odabir instanci dodatnih modula ovisi o odabranoj instanci *Kuciste_1*. Uvjet koji mora biti ispunjen za odabir instanci dodatnih modula glasi da promjer otvora kućišta mora biti jednak vanjskom promjeru štita. Vrijednost promjera otvora kućišta iznosi 940 mm [Tablica 7 - 4], pa su stoga odabrane instance dodatnih modula *Stit_lezajni_PS_1* i *Stit_lezajni_SS_1* [Tablica 7 - 4]. Odabrane instance dodatnih modula prikazane su na slici [Slika 7 - 18] pod rednim brojem 13 i 14.



Slika 7 - 18: Prikaz korisničkog sučelja za odabrane instance temeljnih, izbornih i dodatnih modula

Na temelju odabranih instanci temeljnih, izbornih i dodatnih modula dobiju se sve moguće konfiguracije. Moduli i instance modula mogu se zapisati u vektorskom obliku [7.2.2.4] i to na sljedeći način:

- $Statorski_paket \in \{Statorski_paket_1\}$,
- $Rotorski_paket \in \{Rotorski_paket_1\}$,
- $Vratilo \in \{Vratilo_1, Vratilo_2\}$,
- $Lezaj_SS \in \{Lezaj_SS_1, Lezaj_SS_2\}$,
- $Lezaj_PS \in \{Lezaj_PS_1, Lezaj_PS_2\}$,
- $Kućiste \in \{Kućiste_1\}$,
- $Kapa_kliznih_koluta \in \{Kapa_kliznih_koluta_1\}$,
- $Kapa \in \{Kapa_1\}$,

- $Hladnjak_Z / Z \in \{Hladnjak_Z / Z_1\}$,
- $Stit_lezajni_PS \in \{Stit_lezajni_PS_1\}$ i
- $Stit_lezajni_SS \in \{Stit_lezajni_SS_1\}$.

Ukupan broj konfiguracija dobiva se izračunavanjem kartezijevog produkta svih modula i iznosi $n_{konfig} = 1 \cdot 1 \cdot 2 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 8$. Na slici [Slika 7 - 19] su prikazane sve konfiguracije s odabranim instancama.

Broj konfig	Instanca	Broj konfig	Instanca	Broj konfig	Instanca	Broj konfig	Instanca
1	Stator1	3	Stator1	5	Stator1	7	Stator1
1	Stit_lezajni_PS_1	3	Stit_lezajni_PS_1	5	Stit_lezajni_PS_1	7	Stit_lezajni_PS_1
1	Lezaj_PS_1	3	Lezaj_PS_1	5	Lezaj_PS_2	7	Lezaj_PS_2
1	Rotor_1	3	Rotor_1	5	Rotor_1	7	Rotor_1
1	Kuciste_1	3	Kuciste_1	5	Kuciste_1	7	Kuciste_1
1	Lezaj_SS_2	3	Lezaj_SS_1	5	Lezaj_SS_2	7	Lezaj_SS_1
1	Osovina_2	3	Osovina_2	5	Osovina_2	7	Osovina_2
1	Stit_lezajni_SS_1	3	Stit_lezajni_SS_1	5	Stit_lezajni_SS_1	7	Stit_lezajni_SS_1
1	Hladnjak_ZZ_1	3	Hladnjak_ZZ_1	5	Hladnjak_ZZ_1	7	Hladnjak_ZZ_1
1	Kapa_1	3	Kapa_1	5	Kapa_1	7	Kapa_1
1	Kapa_kliznih_koluta_1	3	Kapa_kliznih_koluta_1	5	Kapa_kliznih_koluta_1	7	Kapa_kliznih_koluta_1
2	Stator1	4	Stator1	6	Stator1	8	Stator1
2	Stit_lezajni_PS_1	4	Stit_lezajni_PS_1	6	Stit_lezajni_PS_1	8	Stit_lezajni_PS_1
2	Lezaj_PS_1	4	Lezaj_PS_1	6	Lezaj_PS_2	8	Lezaj_PS_2
2	Rotor_1	4	Rotor_1	6	Rotor_1	8	Rotor_1
2	Kuciste_1	4	Kuciste_1	6	Kuciste_1	8	Kuciste_1
2	Lezaj_SS_2	4	Lezaj_SS_1	6	Lezaj_SS_2	8	Lezaj_SS_1
2	Osovina_1	4	Osovina_1	6	Osovina_1	8	Osovina_1
2	Stit_lezajni_SS_1	4	Stit_lezajni_SS_1	6	Stit_lezajni_SS_1	8	Stit_lezajni_SS_1
2	Hladnjak_ZZ_1	4	Hladnjak_ZZ_1	6	Hladnjak_ZZ_1	8	Hladnjak_ZZ_1
2	Kapa_1	4	Kapa_1	6	Kapa_1	8	Kapa_1
2	Kapa_kliznih_koluta_1	4	Kapa_kliznih_koluta_1	6	Kapa_kliznih_koluta_1	8	Kapa_kliznih_koluta_1

NASTAVAK

Slika 7 - 19: Prikaz korisničkog sučelja za dobivene konfiguracije

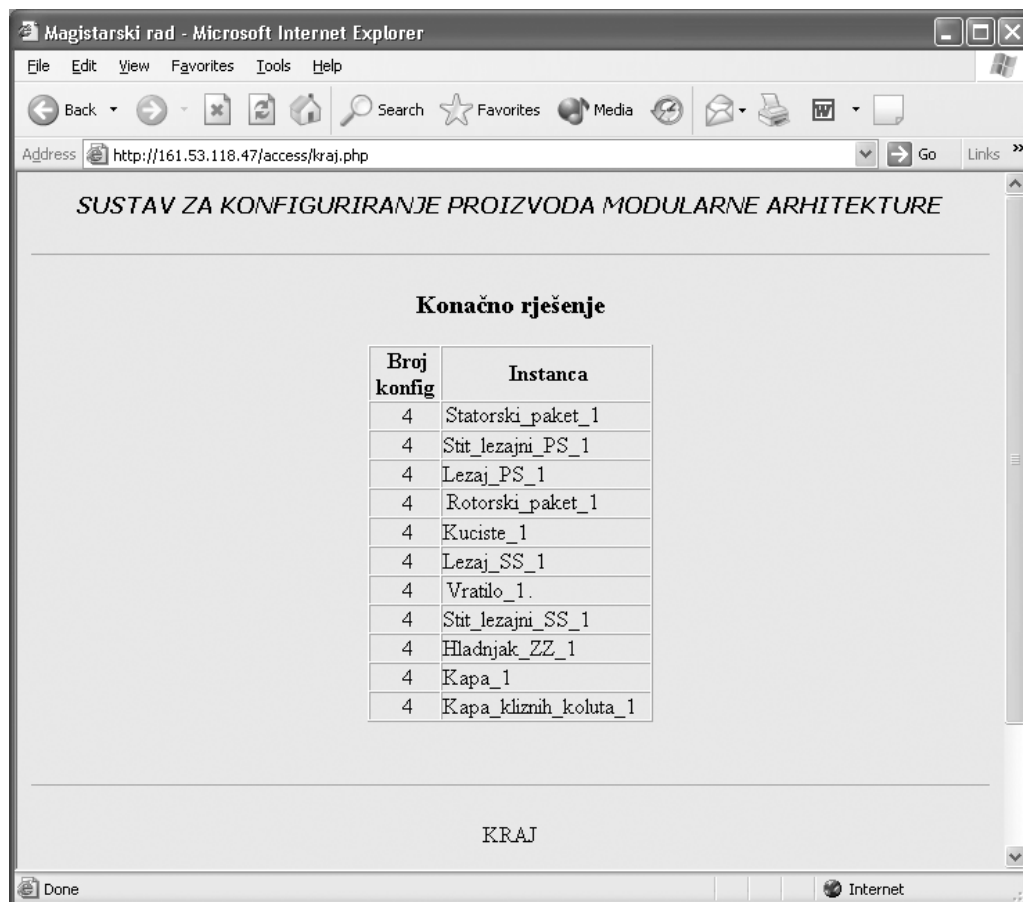
Za dobivene konfiguracije potrebno je provjeriti uvjete nekompatibilnosti između instanci pojedinih konfiguracija. Uvjeti nekompatibilnosti glase:

- 1) *vrijednost unutarnjeg promjera kod štita na pogonskoj strani mora biti jednaka vrijednosti vanjskog promjera kod ležaja na pogonskoj strani* - instance koje ne ispunjavaju ovaj uvjet tj. koje su međusobno nekompatibilne jesu *Stit_lezajni_PS_1* i *Lezaj_PS_2*. Konfiguracije 5, 6, 7 i 8 sadrže navedene instance te se označuju kao konfiguracije koje sadrže nekompatibilne instance;
- 2) *vrijednost unutarnjeg promjera kod štita na slobodnoj strani mora biti jednaka vrijednosti vanjskog promjera kod ležaja na slobodnoj strani* - instance koje ne ispunjavaju ovaj uvjet tj. koje su međusobno

nekompatibilne jesu *Stit_lezajni_SS_1* i *Lezaj_SS_2*. Konfiguracije 1 i 2 sadrže navedene instance te se označavaju kao konfiguracije koje sadrže nekompatibilne instance. Konfiguracije 5 i 6 također sadrže nekompatibilne instance, ali budući su već označene kao konfiguracije koje sadrže nekompatibilne instance, svi ostali uvjeti se ne provjeravaju za označene konfiguracije;

- 3) *vrijednost unutarnjeg promjera kod ležaja na pogonskoj strani mora biti jednaka vrijednosti promjera vratila na ležaju na pogonskoj strani* - instance koje ne ispunjavaju ovaj uvjet tj. koje su međusobno nekompatibilne jesu *Lezaj_PS_1* i *Vratilo_2*. Od preostalih konfiguracija, jedino konfiguracija 3 sadrži nekompatibilne instance te se označava kao konfiguracija koje sadrži nekompatibilne instance;
- 4) *vrijednost unutarnjeg promjera kod ležaja na slobodnoj strani mora biti jednaka vrijednosti promjera vratila na ležaju na slobodnoj strani*.

Konfiguracija broj 4 jedina sadrži instance koje ispunjavaju sve zadane uvjete, te predstavlja konačno rješenje tj. varijantu proizvoda prilagođenu zadanim zahtjevima [Slika 7 - 20].



Slika 7 - 20: Prikaz korisničkog sučelja za odabranu varijantu proizvoda prilagođenu zadanim zahtjevima

8

Zaključak

8.1 Rezultati rada

U današnjim uvjetima globalizacije tržišta, sve je veći broj proizvoda koji su prilagođeni zahtjevima naručitelja. Osim toga, životni se vijek proizvoda neprestano skraćuje. Proizvodi su sve kompleksniji, povećava se broj varijanata proizvoda te je ukupno vrijeme potrebno da se proizvod pojavi na tržištu sve kraće. Ove činjenice su potaknule istraživanje ovog rada. Prema postavljenom je zadatku rada trebalo istražiti i realizirati sustav za konfiguriranje proizvoda modularne arhitekture koji će biti podrška konstruktoru u procesu konfiguriranja proizvoda prema zahtjevima naručitelja.

U drugom poglavlju rada, prikazan je pregled pojmova u području varijabilnosti proizvoda. Objašnjeni su pojmovi familije proizvoda, platforme proizvoda i arhitekture proizvoda. Posebno je stavljen naglasak i objašnjena je modularna arhitektura proizvoda jer predstavlja sadašnjost ali i budućnost u razvoju proizvoda prilagođenih zahtjevima naručitelja. Jedan je od načina realiziranja prilagodbe proizvoda, prema zahtjevima naručitelja, i primjena procesa konfiguriranja u procesu konstruiranja proizvoda. Analiza procesa konfiguriranja, kao i područja koja proces konfiguriranja obuhvaća, a to su konfigurabilni proizvodi, konfiguracijski sustavi, konfiguracijsko znanje i zahtjevi naručitelja, opisani su u trećem i četvrtom poglavlju rada. Ukratko su prikazana glavna teoretska područja: teorija tehničkih sustava, aksiomska teorija konstruiranja te teorija svojstva. Prikazana je također i metodologija razvijanja modula (MDF) koja nije korištena u ovom radu, ali se odnosi na proces modularizacije pa ju je, po mišljenju autora, potrebno spomenuti u radu. Proces modularizacije nije proveden na testiranom primjeru, jer nije predviđeno postavljenim zadatkom, ali je proces modularizacije

potrebno provesti prije stvarne primjene predloženog sustava u industriji.

Kao osnova sustava koji je predmet istraživanja, definiran je informacijski model za konfiguriranje proizvoda modularne arhitekture, te on predstavlja jezgru oko koje je sustav izgrađen. Predloženi informacijski model definiran je u skladu s aplikacijskim protokolom 214, koji je dio ISO 10303 - STEP standarda. Razlog odabira STEP standarda za opisivanje informacijskog modela je u tome što AP 214 opisuje područje razmjene podataka za procese u auto industriji. Pojedini dijelovi tog standarda opisuju i područja koja proces konfiguriranja obuhvaća. Osnovni dijelovi informacijskog modela sadrže entitete za opisivanje:

- zahtjeva i lista zahtjeva naručitelja,
- familije proizvoda i varijante proizvoda,
- značajki varijanta proizvoda i zahtjeva naručitelja,
- vrsta modula,
- instanci modula i
- strukture varijanti proizvoda.

Dio informacijskog modela koji sadrži entitete za opisivanje zahtjeva i liste zahtjeva naručitelja nije obuhvaćen u postojećem aplikacijskom protokolu AP 214, te stoga predstavlja mogućnost proširenja postojećeg aplikacijskog protokola.

Na osnovu predloženog informacijskog modela realizirana je računalna implementacija sustava za konfiguriranje proizvoda modularne arhitekture. Za izbor radnog okruženja računalne implementacije sustava odabrani je mrežni informacijski servis - internet. To je uveliko utjecalo na odabir računalne implementacije sustava. Računalna implementacija sustava realizirana je na osnovama 3-slojne klijent-poslužitelj arhitekture te su opisane značajke pojedinih razina arhitekture: razine poslovne logike, razine trajnog zapisa podataka i klijentske razine. Alati korišteni za realizaciju razine poslovne logike jesu PHP program i SQL jezik, a za realizaciju razine trajnog zapisa podataka korištena je relacijska baza podataka MS Access. Korisnička sučelja preko kojih korisnik komunicira sa sustavom, izrađena su korištenjem programa Dreamweaver MX. Realizirani sustav za konfiguriranje proizvoda modularne arhitekture testiran je na realnom primjeru asinkronog klizno-kolutnog motora s trajno spuštenim četkicama. U bazi podataka su zapisani podaci koji opisuju testirani primjer, te su unijeti podaci o zahtjevima i listi zahtjeva koje se odnose na testirani primjer. Vrijednosti zahtjeva korištenih u testnom primjeru odgovaraju stvarnim vrijednostima. Zbog toga je sustav na temelju unesenih vrijednosti zahtjeva, kao rješenje procesa konfiguriranja, dobio samo jednu konfiguraciju što odgovara stvarnim rezultatima. Dobiveni rezultat prikazuje ispravnost predloženog sustava za konfiguriranje proizvoda modularne arhitekture.

8.2 Smjerovi daljnjeg istraživanja

Nastavak istraživanja moguće je provesti u nekoliko različitih smjerova. Prvi smjer daljnjeg istraživanja odnosi se na istraživanje usmjereno na proces modularizacije. Do sada postoji samo nekoliko metoda, od kojih je metoda opisana u poglavlju 4.5 najpoznatija, koje opisuju načine podijele strukture proizvoda u module. Zbog toga postoji još dovoljno prostora, a i potrebe za istraživanjem u procesu modularizacije.

Drugi smjer istraživanja moguće je usmjeriti na istraživanje koje će se baviti računalnom podrškom zapisa konfiguracijskog znanja. Trenutačno je baš istraživanje zapisa znanja jedno od vodećih područja u istraživanjima u svijetu. Najveći nedostatak zapisa znanja pomoću pravila i ograničenja je upravo održavanje takvog zapisa. Česte promjene znanja (uzrokovane promjenama na tržištu ili promjenama tehnologije) dovode do velikih promjena u računalnom modelu.

Treći smjer daljnjeg istraživanja odnosi se na integrirani razvoj modularnih proizvoda. Integrirani razvoj modularnih proizvoda obuhvaća proširenje postojećeg informacijskog modela za konfiguriranje proizvoda s dodatnim informacijskim modelom koji će opisati sklapanje 3D modela instanci modula u različitim CAD sustavima. Ovakvim proširenjem želi se postići to da se struktura konfiguracije, koja je odabrana kao konačno rješenje, može prikazati kao 3D model u CAD sustavu. Nakon što se na temelju strukture konfiguracije dobije 3D model te konfiguracije, korištenjem alata 3D sustava, omogućilo bi se automatsko dobivanje tehničke dokumentacije. Na taj način, integriranim razvojem modularnog proizvoda, bilo bi moguće zaokružiti proces konstruiranja modularnog proizvoda od liste zahtjeva do tehničke dokumentacije. Pritom je važno voditi računa da se sklapanje 3D modela instanci modula može odvijati u različitim 3D CAD sustavima. Time bi se zadržala neutralnost odabira 3D CAD sustava.

9

Literatura

-
- [1] O'Grady, P., "The Age Of Modularity: Using The New World Of Modular Products To Revolutio Your Corporation", Adams and Steele Publishers, 1999.
 - [2] Riitahuhta, A., Tiihonen, J., Lehtonen, T., Soininen, T., Pulkkinen, A., Sulonen, R., "Modeling Configurable Product Families", Proceedings of the 4th WDK Workshop on Product Structuring, Delft, 1998.
 - [3] Blessing, L.T.M., Chakrabarti, A., Wallace, K.M., "An Overview of Descriptive Studies in Relation to a General Design Research Methodology", Designers - the Key to Successful Product Development, Springer Verlag, 1998.
 - [4] Liedholm, U.,, "Conceptual Design of Products and Product Families", Proceedings of the 4th WDK Workshop on Product Structuring, Delft, 1998.
 - [5] Tichem, M., Andreasean, M. M., Riitahuhta, A., "Design of Product Families", Proceedings of ICED '99, Munich, 1999.
 - [6] Simpson, W.T., "A Concept Exploration Method for Product Family Design", Georgia Institute of Technology, 1998.
 - [7] O'Sullivan, B., "Supporting the Design of Product Families through Constraint-Based Reasoning", Proceedings of the 4th WDK Workshop on Product Structuring, Delft, 1998.
 - [8] Andreasen, M.M., McAloone, T., Mortensen, N.H., " Multi-Product Development - platforms and modularization", Technical University of Denmark, ISBN: 87-90130-34-0, Lyngby, 2001.
 - [9] Elgrad, P., "Designing Product Families", Proceedings of the 13th IPS Research
-

Seminar, Fuglsoe, 1998.

- [10] Sanderson, S.W., Uzumeri, M., "The Innovation Imperative – Strategies for Managing Product Models and Families", Irwin, ISBN: 078631009X, 1997.
- [11] Hildre, H.P., "Mastering Product Variety", Proceedings of the 2th WDK Workshop on Product Structuring, Delft, 1996.
- [12] Andreasen, M.M., "Design for Assembly", IFS Publication & Springer Verlag, 1988.
- [13] Robertson, D., Ulrich, K.T., "Planning for Product Platforms", Sloan Management Review, Summer, 1998.
- [14] Meyer, M.H., Lehnerd, A.P., "The Power of Product Platforms", The Free Press, ISBN 0-684-82580-5, New York, 1997.
- [15] Gonzalez-Zugasti, P.J., Otto, N.K., Baker, D.J., "Assessing Value in Platformed Product Family Design", Research in Engineering Design, vol. 13, pp. 30 - 41.
- [16] Pedersen, P.E., "Organizational Impacts of Platform Based Product Development", Proceedings of ICED '99, Munich, 1999.
- [17] Tichem, M., Storm, T., Andreasen M.M., MacCallum, K.J., "Product Structuring, an Overview", Proceedings of ICED '97, Tampere, 1997.
- [18] Hubka, V., Eder, W.E., "Theory of Technical System", Springer-Verlag, Berlin, 1988.
- [19] Andreasen, M.M., "Machine Design Methods Based on Systematic Approach – Contribution to a Design Theory", Department of Control and Engineering Design, Lund Institute of Technology, Sweden, 1980.
- [20] Štorga, M., "Sustav za razmjenu i upravljanje informacijama o proizvodu", Magistarski rad, Fakultet strojarstva i brodogradnje, Zagreb, 2002.
- [21] Riitahuhta, A., Andreasen M.M., "Configuration by Modularization, Proceeding of NordDesign '98, Stockholm, Sweden, 1998.
- [22] Riitahuhta, A., Pulkkinen, A., Andreasen M.M., "Metrics for Supporting the Use of Modularization in IPD", Proceedings of the 4th WDK Workshop on Product Structuring, Delft, 1998.
- [23] Erens, F., Verhulst, K., "Architecture for Product Families", Proceedings of the 2th WDK Workshop on Product Structuring, Delft, 1996.
- [24] Ulrich, K.T., Eppinger, S.D., "Product Design and Development", McGraw-Hill, New York, 1995.
- [25] Erens, F.J., "The Synthesis of Variety - Developing Product Families", PhD Thesis, TU Eindhoven, 1996.
- [26] Yu, J.S., Gonzalez-Zugasti, P.J., Otto, N.K., "Product Architecture Definition

- Based Upon Customer Demands", Proceedings of 1998 ASME Design Theory and Methodology Conference, Atlanta, 1998.*
- [27] Simpson T.W., Maier J.R.A., Mistree, F., "Product platform design: method and application", *Research in Engineering Design*, vol. 13, pp. 2 - 22.
- [28] Pahl, G., Beitz, W., "Engineering Design a systematic approach", The Design Council, London, 1988.
- [29] O'Grady, P., Liang, W.Y., Tseng, T.L., Huang, C.C., Kusiak, A., "Remote Collaborative Design With Modules", Technical Report TR 97-03, University of Iowa, 1997.
- [30] Sanchez, R., "Strategic Product Creation: Managinig New Interactions of Technology, Markets and organizations", *European Management Journal*, Vol. 14, pp. 121-138
- [31] Ulrich, K., Tung K., "Fundamentals of product modularity", ASME, 1991.
- [32] Tiihonen, J., Soininen, T., "Product Configurators - Information System Support for Configurable Products", Helsinki University of Technology, Finland.
- [33] Hales, H.L., "Automating and Integrating the Sales Function: How to Profit From Complexity and Customization", *Enterprise Integration Strategies*, Vol. 9, no. 11, pp. 1-9.
- [34] Carson, C., "Intelligent Sales Configuration", PC AI, 1997.
- [35] Seelmann-Eggebert, R., Schenk, M., "New Methodologies for Implementing Mass Customization" *Proceedings of the TMC 2002*, Wuhan, China, 2002.
- [36] Riitahuhta, A., "Views and Experiences of Configuration Management", *Design for Configuration - a Debate based on the 5th WDK Workshop on Product Structuring*, Springer, ISBN 3-540-67739-9, 2001.
- [37] Peltonen, H., "Concepts and an Implementation for Product Data Management", PhD Thesis, Helsinki University of Technology, 2000.
- [38] Yu, B., "A Virtual Configuration Workbench for Product Development", PhD Thesis, CAD Center, Department of Design, Manufacture and Engineering Management, University of Strathclyde, 1996.
- [39] Hamou, K.H., Caillaud, E., Lamothe, J., Aldanondo, M., "Knowledge for Product Configuration", *Proceedings of ICED '01*, Glasgow, 2001.
- [40] Niklaus, W., "Algorithms + Data Structures = Programs", Prentice-Hall, Inc., Englewood Cliffs, 1975.
- [41] Faltings, B., Weigel, R., "Constraint-based Knowledge representation for Configuration Systems", Technical Report no. TR-94/95, Departement d'Informatique, Ecole Polytechnique Federale de Lausanne, Lausanne, 1994.

-
- [42] Mortensen, N.H., "Design modeling in a Design's Workbench", PhD Thesis, Department of Control and Engineering Design, Technical University of Denmark, 1997.
- [43] Duffy, A., Andreasen, M.M., "Enhancing the Evolution of Design Science", Proceedings of ICED '95, Praha, Heurista Zurich, 1995.
- [44] Hubka, V., "Theorie der Konstruktionsprozesse", Springer-Verlag, Berlin, 1976.
- [45] Yoshikawa, H., "General Design Theory as a Formal Theory of Design", Intelligent CAD I, ed: Yoshikawa, Gossard, Proceeding of IFIP TC5/WG5.2, North Holland, Boston, 1987.
- [46] Ramakrishnan, R., "Database Management System", McGraw Hill, Singapore 1998.
- [47] Suh, N.P., "The Principles of Design", Oxford University Press, 1990.
- [48] Suh, N.P., "Axiomatic Design - Advances and Applications", Oxford University Press, 2001.
- [49] Gunilla, S., "A Generic Information Platform for Product Families", PhD Thesis, Department of Production Engineering, Royal Institute of Technology, Stockholm, Sweden, 2000.
- [50] Hubka, V., Eder, W.E., "Engineering Design - General Procedural Model of Engineering Design", Heurista, 1992.
- [51] Martin, M.V., Ishii, K., "Design for Variety: A Methodology for Understanding the Costs of Product Proliferation", The 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference, Irvine, California, 1996.
- [52] Erixon, G., "Modular Function Deployment - A Method for Product Modularisation", PhD Thesis, Department of Manufacturing Systems, The Royal Institute of Technology, Sweden, 1998.
- [53] Sullivan, L.P., "Quality Function Deployment - a system to assure the customer needs drive the product design and production process", Quality Progress, June 1986.
- [54] Akao, Y., "QFD - Integrating Customer Requirements into Product Design", Productivity Press, 1990.
- [55] Hubka, V., Andreasen, M.M., Eder, W.E., "Practical Studies in Systematic Design", Butterworth & Co. Ltd, ISBN 0-408-01420-2, 1988.
- [56] Erixon, G., "Evaluation Tool for Modular Design", Int. Forum on DFMA, Newport, USA, 1993.
- [57] Andreasen, M.M., "Design Methodology", PhD Workshop, Technical University
-

-
- of Denmark, Department of Control and Engineering Design, Lyngby, 1998.
- [58] Mortensen, N.H., Hansen, C.T., "System modeling - Theory and Application in design research", PhD Workshop, Technical University of Denmark, Department of Control and Engineering Design, Lyngby, 1995.
- [59] Mortensen, N.H., Andreasen, M.M., "Design in an interplay with a product model - explained by design units", TMCE '96, Budapest, 1996.
- [60] Andreasen, M.M., "Modelling - The Language of The Designer", Journal of Engineering Design, Vol. 5, No. 2, 1994.
- [61] Simonek, R., "Ein Beitrag zur Ermittlung der speziellen Funktionsstruktur in der Konstruktion", PhD Thesis, Fakultät für Maschinenbau und Elektrotechnik, Technischen Universität Carolo-Wilhelmina zu Braunschweig, 1974.
- [62] Birkhofer, H., "Analyse und Synthese der Funktionen Technischer Produkte", Fortschritts-Berichte VDI, Reihe 1, Nr. 70, VDI Verlag 1980.
- [63] Pohl, K. "The three dimensions of Requirements Engineering", Rolland, C. and Bodart, F. and Cauvet, C. (eds), CAISE93, Springer-Verlag, p175-192, 1993.
- [64] Oberšmit, E., "Nauka o konstruiranju, metodičko konstruiranje i konstruiranje pomoću računala", Sveučilišna naklada Liber, Zagreb, 1989.
- [65] Roth, K., Birkhofer, H., Ersoy, M., "Methodisches Konstruieren neuer Sicherheitsschlosser", Z-VDI 117, pp. 613-618, 1975.
- [66] ISO 10303-1, "Industrial data systems and integration - Product data representation and exchange - Part 1: Overview and fundamental principles", ISO, Geneva, 1994.
- [67] ISO 10303, "Descriptive methods: architecture and development methodology reference manual", ISO TC 184/SC4/WG10 N63, 1996.
- [68] ISO 10303-11, "Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: EXPRESS language reference manual", 1994.
- [69] Williams, C., "Professional Visual Basic 6 Databases", Wrox Press Ltd., 1999.
- [70] Welling, L., Thomson, L., "PHP and MySQL Web Development"
- [71] Bakken, S. S., Aulbach, A., Schmid, E., Winstead, J., Wilson, L. T., Lerdorf, R., Zmievski, A., Ahto, J., "PHP Manual", <http://www.php.net/manual/en/>
- [72] Molnar, V., "SQL", Fakultet elektrotehnike računarstva, Zagreb
- [73] Law, K. H., Barslow, T., Widerhold, G., "Management of Complex Structural Objects in a Relational Framework", Engineering with computers, pp.81-92, Vol. 6, Springer-Verlag, New York, 1990.
- [74] "BAZE PODATAKA – OSNOVE PROGRAMIRANJA",
-

-
- <http://stroga.anonima.free.fr/pdf/conception.pdf>, Free @nonima, 2000.*
- [75] Papić, P., "Uvod u teoriju skupova", Hrvatsko matematičko društvo, Zagreb, 2000.
- [76] Piotrovskij, L.M., "Električni strojevi", Tehnička knjiga, Zagreb, 1970.
- [77] Nurnberg, N., "Ispitivanje električnih strojeva", Školska knjiga, Zagreb, 1951.
- [78] Blackenfelt, M., "Managing complexity by product modularisation", PhD Thesis, Department of Machine Design, Royal Institute of Technology, Stockholm, Sweden, 2001.
- [79] Stake, R.B., "A hierarchical classification of the reasons for dividing products into modules", Licentiate Thesis, Department of Manufacturing Systems, Royal Institute of Technology, Stockholm, Sweden, 1999.
- [80] Jensen, T., "Functional Modelling in a Design Support System", Thesis, Department of Control and Engineering Design, Technical University of Denmark, 1999.
- [81] Tiihonen, J., Lehtonen, T., Soininen, T., Pulkkinen, A., Sulonen, R., Riitahuhta, A., "Modeling Configurable Product Families", Proceedings of ICED '99, Munich, 1999.

Davor Pavlić rođen je 1974. godine u Zagrebu, gdje je završio srednju elektrotehničku školu. Fakultet strojarstva i brodogradnje u Zagrebu je upisao 1992. godine. Tijekom akademske godine 1996/97. boravio je na Fakultetu strojarstva i brodogradnje, Tehničkog sveučilišta u Delftu, Nizozemska, kao student-istraživač. Radio je na istraživačkom projektu razvoja novog načina propulzije broda. Diplomirao je 1998. godine na usmjeranju "Strojarske konstrukcije" Fakulteta strojarstva i brodogradnje u Zagrebu.

Od 1998. godine zaposlen je pri Katedri za osnove konstruiranja Fakulteta strojarstva i brodogradnje u Zagrebu kao znanstveni novak na projektu Ministarstva znanosti i tehnologije Republike Hrvatske broj 120015: "Model inteligentnog CAD sustava". Tijekom rada na fakultetu aktivno sudjeluje u nastavi iz sljedećih kolegija: Uvod u računala, Primjena računala-K, Konstruiranje pomoću računala, Osnove konstruiranja i Znanost o konstruiranju. Kao honorarni asistent sudjeluje u izvođenju vježbi pri Studiju Dizajna (Arhitektonski fakultet u Zagrebu) te strojarskom i informatičkom odjelu Tehničkog veleučilišta u Zagrebu. Osim u nastavi, aktivno je sudjelovao u organizaciji međunarodnih znanstvenih skupova DESIGN '98, DESIGN 2000 i DESIGN 2002 koji se održavaju u Dubrovniku.

Tijekom ljeta 2002. godine sudjelovao je na dvotjednom međunarodnom doktorandskom seminaru "Design Methodology" u organizaciji Danskog tehničkog sveučilišta. Kao autor ili koautor objavio je 6 znanstvenih i 9 stručnih radova u Hrvatskoj i inozemstvu. Član je Hrvatskog društva za elemente strojeva i konstrukcije te međunarodnog udruženja The Design Society. Služi se engleskim jezikom. Oženjen je i otac jednog djeteta.

SHORT BIOGRAPHY

Davor Pavlić was born in 1974 in Zagreb where he finished secondary electro technical school. He enrolled in the study of mechanical engineering at the Faculty of Mechanical Engineering and Naval Architecture in Zagreb in 1992. He spent the academic year 1996/97 at the Faculty of Mechanical Engineering and Naval Architecture of the Technical University in Delft, Netherlands, as a student-researcher. He worked on a research project on the development of new modes in marine propulsion. He graduated at FAMENA in 1998 in the field of specialization "Engineering Design".

Since 1998 he has been working as a junior researcher on the project number 120-015 "Model of Intelligent CAD System", supported by the Ministry of Science and Technology of the Republic of Croatia, at the Chair of Theory of Design at FAMENA. In addition to his work on the project, he has been involved in the teaching of the following courses: Programming for Engineers, Computer Applications in Engineering Design, Computer Aided Design, Engineering Design, and Design Science. As a part-time assistant lecturer he is in charge of exercises at the Study of Design (Faculty of Architecture in Zagreb), and at the Study of Mechanical Engineering and the Study of Computer Science (Informatics) at the Polytechnics of Zagreb. In addition to his involvement in teaching, he also took an active part in the organization of the international scientific conferences DESIGN '98, DESIGN 2000 and DESIGN 2002 that took place in Dubrovnik.

Davor Pavlić has participated and passed the "Ph.D. Course - Design Methodology" during the summer 2002, organized by Technical University of Denmark. As the author or coauthor he has published 6 scientific and 9 technical reports in Croatia and abroad. He is a member of The Croatian Association for Machine Elements and Design and a member of the international association The Design Society. He has a good command of English. He is married and a father of a child.