
Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

Matija Hoić

Sustav za navođeno zapisivanje i indeksiranje konstrukcijskog znanja

Zagreb, 2008.

Ovaj rad izrađen je na Katedri za osnove konstruiranja Zavoda za konstruiranje pod vodstvom dr. sc. Nevena Pavkovića, dipl. ing. i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2007./2008.

SADRŽAJ RADA

1.	UVOD	1
1.1.	OPIS ZADATKA	2
1.2.	CILJ ISTRAŽIVANJA I HIPOTEZA	6
2.	PREGLED STANJA I SMJEROVA ISTRAŽIVANJA	9
2.1.	INFORMACIJA, PODATAK, ZNANJE	9
2.2.	SADAŠNJI SUSTAVI KATEGORIZACIJE, SPREMANJA I PRETRAŽIVANJA ZNANJA	10
2.3.	VEZANJE ZNANJA UZ CAD MODEL	14
3.	METODOLOGIJA	18
4.	PRELIMINARNA ISPITIVANJA PREDLOŽENOG PRISTUPA NA KONKRETNOM KONSTRUKCIJSKOM UREDU	21
5.	PROGRAMSKA IZVEDBA PROVOĐENJA PROCEDURA	24
5.1.	CAD ELEMENTI POTREBNI ZA ZAPIS ZNANJA	25
5.2.	PROGRAMSKI ALATI ZA IZVOĐENJE PROCEDURA	28
6.	POČETNA - ZAJEDNIČKA FAZA IZVOĐENJA PROCEDURA	31
6.1.	POKRETANJE PROCEDURE	32
6.2.	ODABIR CAD OBJEKTA	32
6.3.	UPIS PODATAKA O KORISNIKU	36
6.4.	PRIKAZ ACCESS FORME	37
7.	STRUKTURA BAZE ZNANJA	41
8.	PRIJEDLOG KLASIFIKACIJE PROCEDURA	45
8.1.	ODABIR STROJNIH ELEMENATA	45
8.1.1.	PROCEDURA ODABIRA LEŽAJA	46
8.2.	OGRANIČAVANJE AKSIJALNOG POMAKA	49
8.2.1.	ODABIR OPRUGE	51
8.2.2.	ODABIR SVORNJAKA	54
8.2.3.	VEZE S GLAVINAMA	56
8.2.4.	BRTVLJENJE	59
8.2.5.	ODABIR PRIČVRSNOG VIJKA	61
8.2.6.	DODATNI ELEMENTI KONSTRUKCIJA	63
8.3.	DEFINIRANJE FUNKCIONALNO - TEHNOLOŠKIH ELEMENATA	64
8.3.1.	ODABIR DOSJEDA	64

8.3.2.	DIMENZIONIRANJE I POZICIONIRANJE REBARA	67
8.3.3.	SKOŠENA POVRŠINA	69
8.3.4.	HRAPAVOST POVRŠINE	70
8.3.5.	DODATNI FUNKCIONALNO TEHNOLOŠKI ELEMENTI.....	72
8.4.	ODABIR MATERIJALA	73
8.5.	DEFINIRANJA NAČINA GRADNJE CAD MODELA.....	75
8.5.1.	SKLAPANJE PROIZVODA	75
8.5.2.	OBLIKOVANJE DIJELA.....	76
8.5.3.	NAMJENA GEOMETRIJSKIH REFERENCI	77
8.6.	NEKE DODATNE BILJEŠKE	78
9.	PRETRAŽIVANJE BAZE ZNANJA	79
10.	EKSPERIMENTALNA EVALUACIJA I TESTIRANJE PREDLOŽENOG SUSTAVA	81
10.1.	PRVA FAZA: ZAPISIVANJE ZNANJA	82
10.2.	DRUGA FAZA: DOHVAT ZNANJA.....	84
11.	REZULTATI.....	86
12.	RASPRAVA.....	87
13.	ZAKLJUČCI	89
14.	ZAHVALE	91
15.	POPIS LITERATURE	92
16.	SAŽETAK.....	95
17.	SUMMARY	96
18.	PRILOG A - PROGRAMSKI KOD APLIKACIJE ZA DOHVAT IMENA IZ STABLA CAD MODELA.....	97

POPIS SLIKA

Slika 1: Shema sustava za upravljanje znanjem Rohde, Pavković, Aurisicchio, 2008, [16]	3
Slika 3: Struktura C-QuARK sustava, (Ahmed, Wallace, Blessing, 2002, [9]).....	12
Slika 4: Graf relacija između komponenti (K. Case, W. A. Wan Harun, 1997., [5])	15
Slika 5: Primjer „DReD“ grafa, (Bracewell, Wallace, 2007, [19])	16
Slika 6: pristupi načinu generiranja pojmova po hijerarhijskoj strukturi.....	18
Slika 7: Shema rada sustava za upravljanje znanjem (Ahmed, 2008, [13]).....	20
Slika 8: Strukture procedura zapisa obrazloženja odabira	21
Slika 9: Vezanje zapisa znanja uz elemente iz stabla CAD modela	26
Slika 10: Sučelje za odabir procedure	31
Slika 11: Prozor za odabir jednog objekta	33
Slika 12: Prozor za odabir dva objekta.....	34
Slika 13: Upis imena inženjera koji generira zapis	36
Slika 14: Sučelje Access forme	37
Slika 15: Struktura baze znanja.....	41
Slika 16: Princip zapisa i dohvata znanja iz baze znanja	43
Slika 17: Sučelje odabira ležaja	48
Slika 18: Sučelje obrazloženja odabira ležaja	50
Slika 19: Sučelje obrazloženja odabira opruge	53
Slika 20: Sučelje obrazloženja odabira opruge	55
Slika 21: Sučelje obrazloženja odabira veze s glavinama.....	58
Slika 22: Sučelje obrazloženja odabira brtvljenja	60
Slika 23: Sučelje obrazloženja odabira pričvrstnog vijka	62
Slika 24: Sučelje obrazloženja odabira dosjeda	66
Slika 25: Sučelje obrazloženja oblikovanja rebara	68
Slika 26: Sučelje obrazlaganja odabira hrapavosti površine.....	71
Slika 27: Sučelje obrazloženja odabira materijala	73
Slika 28: Sučelje obrazloženja izgradnje CAD modela sklopa.....	75
Slika 29: Struktura sustava za pretraživanje	80

1. UVOD

U razvoju proizvoda, primjena CAD aplikacija donijela je značajne promjene u metodama konstruiranja kao i u samom pojmu tehničke dokumentacije. Ove su promjene omogućile pohranjivanje cjelokupne tehničke dokumentacije u digitalnom formatu što je značajno olakšalo njezino generiranje, izmjenu, pohranu te naročito dupliciranje.

U sprezi sa drugim programskim aplikacijama poput tekstualnih procesora, tabličnih kalkulatora, grafičkih alata i baza podataka, danas je moguće cjelokupno znanje primijenjeno i/ili generirano tokom procesa razvoja proizvoda pohraniti na jednom mjestu neovisno o fizičkom pozicioniranju aktera koji su sudjelovali u samom procesu. Tome je naravno potrebno pridodati znanje generirano tokom proizvodnje, sklapanja, distribucije, eksploatacije i recikliranja proizvoda.

U skladu sa povećanim mogućnostima zapisa znanja javlja se sve veći problem pretraživanja postojećeg (zapisanog) znanja u svrhu pronalaženja trenutno potrebnih podataka. Raznovrsnost formata i tipova zapisa svakako dodatno otežava pretraživanje.

Postavlja se pitanje organizacije prikupljenog znanja i njegovog pohranjivanja u smislenu, jedinstvenu, organiziranu cjelinu koja će omogućiti zapis svih vrsta podataka vezanih uz proizvod kao i njihovo pretraživanje.

Stvaranje jedinstvene baze podataka javlja se kao nužno, ali i tehnički ostvarivo, rješenje ovog problema. Stvaranje same baze te unos podataka riješit će prvi dio problema; pohranu znanja. Ostaje pitanje pretraživanja pohranjenog.

Rečeno je kako se radi o najrazličitijem mogućem skupu tipova podataka koji uključuju tekstualne, numeričke i grafičke, pa i multimedijalne podatke. Bit će stoga potrebno razviti novi sustav kategorizacije podataka koji će omogućiti pretraživanje jer će korisnik prilikom pretraživanja baze podataka tražiti podatak pretraživanjem kategorija što će uvelike ubrzati proces pronalaska traženih podataka.

Važno je napomenuti kako uz zapisano znanje postoji i velika količina znanja koja ostaje nezapisana i nalazi se samo u glavama ljudi koji su radili na proizvodu, barem dok ne bude zaboravljeno. Oni ovo znanje ne zapisuju iz jednostavnog razloga što nemaju načina da ga zapišu u smislenom obliku koji bi im omogućio jednostavno naknadno pretraživanje zapisanog.

Stoga je drugi, možda i važniji cilj razvoja modela podataka, a s njim i jedinstvene baze podataka, omogućiti prikupljanje znanja koje je generirano tokom procesa razvoja, ali nikad ne bi bilo zapisano.

1.1. OPIS ZADATKA

Na samom početku razmatranja problema stvaranja baze podataka u procesu razvoja proizvoda, u prvi plan iskaču CAD alati koji omogućavaju kreiranje trodimenzionalnog modela proizvoda uz kojeg se vežu podaci i karakteristike materijala od kojeg će određeni dio biti izrađen, simulacije proizvodnje te organizacije proizvodnog pogona.

Međutim, uz model ni na koji način nisu vezani brojni drugi podaci. Dobavljači od kojih se kupuju podsklopovi dat će CAD modele svojih proizvoda, ali uz njih neće biti vezani podaci o dobavljalivosti, vremenu između slanja narudžbe i prispjeću proizvoda, pa čak ni telefonski broj dobavljača. CAD model neće imati uz sebe vezane dizajnerske skice vanjske forme, dopise o dogovorenom smjeru razvoja ili statistički obrađene podatke o istraživanju tržišta. Svi ovi podaci postoje u digitalnom obliku, ali se postavlja pitanje gdje ih pronaći.

Zašto razmatrati CAD model zasebno od ostalih podataka?

Za razliku od tekstualnih datoteka, tablica, skica i drugih datoteka, između dijelova, sklopova, crteža, simulacija konačnim elementima, simulacija izrade i sklapanja te drugih tipova podataka čije stvaranje i pohranjivanje omogućavaju pojedine CAD aplikacije, postoje čvrste veze stvorene unutar same CAD aplikacije. S druge strane, ne postoji nikakva veza između svih ostalih datoteka, ni prema CAD modelu, a ni međusobno.

Također je potrebno naglasiti kako su sve ove datoteke nosioci znanja vezanog uz određene dijelove proizvoda ili proizvod u cjelini.

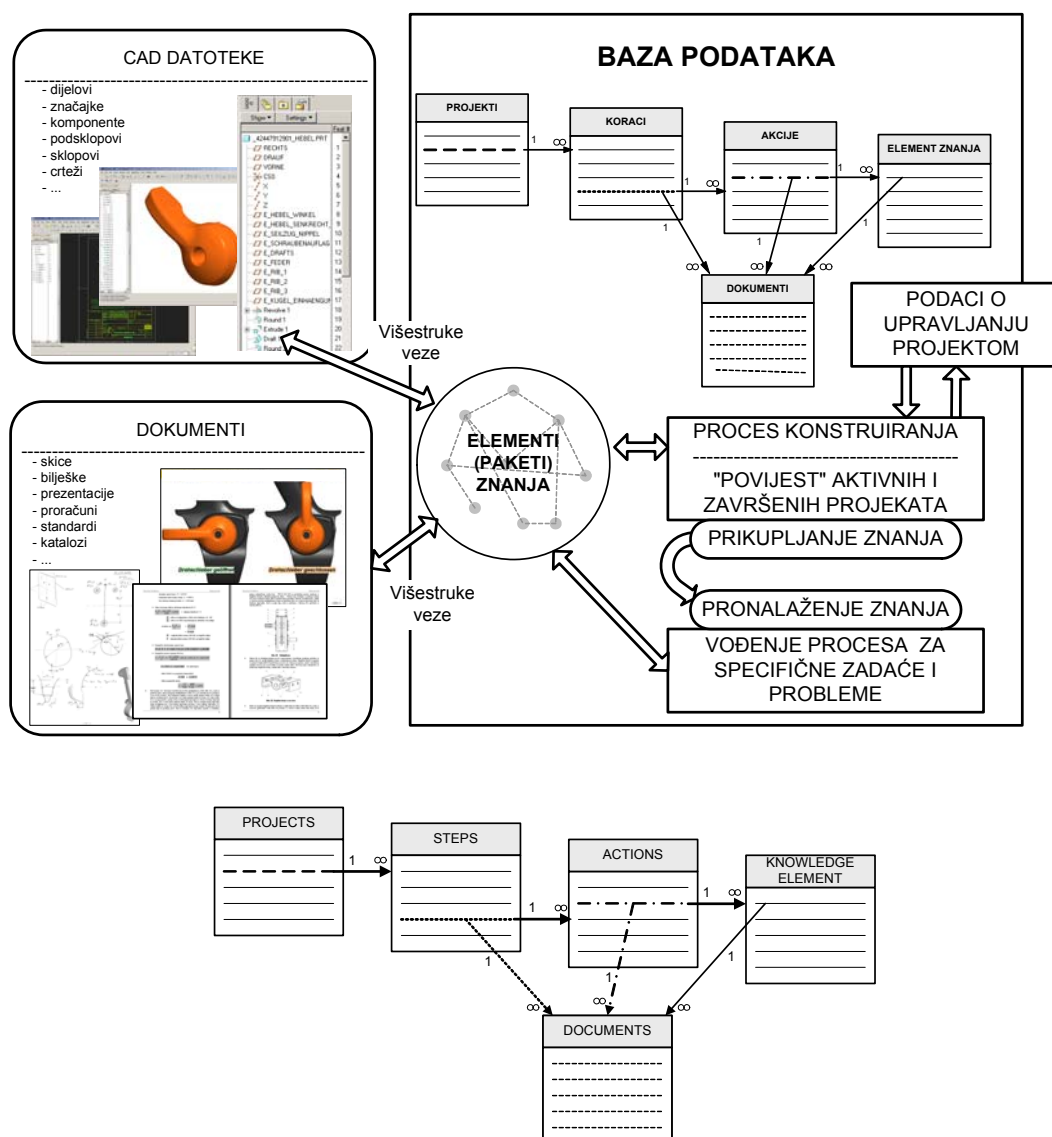
Kako su svi, u ovom ili onom formatu, zabilježeni podaci vezani za budući proizvod, a CAD model predstavlja trodimenzionalni digitalni model budućeg proizvoda, čini se logičnim vezati sve druge podatke uz CAD model. Ovakva organizacija će u konačnici značiti stvaranje tri grupe zapisa u računalnoj memoriji povezanih bazom podataka.

Prvu grupu sačinjava CAD model sa svim dijelovima, sklopovima, crtežima i ostalim datotekama poput analiza, simulacija i sl. Ova je grupa integrirana unutar CAD aplikacije te nije moguće, a ni potrebno, vršiti reorganizaciju ili neku posebnu kategorizaciju ovih podataka. U konačnici, ako se pokaže potreba, tvrtke mogu od svojih dobavljača CAD

aplikacija dobaviti cjelokupna PDM (Product Data Management) ili PLM (Product Lifecycle Management) rješenja koja će im omogućiti učinkovitije rukovanje podacima generiranim u okruženju CAD aplikacija.

Drugu grupu sačinjavaju datoteke drugih formata spremljene negdje unutar računalne mreže tvrtke (ili računala pojedinca, ako čitav projekt izvodi jedan čovjek na svom računalu) ili pak na internetu. Ova nam grupa podataka zapravo i generira inicijalni problem jer ju zbog nepostojanja strukturiranosti nije moguće pretraživati na učinkovit način.

Treću grupu čini znanje koje se planira pohraniti izravno u bazu podataka. Očekuje se da će se raditi o znanju koje se inače ne bi nigdje drugdje pohranilo, stoga je već i pohrana podataka na ovom mjestu značajan dobitak.



Slika 1: Shema sustava za upravljanje znanjem Rohde, Pavković, Aurisicchio, 2008, [16]

Sama baza podataka organizacijski se nalazi između ove tri grupe, tj. veže vanjske datoteke i podatke pohranjene u samoj bazi sa CAD modelom, tj. sa stablom CAD modela od razine sklopa do pojedinačne značajke.

Naravno, pohranjivanje svih podataka na jednom mjestu samo je dio posla. Dapače, već je i spremanje svih dokumenta u digitalnom obliku na računalnu mrežu tvrtke zapravo odradilo tu zadaću. Problem koji je zapravo potrebno riješiti jest kako stvoriti uređenu strukturu tih podataka koja će omogućiti brzo i učinkovito pretraživanje podataka.

U konačnici, razvoj baze podataka koja će omogućiti efikasno spremanje i pretraživanje znanja moguće je ostvariti i bez veze sa CAD sustavom. Postavlja se pitanje, ukoliko je moguće stvoriti istu bazu podataka bez upotrebe dodatne posebne aplikacije, zašto inženjeri ne koriste neku vrstu baze podataka kako bi spremili znanje generirano tokom razvoja proizvoda?

Razlog za nepostojanjem bilo kojeg oblika strukturiranog zapisa leži u jednostavnoj činjenici što svaki zapis zahtijeva veliku količinu uloženog vremena kojeg inženjeri nemaju na raspolaganju. Projekti su pokazali kako postoji želja za zapisivanjem znanja u bazu koja bi omogućila lako pronalaženje zapisanog, ali u nedostatku takve računalne podrške, većina znanja ostaje nezapisana ili se zapisuje u bilježnice prema datumu nastanka što, čak i onome koji je napravio zapis, ne daje velike mogućnosti pronalaska zabilježenog.

Također se za postojeće sustave zapisa znanja pokazalo kako su za industrijsku primjenu previše apstraktni te njihovo korištenje zahtijeva dodatni trud i vrijeme, što dodatno obeshrabruje inženjere kad je u pitanju korištenje ovih sustava.

Cilj je stoga za jedan manji broj poznatih i uobičajenih procedura opisati sam postupak zapisa kao teoretsku osnovu za izgradnju aplikacije koje će kroz stvaranje veza između CAD modela i baze podataka na razini računalnog programa omogućiti brzo i učinkovito prikupljanje podataka iz CAD modela koje bi korisnik u suprotnom morao ručno upisivati. Posljedično očekuje se da će rad sa sustavom koji ima viši stupanj automatiziranosti dati i bolje rezultate tokom rada, ne samo zbog manjeg utroška vremena kod upisivanja znanja već i zbog očekivanog pozitivnog stava inženjera prema sustavu koji od njih zahtijeva minimum uloženog truda.

Samu aplikaciju svi bi inženjeri u konstrukcijskom uredu, neovisno o radnom iskustvu, mogli koristiti na isti način što se tiče samog unosa znanja. Razlika se očekuje u trenutku kad baza bude sadržavala dovoljan broj zapisa za efikasno pretraživanje. Kako baza ne zamjenjuje ljudsku memoriju, iskusni inženjeri i dalje će se moći oslanjati na vlastitu prosudbu i iskustva bez referenciranja na bazu. Koliko će se htjeti referencirati u svrhu dodatne provjere, ovisi o

njima i njihovom načinu rada. Kod inženjera sa manje radnog iskustva koji nemaju iskustvenu bazu za pojedine odluke, očekuje se veći broj referenci na prethodna rješenja.

U pogledu donošenja odluka na temelju prethodnih iskustava, u osnovi se ništa ne mijenja. Inženjeri će i dalje svoje odluke temeljiti na iskustvenom znanju, samo što do njega više neće ići neposredno postavljanjem pitanja starijim kolegama, već posredno, preko zapisa koje su njihovi stariji kolege napravili u bazi.

Kako popis procedura neće biti konačan čak ni po pitanju uobičajenih procedura, a kamoli donošenja odluka u novim slučajevima, savjetovanje sa starijim kolegama neće nestati. Očekuje se ipak kako će stariji inženjeri zaprimati manji broj istih pitanja o istim temama.

Da bi se omogućilo samo pretraživanje na učinkovit način, potrebno je stvoriti strukturu zapisa. Kako je početak stvaranja sustava na opisivanju postojećih postupaka donošenja odluka, nije utvrđena nikakva struktura koja bi u određeni međusobni odnos postavila ove procedure. Uslijed nepostojanja kategorizacije, čini se kako bi pretraživanje upisanih odluka bilo izrazito sporo i neučinkovito.

Ipak, ako se razmotri struktura zapisa uočavaju se potencijalni zajednički parametri koje bi trebalo unositi u sam zapis. U prvom redu to su same procedure. Ukoliko se u zapis zabilježi kojom procedurom je napravljen zapis, korisnik će moći pregledavati bazu prikazujući samo zapise pojedine procedure. Na isti način unošenjem imena projekta, vremena nastajanja zapisa, imena osobe koja je napravila zapis te dijela ili sklopa uz koji je vezan zapis, korisnik bi u kratkom vremenu mogao od baze dobiti popis zapisa koji se tiču samo određenog projekta ili pojedinog dijela. Posljedično, čak i bez uređene strukture samih procedura, korisnik će moći na brz i efikasan način pretraživati bazu.

Jednom kad ovakva aplikacija bude realizirana, slijedeći korak biti će testiranje same aplikacije u realnom okruženju konstrukcijskog ureda. Pokaže li se ovakav pristup prihvatljivim za spremanje i pretraživanje znanja, barem u onom segmentu u kojem je definiran, to će biti osnova za daljnji rad na razvoju sustava.

Razvoj sustava kao primjenjive aplikacije podrazumijeva ugradnju dodatnih procedura definiranih u praksi. Iako se očekuje određeni broj procedura koje će biti vrlo slične u većini konstrukcijskih ureda, razlike će postojati zbog različitih organizacijskih struktura unutar svakog pojedinog ureda, dostupne proizvodne tehnologije, iskustva s određenim tipovima rješenja te ponudom dobavljača. Naravno, postojat će procedure koje su svojstvene određenom području inženjerske struke kao i procedure koje su jedinstvene za svaki pojedini ured. Stoga se u daljem budućem razvoju sustava očekuje omogućavanje svakom korisniku da, uz minimalno poznavanja principa rada sustava, može stvoriti vlastite procedure.

Uzevši u obzir širinu proizvoda koje tvrtke diljem svijeta proizvoda, očito je kako osmišljavanje općeg sustava višestruko nadilazi opseg ovog rada. Stoga će se ovaj rad usredotočiti na jedan segment znanja za koje postoji želja i mogućnost pohrane tj., cilj će biti definirati prijedlog užeg skupa obrazlaganja odabira konstrukcijskih rješenja sa kojima se svakodnevno susreću brojni konstruktori.

1.2. CILJ ISTRAŽIVANJA I HIPOTEZA

U prethodnom poglavlju rečeno je kako će cilj ovog istraživanja biti definirati prijedlog užeg skupa obrazlaganja odabira konstrukcijskih rješenja koje se opetovano pojavljuju pred konstruktorima. Naravno, stariji i iskusniji konstruktori imaju veliko iskustvo po pitanju provođenja ovih procedura iz jednog jednostavnog razloga; ove su procedure izvršili nebrojeno mnogo puta.

S druge strane, inženjeri sa malo ili nimalo radnog iskustva suočeni su sa problemom donošenja odluka čiji je postupak njihovim starijim kolegama dobro poznat. U većini slučajeva, sam postupak nije nepoznanica. Ono što se pokazuje problematičnim jest pronalazak pojedinih podataka koji bi poduprli određenu odluku.

U radu će se predložiti pristup procesu dohvaćanja znanja u kojem će se nastojati razviti posebna sučelja i programske procedure za pojedine konkretne i usko definirane slučajeve zapisa znanja.

Pri tome će sučelja i procedure biti predložene tako da zahtijevaju što manje vremena pri stvaranju novog zapisa znanja na način da se „nude“ unaprijed definirani elementi i strukture zapisa za određeno usko područje.

Rad polazi od hipoteze da takav pristup razvoju sustava za upravljanje konstrukcijskim znanjem treba omogućiti slijedeće:

- 1.) Efikasniji proces dohvaćanja i zapisa znanja zbog manjeg utroška vremena konstruktora
- 2.) Efikasnije i lakše uvođenje sustava u praksu, manji „otpor“ potencijalnih korisnika
- 3.) Mogućnost razvoja novih mehanizama indeksiranja i pretraživanja, različitih od predloženih u dosadašnjem istraživanju. Pretpostavlja se da će takvi sustavi pretraživanja biti efikasniji i jednostavniji za provedbu u praksi.

Osnovni pristup generiranja ovakvog sustava polazi od stvaranja konkretnih slučajeva objašnjenja kroz procedure zapisa objašnjenja donesenih odluka gdje su procedure definirane kao konkretni termini, a ne kao niz apstraktnih pojmova kakvim se indeksiraju zapisi u postojećim sustavima.

U skladu sa pretpostavkom od koje kreće, cilj je ovog rada formirati procedure na temelju

- 1) vlastitog promišljanja
- 2) analize rada konstrukcijskog ureda

pri čemu će osnovne skice procedura proizaći iz prvog izvora dok se od drugog izvora očekuje kako će omogućiti detaljiranje samih procedura. Poseban doprinos razvoju metode očekuje se od iskusnijih inženjera od kojih bi se putem intervjua trebali prikupiti uobičajeni razlozi za pojedini izbor od mogućih izbora unutar svake pojedine procedure.

Od inicijalne zamisli ovih procedura bilo je jasno da će zbog stroge definiranosti pojmova procedura biti nemoguće obuhvatiti cjelokupno područje inženjerskog znanja, ali se željelo inženjerima ponuditi sustav koji će njima biti prihvatljiv za uporabu i time osigurati efektivan zapis barem jednog dijela inženjerskog znanja, a u kontrastu sa postojećim sustavima koji omogućavaju zapis znanja gotovo cjelokupnog potrebnog područja, ali su složeni za uporabu.

Naravno, kako je smisao zapisa znanja njegovo kasnije dohvaćanje, od sustav koji očekuje jednostavnije indeksiranje tokom upisa, očekuje se i mogućnost jednostavnijeg pretraživanja i dohvaćanja znanja.

Za ovaj skup sučelja i procedura očekuje se nadalje kako će pored bržeg zapisa obrazloženja odabira omogućiti i brže pretraživanje i pronalaženje svih zapisa relevantnih za postavljeni upis uz uklonjenu sumnju u postojanje zapisa koji su relevantni, ali nisu ispisani zbog pogrešno postavljenog upita ili pogrešno postavljene kategorizacija zapisa. Ova nedvosmislenost u pretraživanju ujedno bi bila i glavna prednost pred apstraktnijim sustavima kategorizacije.

Za potvrdu ove hipoteze biti će potrebna provjera u realnim uvjetima, tj. opisane procedure treba dati inženjerima kako bi se utvrdilo mogu li ih upotrijebiti tokom procesa konstruiranja.

Inicijalnu verifikaciju moguće je ostvariti kroz razgovor sa inženjerima koji bi ovakav sustav trebali koristiti u praksi, međutim, za potpunu provjeru hipoteze bit će potrebno

izgraditi sustav te ga postepeno implementirati kroz iterativne faze korištenja, zapisa primjedbi, analize rezultata, modifikacija i nadogradnje te ponovnog korištenja.

Prođe li sustav određeni broj faza iterativnog razvoja pokazujući mogućnost za aplikativnu primjenu, industrijska verzija sustava prolazila bi kroz identične iterativne faze počevši sa općim dijelom sustava koji bi se kroz navedena faze usuglašavao sa potrebama pojedine tvrtke.

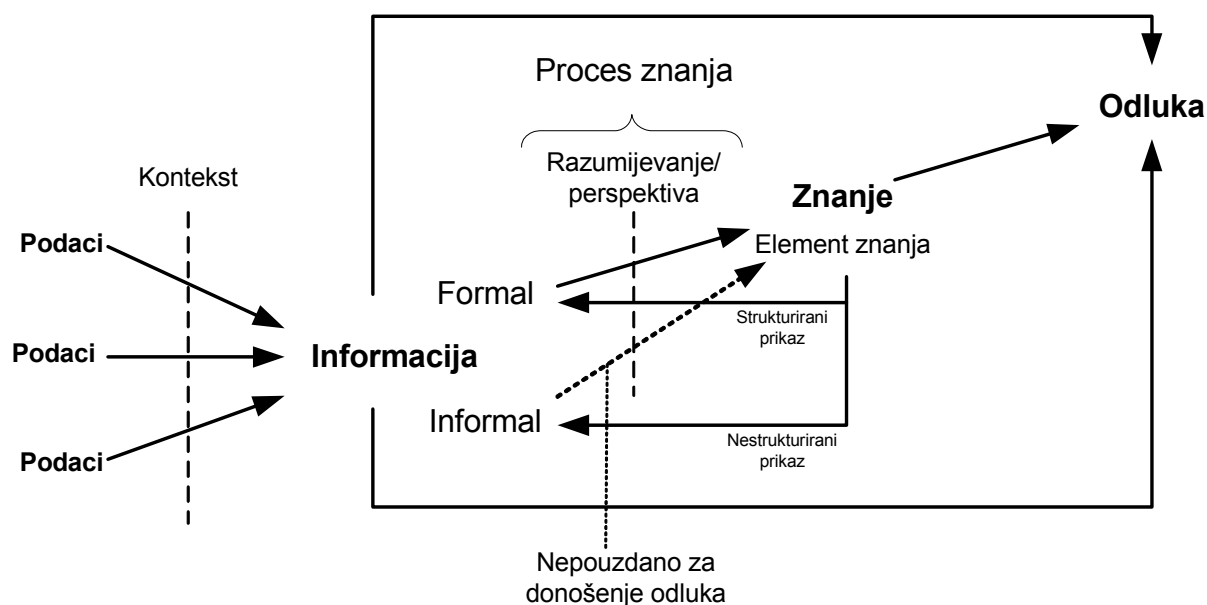
2. PREGLED STANJA I SMJEROVA ISTRAŽIVANJA

Problematikom spremanja inženjerskog znanja bavi se relativno veliki broj radova. Međutim, koliko ima radova, toliko je i pristupa problemu.

2.1. INFORMACIJA, PODATAK, ZNANJE

Za početak, postavlja se pitanje odnosa pojmova informacije, podatka i znanja. Općeniti zaključak, do kojeg se došlo pregledom literature, je da ne postoji jedinstveno razumijevanje ovih pojmova tj., da se u većini slučajeva pojmovi informacija i podataka preklapaju ili se pak koriste kao sinonimi (Hicks, Culley, Allen, Mullineux, 2002., [2]). U duhu sive granice između ova dva pojma, ali i nejasne definicije znanja, opća kategorizacija inženjerskih informacija svodi se na četiri kategorije: opisne, mjerljive, za donošenje odluka i razdiobne (kategorizacijske), dok se znanje dijeli na četiri grupe: opće, dobiveno zaključivanjem, specifično za područje i pojedinačno za jedan slučaj.

Ovakva klasifikacija je izrazito općenita te se postavlja pitanje njezine primjenjivosti u inženjerskoj praksi u svrhu spremanja znanja.



Slika 2: Odnos podatka, informacije i znanja, (Hicks, Culley, Allen, Mullineaux, 2002., [2])

2.2. SADAŠNJI SUSTAVI KATEGORIZACIJE, SPREMANJA I PRETRAŽIVANJA ZNANJA

Bilo bi pogrešno misliti kako se tokom razvoja proizvoda nikakvo znanje, ako se izuzme konačna projektna dokumentacija, ne zapiše u strukturiranom obliku. Zapisi znanja javljaju se u obliku osobnih zapisa inženjera (Jerzy Pokojski, 2004., [20]). Bilo bi korisno omogućiti inženjerima zapis svojih osobnih bilježaka u digitalnom obliku sa strukturom koja bi omogućavala pretraživanje. Dapače, zapisi koji nose razloge zbog kojih je određeno rješenje primijenjeno mogu poslužiti kao nositelji aplikacije koja vodi inženjera kroz postupak oblikovanja dijela sličnog onim dijelovima koji za koje već postoji zabilježeno znanje. U konačnici, sustav koji bi mogao prepoznati sličnosti između pojedinih koraka oblikovanja različiti konstrukcija imao bi još dvije pozitivne karakteristike koje bi proizlazile iz njegove mrežne strukture vezanih postupaka oblikovanja različitih objekata naspram standardnih linijskih nepovezanih postupaka.

Kao prvo, smanjio bi se prostor koji zapisi zauzimaju jer ne bi na dva mjesta pamtio isto znanje. Drugo, sa vezama koje isti korak ima sa različitim projektima, mogao bi inženjeru ponuditi izbor koji se standardno ne razmatra u razvoju određenog tipa proizvoda.

Sam sustav koji bi mogao prepoznavati sličnosti između pojedinih koraka mora imati određena svojstva inteligencije. Stoga je pri razvoju aplikacije za upravljanje bazom podataka zadužena umjetna inteligencija.

Razvijane su i aplikacije za ponovnu upotrebu inženjerskog znanja tokom razvoja nove generacije određenog proizvoda, gdje se pristup očekuje to korisnijim što je nova generacija proizvoda sličnija prethodnoj (Baxter, Gao, Case, Harding, Young, Cochrane, Dani, 2007., [3]). Aplikacija primjenjuje metodu koja predviđa sustav za ponovno upotrebljavanje znanja na bazi dva modela: Procesnog i Proizvodnog modela, kojima dodaje podatke zadane zadatkom i opće znanje. Ove su tri grupe postavljene u zasebne listove Excela te povezane kodom koji obrađuje podatke na način koji definira proces oblikovanja određenog tipa proizvoda. U konačnici aplikacija daje podatke potrebne za oblikovanje.

Ponovna upotreba znanja kroz pravila i podatke koji su trajno spremljeni u gore navedenoj aplikaciji nije upitna, ali postoje određena ograničenja. Kao prvo, ako se proizvod u novoj generaciji samo malo izmijeni, a što u standardnom procesu povećanja broja funkcija tokom razvoja nove generacije proizvoda nije ništa neobično, upotrjebljivost aplikacije

postaje vrlo upitna jer se neće moći uzeti u obzir nove funkcije. Isto tako, ako se želi razmatrati samo jedan segment proizvoda, ni to neće biti moguće jer je potrebno unijeti podatke za cijeli proizvod.

Stoga se postavlja pitanje razbijanja aplikacije na manje module koji će svaki podsustav oblikovati zasebno. Ovo bi omogućilo jednostavnije praćenje cijelog procesa konstruiranja te upotrebu istih komponenti na različitim sustavima. Također, uputno bi bilo razviti veze između Excela i CAD aplikacije kako bi se svaka promjena u geometrijskim podacima koje daje ovaj sustav odrazila na geometriju CAD modela proizvoda.

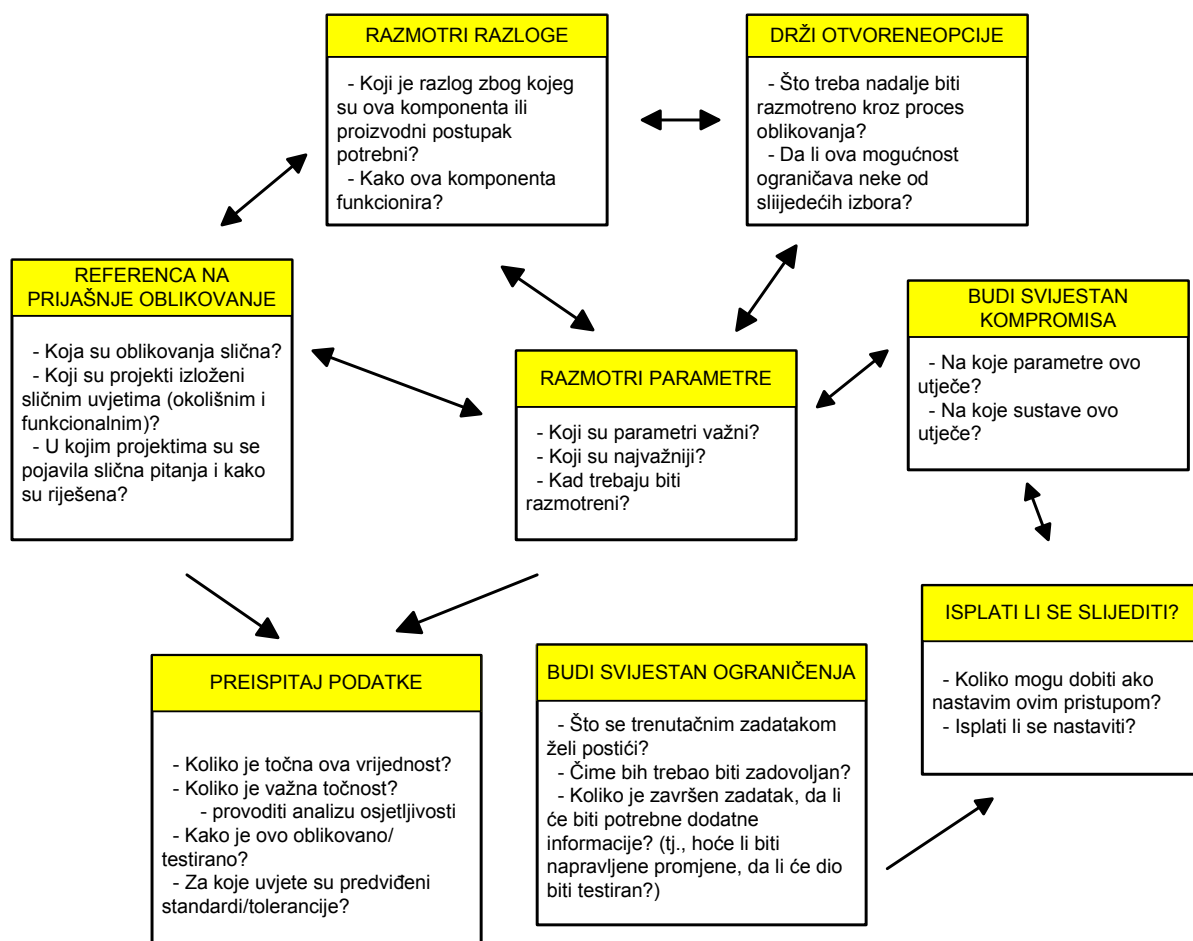
U doba kad je informatička tehnologija postala uobičajeni dio svakodnevnog života, generiranje dokumentacije u digitalnom obliku preuzelo je primat u odnosu na pisane dokumente te zauzima oko 90% svih dokumenata generiranih tokom razvoja proizvoda (Kim, Bracewell, Wallace, 2007., [7]). S obzirom na veliki broj dokumenata koji se generira, aplikacije za pretraživanje imaju očitu prednost pred ručnim pretraživanjem dokumenata. U slučaju jednostavnih pretraživanja (ključne riječi), preciznost pretrage na razini je od 80%, međutim, korisnici teško prihvataju dobivene odgovore zbog nedostatka konteksta. Iako su razvijeni složeniji sustavi koji ovaj problem rješavaju nuđenjem veza na daljnje dokumente, čak ni ovi sustavi ne mogu riješiti problem usporedbe i spajanja informacija iz više dokumenata. Razvijaju se noviji algoritmi koji na bazi analize značenja rečenica omogućavaju njihovo povezivanje sa ciljem davanja združene informacije.

Jedan od takvih algoritama koristi obogaćene lingvističke značajke bazirane na Natural Language Technology (NLP), a primijenjen je u klasificiranju izvještaja o kvarovima i nepravilnostima u zračnom prometu koji su dotada bili klasificirani u samo dvije kategorije; ljudska greška i tehnički kvar. Primjenom algoritma omogućena je automatska klasifikacija u 7 kategorija što je omogućilo povećanje kvalitete analize grešaka u zračnom prometu (Kim, Wallace, 2007., [18]).

U načelu, jednom kad se ovakvi sustavi nađu u inženjerskoj praksi, njihovi korisnici će samo morati postaviti zahtjev za znanjem. Istraživanja su međutim pokazala kako postoji razlika između znanja kojeg inženjeri traže i znanja kojeg su zapravo zatražili tj., inženjeri u većini slučajeva postavljaju krivo ili neprecizno postavljeno pitanje. (Ahmed, Wallace, 2004. [17]).

Kako bi se pomoglo mladim inženjerima u postavljanju pravilnog zahtjeva za znanjem koji su, s obzirom na učestalost postavljanja pravilnog zahtjeva u 35% slučajeva, rijetki, razvijena je metoda pod nazivom C-QuARK koja se bazira na postavljanju pitanja, a razvijena

je promatranjem iskusnih inženjera, tj. načina na koji oni formiraju pitanja kada traže određeno znanje.



Slika 3:Struktura C-QuARK sustava, (Ahmed, Wallace, Blessing, 2002, [9])

Proučavanjem načina na koji rade iskusni inženjeri uočeno je korištenje 9 različitih strategija koje se temelje na odgovorima na 9 pitanja. Ova pitanja odgovaraju na pitanja važnosti pojedinih tema i parametara (**Consider Issues**), preispituju pouzdanost informacija/specifikacija i isplativosti razmatranja određenog rješenja (**Question data/specification, is it worth pursuing**), trude se biti svjesni razloga, ograničenja i omjera dobivenog/izgubljenog (**Aware of reason, limitations, trade-offs**), referenciraju se na prethodne projekte (**Refer to past projects**) te uvijek razmatraju druge opcije (**Keep options open**). Iz ovih strategija je nastalo i ime cijele metode.

Međutim, prije nego se metoda primijeni, potrebno je prvo proučiti način na koji inženjeri sa različitim iskustvom postavljaju pitanja tj., kako postavljaju zahtjev za znanjem. Istraživanja su pokazala kako se inženjeri sa malo praktičnog iskustva najviše pouzdaju u

starije kolege koje osim kao izvor informacija služe i kao savjetnici po pitanju potvrđivanja pravilnosti određene odluke, pomažu u preformuliranju zahtjeva za znanjem kako bi se u drugim izvorima lakše našla informacija koja se uistinu traži te kao izvori dodatnih informacija vezanih uz pitanje kojim se trenutačno bavi neiskusni inženjer (Ahmed, 2006., [23]).

Određena istraživanja doduše ukazuju na smanjenje oslanjanja mladih inženjera na starije po pitanju pronalaska odgovora. Brojke pokazuju blagi pad sa 90% koliki je udio pribavljanja informacija od starijih kolega u odnosu na sva pribavljanja informacija bio 1996. na 75% u 2003. godini (Wallace, Ahmed, Bracewell, 2005, [11]). Istraživanje je provedeno u istoj tvrtci, a razlika u rezultatima pripisuje se razvoju IT tehnologije.

Razvijeni sustav kategoriziranja znanja pod nazivom (Ahmed, 2001., [27]) određuje za svaki dokument veze prema drugim dokumentima pomoću četiri grupe karakteristika. Ove grupe su; proizvod, parametri, funkcije, proces konstruiranja. Svaki put kad inženjer postavi zahtjev za znanjem, bilo po točno određenom pojmu ili odabirom iz kategorija, sustav ga upućuje na daljnje vezane dokumente čime se inženjer vodi kroz postojeće zapise do onog koji je njemu potreban.

Pored kategorizacije samog znanja, postoje i drugačiji pristupi. Jedan od njih je kategorizacija zahtjeva za znanjem (Auricchio, 2005., [10]). Ovaj sustav klasificira zahtjev za znanjem prema karakteristikama zahtjeva u 12 kategorija od kojih svaka ima između 2 i 10 tipova. Svaka kategorija označena je slovom, a svaki tip brojkom. Zahtjev za znanjem se karakterizira putem koda od 12 alfa-numeričkih oznaka. Iako su razvijena i dva skraćena sustava kodiranja od kojih jedan karakterizira zahtjev pomoću 6, a drugi pomoću samo 3 kategorije, i dalje ostaje veliki broj mogućih kombinacija od kojih većina nema smisla, a značajnom dijelu nedostaje jednoznačnost. Kako je i sam autor naznačio, cijeli sustav je razvijen u akademskom okruženju sa svrhom proučavanja samog procesa nastajanja zahtjeva za znanjem, te stoga nije primjenjiv u inženjerskoj praksi.

Problem dvosmislenosti i prekomjerne apstraktnosti za inženjersku praksu uočen je i u pokušaju primjene na stvarni industrijski primjer (Rohde, Pavković, Auricchio, 2008., [16]). Zbog očite neprimjerenosti za praktičnu primjenu razvoj sustava za pohranjivanje i pretraživanje znanja krenuo je u drugom pravcu. Umjesto općenite, ali apstraktne kategorizacije, odlučeno je da se krene iz suprotnog smjera. Za početak će se oblikovati određeni dio uobičajenih procedura odabira rješenja u čijoj će definiciji značajan udio imati konstrukcijski uredi uz čiju pomoć će ove procedure biti detaljirane. Na ovaj način, neće se krenuti od vrha hijerarhije, već od dna. Ovaj pristup neće moći pokriti cijelu razinu, već samo

ograničeni te relativno mali broj pojmova sa najniže razine kojima će usto nedostajati nadređeni pojmovi, tj. neće postojati kategorizacija. Međutim, pozitivna strana ovog pristupa je potencijalna primjenjivost sustava u inženjerskoj praksi.

U konačnici, sa pristizanjem povratnih informacija iz primjene, biti će moguće modificirati pa i proširivati sustav te ga prilagoditi određenom dijelu općenitih potreba za zapisivanjem znanja u konstrukcijskim uredima.

2.3. VEZANJE ZNANJA UZ CAD MODEL

Kako je cilj ovog rada tokom spremanja znanja ujedno i njegovo vezanje uz CAD model kao digitalnog predstavnika stvarnog proizvoda, postavlja se pitanje postojanja aplikacija razrađenih od strane samih proizvođača CAD aplikacija.

Vodeći proizvođači CAD aplikacija svojim klijentima nude PDM (Product Data Management) sustave koji će omogućiti upravljanje svim datotekama vezanim uz pojedini projekt (Bojčetić, [26]) na intuitivan način, oslobađajući korisnika od uređivanja i organizacija dokumenata. Ipak, PDM sustavi ne mogu voditi korisnika, tj. ne mogu mu predstaviti logiku koja stoji iza određenih odluka jer PDM sustavi upravljaju samo podacima koji spadaju u kategoriju „ŠTO“ (konstrukcijski) i „KAKO“ (proizvodni), ali ne i „ZAŠTO“. Upravo je ova zadnja kategorija važna te se odnosi na pojam logičkog zaključivanja koji stoji iza odluke konstruktora. U engleskom govornom području poznata kao „Design Rationale“, kategorija razloga za i protiv određene odluke važna je za sam proces odlučivanja. Mogućnost zapisa ovih razloga osnovna je prednost sustava baze podataka koji bi uz CAD model proizvoda vezao znanje, jer bi iz razmatranja odluka vezanih uz prethodne projekte bilo moguće tokom budućih projekata konstruktoru ponuditi ova rješenja kao izvore korisnih informacija na temelju kojih će donijeti vlastite odluke.

Ideja spremanja cjelokupnog znanja uz stvaranje veza razrađena je na razini značajki CAD modela (Raimundo R.M. da Cunha, A. Dias, [1]) bazirajući se na sličnosti procesa izmjene podataka računalnim funkcijama u procesu objektnog programiranja i procesa izmjene značajki geometrije proizvoda tokom donošenja odluka u procesu konstruiranja. Razvijen je model grupacija značajki u grupe koje odgovaraju generičkom procesu razvoja proizvoda pri čemu se značajke općenito dijele na geometrijske (CAD model) i ne-geometrijske (svi ne-geometrijski podaci). Aplikacija pisana u Javi trebala bi potom

omogućiti stvaranje veza između ovih značajki te tako stvoriti kompletnu strukturu podataka generiranih tokom cijelog procesa razvoja proizvoda.

Pored vezanja znanja uz značajke pojedinog dijela ili pak dio u cjelini, javlja se pitanje vezanja znanja vezanog uz podsklop ili sklop čitavog proizvoda. Jedan primjer takvog znanja je postupak sklapanja za kojeg postoje metode generiranja grafova relacija između komponenti (K. Case, W. A. Wan Harun, 1997., [5]). Prethodno generiranje ovakvog grafa koji bi prikazivao kakve veze trebaju postojati između pojedinih komponenti proizvoda je kod specijaliziranih proizvođača standardni postupak, a može biti jednak za cijelu obitelj proizvoda. Očito postoji valjani razlog vezanja ovakvog grafa uz postojeći proizvod kako bi se olakšalo planiranje sklapanja nove generacije proizvoda.

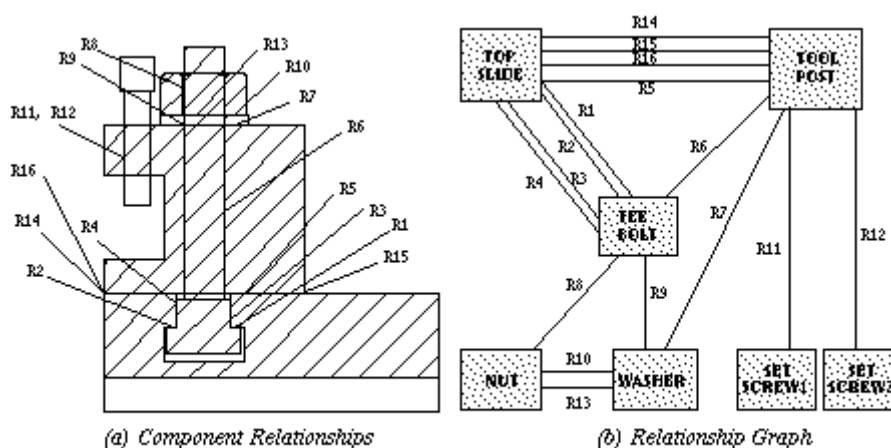
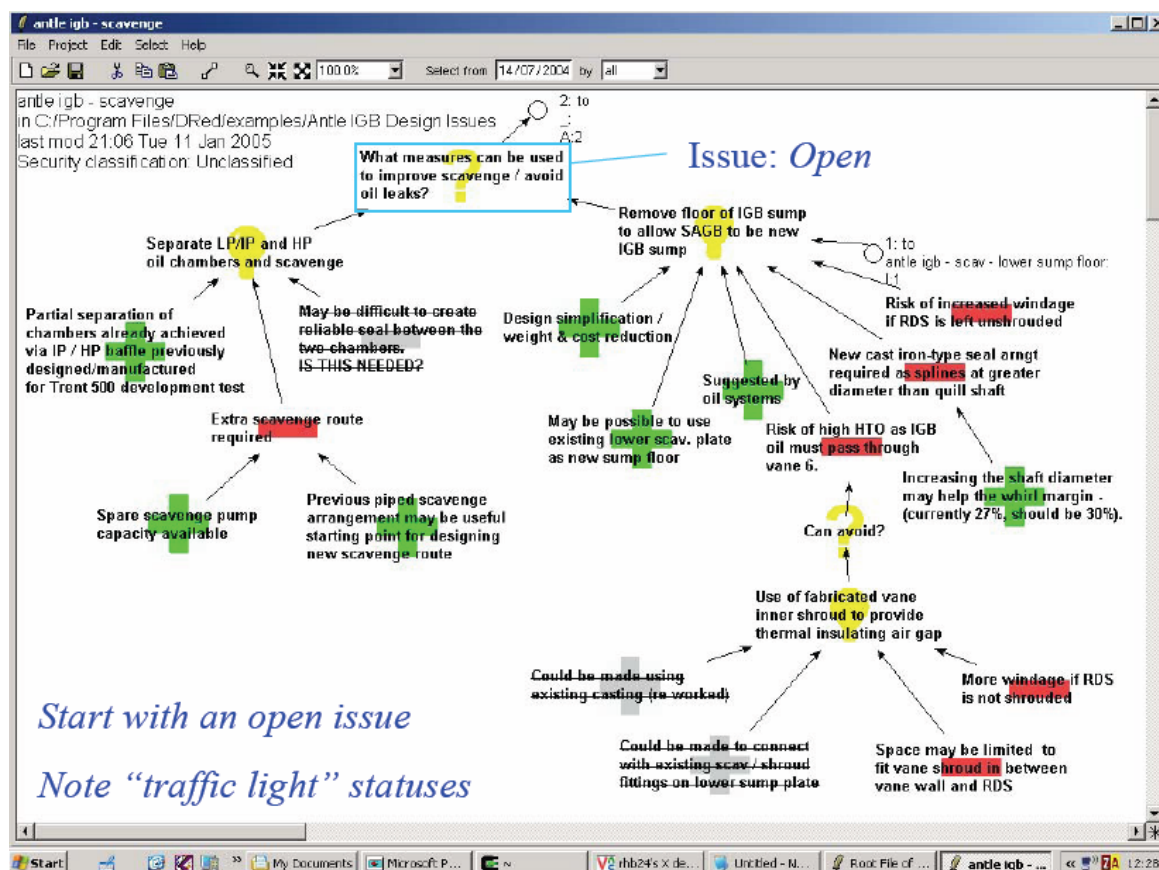


Figure 2. Component Relation Graph for the lathe tool post assembly

Slika 4: Graf relacija između komponenti (K. Case, W. A. Wan Harun, 1997., [5])

Postoji još jedan alat koji je u upotrebi, a njegovo vezanje uz CAD elemente je korisno. Riječ je o „DreD“ grafu (Bracewell, Wallace, 2007, [19]), metodi zapisa inženjerskog razmišljanja o složenom problemu koji se prikazuje razbijen na veliki broj malih problema pri čemu svaki problem koji se rastavlja na manje probleme predstavlja čvor u grafu. Metoda je razvijena u zrakoplovnoj industriji, a u obliku računalne aplikacije, čije komercijalne forme postoje na tržištu od 2001., „DReD“ graf omogućava jednostavan pregled čak i vrlo složenih problema koji se rastavljaju na veliki broj manjih problema. Graf je, ovisno o veličini problema koji se razmatra, prikazan na nekoliko listova gdje su problemi između kojih postoji određeni odnos, a ne mogu stati na isti list, vezani „tunelom“, grafičkim prikazom koji omogućava praćenje veze jednim klikom miša.

Aplikacija omogućava vezanje svakog čvora za neku vanjsku datoteku ili URL adresu, pa je česta veza na digitalne slike (Screenshot) CAD modela (Wallace, Ahmed, Bracewell, 2005, [11]). Međutim, korisnik koji promatra CAD model, ne može znati da je on referenca za „DReD“ graf jer ne postoji povratna veza. U slučaju da se u bazu podataka vezanu na CAD model postavi zapis o tome da se određeni čvor iz „DReD“ grafa referencira na CAD model, tada bi ta povratna veza bila stvorena.



Slika 5: Primjer „DReD“ grafa, (Bracewell, Wallace, 2007, [19])

Razvijeni su pojedini prijedlozi strukture CAD modela proizvoda sa vezanim znanjem, ali su pretežito ostali na teoretskoj razini bez pokušaja stvarne izvedbe predloženog sustava (László Horváth, Imre J. Rudas, 2004., [8]). Iako su dani odnosi između čovjeka, CAD aplikacije, CAD modela te samog vezanog znanja, kao i tokovi informacija uslijed promjena koje se odvijaju tokom razvoja i modifikacije, nije ponuđena struktura znanja niti postupak kojim bi se predloženi model proveo u stvarnim uvjetima što bi omogućilo promatranje procesa zapisa znanja i posljedično, stvaranje strukture koja bi omogućavala pretraživanje.

Literatura pored teoretskih razmatra i tehničku stranu izvedbe sustava zapisa znanja. Rečeno je u uvodu kako je ideja stvaranja strukture vezati dio konstrukcijskog znanja na CAD model kao digitalnog predstavnika stvarnog proizvoda. Uslijed velikog broja CAD aplikacija koje je danas moguće naći na tržištu, postavlja se pitanje razlike među njima, tj. da li će u svakoj aplikaciji CAD model biti građen od istih elemenata. Usporedbom istovjetnih modela izrađenima u trima vodećim CAD aplikacijama (Rappoporty, Spitzz, Etzionx, [22]), različite aplikacije predstavljaju granice trodimenzionalnih tijela različitim setovima jedno- i dvodimenzionalnih geometrijskih elemenata. Razvijen je algoritam koji će za CAD model prenesen iz bilo koje CAD aplikacije u UPR standard generirati isti set rubnih elemenata.

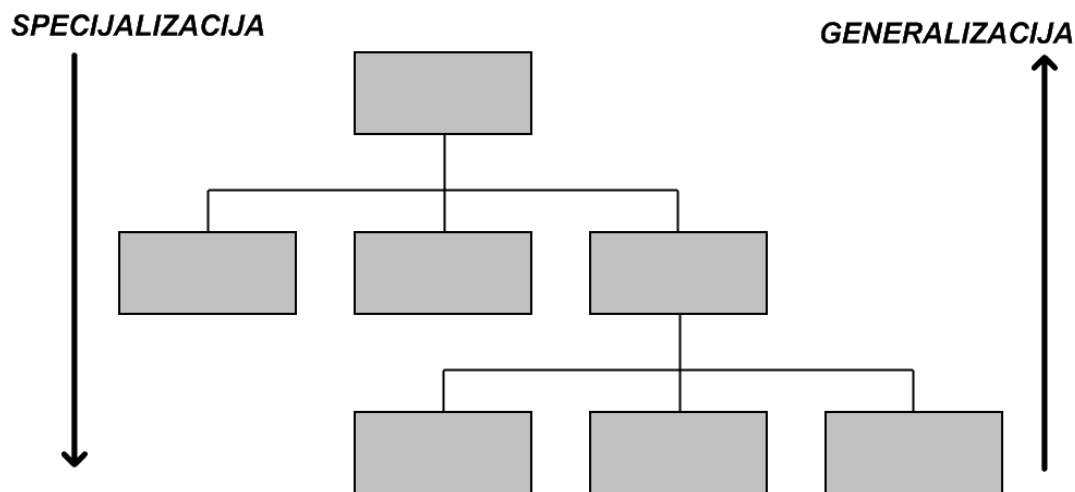
Iako bi ovo pojednostavilo zapis i pretraživanje znanja u slučaju da istu bazu podataka dijele odjeli ili tvrtke koje koriste različite CAD aplikacije, javlja se nekoliko problema koji poništavaju ovu prednost. Prva je potreba za izmjenama pojedinih dijelova nakon što je za njih već vezan određen zapis. Sa promijenjenom geometrijom mijenjao bi se i set rubnih elemenata koje bi algoritam generirao što bi rezultiralo gubitkom veze zapisa sa CAD modelom. Čak i ako se izostavi sam proces vezanja znanja, sama potreba za izmjenama modela tokom zajedničkog rada diktira korištenje istovjetne CAD aplikacije.

Čini se kako bi u realnom procesu bilo nužno korištenje iste CAD aplikacije, pa potrebe za spremanjem u druge formate nema. Ostaje ipak problem promjena u generiranju trodimenzionalnog modela koje se stvaraju tokom razvoja nove generacije iste CAD aplikacije.

3. METODOLOGIJA

Formiranje taksonomije podrazumijeva dva koraka: generiranje kategorija pojmova te uspostavljanje sustava hijerarhije između ovih pojmova. Nasumično generiranje kategorija te njihovo ubacivanje u prostor taksonomije da bi ih se kasnije organiziralo nije preporučljivo zbog nesistematičnosti koja će rezultirati greškama u taksonomiji u vidu nedostajućih kategorija. Stoga se tokom razvoja taksonomije određene grupe pojmova primjenjuje neka vrsta sistematičnog pristupa. Svi pristupi započinju na isti način; određivanjem gornje i donje rubne razine kategorije pojmova.

Pristupi se razlikuju prema načinu generiranja svih pojmova po svim razinama između ove dvije granice, a dijele se u dvije velike grupe; odozgo prema dolje (top- down) i odozdo prema gore (bottom up). Unutar ovih dviju grupa moguće su različite metode popunjavanja svih razina, a koje uključuju popunjavanje razine jednu po jednu te različite kombinacije redoslijed popunjavanja po granama. Međutim, za konačnu taksonomiju pristup popunjavanju grana je manje bitan. Krucijalna odluka leži u odabiru smjera popunjavanja.



Slika 6: Pristupi načinu generiranja pojmova po hijerarhijskoj strukturi

Pristup odozgo prema dolje nudi prednost preglednosti jer od samog početka paralelno stvara i pojmove i njihove međusobne odnose. Na početku stvaranje taksonomije generiran je vršni pojam, koji je sa stajališta inženjera vrlo općenit, te se potom krenulo u proces specijalizacije pojmova tj., generiranja nižih razina pojmova koji pokrivaju uže područje sve

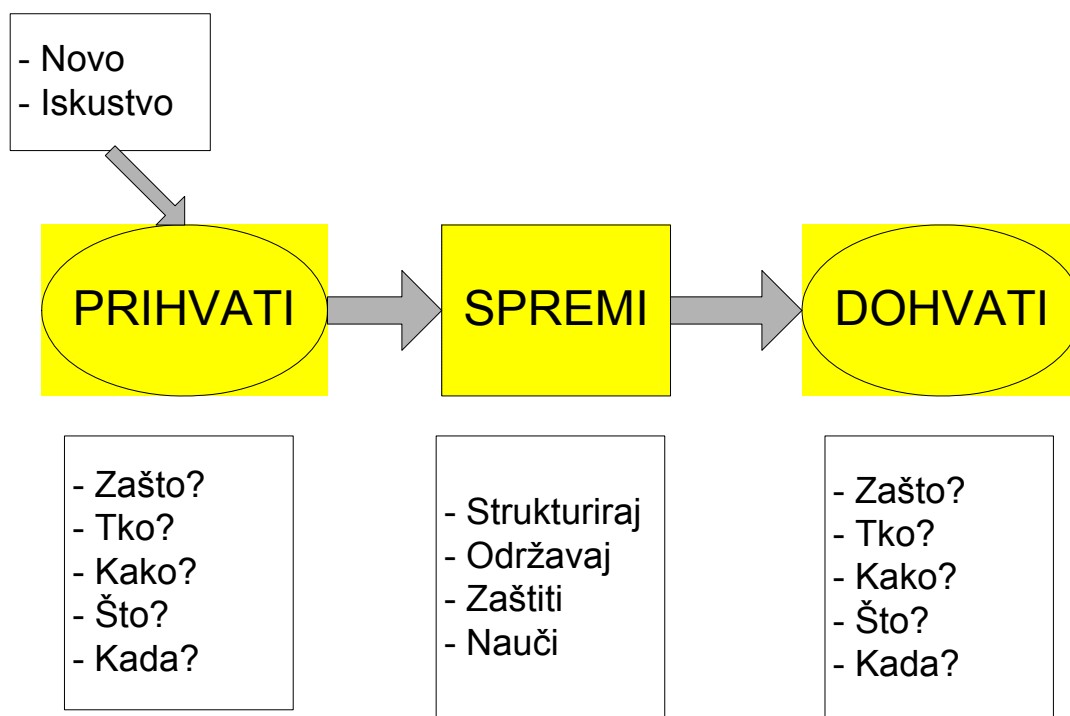
do razine koja se u inženjerskoj praksi smatra zasebnim pojmom te se ne dijeli dalje. Generiranje kategorizacije na ovaj način je provedeno u dva odvojena znanstvena projekta čime su stvorene dvije odvojene taksonomije. Iako su obje taksonomije krenule od najviše razine pojmova prema nižim, osnovna razlika u njihovom generiranju leži u pristupu generiranja teorije koja objašnjava praksu. Jedan projekt krenuo je od proučavanja prakse iz čega je kreirana teorija, dok je drugi krenuo od teorije te ju potom provjerio u praksi. U postupku je kreiranje združene taksonomije (Ahmed, Štorga, 2008., [21]) kombiniranjem ovih dviju taksonomija. Rad koji predstavlja ovu združenu taksonomiju u trenutku pisanja ovog rada nalazi se u procesu recenziranja s ciljem objave u znanstvenom časopisu.

U ovom radu biti će međutim primijenjen drugačiji kut gledanja. Umjesto od gornje granice pojmova, krenut će se od donje granice tj., krenut će se od pojmova koji se razmatraju kao pojedinačni pojmovi na razini konstruiranja u strojarstvu. Ovi pojmovi biti će skup uobičajenih procedura odabira koje nemaju nikakvu postojeću kategorizaciju. Obzirom na broj mogućih procedura, kreirat će se, iz jednostavnih organizacijskih potreba aplikacije, određene kategorije na osnovi pojedinih zajedničkih svojstava. Iako bi provedena podjela opisanih procedura mogla činiti prijedlog njihove kategorizacije, za pravu kategorizaciju bila bi potrebna znanstvena analiza koja nadilazi ovaj rad kako zbog njegove namjene, tako i zbog relativno malenog broja opisanih procedura.

Iako nije postavljen kao uvjet, jedan od ciljeva ovog rada je uspostavljanje veza i što veće podudarnosti sa navedenom združenom taksonomijom. Međutim, uslijed razlike u namjeni ovog rada (razvoj sustava za što jednostavnije korištenje u precizno definiranim slučajevima koji se učestalo javljaju u praksi) te postojeće taksonomije (stvaranje općeg sustava primjenjivog u svim slučajevima), za određene kategorije iz postojeće taksonomije neće postojati ekvivalentne kategorije u ovom radu.

Konačni je cilj ovog rada verificirati pristup odozdo pri čemu se prvo definiraju najniži elementi, a tek se potom određuju više razine strukture. Verifikacija će se naravno provesti u praksi kroz pokusnu upotrebu ovih procedura u konstrukcijskom uredu. Pored provjere primjenjivosti kao takve, rezultati pokusne primjene omogućiti će usporedbu sa kategorizacijom koja je kreirana od najvišeg pojma prema dolje. Međutim, svrha verifikacije nije utvrditi koji je od pristupa bolji već utvrditi poveznice ovih dviju sustava i ostvariti spajanje u jednu kategorizaciju koje će istovremeno biti primjenjiva u praksi i imati valjanu znanstvenu podlogu.

Pristup koji je odabran za ovaj rad rezultirat će stvaranjem sustava za upravljanje znanjem u skladu sa postojećim shvaćanjem principa rada ovakvog sustava (Ahmed, 2008, [13]), ali uz dvije značajne razlike u odnosu na sustave koji su svoju kategorizaciju znanja generirali „top-down“ pristupom. Prva razlika tiče se nepokrivenosti cijelog područja koje spada u kategorizaciju s obzirom na dane granice, kako gornje, tako i donje. Ovaj pristup će pokriti jedan mali segment cijelog područja što mu je i osnovna mana.

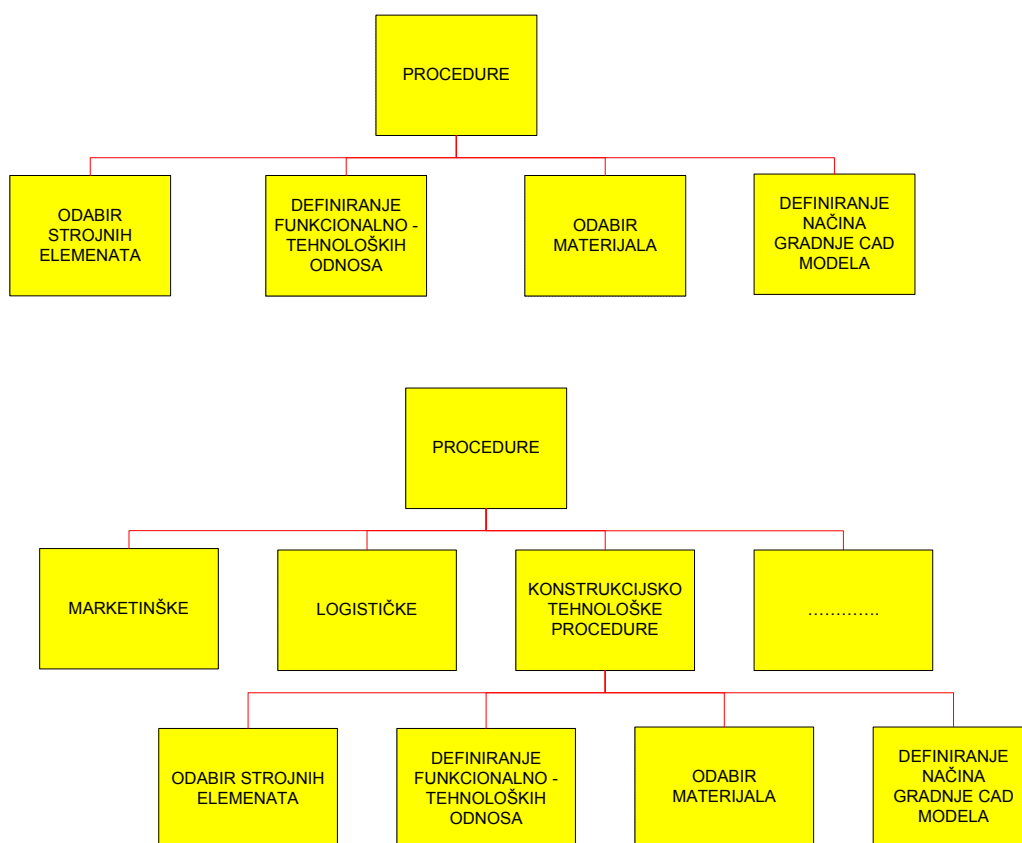


Slika 7: Shema rada sustava za upravljanje znanjem (Ahmed, 2008, [13])

Druga razlika leži u nedvosmislenosti pretraživanja koja osigurava korisniku da kod pretraživanja dobije, odabirom kriterija, uistinu sve zapise vezane uz pojedinu kategoriju zapisa. Punu primjenjivost ovog pristupa biti će, naravno, potrebno potvrditi u praksi, međutim, pokaže li se sustav primjenjivim, nedvosmislenost sustava rezultirat će jednostavnošću korištenje što će biti glavna prednost ovog sustava.

4. PRELIMINARNA ISPITIVANJA PREDLOŽENOG PRISTUPA NA KONKRETNOM KONSTRUKCIJSKOM UREDU

Tokom razvoja sustava, u trenutku kad su procedure počele poprimati oblik i dobivati dijelove koji su se mogli generirati na temelju dostupne literature, izvršen je intervju sa direktorom tvrtke „KLEX“ d.o.o. Zagreb, koji ima 20 godina radnog iskustva na području razvojnih projekata iz područja strojarstva. Tvrtka KLEX bavi se projektiranjem i razvojem proizvoda sa područja malih poljoprivrednih strojeva, uređaja na motorni i električni pogon.



Slika 8: Strukture procedura zapisa obrazloženja odabira

U intervjuu je naglašeno kako su procedure opisane u ovom radu samo dio procedura koje su uobičajene u procesu razvoja proizvoda te da pokrivaju samo područje neposredne tehničke razrade. Pored ovih procedura bilo bi moguće definirati i veliki broj procedura sa područja marketinga ili logistike. Definiranjem ovih dodatnih procedura, sve procedure opisane u ovom radu ušle bi u zasebnu kategoriju „konstrukcijsko - tehnoloških“ procedura

zapisa obrazloženja odluka. Uz ovu kategoriju bile bi definirane i druge kategorije čime bi procedure iz ovog rada iz druge organizacijske razine prešle u treću (Slika 8).

Pored mogućnosti definiranja novih vrsta procedura, utvrđeno je kako postoji i čitav niz novih vrsta uobičajenih razloga za odabire unutar procedura koje su tema ovog rada. Na ovaj će se način pojaviti novi tipovi razloga; marketinški, logistički, dizajnerski itd. Ove će nove vrste tipova razloga odabira biti objašnjene u poglavljima u kojima se budu objašnjavale pojedine procedure.

Također je upozoreno na značajne izmjene dijelova tokom razvoja proizvoda, posebice nakon testiranja prototipova i pokusne serije pri čemu objekti koji su ranije postojali u CAD modelu, više ne postoje. Zapisi vezani uz objekte koji više ne postoje jer su zamijenjeni drugim objektima, postaju nevezani zapisi. Druga opcija je da objekt i dalje postoji ali je značajno izmijenjen uslijed spoznaja koje su dobivene nakon stvaranja inicijalnih zapisa. U ovom slučaju određeni zapisi neće biti potpuno vezani jer jedna od dvije reference više ne postoji. Postoje dvije mogućnosti kako tretirati ovakav zapis. Prvi je da se izvrši dopuna zapisa uz kreiranje novih referenci. Na ovaj način na jednom mjestu zabilježeno je objašnjenje za posljednju odluku uz zapisanu povijest odluka i razloga zbog kojih su donesene. Ovakvi zapisi imali bi posebnu vrijednost zbog mogućnosti pregleda povijesti razvoja koja bi mogla ubuduće omogućiti zaobilaženje slijepih ulica razvoja pojedinih rješenja.

Druga mogućnost je da se zapis ostavi takvim kakav jest. Iako ovaj zapis više nema referenci, on ostaje kao zabilješka određene odluka koja je donesena u nekom trenutku da bi ju se kasnije moralo mijenjati. Kao i kod nadopunjenog zapisa moguće je formirati određena zaključke, međutim, za to je potrebno pronaći i novije zapise koji su na drugom mjestu.

Čini se kako bi općeniti zaključak po ovom pitanju trebao biti kako je bolje nadopunjavati isti zapis nego stvarati nove zbog toga što u ovom slučaju različiti zapisi sačinjavaju jedan zapis tj., „paket“ znanja će se formirati od nekoliko zapisa.

Uz problem naknadnog zapisivanja vezan je i problem višestrukih mogućih rješenja čiji odabir nije u nadležnosti tvrtke već se sva rješenja trebaju ponuditi poslovnim partnerima koji donose odluke. U inicijalnoj ideji sučelja predviđena je mogućnost upisa alternativnog odabira koji bi odgovarao traženoj mogućnosti upisa varijanti. Postavlja se samo pitanje broja mogućih varijanti. Jednom kad je donesena konačna odluka o varijanti koju se proglašava rješenjem, upisuje se ova konačna odluka kao naknadna dopuna zapisa.

U konačnici u razgovoru je utvrđeno kako se u praksi javljaju dvije vrste pretraživanja, specifično pretraživanje objašnjenja konkretne odluke gdje je poznato uz koji projekt, dio i

značajku je vezan zapis, te opće pretraživanje u kojem se pretražuju kategorije sa ciljem istraživanja niza sličnih prethodnih rješenja kako bi se na osnovu zabilježenog iskustva donijela odluka. O ovim tipovima pretraživanja bit će više riječi u poglavlju o pretraživanju baze.

U danima nakon izvršenog intervjua, direktor tvrtke sazvaio je sastanak svojeg razvojnog tima na kojem je metodom „olujna ideja“ (brainstorming) sastavljen popis najčešćih razloga za određeni odabir iz tri područja dogovorenih prethodno tokom intervjua.

Područja za koja su sastavljeni popisi su odabiri steznog spoja (dosjedi), ležajeva te konstrukcije orebrenja. Dani su razlozi za odabire navedenih pojmova sa područja razvoja projekata koje karakteriziraju da su izrađeni Većinom od lakih legura (Al, Mg) u tlačnom lijevu, tehničkih polimera, uz velika dinamička opterećenja i vibracije, sa zahtjevom za minimalnom težinom., predviđeni za proizvodnju u velikim serijama (od 10 000 – 500 000 komada godišnje) uz nužnu optimalizaciju troškova te održavanje konstantne kvalitete.

Popisi razloga biti će dani u poglavlju koje opisuje same procedure zapisa. Prikazat će se razlika između popisa razloga koji su nastali promišljanjem i pregledom literature s jedne, te razloga dobivene iz stvarne industrijske prakse s druge strane.

Na ovom mjestu upozoriti će se samo na važnost dobivenih podataka. U prvom redu podaci su važni jer ukazuju na postojanje učestalih razloga zbog koji se donosi određeni odabir. Ovime je djelomično potvrđena teza o postojanju uobičajenih procedura obrazlaganja odabira čijom bi definicijom stvorili preduvjeti za razvoj računalne aplikacije koja bi omogućila brzo i efikasno pohranjivanje znanje te njegovo kasnije dohvaćanje.

Druga karakteristika ovih podataka zbog kojih se pokazuju važnima je njihova različitost od standardnih razloga kakvi se mogu naći u literaturi, kako u različitim katalozima tako i u sveučilišnim udžbenicima. Industrijska praksa očito poznaje razloge koji su specifični za određeno područje. Usporedbom prikupljenih podataka bilo bi moguće sve razloge podijeliti na dvije velike skupine. U jednoj bili bi razlozi koji su univerzalne prirode, a koji bi se nalazili kao sastavni dio sustava, te na razloge specifične za pojedino područje, a koje bi svaka tvrtka trebala moći sama unijeti u sustav.

5. PROGRAMSKA IZVEDBA PROVOĐENJA PROCEDURA

U uvodu je rečeno kako je ideja zapisa znanja izrada programske aplikacije koja će nuditi obrazloženje skupa procedura donošenja odluka. Ovime će se omogućiti brže izvođenje ovih procedura zbog automatizacije unošenja jednog dijela podataka te omogućavanjem unošenja drugog dijela odabirom iz seta unaprijed definiranih odabira. Više govora o izvoru ovih pre-definiranih odabira biti će u slijedećem poglavlju u kojem će se, između ostalog objasniti i izgled Access forme.

Iz navedenog je moguće zaključiti kako ovakav zadatak ima dva dijela. Prvi dio je osmišljavanje procedura što uključuje definiciju procedura, njihovu razradu uključujući prikupljanje izvora podataka te generiranje algoritma za njihovo provođenje.

Drugi dio zadatka sastoji se od izgradnje računalne baze podataka koja predstavlja organizaciju znanja u digitalnom obliku, te od programiranja aplikacije koja će prikupiti potrebne podatke iz CAD modela i spremiti ih u odgovarajuće tablice baze podataka.

U opisu zadatka navedeno je kako je jedan od ciljeva ovog rada davanje prijedloga tehničke realizacije izvođenja procedura. Iz navedenog se može zaključiti kako ovaj rad neće sadržavati tehničku izvedbu koja se, zbog vremenske ograničenosti, nalazi van okvira ovog rada. Stoga će se rad zadržati na teoretskoj definiciji procedura te obrazloženju algoritma.

Na ovom mjestu objasniti će se zadnja stavka ciljeva navedenih u opisu zadatka; prijedloga tehničkog rješenja. Razlog za ovaj redoslijed slijedi iz činjenica da će način izvođenja procedura uvelike ovisiti o aplikacijama koje su upotrijebljene pri izvođenju, pa je stoga i sam opis procedura nešto drugačiji. Važno je naglasiti kako se same procedure ne mijenjaju, ali izgled aplikacije ovisi o mogućnostima alata kojima je rađena, programerskom i organizacijskom umijeću osobe koja je izradila aplikaciju te organizacijom same baze podataka.

5.1. CAD ELEMENTI POTREBNI ZA ZAPIS ZNANJA

Rečeno je kako će aplikacija automatizirati dio prikupljanja podatka u njihov upis u bazu tako što će omogućiti korisniku odabir elemenata CAD modela uz koje se želi vezati zapis te ih potom automatski smjestiti u bazu zajedno sa drugim relevantnim podacima iz CAD modela.

CAD je širok pojam koji podrazumijeva korištenje neke od programskih aplikacija za konstruiranje tj., općenito oblikovanje budućeg proizvoda. Iako postoji više tehnologija kojima se koriste različite aplikacije u stvaranju predodžbe izgleda budućeg proizvoda, pod CAD tehnologijom na ovom mjestu nećemo podrazumijevati grafiku generiranu rasterskom ili vektorskom grafikom jer je korištenje 2D prikaza na području strojarskih tehnologija u izrazitom padu, posebice u propulzivnim tehnologijama i velikim tvrtkama.

Budući da su CAD aplikacije na bazi tehnologije značajki (Feature Based Design - FBD) koja omogućava jednostavno generiranje 3D objekata danas praktički postale industrijski standard, planirani sustav procedura za zapis znanja vezat će zapisano znanje uz trodimenzionalni CAD model proizvoda i njegovih dijelova.

Važno je istaknuti kako bi eventualna aplikacija koja bi služila za vezanje znanja uz vektorski dvodimenzionalni prikaz generirala veze upitne smislenosti jer bi u tom slučaju veza postojala prema određenoj liniji, krivulji ili točki čime bi bilo praktički onemogućeno vezanje znanja uz određenu površinu ili cijeli strojni dio. Također, kako ne postoji veza između dijelova u radioničkim crtežima i njihovim prikazom u sklopu, svaki bi zapis vezan uz neki dio trebalo ručno kopirati i na svaku primjenu tog dijela u svim projektima u kojima bude korišten. Ovakav posao može se uz veliku sigurnost proglasiti uzaludnim.

Stoga FBD tehnologija nije odabrana samo zbog toga što je jedan od najraširenijih inženjerskih računalnih alata, već i zbog toga što je upitna realizacija aplikacije koja bi se vezala uz neku drugu CAD tehnologiju.

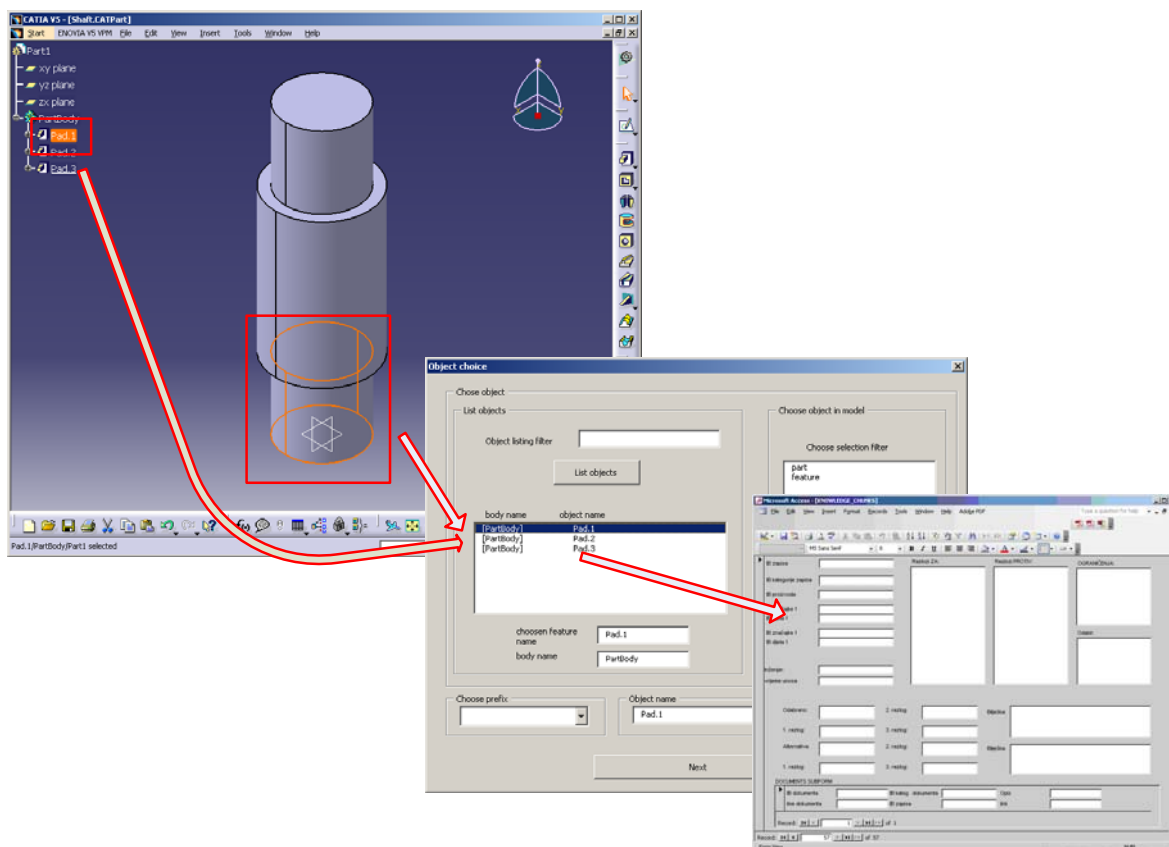
Prednosti parametarske tehnologije u ovom slučaju očitovat će se u elementima koji su omogućili izgradnju trodimenzionalnog modela proizvoda. Ovi isti elementi će u ovom slučaju biti objekti uz koje će se vezati znanje.

U poglavlju koje opisuje same procedure biti će navedeni za svaku proceduru prijedlozi elemenata uz koje bi se vezalo znanje. Na ovom mjestu navest će se svi elementi uz koje se može vezati znanje.

U najjednostavnijoj strukturi trodimenzionalnog CAD modela pojedini model strojnog dijela gradi se korištenjem niza značajki koje su definirane svojim atributima i parametrima kao vrijednostima tih atributa. Cijeli sklop dobiva se sklapanjem pojedinih dijelova pri čemu se definiraju odnosi između pojedinih značajki različitih dijelova kako bi se ostvarile veze između pojedinih dijelova u sklopu.

Prilikom stvaranja zapisa, veza će se u općem slučaju moći stvoriti prema pojedinoj značajki dijela, dijelu kao zasebnoj cjelini ili kao instanci u sklopu te sklopu u cjelini.

Pod značajkom dijela podrazumijevaju se kako značajke koje neposredno definiraju geometriju dijela, tako i značajke koje služe kao pomoćni elementi poput dodatnih geometrijskih ravnina, pravaca i točaka.



Slika 9: Vezanje zapisa znanja uz elemente iz stabla CAD modela

Za pojedine procedure biti će poželjno vezati zapis uz neki od rubnih elemenata modela kao što su plohe, krivulje i točke. Međutim, kako ovi entiteti ne pripadaju klasi značajke, oni se ne nalaze u stablu, te im se stoga ne daju posebna imena niti ih korisnik može vidjeti u stablu modela. Stoga će se u slučajevima kada se bude pokazalo najprimjerenijim vezati zapis uz neki od nabrojanih elemenata, pronaći alternativno rješenje.

Pored samog elementa, tj. imena elementa uz koji se veže zapis, aplikacija će prikupiti još nekoliko podataka iz CAD modela. Primjerice, u slučaju vezanja zapisa uz pojedinu značajku, aplikacija će u bazu pored imena značajke spremiti i naziv body-a, naziv cijelog dijela te puni put do datoteke u kojoj se nalazi taj dio. Isto tako, ako se je element pojedini dio u sklopu, aplikacija će spremiti naziv dijela, naziv sklopa u kojem se nalazi taj dio, koliko puta je taj dio korišten u sklopu te pune putove do datoteka koje sadrže dio i sklop.

Ovakvo prikupljanje podataka iz CAD modela i njihovo spremanje u bazu planirano je kako bi se omogućilo što brže i efikasnije formiranje „Paketa“ znanja (Knowledge Chunk KC), kako je prikazano na Slici 9. Dakako, moguće je formirati KC u bazi i bez automatskog prikupljanja podataka iz CAD modela, jednostavnim ručnim upisivanjem svih podataka. Međutim, uzevši u obzir količinu podataka koji se tiču CAD modela, a koje je potrebno upisivati u samo jednom zapisu, te potencijalno veliki broj zapisa koje će inženjeri unositi svakodnevno, ovakav polu-automatizirani sustav trebao bi donijeti značajnu uštedu u vremenu, ali i motivirati inženjere svojom jednostavnošću na njegovo korištenje.

Na ovom mjestu opisani su podaci koji aplikacija treba prikupiti iz CAD modela i spremiti u bazu podataka. U slijedećem poglavlju biti će riječi o pojedinim programskim alatima kojima se planira realizirati cijeli sustav.

5.2. PROGRAMSKI ALATI ZA IZVOĐENJE PROCEDURA

Izrada tehničke izvedbe predviđena je za diplomski rad autora, te je u ovom trenutku u izradi. Do završetka ovog rada, autor je usvojio (nužna) znanja potrebna za rad u CAD aplikaciji, za izgradnju baze podataka te izradu aplikacije za povezivanje jednog i drugog (PRILOG A). Na ovom mjestu je važno napomenuti kako je „veza“ između CAD modela i baze vrlo slaba, naime uspješno je ostvaren prijenos potrebnih podataka iz CAD aplikacije u bazu, međutim, nije ostvarena trajna veza između CAD modela i zapisa u bazi što predstavlja značajan problem, jer ograničava slobodu po pitanju manipulacije CAD modelima.

Naime, kako bi se zadržala smislenost zapisa koji sadrži imena objekata uz koje je vezan, ovi se objekti nakon stvaranja zapisa ne smiju preimenovati, premještati u drugu datoteku ili brisati. Kako baza podataka nije u mogućnosti pratiti promjene nastale nakon stvaranja zapisa, zapis vezan uz objekt nad kojim je izvršena neka od navedenih promjena gubi smisao, jer više nije vezan uz objekt.

U industrijskoj praksi ovakav nedostatak učinio bi cijeli sustav beskorisnim. Bilo bi stoga nužno stvoriti aplikaciju koja će stvoriti trajnu vezu zapisa i CAD modela kako bi se sa svakom promjenom CAD modela ažurirale i baza podataka. Uzevši u obzir prirodu zadatka, kompetencije autora, zadane rokove te tehnička ograničenja, na ovom mjestu uvest će se značajno pojednostavljenje industrijske prakse, tj. pretpostaviti će se da se CAD modeli na koje će se odnositi zapisi neće mijenjati na neki od prethodno opisanih načina.

Što se tiče samih programskih alata koji su planirani za tehničku izvedbu, oni uključuju CAD aplikaciju CATIA-u V5R18 tvrtke Dassault Systems (u daljnjem tekstu CATIA) te aplikaciju za izradu baza podataka, Microsoft Office Access 2003 (u daljnjem tekstu Access).

CATIA je jedan od vodećih CAD aplikacija na tržištu, a za ovaj projekt odabrana je iz dva razloga. Prvi je što CATIA u sebi sadrži programski jezik Visual Basic for Applications (u daljnjem tekstu VBA) koji je zapravo pojednostavljena verzija Microsoftovog Visual Basic 6.0 programskog jezika. Jednostavnost pisanja koda omogućit će relativno jednostavnu izradu aplikacije za prikupljanje podataka i njihovo prebacivanje u bazu. Kako su i CATIA i Visual Basic jako rašireni, relativno je jednostavno pretraživanjem interneta pronaći rješenja za veliki broj problema.

Drugi razlog za odabir CATIA-e povezan je s odabirom aplikacije za gradnju baze podataka. Access je odabran zbog jednostavnosti korištenja budući da se relacije između

tablica koje sačinjavaju bazu postavljaju grafički, povlačenjem veza između tablica. Također vrlo je jednostavno stvoriti i brojne druge objekte poput formi ili izvještaja.

Access nije najmoćniji alat za izgradnju baze podataka, dapače, jedan je od najslabijih kad su u pitanju veličina i kompleksnost baza kojima može rukovati. Međutim, uzevši u obzir bazu podataka kojom je potrebno rukovati u ovom slučaju, prednosti jednostavnosti korištenja će višestruko premašiti ograničenja po pitanju veličine, posebice stoga što se ne očekuje da će baza narasti ni približno blizu mogućnosti Accessa.

Kako je Access proizvod Microsofta, ne očekuju se problemi u komunikaciji VBA, budući da su oba programa proizvodi iste tvrtke. Nadalje, u Access je također ugrađen VBA te će stoga biti jednostavnije programirati sve one algoritme koje neće biti moguće automatizirati standardnim Accessovim alatima.

Naravno, CATIA nije jedini CAD sustav za koji je moguće napraviti aplikaciju koja će prikupljati podatke i slati ih u bazu, niti je Access jedini alat za izgradnju baze koji je primjenjiv. U slučaju razvoja stvarne industrijske verzije aplikacije i baze, u obzir bi došlo korištenje mnogo jačih alata. Ponovo, s obzirom na pretežito akademsku prirodu ovog projekta, navedeni alati su optimalni.

Iako je svaka procedura zasebna, te se svaka sastoji od svojih specifičnih koraka, među njima postoji relativno veliki broj identičnih koraka, a među ostalim koracima postoji i veliki broj relativno sličnih koraka. Stoga će, prije opisa provedbe samih procedura, biti dani opisi ovih zajedničkih koraka uz predočenje idejnog rješenja sučelja aplikacije prema korisniku.

Kao što je već rečeno, razlog za objašnjavanje prijedloga tehničke izvedbe prije samih opisa procedura je specifičnost provođenja procedura i izgled sučelja koji su uvelike uvjetovani mogućnostima primijenjenih alata. Navedeni alati su upotrijebljeni za dizajn sučelja čiji će dijelovi biti dani, kako uz opis zajedničkog dijela procedura, tako i uz svaku proceduru zasebno.

U konačnici, prije nego se krene na same opise, važno je naglasiti kako su dana sučelja idejno rješenje. U konačnoj verziji, kada sama aplikacija te baza podataka budu izrađene, vjerojatno će sučelje biti ponešto izmijenjeno. Međutim, u osnovnoj strukturi, ne očekuju se korijenite promjene izgleda sučelja.

Na ovom mjestu je važno naglasiti kako sam sustav neće davati odgovore na pitanje što inženjeru treba tj., ovaj sustav nema sposobnost donošenja odluka. Procedure koje će biti definirane u ovom radu poslužiti će kao teoretska osnova kreiranja aplikacije za spremanje znanja, ne njegovog generiranja. Aplikacija neće generirati rješenje za zadani problem niti ga

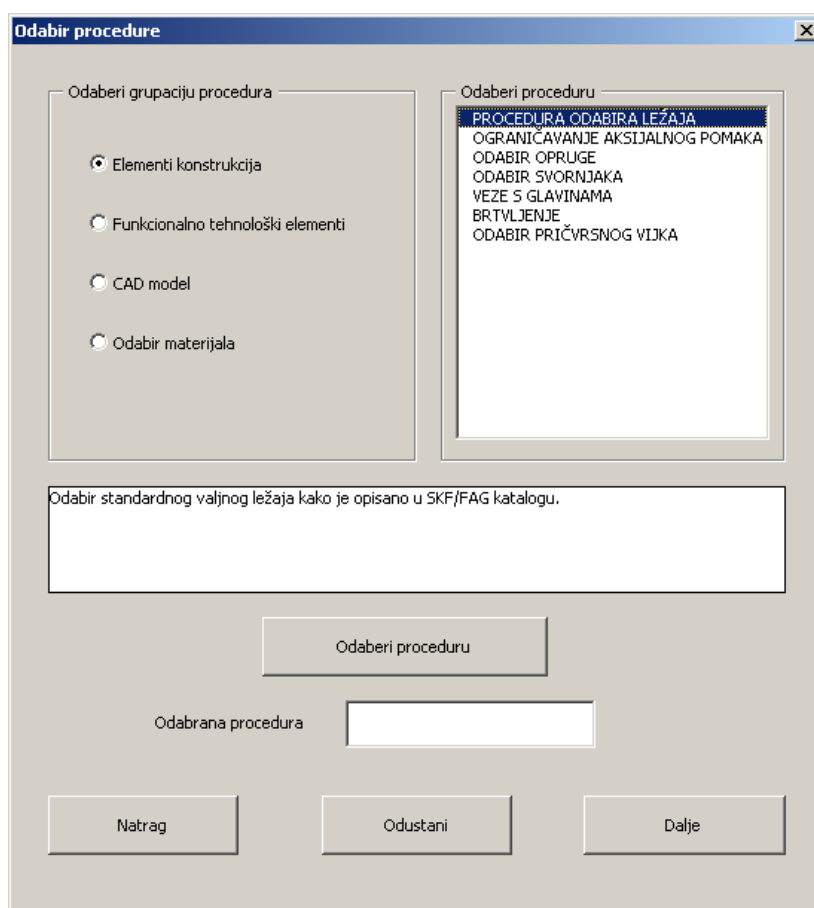
obrazlagati već automatski spremiti dio podataka, ponuditi mogućnost odabira standardiziranih rješenja te olakšati upisivanje preostalih podataka.

Potporna donošenju odluka proizlazi iz zapisa koji će biti unijeti. Jednom kad baza bude sadržavala dovoljan broj zapisa korisnik će moći pretraživati bazu u potrazi za prethodnim rješenjima za određene probleme koja mu mogu pomoći u donošenju odluke.

6. POČETNA - ZAJEDNIČKA FAZA IZVOĐENJA PROCEDURA

Namjena planirane aplikacije je omogućavanje korisniku da sam postupak provođenja procedure donošenja i zapisa standardnih odluka izvede u jednom sučelju sa što je moguće manje vlastitog angažmana po pitanju potrebe za zapisivanjem te istovremenim paralelnim radom sa CAD sustavom i bazom podataka. Stoga je aplikacija planirana za pokretanje neposredno iz prozora CATIA-e.

Idejno je zamišljeno da se inženjer nalazi u procesu modeliranja nekog strojnog dijela, te je došao do točke u kojoj mora donijeti odluku koja spada u standardne procedure, što znači da će aplikacija moći izvršiti zapis obrazloženja donesene odluke.



Slika 10: Sučelje za odabir procedure

6.1. POKRETANJE PROCEDURE

Kako je aplikacija koja će inženjeru pomoći u procesu zapisa obrazloženja pisana u CATIA-inom VBA, pokretanje se vrši iz CATIA-e. Postoje dvije mogućnosti pokretanja. Prva je da se aplikacija pokrene kao makronaredba iz izbornika makronaredbi, Druga opcija je da se aplikacija pokrene pritiskom na ikonu postavljenu u neki od izbornika. Ovisno o učestalosti korištenja aplikacije, jedan od ova dva načina će imati prednosti. Ukoliko korisnik bude često stvarao zapise, biti će funkcionalnije pokretati aplikaciju iz glavnog prozora kao i sve ostale često korištene naredbe. Dijametralno suprotno, ukoliko stvaranje zapisa nije često, dodatna ikona na osnovnom sučelju samo bespotrebno zauzima korisnu površinu ekrana.

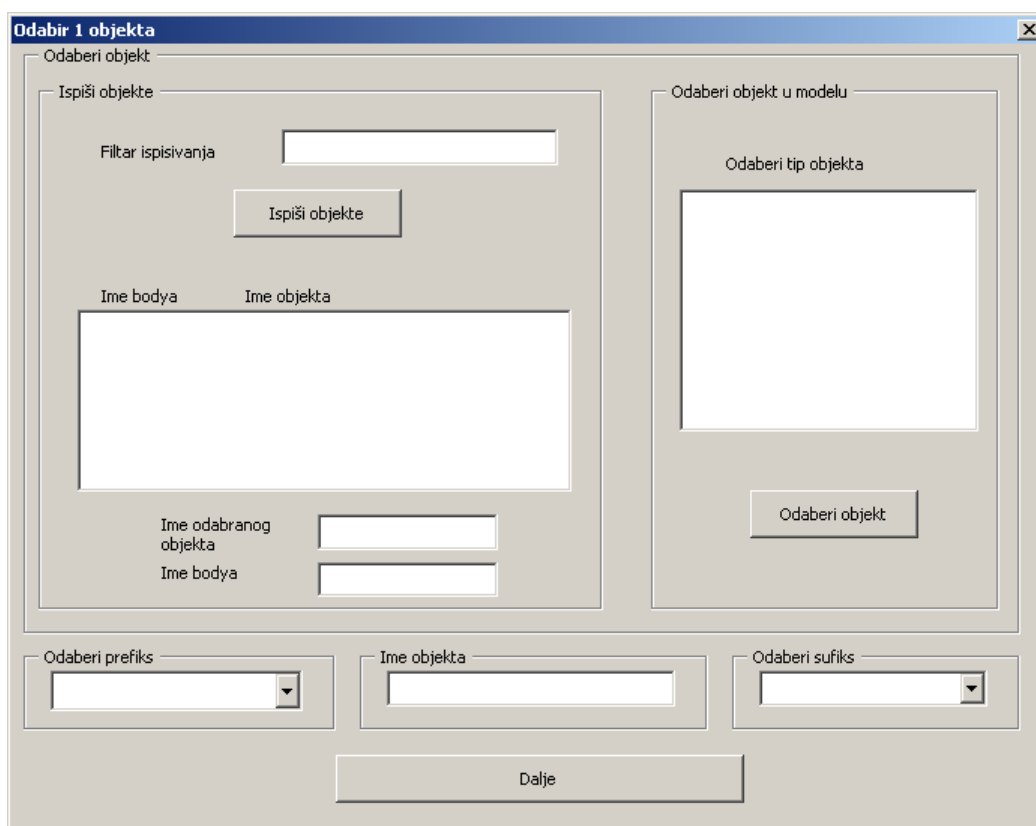
Neovisno o načinu pokretanja, korisniku se prikazuje ekran za odabir procedura zapisa. Odabirom grupacije procedura u desnom okviru ispisuje se popis procedura koje postoje u toj grupaciji. Korisnik označuje željenu proceduru u danom popisu te pritišće tipku „Odaberi proceduru“. U polju „Odabrana procedura“ ispisuje se ime, a u srednjem okviru kratak opis trenutno odabrane procedure. Kad je korisnik zadovoljan odabirom, započinje provođenje procedure koja će u konačnici rezultirati stvaranjem novog zapisa.

6.2. ODABIR CAD OBJEKTA

Prvi korak u provođenju svake procedure je odabir objekta u CAD modelu za kojeg će se vezati zapis. Tip objekta varira od procedure do procedure, dapače, čak i u slučaju iste procedure postoje logični razlozi za vezanje znanja uz različite elemente. Uz svaku proceduru razmotrit će se mogućnosti tipa objekata za koje postoji valjani razlog za vezanje elemenata kao i uočene prednosti i nedostatke pojedinih mogućnosti, međutim ovaj rad neće davati konačne preporuke iz jednostavnog razloga što za većinu mogućnosti ne postoji neki presudni razlog za ili protiv pojedine mogućnosti. U konačnici opći zaključak jest kako u slučajevima kad postoji više mogućnosti treba donijeti odluku o tipu objekta uz koji će se vezati određeni tip zapisa, međutim samu odluku treba donijeti na razini tvrtke jer nije moguće donijeti opći zaključak.

Neovisno o tipu objekta, postavlja se pitanje broja objekata uz koje će biti potrebno vezati zapis. Tokom osmišljavanja procedura uočeno je kako se donesene odluke tiču ili

jednog ili dva objekta u CAD modelu. U određenim slučajevima postoji tendencija za vezanje zapisa uz potencijalno puno veći broj objekata kao što su primjerice obrazloženje tehnološkog zaobljenja na lijevanim kućištima. Ovih zaobljenja doduše može biti i po nekoliko stotina. Međutim, dotična zaobljenja ili su oblikovana jednom značajkom koja u sebi sadrži reference na veliki broj rubova što je onda zapravo jedan objekt, ili se radi o toliko velikom broju značajki da je jednostavnije i smislenije vezati zapis uz cijeli dio nego odabirati svako pojedino zaobljenje.



Slika 11: Prozor za odabir jednog objekta

Stoga će korisniku za odabir objekta ili objekata, ovisno o zahtjevima procedure biti prikazan jedan od dva prozora; prozor za odabir jednog objekta ili prozor za odabir dva objekta.

Korisnik može odabrati objekt na dva načina. Prvi način je da aplikacija ispiše sve objekte trenutno aktivnog CAD modela te potom korisnik bira objekt sa popisa. Druga mogućnost je da korisnik odabere objekt neposredno u CAD modelu.

Rečeno je kako tip objekata uz koji se veže pojedini tip zapisa ovisi o proceduri, te da ga treba definirati na razini tvrtke. Vrsta objekta koji se bira ne mijenja značajno prozor. Ako

se razmotri Slika 12, uočava se da je uz svaki objekt u popisu objekata (lijevi okvir) postavljeno mjesto i za naziv body-a objekta. Body je prva više organizacijska instanca značajke dijela u CATIA Part dokumentu. Dakle, u slučaju kad je tip objekta značajka nekog dijela, uz ime značajke će biti ispisan naziv pripadajuće body-a jer je moguće u istom dijelu imati dvije značajke istog imena ako nisu u istom body-u. U slučaju odabira jednog dijela iz podsklopa nekog glavnog sklopa, prva viša instanca neće biti body, već podsklop.

Odabir 2 objekata

Odaberi objeat

Ispisi objekte

Filtar ispisivanja

Ispisi objekte

Ime bodya Ime objekta

Odabrani objekt

Body

Odaberi objekt u modelu

Selekcijski filter

Odaberi objekt

Odaberi prefiks

Ime objekta 1

Odaberi sufiks

Odaberi prefiks

Ime objekta 2

Odaberi sufiks

Dalje

Slika 12: Prozor za odabir dva objekta

Filtar za ispisivanje omogućava korisniku da odabere koje će objekte ispisati prema dijelu imena koje korisnik unese. Kod složenih modela koji se sastoje od nekoliko stotina značajki, bilo bi korisno unijeti tip značajke za koju se želi vezati zapis jer bi značajno smanjilo broj objekata koji će biti ispisani. Isto tako, u vrlo složenim sklopovima broj dijelova je vrlo velik pa ako se zna tip dijela uz koji se želi vezati zapis (neki zupčanik u višestupanjskom mjenjaču brzina), ispisana lista se može značajno skratiti što olakšava pretraživanje.

Ako korisnik odluči odabrati objekt neposredno u CAD modelu, morat će odabrati tip objekta koji želi odabrati i potom kliknuti „Odaberi objekt“ (desni okvir). Aplikacija će prikazati trenutno aktivni objekt, te jednom kad je objekt odabran, vratiti se na prozor za odabir objekta ispisujući ime te druge relevantne podatke za objekt koji je odabran u pripadajuće Textboxove.

U konačnici, jednom kad je korisnik zadovoljan sa odabirom, pritiskom na tipku “Dalje”, korisnik prelazi na slijedeći korak provođenja procedure.

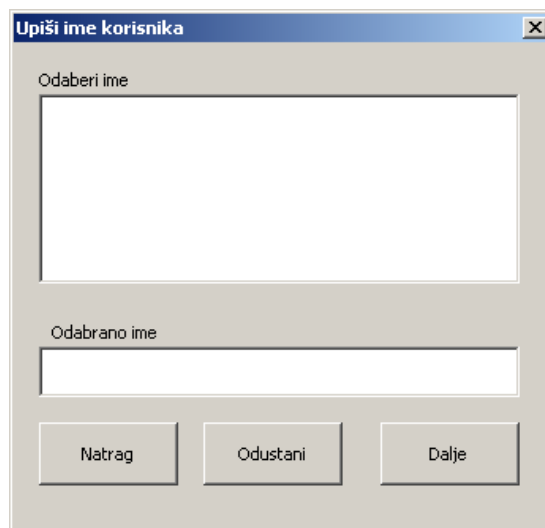
Ime koje stoji u polju „Ime objekta“ je ime koje će se zapisati u bazu znanja, ali i u sam CAD model. Stoga, ako korisnik želi, može na ovom mjestu promijeniti ime objekta uz koji veže zapis. Kako bi se standardizirala promjena imena, uz svako ime moguće je staviti neki od zadanih prefiksa ili sufiksa koji će biti uneseni u bazu kao dio imena, ali i u sam model. Namjena prefiksa i sufiksa je postavljanje jednostavne oznake u ime objekta u CAD modelu koja će naznačivati da uz dotični objekt postoji zapis unesen u bazu podataka. Sam sadržaj prefiksa i sufiksa, te pravila njihovih dodjeljivanja treba donijeti na razini tvrtke. Ukoliko se za neke procedure eksplicitno definira način preimenovanja objekta uz koji se veže zapis, moguće je dodatno automatizirati postupak preimenovanja. Također, automatsko preimenovanje služit će i za pretraživanje baze jer će korisnik moći tražiti da mu se ispišu samo zapisi vezani uz ime jednog elementa ili imena koja sadrže pojedini prefiks i/ili sufiks.

Nakon odabira objekta (objekata), korisniku se daje prozor sa mogućnostima odabira koje su specifične za svaku pojedinu proceduru. Ovi prozori biti će prikazani uz objašnjenja tijeka procedura.

Na ovom mjestu prikazat će se još dva prozora koja su (približno) istovjetna za sve procedure, a nalaze se na kraju izvođenja procedura. Prvi prozor služi za upisivanje podataka o osobi koja je unijela zapis dok je drugi prozor forme Accessa koji prikazuje sve dotad upisane podatke te omogućava odabir i upis dodatnih podataka.

6.3. UPIS PODATAKA O KORISNIKU

Prozor upisa podataka o korisniku koji unosi zapis omogućava izbor imena s popisa osoba koje rade na projektu. Sam popis je sa svim podacima o osobama spremljen u bazi, a na ovom mjestu prikazani su samo osnovni podaci; ime, prezime te funkcija u tvrtci.



Slika 13: Upis imena inženjera koji generira zapis

Izgled prozora za odabir korisnika koji unosi zapis ovisi o broju ljudi koji radi na nekom projektu. Ukoliko se radi o manjem broju konstruktora, pronalaženja svog imena na popisu neće biti teško. Međutim, ako se radi o projektu na kojem sudjeluje nekoliko desetaka ljudi, javit će se potreba za nekom vrstom filtra koji će prema zadanom kriteriju ispisati smanjenu listu ljudi zaduženih za neki projekt.

6.4. PRIKAZ ACCESS FORME

Prozor na Slici 14, ujedno je i zadnji prozor koji će se prikazati korisniku. Prikazat će se opći oblik koji uključuje sve moguće podatke koji mogu sačinjavati jedan zapis.

Forma prikazuje slučaj kada se zapis veže uz dva objekta koji su značajke dvaju dijelova koji se nalaze u istom proizvodu. U slučaju manjeg broja objekta za koje je vezan zapis, ovisno o tipu objekta, određena polja se neće prikazivati korisniku. Dakle, u svakoj proceduri biti će prikazana samo ona polja koja je automatski ispunila aplikacija nakon odabira objekta (objekata) uz koje je vezan zapis.

The screenshot shows the Microsoft Access application window titled "Microsoft Access - [KNOWLEDGE_CHUNKS]". The menu bar includes File, Edit, View, Insert, Format, Records, Tools, Window, Help, and Adobe PDF. The toolbar contains various icons for file operations, editing, and formatting. The form is displayed in Form View and contains the following elements:

- Left Panel:** A list of fields for data entry:
 - IB zapisa
 - IB kategorije zapisa
 - IB proizvoda
 - IB značajke 1
 - IB dijela 1
 - IB značajke 1
 - IB dijela 1
 - Inženjer:
 - vrijeme unosa
- Main Area:** Large text boxes for "Razlozi ZA:" and "Razlozi PROTIV:". To the right, there is a section for "OGRANIČENJA:" and an "Odabir:" section.
- Bottom Section:** Fields for "Odabrano:", "1. razlog:", "Alternativa:", and "1. razlog:". To the right, there are fields for "2. razlog:", "3. razlog:", and "Bilješka:".
- DOCUMENTS SUBFORM:** A subform containing fields for "IB dokumenta", "Ime dokumenta", "IB kateg. dokumenta", "IB zapisa", "Opis", and "link".
- Status Bar:** Shows "Record: 1 of 1" and "Record: 57 of 57".

Slika 14: Sučelje Access forme

Dva najveća polja namijenjena su ispisivanju unaprijed definiranih razloga za i protiv određene odluke. Očekuje se da za određene odabire u procedurama postoje razlozi za i protiv

koji se ponavljaju, tj. očekuje se kako se pojedine odluka donose u obliku odabira jednog od konačnog broja razloga. Također, očekuje se kako će se pojedini razlozi pojavljivati u relativno velikom broju procedura. Tehnička izvedivost ili niska cijena samo su dva primjera ovakvih razloga.

Postavlja se pitanje izvora ovih razloga. Očekuje se kako će iskusni inženjeri konstrukcijskog ureda biti glavni izvor uobičajenih razloga za ili protiv određenog odabira. Naravno, da bi se prikupili ti razlozi, bila bi potrebna dugotrajna suradnja kako bi se razmotrile sve procedure i mogućnosti odabira unutar njih što uvelike nadilazi opseg ovog rada.

Međutim, dok je sam proces prikupljanja i obrazlaganja pojedinih razloga za određeni odabir dugotrajan posao, njihov upis u bazu je vrlo jednostavan. Polja razloga za i protiv i te razloga ograničenja prikazuju podatke koje su prikupile iz određenih tablica koje čine strukturu baze. Popunjavanje ovih tablica je izrazito jednostavno čime je omogućeno formiranje baze bez zapisa te njezino predavanje korisniku na upotrebu. Korisnik će moći sam unijeti sve razloge čime je maksimalno osigurao svoje znanje od neovlaštenog korištenja.

Kod složenih slučajeva pojedini odabiri nisu nastali na temelju jednog razloga, već su odabrani kao optimalna rješenja za čitav niz parametara. Donošenje ovakvih odluka je samo za sebe poprilično složeno i najčešće uključuje korištenje neke od složenijih metoda odabira čiji je tijek zabilježen u zasebnom dokumentu ili određenom skupu dokumenata.

Jedna od metoda odabira koja je u suštini jednostavna, a postaje složena zbog velikog broja zapisa, je crtanje „DreD“ grafa (Bracewell, Wallace, 2007, [19]). Danas postoje komercijalne aplikacije koje omogućavaju jednostavno crtanje ovakvih grafova. Također, u trenutku pisanja ovog rada, aplikacija za crtanje istog tipa grafa u razvoju je na Fakultetu strojarstva i brodogradnje Sveučilišta u Zagrebu (Vadla, 2008 [15]).

Kako je „DReD“ graf zapravo objašnjenje samog odabira, ponovno objašnjavati razloge za odluke koje prikazuje ovakav graf je nepotrebno. Stoga bi u slučaju da je odluka donesena tokom izgradnje ovakvog grafa, bilo najučinkovitije stvoriti vezu na dokument ili skup dokumenata koji objašnjava kako je napravljen odabir.

Dio zapisa u kojem se nalazi obrazloženje za pojedini odabir je srž cijelog projekta, jer je cilj bio da se uz zapis o donesenim odlukama (koji i inače postoji jer se u tehničkoj dokumentaciji vidi što je napravljeno) veže zapis razloga zbog kojih je donesena određena odluka. Odabrani razlog upisuje se u polje razloga. Dodatno objašnjenje korisnik može unijeti u polje bilješke.

Sami razlozi za pojedini odabir neće se upisivati na ovom mjestu već će biti odabrani na prozorima oblikovanim za svaku proceduru zasebno. Na ovom mjestu nalaze se ispisani zajedno sa svim ostalim zapisanim podacima kako bi se korisniku pružio pregled nad cijelim zapisom.

Ispod područja u kojem se nalaze podaci o donesenoj odluci nalazi se identično područje u kojem je moguće upisati alternativno rješenje sa objašnjenjem zašto nije primijenjeno tj., u kojim bi uvjetima alternativno rješenje bilo bolje od trenutno odabranog.

Na dnu forme nalazi se područje u kojem je moguće stvoriti „hiperlinkove“ na vanjski dokument ili URL adresu. Korisnik može postaviti jedan „hiperlink“ ili više. Koliko će ih biti, ovisi o pojedinom slučaju te o procjeni inženjera da li je neka datoteka bila dovoljno važna za donošenje odluka. Zatvaranjem forme zaustavlja se procedura, podaci se spremaju u bazu te se korisnik vraća u okruženje CATIA-e.

Tokom prikaza zajedničkih koraka procedura prikazano je zapravo unošenje podataka čiji je tip istovjetan svim predloženim procedurama. U ove podatke spadaju identifikacijski broj (IB) zapisa, identifikacijski broj kategorije zapisa (tip procedure), CAD objekt uz kojeg je vezan zapis, vrijeme unosa te ime inženjera biti će prikazano kako će ovi zapisi omogućavati pretraživanja svih baza, tj. biti će moguće pretraživanje zapisa kojima nije jedina zajednička karakteristika da su zapisani istom procedurom.

Postoji još jedna zajedničko svojstvo koje je moguće za sve procedure. Postoji mogućnost da je napravljan pogrešan odabir u proceduri. Pod pogrešnim odabirom na ovom mjestu ne podrazumijeva se da je korisnik odabrao krivu proceduru zapisa odabira ili krivi CAD objekt uz koji će vezati zapis. Ovakve greške mogu se ispraviti tokom samog provođenja procedura (tipke dalje - natrag) ili popravkom unosa u samoj bazi.

Greške o kojima se ovdje govori su greške u samom odabiru. Sustav koji je primijenjen ne može biti kriv za ovakve greške jer sustav ne generira rješenja već ih samo zapisuje. Korisnik je dakle, odgovoran za svoj krivi odabir. O razlozima zbog kojih je došlo do pogrešnog odabira na ovom mjestu neće se raspravljati jer to nije tema ovog rada, međutim ustvrdit će se kako se tokom procesa konstruiranja donosi i određeni broj loših odluka.

Također, ako se uzme u obzir materijalna i krivična odgovornost pojedinca koji je donio krivu odluku, uvijek postoji opasnost od naknadne promijene zapisa odluke kako bi se izbjegla odgovornost za nastalu štetu ili ozljeđivanje. Onemogućavanje promijene zapisa ili neki drugi vid zaštite baze podataka svakako će trebati razmotriti prije eventualnog ulaska sustava u praktičnu primjenu.

Ako se zasad ostavi postrani ovakav scenarij, ostaje još uvijek veliki broj slučajeva u kojima je donesena odluka za koju se tek u praksi moglo pokazati da li je dobra ili ne, tj. da li se trenutačna odluka pokazala apsolutno lošom, izvanredno dobrom ili se pak možda naknadno utvrdilo kako postoji bolja opcija.

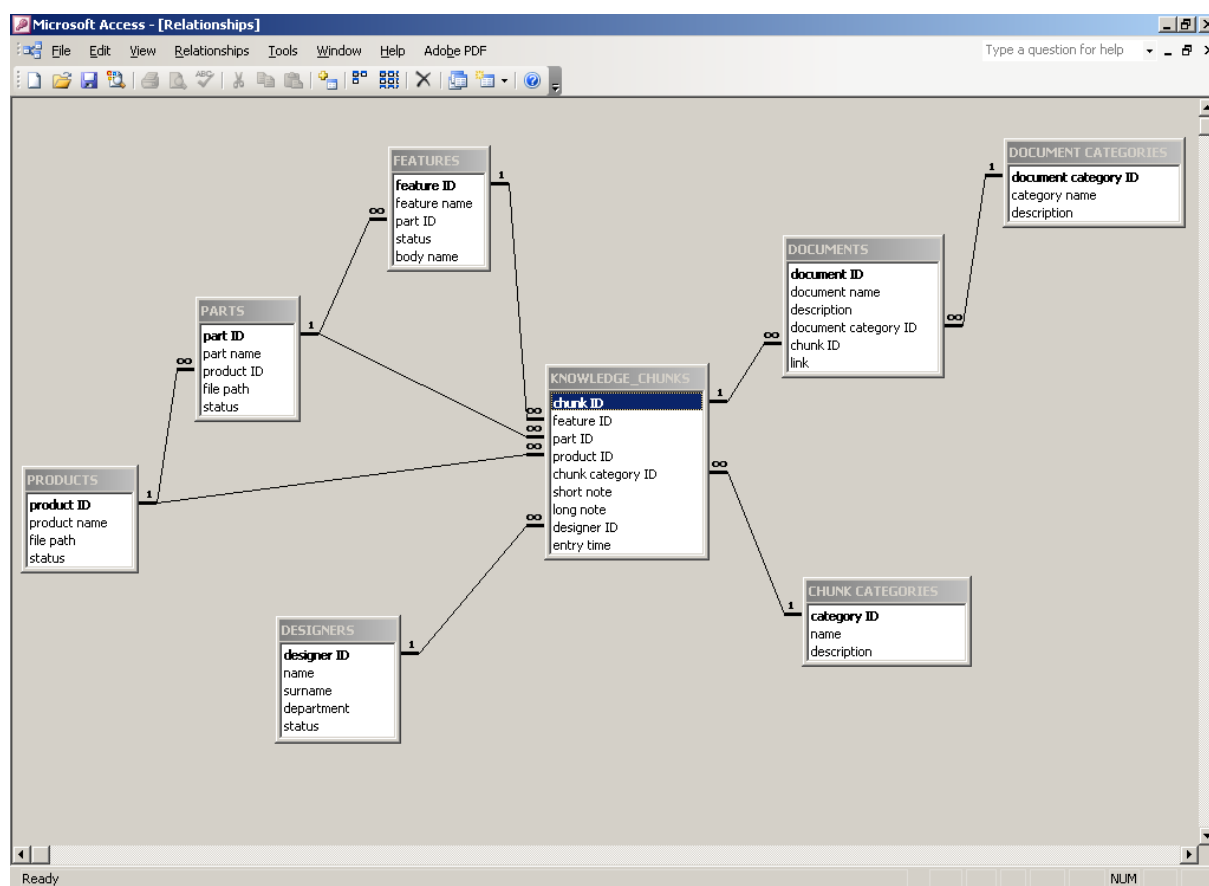
U svakom slučaju, potrebno je omogućiti upisivanje dodatnog zapisa koji će, sve koji budu pregledavali zapis, upozoriti da postoje određene izmjene, što je izmijenjeno i zašto je izmijenjeno.

Što se tiče tehničke izvedbe, najjednostavnije rješenje je postavljanje dodatna tri polja ispod polja za odabir i alternativu. Polja za izmjenu odluke izgledaju identično onima za odabir i alternativu, a imaju i identičnu namjenu. Korisnik će u njih unijeti koja je izmijenjana odluka, razlog za nju te, opcionalno, dodatnu bilješku. Ukoliko je potrebno vezati dodatni dokument koji je povezan sa izmjenom odluke, kao što su izvješća iz proizvodnje, testiranja ili servisa, moguće je postaviti hiperlink u pod-formu koja već postoji.

7. STRUKTURA BAZE ZNANJA

U opisu zadatka rečeno je kako je zapise znanja potrebno spremiti u računalnu bazu podataka kako bi se omogućilo brzo i efikasno pretraživanje. U prethodnom poglavlju objašnjeno je koji se podaci žele upisati u bazu te kako se planira izvesti njihovo upisivanje. U ovom poglavlju biti će više riječi o strukturi baze te načinima pretraživanja.

Specifičnost Microsoft Access baza podataka je stvaranje čitave baze u jednoj datoteci, tj. sve tablice, forme, upitnici i izvješća nalaze se na jednom mjestu. Iako ovakav pristup ima svojih prednosti kod manjih baza podataka čijoj je izgradnji i namijenjen posebice zbog toga što za izgradnju baze nije potrebno znanje sa područja programiranje baza podataka, u ovom slučaju pojavit će se problem sa veličinom cijele baze.



Slika 15: Struktura baze znanja

U sklopu zadatka Diplomskog rada autora ovog rada formirana je baza znanja u Microsoft Accessu s ciljem dokazivanja izvedivosti formiranja aplikacije koja će povezati CAD aplikaciju sa bazom podataka. Na Slici 15 prikazana je struktura ove baze. Sama baza

formirana je oko tablice „paketa“ znanja (Knowledge Chunk - KC) prema kojoj su formirane sve ostale veze.

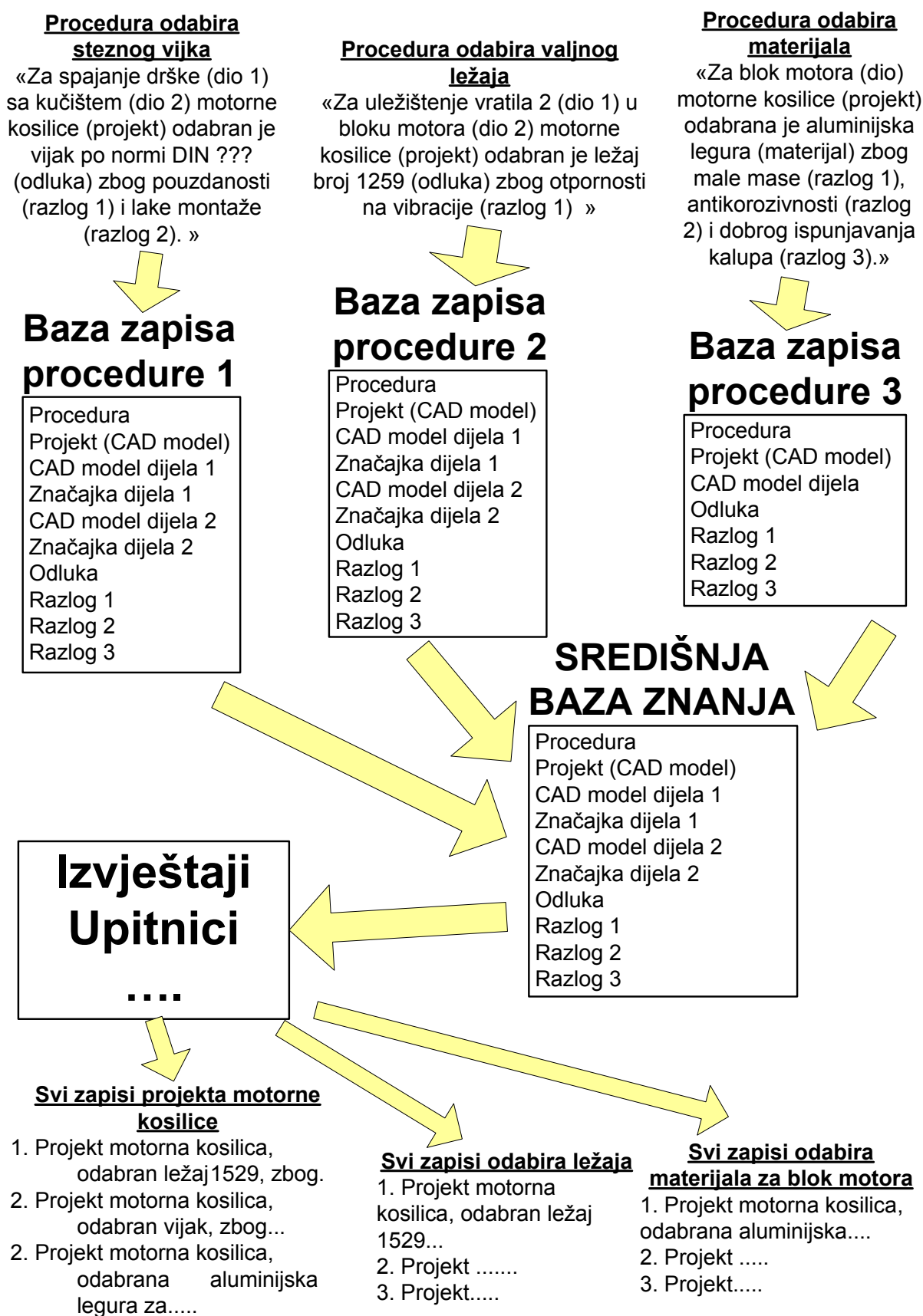
Središnja baza znanja procedura koja bi sadržavala „pakete“ znanja imala bi vrlo sličnu strukturu uz razlike u samoj tablici KC-ova gdje bi morala biti dodana dodatna polja za određeni najveći broj razloga koji se pojavljuju kao obrazloženja odabira, određeni broj mogućih upisa alternativa kao i polja za naknadne dopune i izmjene zapisa.

Pored same baze koja sadrži tablice „paketa“ znanja sa upitnicima, formama za upis te izvješćima koji omogućavaju pretraživanje, biti će potrebno uvesti dodatne tablice u kojima će se spremati podaci vezani za karakteristične procedure. Unošenje svih tablica u bazu sa „paketima“ znanja zajedno sa svim ostalim elementima, uzrokovalo bi nepreglednost cijele baze zbog velikog broja veza koje bi bilo potrebno uspostavljati unutar jedne baze. Gubitkom preglednosti nad bazom, izgubio bi se i jedan od primarnih razloga za korištenje Microsoft Accessa za njezinu izgradnju.

Pri izgradnji je stoga baza podijeljena na manje baze, po jedna za svaku proceduru zapisa znanja te jednu bazu koja će objedinjavati zapise, tj. prikupljati zapise iz njihovih matičnih baza prilikom generiranja odgovora na upite korisnika.

Svaka od baza znanja za pojedinu proceduru ima dva dijela. Prvi dio sačinjavaju tablice koje čine bazu znanja vezanu uz pojedinu proceduru. Ovo znanje uključuje mogućnosti odabira te uobičajene razloga za ove odabire. Drugi dio sačinjava tablica zapisa znanja generiranih procedurom kojoj pripada. Prvi dio sačinjavaju dakle tablice statičnog znanja, znanja koje se ne mijenja, barem ne na svakodnevnoj bazi. Tablica dinamičkog znanja prema zapise objašnjenja odluka koje su većim dijelom kombinacije zapisa iz tablica statičkog znanja uz dodatak naziva CAD objekata uz koje je vezan zapis, podataka o osobi koja je napravila zapis te eventualnih dodatnih bilježaka "hiperlinkova" prema vanjskim dokumentima.

Postavlja se pitanje razloga razdvajanja zapisa na više tablica, po jednu za svaku proceduru. Bilo bi jednostavnije pretraživati samo jednu tablicu, međutim postavlja se problem izgradnje takve tablice i njezino povezivanje sa drugim tablicama relacijama koje definiraju mogućnosti unosa u pojedina polja tablice. Pored izbjegavanja problema kompleksnosti baze, ovakvim načinom organizacije omogućava se i jedna prednost – nadogradnja i modifikacija tablice.



Slika 16: Princip zapisa i dohvata znanja iz baze znanja

Rečeno je u samom opisu zadatka kako ovaj rad neće pokriti sve procedure koje je moguće zapisati. Ustrojem baza za svaku pojedinu proceduru omogućeno je jednostavno dodavanje procedura izgradnjom nove baze, neovisne o svim ostalim bazama. Isto tako, svaku modifikaciju zasebnih baza bilo bi jednostavnije provesti. Do potrebe za modifikacijama bi, u slučaju da se sustav pusti u probni rad, sigurno došlo u onom trenutku kad se pojedina procedura koristi dovoljan broj puta da je moguće ustvrditi koje je izmjene potrebno provesti.

Prilikom opisa zajedničkog dijela izvođenja procedura zapisa navedeno je kako se za svaku odluku unose do tri razloga. Međutim, popis uobičajenih razloga za svaku je pojedinu proceduru drugačiji, tj. postoji druga tablica koja definira moguće odabire odgovora. Kako nije moguće vezati redove iz više tablica na isti redak u istoj tablici, postavlja se pitanje kako uspostaviti vezu. Jedna od mogućnosti je svakako izgraditi tablicu sa svim razlozima koji se pojavljuju u svim procedurama. U tom slučaju bilo bi potrebno odrediti dodatni mehanizam koji bi omogućavao ispis samo onih zapisa koji su relevantni za trenutno aktivnu proceduru.

Ovo međutim nije jedini problem u organizaciji gdje će broj tablica i veza i dalje izrazito rasti. Stoga je odabran model višestrukih baza koje svoje zapise daju središnjoj bazi prilikom pretraživanja kako bi se korisniku mogli na jednom mjestu svi zapisi koji imaju zajedničku osobinu koja nije karakteristična za samo jednu proceduru. Primjer je ispisivanje svih zapisa koji su unijeti za isti projekt, strojni dio, uneseni su u nekom vremenskom razdoblju ili ih je zapisala ista osoba.

8. PRIJEDLOG KLASIFIKACIJE PROCEDURA

Tokom razmatranja predloženih primjera procesa odabira koji su dovoljno česti i u praksi uvriježeni da bi im se mogao odrediti algoritam izvođenja, uočene su određene sličnosti između procedura na temelju kojih je izvršeno grupiranje procedura u četiri skupine. Primjenom grupiranja procedura u kategorije olakšava se izrada osnovnog sučelja i odabir procedure koja se želi provesti. U protivnom riskira se gubitak interesa korisnika za korištenjem aplikacije u samom početku kad ne mogu naći proceduru koja im je potrebna.

Četiri predložene kategorije procedura su slijedeće:

- Odabir strojnih elemenata
- Definiranje Funkcionalno - tehnoloških elemenata
- Odabir materijala
- Definiranje načina gradnje CAD modela

Detaljniji opis svake kategorije biti će dan prije popisa i objašnjenja svake procedure.

8.1. ODABIR STROJNIH ELEMENATA

Kategorija elemenata konstrukcija obuhvaća standardne strojne elemente za koje je standardom definirana funkcija, dimenzije i način proračuna, a za elemente koji postoje kao zasebni dijelovi ili podsklopovi (vijci, valjni ležajevi) postoje specijalizirani proizvođači koji svojom pouzdanošću, cijenom i brzinom isporuke uklanjaju svaku potrebu za vlastitom proizvodnjom. Cilj svake procedure koja spada u ovu kategoriju je pružiti korisniku na jednom mjestu cjelokupnu lepezu varijacija pojedinog elementa uz osnovne prednosti i nedostatke koji će olakšati odabir, a time postati i najvjerojatniji kandidati za razlog za ili protiv određenog odabira.

Proračun, tj. odabir pojedinog elementa po svim stavkama izlazi van okvira standardnih procedura za što postoje dva razloga. Prvi je što je na ovoj razini funkcija procedura omogućavanje zapisivanja razloga za određeni odabir, što ne uključuje detaljan proračun. U budućnosti, pokaže li se sustav ovih procedura primjenjivim i korisnim, nema razloga da se njihov djelokrug ne proširi i na sam proračun, tj. detaljan odabir i

dimenzioniranje. Drugi razlog je što se dio proračuna elemenata provodi očitavanjem vrijednosti iz dijagrama koji su dobiveni na iskustvenoj bazi, tj. nisu nastale kao grafovi poznatih matematičkih funkcija. Naravno, moguće je izračunati aproksimativnu funkciju, međutim, radi se o zadaći koja potrebnim vještinama i opsegom daleko nadmašuje okvire ovog projekta.

U nastavku će biti opisane procedure odabira elemenata, naravno planirane u skladu sa alatima kojima se trebaju ostvariti.

8.1.1. PROCEDURA ODABIRA LEŽAJA

Kako je ležaj standardni strojni element vezan uz osovinu ili vratilo, točnije uz jedan njegov stupanj, ukoliko se radi o stupnjevanom vratilu, objekt uz koji se veže zapis može biti ili cijeli dio, tj. cijelo vratilo, ili jedan njegov stupanj, naravno pod uvjetom da je taj stupanj zasebna značajka unutar CAD modela. Naravno, u slučaju da je korisnik prvo ugradio ležaj, a potom pokrenuo proceduru zapisa, moguće je zapis vezati uz sam ležaj. Ovisno o tipu CAD objekta koji predstavlja valjni ležaj, korisnik bi zapis vezao uz pojedini dio (kao što je STEP ili IGES datoteka) ili cijeli sklop modela valjnog ležaja. Ovisno o standardizaciji unutar tvrtke, moguća je neka od ovih opcija, međutim, u svakom slučaju radi se o jednom objektu, te će stoga korisnik sigurno za odabir objekta koristiti formu za odabir jednog objekta.

Nakon odabira objekta, potrebno je navesti određene specifikacije koje se traže od ležaja. Proizvođači valjnih ležaja (FAG, 2004. [14]) naglašavaju kao prednosti pojedinih tipova ležaja slijedeće karakteristike:

- Prenosnje radijalne sile
- Prenosnje Aksijalne sila
- Broj okretaja između dva servisa
- Mogućnost promjene smjera okretanja)
- Pogodnost za udarna opterećenja (da, ne)
- Potrebna veličina ugradbenog prostora (mala, velika)
- Radna temperatura
- Osjetljivost na utjecaj okoliša (nečistoća, agresivni mediji, vibracije)
- Mogući poredak osovinu (horizontalno, vertikalno, nagnute)
- Krutost ležaja (zakretanje)

-
- Točnost (odstupanje od osi)
 - Vijek trajanja
 - Buka
 - Iskoristivost (gubici trenja, zagrijavanje)
 - Podmazivanje i servis (složeno ili jednostavno, da li je potreban sustav dobave maziva itd.)
 - Montaža i demontaža (jednostavno ili složeno)
 - Komercijalni podaci (dobavlјivost, broj komada, skladištenje)

Tvrtka „KLEX“ d.o.o. dala je slijedeći popis razloga koji se učestalo javljaju kao kriteriji za odabir pojedinog tipa ležaja, a koje proizvođači ne navode jer nisu dio opće važećih kriterija:

- Smjer opterećenja (radijalno, aksijalno, kombinacija)
- Smještaj ležaja u jednodijelno ili dvodijelno kućište
- Optimiranje težine, cijene, servisa, vijeka trajanja
- Vrsta rotacije – potpuna rotacija ili samo za određeni kut
- Tolerancijska analiza – statičko toleriranje
- U slučaju uprešavanja u tvrde materijale i/ili sa velikim preklapom specijalni ležajevi (C3, EC..)
- Uprešavanje ležaja u kućišta sa ili bez temperaturnih razlika
- Povratna informacija sa montažne linije o mogućnosti/jednostavnosti uprešavanja
- Osiguravanje ležaja protiv aksijalnih pomaka – uprešavanje na oslonac ili uprešavanje na dimenziju – problem velike tolerancije širine ležaja
- Nužnost mehaničke obrade nakon lijevanja
- Moguće je postići toleranciju dosjeda čak i bez mehaničke obrade ako se radi dvodijelno kućište (vertikalna podjela) u tom slučaju se mora kontrolirati alat a ne odljevak tj. mjeri se alat; iz te mjere se gleda kakav je dosjed ležaja (mjerenjem okretnog momenta)
- Broj i raspored vijaka oko ležajnih mjesta u slučaju dvodijelnih kućišta
- Da li je potrebno specijalno brtvljenje (prašina, voda i slično)
- Kod viših temperatura koristiti „Viton“ brtve
- Smjer uprešavanja kod igličastih ležaja s vlastitim kućištem

Ove će se dvije liste razloga objediniti i ponuditi inženjeru na sučelju kao moguća objašnjenja. Korisniku se prikazuje prozor sa Slike 17. Na ovom prozoru korisnik unosi zapis o tipu ležaja koji je odabran te karakteristike zbog kojih je odabran.

Slika 17: Sučelje odabira ležaja

Ležajevi se uobičajeno dijele na radijalne i aksijalne prema smjeru sile opterećenja za čije su prenošenje primarno namijenjeni, te potom prema tipu valjnog tijela koja je ugrađeno. Odabirom sa popisa koji se otvaraju klikom miša na odgovarajuće padajuće popise, korisnik odabire do tri razloga zbog kojih je ležaj odabran. Nakon odabira do najviše tri predefinirana razloga, korisnik se prebacuje na prozor Access forme gdje će svi njegovi odabiri biti preneseni.

8.2. OGRANIČAVANJE AKSIJALNOG POMAKA

Aksijalni pomak po osovini ili vratilu objekata poput zupčanika ili kotača je nužan tokom sklapanja i rasklapanja, ali je tokom rada potrebno onemogućiti ga. Postoje brojni načini kako onemogućiti aksijalni pomak u koje spadaju zavarivanje, stezni spoj, korištenje vijaka, uskočnika, oblikovanjem vratila itd. Sve opcije nisu uvijek primjenjive, već za svaku postoje situacije u kojima se pokazuju optimalnima.

Pojedina tehnička rješenja razlikuju se prema svojim karakteristikama pri čemu nijedno tehničko rješenje nema po svim karakteristikama prednost pred ostalim rješenjima. Stoga se odabir rješenja svodi na odabir rješenja sa optimalnim karakteristikama za slučaj na koji se kani primijeniti. Korisnik dakle prvo mora odrediti koje karakteristike rješenje treba imati, te potom bira rješenje. Karakteristike problema prema kojima se bira rješenje su:

- Stupanj aksijalne sile
- Stupanj prenošenja zakretnog momenta
- Mogućnost promjene smjera djelovanja
- Koliko često je potrebno rastavljati sklop
- Veličina udarnih opterećenja
- Kolika je potrebna točnost položaja
- Jednostavnost montaže
- Upotrebljivost nakon demontaže
- Cijena

Procedura “Ograničavanje aksijalnog pomaka” namijenjena je upravo vođenju inženjera kroz postupak upisa odabira pravilnog pristupa ovisno o karakteristikama sklopa.

Korisnik stoga na prozoru odabire karakteristike koje vrijede u njegovom slučaju. Aplikacija će potom na temelju baze podataka ispisati načine koji se preporučuju za tu situaciju. U polju pod nazivom „Opis“ za svaku odabranu karakteristiku bit će dan kratak opis i primjeri.

Slika 18: Sučelje obrazloženja odabira ležaja

Jednom kad je korisnik odabrao način osiguranja i razloge za svoj odabir, prebacuje ga se na formu Accessa.

8.2.1. ODABIR OPRUGE

Iako klasična mehanika promatra tijela kao krute objekte koji se ne deformiraju neovisno o iznosu opterećenja, pa čak i proračuni strojnih dijelova koji su bazirani na teoriju Nauke o čvrstoći u pravilu ne uzimaju u obzir deformacije dijelova, u stvarnom svijetu svaki se mehanički objekt pod utjecajem sila deformira.

Dok je kod većine strojnih dijelova ova deformacija nepoželjna, kod opružnih elemenata upravo je deformacija pod opterećenjem glavno svojstvo koje omogućava izvršavanje funkcija opruga kao što su ublažavanje udara i smanjenje utjecaja vibracija.

S obzirom veliki broj tehničkih rješenja opružnih elemenata, razvijen je i veći broj sustava klasifikacija koji uključuju podjelu po naprezanju, po oblikovanju i materijalu, po opterećenju i po izgledu (obliku).

Sa stajališta konstruktora opruga, svaka od ovih kategorija je izrazito važna, međutim, sa stajališta konstruktora koji bira oprugu kako bi riješio problem vršenja određene funkcije razvijanog sustava, važna su samo posljednje dvije kategorije; način opterećenja i izgled(oblik).

Način opterećenja i oblik opruge važni su zbog toga što korisnik zna kakvo je djelovanje i iznos opterećenja na oprugu, koliko im prostora te kakvi su okolišni uvjeti te mu je stoga najvažnije ponašanje opruge u takvom okruženju.

Kombinacijom navedenih dviju kategorija dobiva se slijedeća kategorizacija prema (Cvirić, 2000., [24]).

- Opruge za tlačno opterećenje
 - Cilindrična zavojna opruga okruglog presjeka žice
 - Cilindrična zavojna opruga pravokutnog presjeka žice
 - Tanjuraste opruge
 - Prstenaste opruge
- Opruge za vlačno opterećenje
 - Cilindrična zavojna opruga okruglog presjeka žice
- Opruge za torzijsko opterećenje
 - Šipka kružnog presjeka
 - Cilindrična zavojna opruga
 - Spiralna opruga

-
- Opruge za savojno opterećenje
 - Jednolisna opruga
 - Složena lisnata opruga

Pored navedene klasifikacije, jedno od najvažnijih svojstava opruga je njihova karakteristika. Karakteristika opruge iskazuje funkciju veličine deformacije opruge u ovisnosti o povećanju opterećenja. Opruge prema karakteristici mogu biti progresivne, proporcionalne i regresivne ovisno o tome kako se mijenja odnos deformacija/opterećenje sa povećanjem opterećenja. Naravno, za odabir opruge je bitan i vršni rezultat, tj. kolika je najveći omjer deformacije i opterećenja te iznos najvećeg opterećenja

Neke od ostalih svojstava koje utječu na primjenjivost u specifičnom slučaju uključuju:

- Dimenzije
- Sposobnost prigušivanja vibracija
- Jednostavnost ugradnje
- Demontaža i servis
- Cijena
- Radna temperatura
- Utjecaj okoliša

U pogledu izvođenja same aplikacije, postavlja se pitanje objekta uz koji će biti vezan zapis. Čini se kako postoje više logičnih mogućnosti. Prva je da se zapis veže uz sklop u kojem će opruga biti ugrađena. Druga opcija je vezanje zapisa uz objekte s kojima opruga dolazi u kontakt, tj. objekt kojeg treba amortizirati i objekt koji služi kao oslonac. Treća mogućnost je vezanje samo uz objekt koji treba amortizirati.

Odluku o tome uz koji će se objekt vezati zapis treba, kao i prijašnjim slučajevima, donijeti na razini tvrtke. Ovisno o odluci, korisnik će odabir elementa vršiti u prozoru odabira jednog ili dva elementa.

Slijedeći je korak zapis odabira tipa opruge. Prema tipu opterećenja koje se javlja za njegov slučaj, korisniku se nudi popis odgovarajućih opruga. Daljnji upis izbora vrši se prema karakteristici opruge te ostalim svojstvima.

8.2.2. ODABIR SVORNJAKA

Svornjak je strojni element koji se ugrađuje kao veza između dva strojna dijela kod kojih postoji oscilatorni kutni pomak. Ponekad se za svornjak koristi i naziv osovinica što ima opravdanja jer je svornjak zapravo mala osovinica. Temeljna razlika u odnosu na osovinu je nepostojanje ležaja zbog čega se svornjak koristi za omogućavanje manjeg kutnog pomaka kao što je okretanje ručice, otvaranje vrata i sl. Ako se kod kutnog pomaka strojnih dijelova očekuje veliki broj okretaja, potrebno je, zbog velikih gubitaka, koristiti osovinu sa ležajima. Drugi način gledanja na ovu situaciju je da se kod malog kutnog pomaka u jedinici vremena financijski ne isplati ugrađivati ležajeve. Primjer je motorna kosilica (hobi modeli) čiji se kotači okreću oko svornjaka, a ne na osovini, tj. kod kotača na kosilicama ne postoje ležajevi.

Pretpostavka kod ove procedure je da korisnik zna koje su prednosti svornjaka pred osovinom te da se svjesno odlučio za upotrebu svornjaka. Ovakva odluka u pravilu nije teška jer se u većini slučajeva iz industrijske prakse te jednostavne logike zna da li je bolja osovinica ili svornjak. U slučajevima kad odluka nije tako jednostavna, ona se mora donositi na razini voditelja projekta (u slučaju kosilice – hobi ili profesionalna) te stoga izlazi van okvira namjene ove aplikacije. Naravno, iako je odluka o upotrebi svornjaka u takvim slučajevima potekla iz procesa odlučivanja koji nadilazi namjenu ove aplikacije, korisnika ništa ne sprečava da razloge za ovu odluku upiše u bazu.

Literatura ne daje podjelu svornjaka po kategorijama zbog relativno malog broja tipova. Ipak, općenito se može izvršiti podjela po dvije kategorije koje razlikuju svornjake prema načinu na koji je svornjak osiguran od ispadanja.

Na jednoj strani osiguranje od ispadanja moguće je izvršiti oblikom, tj. na jednoj strani svornjak može imati glavu (šesterokutnu, okruglu, okruglu sa dvije usječene bočne plohe). Prednost glave je potreban manji broj dijelova, negativnost je definiranost smjera ugradnje. Dakle, prema prvoj kategoriji svornjak može biti sa ili bez glave.

Druga kategorija dijeli svornjake prema tipu elementa koji je upotrijebljen za osiguranje od ispadanja. Naravno, kod svornjaka sa glavom ovaj element se ugrađuje na suprotnoj strani od glave, a kod svornjaka bez glave s obje strane. Elementi za osiguranje uključuju podložnu pločicu u kombinaciji sa rascjepkom, vanjskim uskočnikom, prstenom od elastične žice ili maticom (potreban navoj na svornjaku) te sigurnosnu pločicu. Prednost vijka je što omogućava pritezanje dijelova koji se spajaju svornjakom te može izdržati veliku aksijalnu silu. Negativnost je veća cijena zbog potrebe narezivanja navoja. Svi ostali elementi

su jeftiniji, ali ne mogu podnijeti veliku aksijalnu silu, tj. koriste se samo u slučajevima kad je potrebno samo osigurati da svornjak ne ispadne.

Iako su u načelu moguće sve kombinacije, svornjaci bez glave izrađuju se sa istim elementom za osiguranje sa obje strane. Također, ukoliko je odabrana šesterokutna glava, element za osiguranje od ispadanja s druge strane je matica sa podložnom pločicom.

Iz razmotrenog izvlači se zaključak kako bi korisnik za odabir svornjaka trebao navesti da li svornjak ima glavu ili ne, te koji je element za osiguranje od ispadanja primijenjen.

Kako je svornjake svih postojećih debljina i duljina moguće izvesti sa glavom ili bez, te sa upotrebom bilo kojeg tipa elementa za osiguranje, jedine karakteristike o kojima može ovisiti odabir su one koje proizlaze iz (ne)postojanja glave i tipu elementa za osiguranje.

Karakteristike su prema razmotrenom slijedeće:

- Potreba za izvlačenjem svornjaka s obje strane
- Veličina aksijalne sile

Kod izvođenja same aplikacije, biti će potrebno prvo odabrati objekte uz koje se veže zapis. S obzirom na činjenicu da svornjak služi kao veza između dva strojna dijela, pred korisnika će biti postavljen prozor za odabir dva objekta.

Nakon odabira objekata, korisnik upisuje odabir tipa svornjaka te razloga za odabir na prozoru prikazanom na Slici 20.

Slika 20: Sučelje obrazloženja odabira opruge

Nakon odabira svornjaka, korisnik upisuje svoje ime i prelazi na Access formu.

8.2.3. VEZE S GLAVINAMA

Veze s glavinama važan su dio svakog uređaja kod kojeg se snaga prenosi na vratilo i/ili sa vratila na neki drugi element kao što su zupčanici, remenice, kotači ili neka vrsta alata poput glave glodala i sl. jer ove veze omogućuju prenošenje pogonskog momenta i brzine vrtnje.

Literatura raspoznaje velik broj načina prenošenja momenta i brzine vrtnje koji se nalaze u upotrebi, svaki sa svojim prednostima i manama. Proračuni nekih od ovih elemenata uključuju očitavanje podataka iz dijagrama što značajno otežava izradu aplikacije koja bi dala potpuno rješenje. Dapače, na ovoj razini razvoja aplikacija postavlja se pitanje da li je uopće moguće izvesti tako nešto, posebice ako se uzme u obzir da se radi o načelnim rješenjima kod kojih treba računati sa posebnostima specifične primjene kao i tehnološkog ograničenja pojedinih proizvodnih pogona.

Procedura obrazloženja odabira veze s glavinama treba omogućiti korisniku da upiše razloge zbog kojih je odabrao pojedino rješenje. Zapisi koji budu uneseni omogućiti će ubuduće lakše donošenje odluke inženjerima koji ne posjeduju značajno iskustvo u projektiranju.

Također, kategorizacija uključuje stavke koje su na drugim mjestima navedene kao zasebne procedure (pero, vijak). Na ovom mjestu stoga neće biti proveden detaljni odabir tih elementa, već samo odabir generalnog principa uz koji će korisnik upisati objašnjene odabira. Moguće je, nakon završavanja zapisa odabira veze s glavinom, prebaciti korisnika na postojeću proceduru odabira primjerice vijka, ako je korisnik odlučio spojiti vratilo sa glavinom primjenom steznog vijka.

Spojevi s glavinama podijeljeni su tri kategorije:

Prijenos momenta oblikom

- o Perom
- o Oblikom vratila
- o Zatikom
- o Vijkom
- o Ozubljenjem
- o Uskočnikom

Prijenos momenta trenjem

- Neposredni spojevi
 - Stezni spoj
 - Konusni spoj
 - Spoj steznom glavčinom
- Posredni spojevi
 - Prstenaste stezne opruge
 - Četverostruko konusni stezni element
 - Jednostruko konusni stezni element
 - Dvostruko konusni stezni element
 - Stezne prirubnice
 - Pritezne čahure
 - SPIETH tlačna čahura
 - Hidraulički (ETP) stezni elementi
 - Zvezdasto prstenasta ploča
 - Veza tolerancijskim prstenima
 - Zaobljeni klin

Utvrđene su slijedeće karakteristike (Ivandić, 1996., [6]) prema kojima se bira način spoja glavine s vratilom:

- Prenosivi moment vrtnje
- Prijenos momenta ovisan
- Preuzimanje aksijalne sile
- Posljedica preopterećenja
- Mogućnost usredištenja
- Aksijalna pomičnost
- Postavljivost glavine
- Sastavljanje i rastavljanje
- Upotrebljivost nakon rastavljanja
- Trošak izrade i sastavljanja

Kao i kod svih ostalih procedura, korisnik na početku bira objekt uz koji će vezati zapis. Kako se u ovom slučaju radi o spoju dva elementa (glavine s vratilom), korisniku se

prikazuje prozor za odabir dva elementa. Odabir će naravno morati biti napravljen u dokumentu sklopa.

Nakon odabira elemenata korisnik se prebacuje na prozor odabira tipa spoja s glavinom (Slika 21).

Slika 21: Sučelje obrazloženja odabira veze s glavinama

Nakon izvršenog odabira korisnik se prebacuje na prozor za upis imena korisnika te potom na Access formu gdje unosi razloge za svoj odabir, unosi alternativu i veže vanjske dokumente.

8.2.4. BRTVLJENJE

Brtvljenje je postupak onemogućavanja istjecanja fluida kroz prostor između dva strojna elementa. Ovisno o međusobnom relativnom gibanju ova dva strojna elementa te temeljnom principu brtvljenja, postupci brtvljenja podijeljeni su u tri kategorije prema (Herold, 1997., [25]).

- Dodirno brtvljenje dijelova koji miruju
 - Nerastavljivo brtvljenje
 - Brtvljenje masama za brtvljenje
 - Rastavljivo brtvljenje
- Dodirno brtvljenje dijelova koji se pokreću
 - Glavno brtvljenje dijelova koji se pokreću
 - Glavno brtvljenje na radijalnoj površini
- Bezdodirno brtvljenje
 - Brtvljenje prigušivanjem
 - Brtvljenje pomoću tekućine
 - Membranske brtve

Alternativna klasifikacija je prema (Decker, 2006., [12])

- Brtve protiv izlaza masti
 - Brtve s brusnim djelovanjem
 - Bezdodirno brtvljenje
- Brtve protiv istjecanja ulja
 - Brtvljenje brusnim djelovanjem
 - Bezdodirno brtvljenje

Ukoliko je odabrana prva klasifikacija, sučelje za odabir dano je na Slici 22. Podloga prozora je slika preuzeta iz literature (Herold, 1997., [25]).

Odaberi način brtvljenja

BRTVE I BRTVENJE

Dodirno brtvenje dijelova koji miruju

Nerastavljivo brtvenje ...

Brtvenje zavarivanjem ...

Brtvenje prešanjem ...

Brtvenje masama za brtvenje ...

Rastavljivo brtvenje ...

Brtvenje pretežno vanjskim silama ...

Plosnate brtve ...

Profilne brtve ...

Brtvenje brtvenicama ...

Brtvenje pogonskim tlakom (automatsko) ...

Dodirno brtvenje dijelova koji se pokreću

Glavno brtvenje na cilindričnoj površini ...

Stlačiva brtvila ...

Hidraulič brtve ...

Brtvenje manšetama ...

Profilne brtve ...

Brtvila postojećeg oblika ...

Opružni prsteni ...

Višedjelni prsteni ...

Glavno brtvenje na radijalnoj površini ...

Rad - suho ili mješovito trenje ...

Rad - tekuće trenje ...

Bezdodirno brtvenje

Brtvenje prigušivanjem ...

Bezdodirne brtve sa zračnošću ...

Radijalna zračnost ...

Aksijalna zračnost ...

Labirintne brtve ...

Radijalni labirinti ...

Aksijalni labirinti ...

Brtvenje pomoću tekućine ...

Potreban tlak stvoren u brtvi ...

Navojna brtva ...

Centrifugalna brtva ...

Potreban tlak stvoren izvan brtve ...

Membranske brtve ...

Karakteristike

Tlak fluida

Složenost proizvodne tehnologije

Učestalost servisa

Upotrebljivost nakon demontaže

Cijena

Odabrano

Odabrana brtva

1. razlog

2. razlog

3. razlog

Natrag

Odustani

Dalje

Slika 22: Sučelje obrazloženja odabira brtvljenja

Druga klasifikacija može se promatrati kao lista karakteristika, pa bi korisnik birao da li mora zabrtviti izlazak masti ili ulja, te da li želi brtvljenje brusnim djelovanjem ili bezdodirno brtvljenje koje se razlikuju po tome što je brtvljenje brusnim djelovanjem jednostavnije za izvođenje i jeftinije dok je održavanje kod bezdodirnog brtvljenja rjeđe.

Pored odabira jedne od klasifikacija, postavlja se mogućnost i njihovih spajanja u jednu klasifikaciju.

60

8.2.5. ODABIR PRIČVRSNOG VIJKA

Pričvrsni vijci jedan su od najčešće korištenih standardnih strojnih elemenata, a služe za spajane dijelova koji se u nekom trenutku u budućnosti trebaju moći rastaviti. Ovisno o tome da li barem jedan od dva (ili čak i tri) dijela želimo spojiti ima prikladan navoj, bit će ili neće biti potrebna matica.

Postoje još dvije značajke koje je bitno razmotriti prije odabira vijka. Prva se odnosi na smjer sile o odnosu na smjer pružanja osi vijak. Ako sila djeluje paralelno sa osi, to je idealni slučaj jer nema opasnosti od kontakta navoja sa rupom.

Ako pak sila djeluje paralelno sa kontaktnim ravninama dijelova koji se spajaju, tj. Okomito na smjer pružanja osi vijka, biti će potrebno odrediti kako će vijak prenositi silu. Jedna opcija je da ju prenosi oblikom opri čemu vijak dolazi u kontakt sa stjenkama rupa u dijelovima koji se spajaju. U tom slučaju biti će potrebno koristiti posebne vijke ili tuljce. Druga opcija je da vijci toliko stegnu dijelove da silu prenese trenje koje se javlja na kontaktnim površinama. U svakom slučaju, potrebno je provesti proračun naprezanja u tijelu vijka.

Postavlja se pitanje razloga odabira vijka na određenom mjestu. Naravno, određeni tipovi vijaka namijenjeni su za određene funkcije za što su i posebno oblikovani. Iz tipa vijka koji je odabran je stoga jasno čemu vijak služi. Postavlja se međutim pitanje zašto je odabran upravo vijak koji je odabran iz određene skupine. Ovo je posebice važno za matične vijke koji su najčešći u uporabi, a kako postoji veliki broj varijacija i veličina postoji u većini slučajeva njihove primjene više od jednog mogućeg izbora. Neki od razloga za odabir određenog vijka mogu biti:

- Konstrukcijski – potreban je vijak određene duljine kako bi mogao učvrstiti dijelove veće ukupne debljine
- Funkcionalni – odabran je vijak koji neće sprečavati izvršavanje određenih funkcija proizvoda ili predstavljati sigurnosni problem kao što su imbus vijci koji mogu cijelom glavom ući u osnovni materijal čime se smanjuje opasnost od zahvaćanja stranih tijela
- Montažni – vijak je odabran jer olakšava montažu (vijak sa utorom sa odvijač koji se nalazi na mjestu gdje je otežan pristup ključem)
- Ekonomski – odabran je isti vijak kao i na drugom mjestu u sklopu kako bi se smanjili troškovi nabavljanja dvije vrste vijaka

Korisnik će odabrani vijak vezati uz dva objekta, dva dijela koji se vijkom spajaju. Prema tome, korisniku treba dati prozor za odabir dva objekta. Naravno, postoji i druga opcija, a ta je da se zapis veže uz sam vijak što znači da korisniku treba dati prozor za odabir jednog objekta.

Nakon odabira objekta (ili objekata) korisnik unosi podatke o odabranom vijku te razloge za odabir.

Na prvom prozoru korisnik će odabrati tip vijka koji želi upotrijebiti. Uz matični, elastični i rastojni tip vijka moguće je korištenje matica. Ako korisnik uz odabir nekog od ta tri tipa vijka želi koristiti maticu, mora to naznačiti klikom na odgovarajuću kućicu.

Odabir vijka

Odaberi tip vijka

Matica

☐ ☒ Matični vijak

☐ Zatični vijak

☐ Uvrtni i zatični vijak

☐ Očni vijak

☐ ☐ Elastični vijak

☐ Zatezni vijak i matica

☐ Temeljni vijak

☐ Preklopni vijak

☐ ☐ Rastojni vijak

☐ Brtveni vijak

Karakteristike

Smjer sile u odnosu na os vijka

Prijenos opterećenja

Pritezanje

Odabrano

Odabrani ležaj

1. razlog

2. razlog

3. razlog

Natrag

Odustani

Dalje

Slika 23: Sučelje obrazloženja odabira pričvrsnog vijka

Nakon odabira vijka (i matice), korisnik se prebacuje na Access formu.

U bilješci će aplikacija automatski unijeti zapis da li je korištena matica ili ne. Korisnik može dodati još bilješki ako želi.

8.2.6. DODATNI ELEMENTI KONSTRUKCIJA

Pored opisanih procedura za obrazloženja odabira standardnih strojnih dijelova, moguće je na sličan način formirati procedure i za ostale strojne elemente (Decker, 2006. [12]): čahure, čepove, klinove, lance, lančanike, osovine, pera, podloške osiguranja, pokretne vijke (vijčane pogone), profile, puževe, pužna kola, remenje, remenice, ručice, rukavce, spojke, tarenice, ventile, vratila, zaklopke, zakovice, zasune, zatike i zupčanike.

8.3. DEFINIRANJE FUNKCIONALNO - TEHNOLOŠKIH ELEMENATA

U grupu funkcionalno-tehnoloških elemenata spadaju pojedine značajke strojnih dijelova čija je geometrija određena njihovom funkcijom ili tehnologijom proizvodnje. Iz same geometrije često nije neposredno jasno zbog čega je ona odabrana u ovom ili onom obliku tj., zašto je određena dimenzija postavljena na određeni iznos, te da li je moguće neko drugo rješenje.

Stoga je cilj ovih procedura omogućiti korisniku da zapiše razloge za svoje odabire pojedinih značajki dijela, posebice po pitanju mogućih alternativa. Dok su kod elemenata strojeva u zapisu postojala dva objekta; sam strojni element o kojem je zapis govorio te CAD objekt za kojeg se vezao zapis, a s kojim je strojni element u kontaktu, kod ove grupe procedura, objekt za koji se veže zapis ujedno je i objekt o kojem sam zapis i govori.

8.3.1. ODABIR DOSJEDA

Kad se odabire dosjed, korisnik bira procedure “Odabir dosjeda” pri pokretanju aplikacije. Kako je dosjed zapravo odnos dimenzija dvaju strojna dijela, glavine i vratila, zapis je potrebno vezati uz dva objekta, te će stoga korisniku biti prikazan prozor za odabir dvaju objekata.

Nakon odabira objekata otvara se prozor za odabir dosjeda. Ovisno o broju dosjeda koji se koriste unutar tvrtke, moguće je postaviti različiti broj kategorija. Na ovom mjestu koristit će se osnovna podjela dosjeda na labave, prijelazne i čvrste. Uz osnovni tip dosjeda korisnik može odabrati želi li koristiti sustav jedinstvenog provrta (provrta na nazivnoj mjeri ili veći za iznos tolerancijskog polja) ili sustav jedinstvene osovine (osovina na nazivnoj mjeri ili manja za iznos tolerancijskog polja). Općenito se preporuča sustav jedinstvenog provrta što bi trebalo biti naznačeno.

Nakon odabira tipa dosjeda, bira se podvrsta dosjeda. Literatura (Cvirm, 2000., [24]) navodi slijedeće podvrste dosjeda

LABAVI

- Klizni dosjed, uz mazanje pomičan rukom
- Polupovodljivi dosjed, mala zračnost
- Povodljivi dosjed, osjetna zračnost
- Polupomičan dosjed, znatna zračnost
- Pomičan dosjed, veća zračnost
- Poluprostran i prostran dosjed, velika zračnost
- Vrlo prostran dosjed, jako velika zračnost

PRIJELAZNI

- Pokretni dosjed, rastavljiv drvenim čekićem
- Prilegli dosjed, rastavljiv drvenim čekićem
- Stegnuti dosjed, teško rastavljiv ručnim čekićem
- Jako stegnuti dosjed, rastavljiv tlakom

ČVRSTI

- Zažeti i uglavljeni dosjed, srednji preklopi
- Čvrsto zažeti i ukrućeni dosjed, veliki preklopi

Pored funkcije dosjeda, literatura ne daje dodatne razloge za odabir pojedinog tipa dosjeda. Tvrtka „KLEX“ d.o.o. dala je slijedeći popis razloga za odabir pojedinog dosjeda:

- Vrsta spoja – labavi, prijelazni, čvrsti
- Vrste materijala od kojih se spoj sastoji (tvrdi materijali – čelik, meki materijali – Mg i Al legure)
- Temperature uprešavanja (zagrijavanje, pothlađivanje, bez temperaturnih razlika)
- Opterećenje spoja
- Tolerancijska i statistička analiza spoja
- Utjecaj vibracija na vezu
- Način uprešavanja (naprave)
- Što u slučaju havarije (servis) – specijalni alati
- Ekologija (reciklaža) – da li je potrebno odvojiti različite materijale?

Slika 24: Sučelje obrazloženja odabira dosjeda

Dosjedi se u pravilu biraju kako bi izvršili određenu funkciju. Iz samog naziva moguće je očitati tu funkciju čime daljnje oznake po tom pitanju postaju nepotrebne. Na ovom mjestu, kako se potencijalno u uporabi u nekoj tvrtci može naći veliki broj dosjeda, kako bi se olakšao zapis odabira, korisnik može odabirom određenih filtara smanjiti popis mogućih odabira.

Nakon upisa svih podatka, korisnik se prebacuje na Access formu.

8.3.2. DIMENZIONIRANJE I POZICIONIRANJE REBARA

Rebra se koriste za povećanje krutosti i nosivosti konstrukcije uz minimalno povećanje mase. Za kreiranje rebara CAD aplikacije imaju integrirane zasebne naredbe koje kreiraju rebro u jednom koraku (u CATIA-i – Stiffener, u drugima - Rib). Međutim, kako je rebra moguće oblikovati i korištenjem drugih naredbi, a ponekad korištenje naredbe za oblikovanje rebara nije ni moguće (ili efektivno), proceduru dimenzioniranja i pozicioniranja rebara nije preporučljivo vezati isključivo uz korištenje ovog tipa naredbi. Ipak, samo rebro je jedna značajka u dijelu pa prema tome radi se o jednom objektu, a ukoliko se žali stvoriti zapis o svim rebrima u jednom dijelu, tada je moguće zapis vezati uz cijeli dio. U svakom slučaju, korisnik bira jedan element, pa mu se stoga nakon odabira procedure “Dimenzioniranje i pozicioniranja rebara” daje prozor za odabir jednog objekta iz CAD modela. Ovaj objekt bi trebao biti značajka u modelu dijela.

Tvrtka „KLEX“ dala je slijedeći popis uobičajenih razloga za pozicije i dimenzije rebara:

- Konstrukcijski uvjeti
- Opterećenje komada
- Gomilanje materijala
- Mogućnost izrade – što prije razgovarati s alatničarom
- Odnos visine i širine rebara (gomilanje materijala radi kuta izvlačenja)
- Smjer ulijevanja materijala
- Debljina stjenke
- Vibracije
- Optimizacija težine
- Manji broj rebara zbog moguće modifikacije alata
- Moguće koristiti NC glodanje
- Pozicija izbacivača
- Veći stupanj slobode zbog nemogućnosti nošenja opterećenja
- Potrebe dizajna
- Povećanje torzijskog momenta uz minimalnu težinu
- Rebra koja završavaju u nuli tj., nemaju daljnju vezu s okolinom su beskorisna
- Tok sila i opterećenja
- Dinamička opterećenja i vlastite frekvencije

-
- Iskustvena trzajna čvrstoća veća od rezultata FEM analiza

Nakon odabira objekta korisniku se pokazuje prozor u Accessu u kojem će korisnik, kao i kod ranijih procedura, unijeti dodatne podatke.

Slika 25: Sučelje obrazloženja oblikovanja rebara

Nakon odabira razloga, korisnik se prebacuje na Access formu.

8.3.3. SKOŠENA POVRŠINA

Skošene površine postavljaju se iz nekoliko mogućih razloga. Jedan je olakšano vađenje iz kalupa kod lijevanih dijelova. Druga je olakšano sklapanje rotacijskih dijelova (osovina, svornjaka). Treća je omogućavanje funkcije klina.

Kao i kod rebara, većina CAD aplikacija ima naredbe za jednostavno izradu iskošenja. Postoji međutim veliki broj skošenih površina koje se ne oblikuju na taj način. Također, u vezi sa prethodno opisanim procesom, moguće je vezati isti zapis uz istu skošenu površinu korištenjem različitih procedura. Ako se radi o skošenju koje olakšava sklapanje, moguće je zapis izvesti procedurom “Sklapanja proizvoda”, kao i procedurom “Skošena površina”. Međutim, kako je sam zapis isti jer se i u jednom i u drugom slučaju uz skošenje veže njena funkcija, koja je procedura omogućila zapis je manje bitno.

Korisnik će morati odabrati značajku samog skošenja kao objekt uz koji se veže zapis, pa je logično da odabere objekt iz prozora za odabir jednog objekta. Slijedi podizanje Accessa i unos ostalih podataka.

Glavni podatak vezan uz skošenje je nagib skošenja. Uz sam nagib potrebno je odabrati razlog za odabir upravo te vrijednosti. Pretpostavka je da se unutar tvrtke koristi standardni set nagiba i razloga zbog kojih se koriste.

8.3.4. HRAPAVOST POVRŠINE

Hrapavost određene površine u ovisnosti je o dva parametra površine. Prvi parametar je funkcija površine koja određuje maksimalnu dozvoljenu hrapavost. Drugi parametar je primijenjena tehnologija izrade koja definira ostvarivu minimalnu hrapavost. Očito će preko hrapavosti funkcija površine definirati tehnologiju obrade površine.

Konstruktivski inženjeri upoznati su sa funkcijama površina dijelova na kojima rade jer su oni sami te funkcije zadali. Međutim, kako konstruktorima nedostaje iskustva u proizvodnim tehnologijama kao i podataka o svojstvima površina ovisno o njihovom stanju, često se javljaju pitanja o najvećoj hrapavosti dovoljnoj za izvršavanje funkcija kao i mogućnostima proizvodnog pogona da te hrapavosti ostvari.

U pravilu, konstruktori se pri određivanju hrapavosti površina vode jednostavnim pristupom traženja najmanje hrapavosti koju proizvodnja može ostvariti i potom je traže na svim površinama za koje znaju da je potrebna manja hrapavost. Ovakav pristup doduše olakšava posao konstruktoru, međutim, kako manja hrapavost traži primjenu preciznije i skuplje tehnologije proizvodnje, on je ujedno i razlog povećanja troškova proizvodnje bez povećanja funkcionalnosti i pouzdanosti proizvoda.

Neiskusni inženjeri koji traže optimalnu površinsku hrapavost, podatke u pravilu dobivaju od starijih kolega koji imaju iskustva sa zadavanjem potrebne hrapavosti te znaju koliko je neka površina, za koju je numerički zadana hrapavost, zapravo "glatka". Zapis svakog pojedinog odabira hrapavosti bi, uz zapis funkcija površine, olakšao buduće odabire hrapavosti jer bi inženjeri imali referencu na koju bi se mogli osloniti tokom donošenja odluka.

Ova procedura će sa svakim izvršenim odabirom povećavati bazu podataka koja će pak poslužiti kao temelj budućih odabira. Naravno, važno je prikupljanje podataka iz prakse koji će omogućiti ocjenjivanje donošenih odluka. Unošenje ocjena u bazu će pak pružiti dodatne podatke koji će omogućiti donošenje kvalitetnije odluke.

Sa stajališta stvaranja same aplikacije potrebno je naglasiti jedno važno ograničenje. Stablo CAD aplikacije ne sadrži površine koje omeđuju prostor trodimenzionalnog modela proizvoda, što znači da nije moguće neposredno u stablu proizvoda naznačiti da je uz određenu površinu vezan zapis. Samu površinu je doduše moguće odabrati u modelu no njezino ime se sastoji od niza brojki i zagrada i kao takvo nije uopće prepoznatljivo.

Pored naznačivanja postojanja zapisa postavljanjem određene oznake samo ime značajke u stablu, moguće je uz površinu vezati određenu bilješku (note). Međutim, ako korisnik ne uključi prikaz bilješki, opet neće biti svjestan postojanja zapisa u bazi.

Kao sigurno rješenje se stoga nameće vezanja zapisa uz značajku modela koja sadrži površinu koja treba biti dodatno obrađena.

Pod pretpostavkom da je stoga korisnik odabrao značajku kao objekt uz koji će biti vezan zapis, biti će prikazana slijedeća forma.

The image shows a Windows-style dialog box titled "Stanje površine". It contains three columns of options. The first column, "Vrsta kontakta", has four radio buttons: "Bez kontakta", "Statički kontakt", "Dinamički kontakt", and "Vizualni kontakt". The second column, "Pritisak kontakta", has three radio buttons: "Nema pritiska", "Mali pritisak", and "Visoki pritisak". The third column, "Kvaliteta površinske obrade", features a large empty rectangular box, a button labeled "Odaberi", and a text input field below it. At the bottom center of the dialog is a button labeled "Dalje".

Slika 26: Sučelje obrazlaganja odabira hrapavosti površine

Korisnik odabire tip kontakta koji površina treba ostvariti. Općenito, što je kontakt dinamičniji uz što veći pritisak, biti će potrebna manja hrapavost površine. Ovisno o karakteristikama kontakta koje korisnik odabere, aplikacija će suziti popis odgovarajućih hrapavosti. Jednom kad korisnik odabere hrapavost sa popisa hrapavosti koje je moguće odabrati, otvara se prozor forme u Accessu.

8.3.5. DODATNI FUNKCIONALNO TEHNOLOŠKI ELEMENTI

Kao i kod elemenata strojeva, i kod funkcionalno-tehnoloških elemenata opisane procedure pokrivaju samo mali dio cijelog popisa. Na temelju opisanih procedura moguće je na sličan način definirati procedure i za druge funkcionalno-tehnološke elemente: stjenke, noge, prijelazne radijuse, prirubnice, zavare, lemljene spojeve, lijepljene spojeve, izlaze za alat, navoje, bradavice, proreze, rupe, žlijebove, grafičke oznake itd.

8.4. ODABIR MATERIJALA

Odabir materijala sličan je odabiru ležaja. Nakon pokretanja aplikacije i odabira procedure „Odabir materijala“, korisnik bira element na koji će vezati zapis. U ovom slučaju element koji se bira može biti samo cijeli model dijela jer je materijal isti za cijeli dio. Odabir je moguće izvršiti u sklopu ili u modelu dijela.

Nakon odabira objekta korisniku se omogućava odabir materijala. Kategorije materijala mijenjat će se od tvrtke do tvrtke, ovisno o broju i tipu materijala koji se koriste, ali i primjeni koja je za njih namijenjena.

Slika 27: Sučelje obrazloženja odabira materijala

Slika prikazuje rješenje gdje se materijal bira na temelju sastav, tj. korisnik zna koji materijal želi. Međutim, moguć je i drugačiji pristup. Moguće je materijale sortirati prema karakteristikama. Korisniku se nudi sučelje u kojem će odabrati koje karakteristike su mu važne, a aplikacija će mu ponuditi na izbor sve materijale koje tvrtka koristi, a koji zadovoljavaju sve navedene zahtjeve.

U karakteristike materijala spadaju

- Čvrstoća
- Tvrdća
- Elastičnost
- Mogućnost plastične deformacije
- Gustoća (specifična masa)
- Otpornost na koroziju
- Prikladnost za obradu odvajanjem čestica
- Livljivost
- Duktilnost
- Estetski dojam

Na kraju korisnik se premješta u Access gdje će korisnik unijeti dodatne podatke, razloge za određeni odabir i alternative.

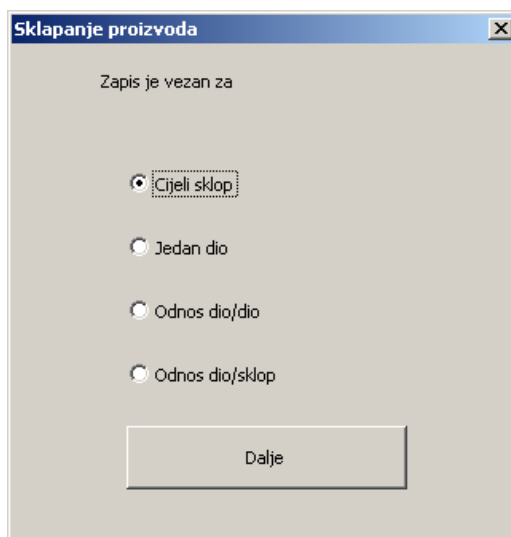
8.5. DEFINIRANJE NAČINA GRADNJE CAD MODELA

8.5.1. SKLAPANJE PROIZVODA

Ako korisnik želi unijeti zapis vezan uz sklapanje proizvoda, tada se zapis može vezati za cijeli sklop (plan sklapanja), pojedini dio sklopa (mjesto za određeni dio ili namjena podsklopa) ili pak za određenu značajku na nekom dijelu (značajka postavljen kako bi olakšala ili omogućila sklapanje). U svakom slučaju, radi se o jednom objektu za koji je vezan paket informacija.

Ako se zapis želi vezati za dva objekta (dva dijela ili dio i sklop), korisniku mora biti omogućen odabir dva objekta.

Slijedeća slika prikazuje prozor odabira tipa zapisa. Ako je odabrana jedna od prve dvije opcije, korisnik će odabrati jedan objekt te mu stoga treba prikazati prozor za odabir jednog objekta. Ako pak odabere neku od druge dvije opcije, treba mu i prikazati prozor za odabir dvaju objekata.



Slika 28: Sučelje obrazloženja izgradnje CAD modela sklopa

Postavlja se pitanje kako formulirati znanje koje se veže uz planiranje sklapanja. Standardne odabire je vrlo teško definirati, pa bi korisniku najvjerojatnije trebale samo mogućnost zapisa neke bilješke i veze prema nekoj vanjskoj datoteci kao što su grafovi relacija između komponenti (K. Case, W. A. Wan Harun, 1997., [5]).

8.5.2. OBLIKOVANJE DIJELA

Glavna namjena ove procedure je omogućavanje korisniku da zapiše kako je oblikovao model u CAD aplikaciji što se pokazuje korisnim kod ponovnog razmatranja dijela koji ima nekoliko stotina značajki.

Iako su općenito tip značajki, njihov raspored i zadani parametri ovisni o različitim faktorima; redoslijedu kojim ih korisnik unosi, broju i tipu značajki koje određena aplikacija podržava, proizvodnim tehnologijama, načinu sklapanja i brojnim drugim, u ovom slučaju zanima nas samo oblikovanje dijela u CAD aplikaciji.

Druge procedure trebale bi pokriti područje proizvodne tehnologije.

Problem koji je u ovom slučaju potrebno razmotriti je mogućnost da se isti zapis želi vezati uz različite značajke. Postoji stoga mogućnost da će korisnik htjeti u jednom potezu vezati zapis uz veliki broj značajki. Tehnički je ovo relativno teško izvesti. Za sada ćemo pretpostaviti da će korisnik ipak vezati zapise jedan po jedan, tj. Da će u takvim slučajevima zapisi biti vezani za cijeli model. U svakom slučaju, korisniku će biti dan prozor za odabir jednog objekta.

Nakon odabir, korisnik se prebacuje na Access formu koja će omogućiti upis ostalih podataka te vezanje vanjskih dokumenata.

Atributi koji se mogu vezati za pojedinu značajku mogu uključivati središnjicu, konstrukcijsku ravninu, redoslijed značajki itd.

8.5.3. NAMJENA GEOMETRIJSKIH REFERENCI

Oblikovanje CAD modela provodi se u trodimenzionalnom prostoru u kojem se model smješta u odnosu na početni Kartezijev koordinatni sustav. Ishodište ovog sustava definira tri osnovne ravnine (uobičajeno nazvane Front, Right, Top). Prvi presjek (Sketch) čijom će se translacijom po krivulji dobiti prva značajka najčešće će biti smješten u jednu od te tri ravnine.

Dijelovi sa složenom geometrijom tražit će korištenje dodatnih geometrijskih referenci poput ravnina za dodatne presjeke ili osi rotacije. Zadavanje ovih referenci jedna je od mogućnosti parametarske tehnologije CAD aplikacija koja omogućava oblikovanje čak i najsloženijih dijelova, međutim naknadno razmatranje CAD modela sa stotinama značajki od kojih ima nekoliko desetaka geometrijskih referenci, je vrlo teško jer nije uvijek jasno čemu određena referenca služi.

Ova procedura namijenjena je upravo omogućavanju korisniku da zapiše namjenu pojedinih geometrijskih referenci. Kako se očekuje zapisivanje za pojedinačnu referencu, korisniku će biti omogućen odabir preko prozora za odabir jednog elementa. Filtar pretraživanja treba biti postavljen na odabir geometrijskih elemenata koji uključuju ravnine, osi i točke.

Kad je korisnik odabrao element, prikazuje mu se prozor Access forme u kojem će moći upisati objašnjenje namjene konstrukcijskog elementa. Kako ovi elementi ne predstavljaju nikakve stvarne objekte, a upotrebljavaju se samo u slučaju kada korisnik nije u mogućnosti oblikovati dio bez njih, postavlja se pitanje mogućnosti definiranja njihovih standardnih funkcija.

S obzirom da ne postoje standardne funkcije elemenata, kao ni standardni set razloga za ili protiv, forma Accessa je vrlo jednostavna.

8.6. NEKE DODATNE BILJEŠKE

Pojedini odabiri u istom slučaju mogu biti preferiraniji od drugih iz samo jedno razloga: standardizacija unutar tvrtke. U takvim slučajevima, mogući izbori trebali bi imati neku dodatnu oznaku koja bi omogućavala inženjeru razlikovanje između ovih, u općem slučaju, jednakopravnih mogućnosti. Ova oznaka može biti u numeričkom obliku koji bi omogućio jednostavno ispisivanje mogućih odabira tako da su korisniku ponuđeni u nizu od onih koji se najviše preporučuju prema onima za koje nema preporuke.

Iz istog razloga, kako inženjeri preferiraju korištenje poznatih rješenja, a svaki odabir se zapisuje, uz svaki mogući odabir bilo bi korisno ispisati koliko je puta odabrana upravo ta mogućnost odabira. Ova brojka neće pružiti puni uvid u okolnosti pod kojima su svi ti

Ponekad je teško spremiti sve informacije u pisanoj formi te će izgled samog CAD modela biti koristan. Kako bi se izbjeglo podizanje CAD modela svaki put kad je određen pogled dovoljan da objasni željeni zapis, bilo bi dovoljno spremiti sliku ekrana (Screenshot) i postaviti ovu slikovnu datoteku kao hiperlink u bazi podataka. Naravno, s ciljem olakšavanja posla inženjera, ovaj algoritam bi trebalo automatizirati do stupnja u kojem korisnik samo mora odabrati želi li snimiti sadržaj ekrana ili ne.

9. PRETRAŽIVANJE BAZE ZNANJA

Svrha zapisivanja znanja je njegovo korištenje u budućnosti što je nekoliko puta naglašeno u ovom radu. Samo zapisivanje znanja nije problematično, iako može biti poprilično zamorno, a može se izvesti na više načina. Problem koji je potrebno riješiti je kako unutar sveukupnog zapisanog znanja pronaći ono koje je u tom trenutku potrebno.

Upravo iz razloga da se ubrza sustav pretraživanja, cijeli je ovaj sustav izgrađen oko računalne baze podataka koja nam omogućava gotovo trenutačno pretraživanje baze i ispisivanje svih podataka koji odgovaraju danom upitu. U prethodnim poglavljima objašnjena je struktura baze i način popunjavanja. Na ovom mjestu objasniti će se način pretraživanja baze. Upravo mogućnost pretraživanja na način na koji će to ovdje biti opisano uvjetovao je način oblikovanja pojedinih dijelova baze.

U poglavlju o preliminarnim ispitivanjima predloženog pristupa na konkretnom konstrukcijskom uredu navedeno je kako je u razgovoru u tvrtci "KLEX" utvrđeno kako se u praksi javljaju dvije vrste pretraživanja:

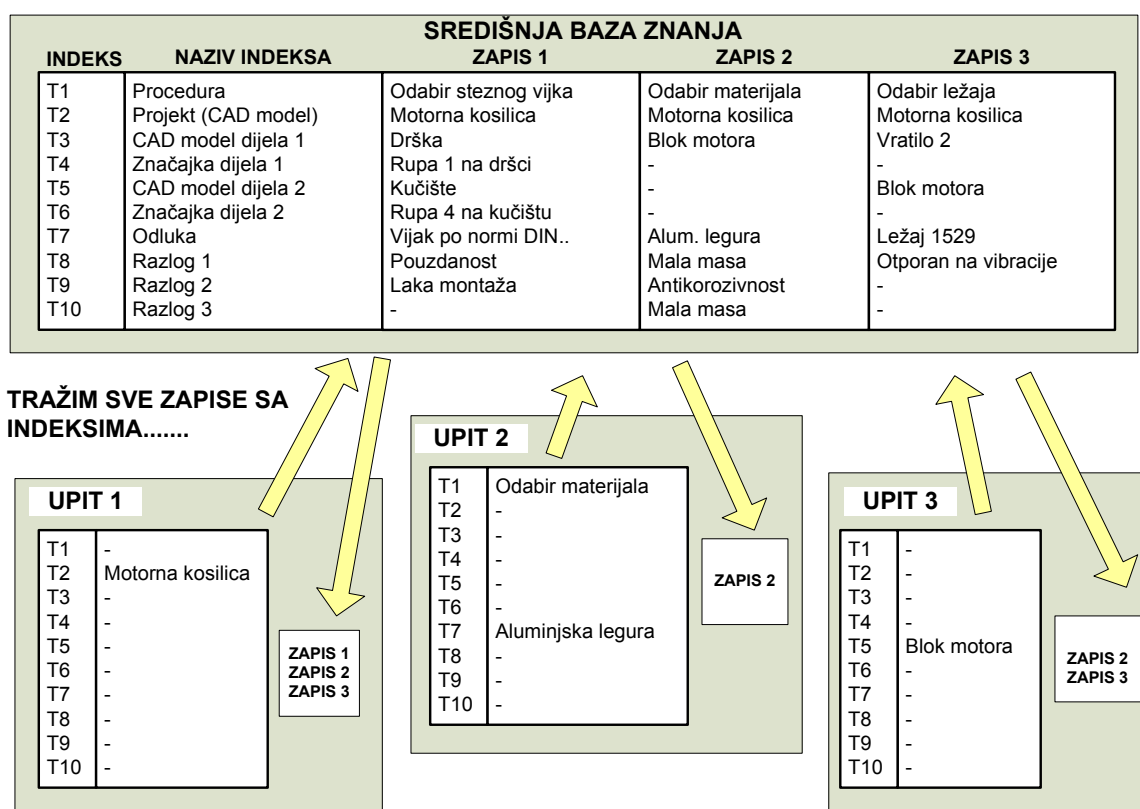
- Specifično pretraživanje za pojedinim podatkom
- Opće pretraživanje za tipovima podataka

Pretraživanje baze s ciljem traženja pojedinog podatka provodi se u slučaju kada se tokom redizajna traži razlog za određenu odluku koju se želi promijeniti. Neovisno o razlogu zbog kojeg se želi izmijeniti donesena odluka, u svakom je slučaju zajedničko poznavanje velikog broja dijelova zapisa. U pravilu poznat je projekt, značajka, dio i/ili sklop uz koji je vezan zapis, približno vremensko razdoblje te najvjerojatnije procedura kojom je zapis generiran.

Opće pretraživanje baze provodi se sa ciljem dobivanja zapisa koji nose informacije iz određenog područja koje se najčešće poklapa sa nekom od procedura. Ovakva se pretraživanja baze provode s ciljem pronalaženja prijašnjih odluka i razloga za te odluka u svrhu donošenja odluke iz istog područja. Broj parametara prema kojem se pretražuje baza je manji, najčešće uključuje samo proceduru i neku od mogućnosti odabira. Korisnik za ovakvo pretraživanje dobiva niz zapisa na temelju kojih može donijeti odluku koju će potom odgovarajućom procedurom zapisati u bazu.

Oba načina oslanjaju se na upite (Query-e) koji koriste indekse, odnosno ključne riječi zapisane procedurama koje su podržavale zapise KC-ova.

Neovisno o tipu pretraživanja uočava se kako korisnik za podatke koje traži poznaje određeni broj parametara. Radi se o parametrima koje je korisnik generirao kroz procedure, a i samim odabirom procedure. Na te parametre koji su preko procedura kategorizirani oslanjaju se upiti za pretraživanje. Takvi upiti mogu se ponuditi gotovi u bazi ili se mogu ponuditi sučelja za dinamičko generiranje upita.



Slika 29: Struktura sustava za pretraživanje

Stoga će i sustav pretraživanja omogućavati korisniku da odabirom parametara sužava popis zapisa koje mu baza nudi kao odgovor na njegov upit. Ovakav način pretraživanja je specifičan pristup ovog rada. U drugim se sustavima zasebno stvara zapis, a zasebno sustav kategorizacije, bilo ručno odabirom iz popisa mogućih indeksa, bilo automatski uporabom ekspertnih sustava koji sami formiraju kategorije i smještaju zapise u njih. U pristupu opisanom u ovom radu, sustav kategorizacije je zadan unaprijed, a zapis se smješta u kategoriju putem samog sadržaja zapisa. Pored smanjenog obima posla pri upisu, olakšano je i samo pretraživanje jer korisnik zna koje su sastavnice samog zapisa prema kojima može pretraživati bazu.

10. EKSPERIMENTALNA EVALUACIJA I TESTIRANJE PREDLOŽENOG SUSTAVA

U prethodnim poglavljima opisana je nekolicina procedura zapisa objašnjenja donesenih odluka kao i opis tehničke realizacije aplikacije provođenja procedure obrazloženja odabira.

Na ovom mjestu se stoga pretpostavlja da je aplikacija u svojoj inicijalnoj izvedbi gotova te da je sustav spreman za korištenje u konstrukcijskom uredu tokom stvarnog razvoja proizvoda.

U planiranju eksperimenta pored ljudskih i materijalnih resursa potrebnih za provođenje, nužno je planirati i vremenski okvir unutar kojeg će se provoditi eksperiment. Što se tiče ljudskih i materijalnih resursa, oni su ograničeni mogućnostima konstrukcijskog ureda s kojim bi se dogovorila suradnja. Naravno, veći broj inženjera koji će generirati što veći broj zapisa na što većem broju što raznolikijih projekata je poželjan zbog mogućnosti prikupljanja što većeg broja uobičajenih razloga za određeni odabir uz veću pouzdanost rezultata statističke obrade zapisa. Međutim, u realnom slučaju, broj inženjera kao i raznolikost zapisa bit će značajno ograničeni.

Ovoj ograničenosti zapisa moguće je doskočiti na dva načina. Prvi uključuje planiranje eksperimenta na dulje vremensko razdoblje, a drugi uključivanje više konstrukcijskih ureda u eksperiment. Međutim, provođenje eksperimenta unedogled sa velikim brojem učesnika može rezultirati i negativnim posljedicama u vidu prikupljanja ponavljajućih podataka i frustriranosti učesnika. Oboje će negativno utjecati na rezultate eksperimenta, uz nepotrebno rasipanje resursa i vremena.

Uzevši u obzir i činjenicu kako jedan eksperiment ne može dati rezultate koji bi rezultirali modificiranjem aplikacije u komercijalno primjenjivu verziju, eksperiment je potrebno planirati kao početni korak verifikacije pristupa koji će dati inicijalne rezultate za ocjenjivanje primjenjivosti kao i podatke nužne za prvi krug unapređivanja sustava. Jednom kad sve zamjerke i prijedlozi budu implementirani u sustav, moguće je planirati slijedeći krug ispitivanja sustav u realnim uvjetima. Na ovaj način, iterativnim koracima razvijat će se primjenjivi sustav prilagodljiv promjenama koje tokom vremena nastaju u procesu razvoja proizvoda.

Ljudski i materijalni resursi biti će, uzevši u obzir sve dosad navedeno, ograničeni mogućnostima konstrukcijskog ureda koji bi inicijalno sudjelovao u ispitivanju primjenjivosti sustava. Ostaje dakle pitanje vremenskog okvira.

Kako se rad sustava dijeli na dvije faze, zapis i dohvaćanje znanja, tako će se i planirana eksperimentalna evaluacija dijeliti na dvije faze. U prvoj fazi utvrdit će se primjenjivost sustava za potrebe zapisivanja znanja, dok će se druga faza sastojati od provjere uspješnosti dohvaćanja potrebnog znanja.

10.1. PRVA FAZA: ZAPISIVANJE ZNANJA

Kako je sustav namijenjen zapisivanju odluka tokom procesa konstruiranja proizvoda, vremenski okvir ne bi smio biti manji od vremenskog razdoblja unutar kojeg se izradi čitav projekt. Koliko je vremena potrebno za jedan cijeli projekt ovisi o ustroju tvrtke te vrsti projekata, međutim, uzevši u obzir potrebnu kompleksnost i tip projekata koji su pogodni za testiranje sustava, očekuje se kako će biti potrebno vremensko razdoblje od minimalno dva mjeseca.

Ova donja granica proizlazi iz iskustva sa razvojem pojedinih dijelova ili podsklopova kućanskih aparata, ručnog alata te specifičnih funkcionalnih jedinica većih mehaničkih sustava. Ovakvi projekti ocijenjeni su prikladnima za inicijalno ispitivanje sustava jer uključuju veliki dio komponenata ili funkcionalnih značajki čiji je odabir opisan danim procedurama ali se sastoje od relativno jednostavnih elemenata i manjeg broja dijelova što olakšava pregledavanje zapisa bez poznavanja specifičnosti visokospecijaliziranih komponenti i veoma kompleksnih sustava.

Što se tiče gornje granice vremenskog razdoblja unutar kojeg ima smisla provoditi eksperiment, nju je sa teoretskog stajališta nešto teže utvrditi zbog toga što ovisi o teže mjerljivim faktorima kao što su interes učesnika za sudjelovanjem. Očekuje se kako će zapravo minimalna granica biti ujedno i glavni orijentir za dogovor o trajanju eksperimenta, pa za realno planiranje eksperimenta gornja teoretska granica nije ni važna.

Pored toga, kako većina konstrukcijskih ureda, a i konstruktora paralelno radi na nekoliko projekata, provođenje eksperimenta unutar vremenskog raspona jednog projekta, omogućit će i prikupljanje i dijelova znanja generiranih tokom početnih ili završnih faza

drugih projekata, ovisno o tome da li su počeli prije ili poslije započinjanja eksperimenta. Na ovaj način, prikupljanje podataka iz različitih podataka je, barem djelomično, ostvareno.

Stoga je zaključak ovog razmatranja kako je eksperiment u trajanju od (približno) šest mjeseci u jednom konstrukcijskom uredu minimalan okvir unutar kojeg je moguće prikupiti potrebne inicijalne podatke.

Ostaje stoga pitanje podataka koje se želi prikupiti. U prvom redu, ovi podaci uključuju samo znanje o odlukama koje su donesene. Ovo znanje poslužit će kao osnova baze znanja uobičajenih razloga za donošenje određenih odluka. Uspješno generiranje zapisa biti će potvrda primjenjivosti sustava kao i izvor razloga koji se učestalo ponavljaju.

Moguće je da upisani razlozi budu krivi, bilo potpuno krivi, bilo krivo primijenjeni. Na ovom mjestu ispravnost upisanih razloga neće se dovoditi u pitanje tokom samog provođenja eksperimenta, već se evaluacija upisanih razloga planira provesti po završetku eksperimenta. Od inženjera će se tokom samog eksperimenta tražiti samo da upišu razloge, te podatke o vremenu zapisa i osobi koja je napravila zapis. Premještanjem evaluacije zapisa na razdoblje nakon završenog eksperimenta žele se izbjeći dvije negativnosti paralelne evaluacije.

Prva negativnost tiče se gubitka vremena i zamor uslijed paralelnog rada dvaju poslova, posebice stoga što bi za evaluaciju razloga bilo potrebno dodatno angažirati iskusnije inženjere čime se značajno smanjuje njihova produktivnost. Druga negativnost potječe iz prirodnog straha od negativne kritike koji se može javiti kod inženjera sa manje iskustva što bi značajno utjecalo na njihovu zainteresiranost za eksperiment kao i na broj zapisa. Očekuje se kako bi inženjeri izbjegavali zapise kod kojih ne bi bili sigurni u točnost razloga za donesene odluke. Pomicanjem ocjenjivanja na neko drugo vremensko razdoblje, kao i „utapanje“ pojedinog zapisa u mnoštvo drugih, nabrojene negativne posljedice bi se trebale izbjeći.

Zapisi razloga odabira prvi su dio informacija koje se žele prikupiti. Drugi dio su iskustva iz primijene samog sustava, kako definiranih procedura, tako i rada same aplikacije. Iz primjedbi, prijedloga i zapažanja inženjera biti će moguće ugraditi određena poboljšanja u sučelju, reducirati broj grešaka u programskom kodu kao i dodati određene funkcije aplikacije koje će olakšati njezinu primjenu. Prikupljanje ovih dodatnih informacija moguće je izvršiti na dva načina. Prvi je prikupljanje ciljano izrađenih bilješki inženjera, a drugi intervjuiranje učesnika. Da li će biti primijenjen jedan od ova dva načina ili neka kombinacija, ovisi o dogovoru sa konstrukcijskim uredom.

Gledano u vremenskom okviru, unutar ova dva mjeseca eksperimenta kojima je postavljeni cilj utvrđivanje sustava zapisa znanja, postoje dvije točke u kojima će biti

potrebno izvršiti dvije provjere. Prva provjera tiče se razloga za pojedine odabire. Nakon prvog tjedna primjene biti će potrebno provjeriti da li popisi uobičajenih razloga sadrže sve potrebne razloge za objašnjavanje. Provjera je relativno jednostavna. Ukoliko se u zapisima ne pojavljuju dodatne bilješke sa učestalim istim razlozima, tada su svi potrebi razlozi uneseni u sustav. Ukoliko se učestalo pojavljuje isti razlog u dodatnoj bilješci, potrebno ga je ubaciti u popis uobičajenih razloga.

Druga provjera tiče se funkcionalnosti sučelja. Nakon približno mjesec dana biti će potrebno prikupiti bilješke koje su inženjeri napravili po pitanju rada u samom sučelju. Ukoliko postoje učestale zamjerke na samo sučelje, njih je potrebno u najkraćem vremenskom roku ispraviti.

Po završetku prve faze predviđenog eksperimenta, davanja aplikacije na korištenje u realnim uvjetima, slijedi evaluacija svih prikupljenih podataka. Obrada podataka omogućiti će modificiranje aplikacije čime će se ostvariti nužni preduvjet za slijedeći krug ispitivanja aplikacije u realnim uvjetima. U tom trenutku, ovisno o mogućnostima biti će potrebno planirati drugu fazu eksperimenta.

10.2. DRUGA FAZA: DOHVAT ZNANJA

Nakon završetka prve faze eksperimentalne evaluacije sustava baza bi trebala sadržavati relativno veliki broj zapisa. U drugoj fazi eksperimenta ova će se baza staviti na raspolaganje inženjerima kao izvor znanja.

Naravno, da bi se u bazi znanja pronašao zapis ili zapisi koji su potrebni inženjeru, potrebno je formirati sustav pretraživanja baze. U poglavlju o pretraživanju baze objašnjen je princip rada sustava pretraživanja kako je inicijalno zamišljen. U ovoj fazi eksperimenta cilj je ispitati predloženi sustav pretraživanja. Kako bi se mogli formirati bilo kakvi zaključci o uspješnosti rada sustava pretraživanja, inženjeri koji sudjeluju u eksperimentu morat će bilježiti kada su tražili koje znanje te jesu li ga našli ili ne.

U slučaju da se pregledom bilješki utvrdi da je traženo znanje pronađeno i primijenjeno, može se izvući zaključak kako cijeli sustav funkcionira jer je uspješno zatvoren krug donošenja odluke na temelju određenog znanja, zapisivanja te odluke u bazu, kasnijeg pronalaženja tog zapisa kao iskustvene osnove za donošenje nove odluke te zapis nove odluke.

Međutim, u slučaju bilješki u kojima stoji da inženjeri nisu našli potrebno znanje, postavlja se pitanje zbog čega je tome tako. Prva opcije je da zapis uopće ne postoji, tj. znanje nije upisano. Nemogućnost stvaranja kvalitetnog zapisa trebala je biti riješena tokom prve faze projekta što znači da bi zapis trebao postojati jer n postoje objektivni razlozi zbog kojih zapis ne bi postojao.

Druga opcija je da zapis postoji ali ga korisnik ne može pronaći zbog toga što nije upoznat sa sustavom ili zbog toga što je sustav krivo postavljen. Neovisno o razlogu, do kraja druge faze eksperimenta, uspješno pronalaženje zapisa trebalo bi činiti vrlo veliki dio rezultata postavljenih upita.

Ostaje stoga samo pitanje trajanja druge faze eksperimenta. Ono će naravno ovisiti o tom kada će inženjeri početi tražiti znanje koje su upisali. Znanje koje je zapisano u određenom trenutku neće biti odmah potrebno za korištenje, već će se za njim javiti potreba tek nakon doglednog vremena, naravno ukoliko nije donesena kardinalna greška u odlučivanju. Izuzev ovakvog kritičnog slučaja, pretraživanje se vrši na dva prethodno opisana načina; pretraživanje s ciljem traženja specifičnog podatka te pretraživanje s ciljem usvajanja novih znanja.

U slučaju traženja specifičnog podatka, traži se obrazloženje konkretne pojedinačne odluke tokom procesa razvoja nove generacije proizvoda kada se traži način da se poboljša stariji model. Razmotri li se učestalost pojavljivanja novih modela istog tehničkog proizvoda primjećuje se kako se na tržištu novi modeli pojavljuju najčešće svakih godinu dana što znači da se i razvoj proizvoda bazira na istom vremenskom okviru, tj. svakih godinu dana vršit će se modifikacija istog proizvoda, pa će i u tom vremenu nastati potreba za dohvaćanjem zapisanog znanja. Drugim riječima, zapis koji je unijet na kraju prethodnog projekta biti će tražen tek na kraju slijedećeg projekta, tj. zapis će biti tražen tek nakon godinu dana.

Pretražuje li inženjer bazu znanja s ciljem usvajanja novih znanja, tada znanje koje traži nije vezano za pojedini projekt. Međutim, kako u ovom slučaju korisnik treba više zapisa na istu temu, potrebno je držati sustav u upotrebi dovoljno dugo da se unese više zapisa odluka za slične probleme. Posljedično, čak i kod proizvoda sa kratkim ciklusom razvoja biti će potrebno približno dvije godine prije nego se prikupi dovoljan broj zapisa za usporedno pregledavanje iz kojeg se mogu izvući opći zaključci kakve traži inženjer koji želi proširiti svoje znanje.

11. REZULTATI

Ovaj rad je definirao 15 procedura zapisa obrazloženja odabira te prikazao popis dodatnih 30-tak procedura koje je potrebno razraditi. Formiranje procedura izvršeno je na temelju promišljanja i izvora dostupne literature. Tri su procedure, odabir ležaja, odabir steznog spoja te pozicioniranje i dimenzioniranje rebara, razrađene u suradnji sa tvrtkom „KLEX“, d.o.o., a jedna je procedura dodatno razrađena u suradnji sa Prof. Heroldom.

Za svaku definiranu proceduru dano je sučelje kao osnova za razvoj aplikacije kao i plan organizacije čitave aplikacije za upis obrazloženja odabira. Također, dana je struktura baze znanja koja će pohraniti sve zapise kao i opis strukture sustava pretraživanja.

Rad je dao prijedlog provedbe eksperimentalne evaluacije primjenjivosti sustava kroz dvije faze. U prvoj fazi u trajanju od 6 mjeseci do 1 godine cilj je potvrditi primjenjivost definiranih procedura kao sustava zapisivanja znanja, te modificirati i nadograditi sustav u skladu sa potrebama korisnika.

Druga faza predviđena je za verifikaciju sustava dohvata znanja putem razvijenog sučelja pretraživanja. Ciljevi su identični ciljevima prve faze; verifikacija primjenjivosti te modifikacija i nadogradnja u skladu sa potrebama i željama korisnika. Trajanje druge faze značajno ovisi o prosječnom vremenu trajanja jednog projekta te broja zapisa koji budu uneseni tokom prva faze, a procjena je na dodatnih 6 mjeseci do godine dana sa početkom u trenutku kad budu potrebni zapisi generirani tokom prve faze.

12. RASPRAVA

Glavna razlika između pristupa zapisivanju inženjerskog znanja koji je primijenjen u ovom radu te pristupa koji su razvijani u drugim projektima je smjer izgradnje sustava kategorizacije. Dok većina literature opisuje razvoj sustava polazeći od vrha hijerarhije, u ovom radu opisan je pristup u kojem se krenula od njezina dna.

Sa akademskog stajališta, polazak od vrha je ispravan jer će omogućiti pravilnu i punu kategorizaciju do razine pojmova koji se za određenu primjenu smatraju nedjeljivima. Polazak od najnižih pojmova, pa prema vrhu prati čitav niz problema.

Prvi problem je generiranje svih pojmova sa najniže razine što je potencijalno zamoran i teško ostvariv posao. Postoji velika vjerojatnost kako ovakav pristup neće omogućiti generiranje svih pojmova što će rezultirati nekompletnošću hijerarhije. Samim time što određena hijerarhije pojmova nije potpuna, ozbiljno je ugrožena njezina svrsishodnost te znanstvena vrijednost.

Nepostojanje hijerarhije u trenutku generiranja pojma drugi je značajan problem. U prvom redu, hijerarhiju će biti potrebno definirati nakon generiranja pojmova, što znači da će postojeće pojmove sa najniže razine hijerarhije trebati podijeliti u skupine te stvoriti njima nadređene pojmove. Ova više razina se ponovo grupira i dodjeljuje višim pojmovima sve dok se ne dođe do jednog, najvišeg pojma.

U realnom slučaju, naknadno generiranje hijerarhije je problematično zbog različitih mogućnosti stvaranja kategorija, potencijalne pripadnosti istog pojma dvama kategorijama te mogućem postojanju inicijalno generiranog pojma koji se može interpretirati kao pojam iz više hijerarhijske razine. Procedure veza s glavinama ili osiguranja aksijalnog pomaka su primjeri takvih pojmova. Iako se radi o elementima strojeva, svaka od ovih procedura uključuje mogući izbor rješenja u vidu odabira pričvrsnog vijka. Ovime bi se naveden procedure mogle smatrati nadređenim pojmom proceduri odabira vijka. Također, kako je i prijenos momenta vrtnje i onemogućavanje aksijalnog pomaka moguće izvršiti ugradnjom vijčanih spojeva, pa bi, iako su sama tehnička rješenja nešto različita, procedura odabira pričvrsnog vijka imala bi dva nadređena pojma. U konačnici, postoje tehnička rješenja koja izvršavaju obje funkcije istovremeno, što dodatno komplicira situaciju.

Postavlja se stoga pitanje, ako je odabrani pristup toliko teoretski problematičan, zašto je odabran? Odgovor na ovo pitanje naglašen je više puta u ovom radu. Odabran je zbog toga

što se očekuje kako će ovaj pristup generirati procedure koje će omogućiti efektivno i brzo zapisivanje i pretraživanje znanja, što dosada razvijene hijerarhije nisu uspjele..

Sa znanstvenog stajališta javlja se čitav niz zamjerki. Zanimare li se aspekti zamjerki koji u industrijskoj praksi nemaju utjecaja, ostaje kao glavne zamjerka nemogućnost kategoriziranja bilo kojeg tipa znanja generiranog tokom procesa razvoja proizvoda, a što kategorizacije razvijane od najvišeg pojma prema nižima omogućuju. Problem koji se, međutim javljao kod korištenja navedenih kategorizacija je apstraktnost pojmova koja proizlazi iz širine područja u kojem se nalaze pojmovi koje treba kategorizirati.

Dvosmislenost kategorizacije prilikom zapisa otežavala je i pretraživanje jer je uvijek ostajala sumnja u postojanje zapisa koji se traže ali ih sustav nije ponudio prilikom pretraživanja jer osoba koja je napravila zapis nema isto shvaćanje sustava indeksiranja kao i osoba koja je postavila upit za znanjem. Na ovaj način, odgovori na gotovo sve upite dolaze uz sumnju u postojanje zapisa koji su relevantni za postavljeni upit, ali ih sustav nije ponudio korisniku.

Prednost pristupa primijenjenog u ovom radu je u jednoznačnosti pojmova koja će omogućiti jednostavnije i brže indeksiranje zapisa prilikom upisa, ali i brže i sigurnije pretraživanje.

U pogledu ovih dviju karakteristika sustava kategorizacije, znanstvene vrijednosti i primjenjivosti, uočavaju se slijedeće dvije mogućnosti.

Sa znanstvenog stajališta, opisani pristup ima upitnu vrijednost. Kako bi se sustav razvio, predloženo je spajanje prijedloga kategorizacije iz ovog rada sa prethodno navedenom združenom taksonomijom (Ahmed, Štorga, 2008., [21]). Sa znanstvenog stajališta, sustav iz ovog rada ostvario bi određeni stupanj verifikacije znanstvene vrijednosti, dok bi sa stajališta primjenjivosti bilo olakšano generiranje nedostajućih pojmova, tj. procedura jer bi iz postojeće združene taksonomije bili dobiveni nadređeni pojmovi koji bi, uz usporedbu sa postojećim procedurama, olakšali generiranje novih procedura.

Sa stajališta primjenjivosti, rasprave o upotrebljivosti mogu se provlačiti unedogled. Međutim, jedini relevantni podatak koji nedvosmisleno pokazuje koji se pristup bolji bili bi podaci z prakse. Stoga bi oba sustav trebalo dati na pokusnu primjenu koja bi pokazala pozitivne i negativne strane svakog od sustava. Rezultati ovakvog eksperimenta dali bi osnovu za zaključak koji je sustav bolji, tj. ukoliko se pokažu značajne pozitivne i negativne karakteristike na oba sustava, kako spojiti sustave.

13. ZAKLJUČCI

Pregledom dostupne literature utvrđeno je kako se postojeći sustavi upravljanja znanjem baziraju na hijerarhiji pojmova koja je generirana od vršnog pojma inženjerskog znanja prema nižima. Ovakav pristup će naravno obuhvatiti čitavo inženjersko znanje ali se u praksi pokazalo kako sustavi ove vrste nisu prihvaćeni s oduševljenjem zbog male dubine razrade pojmova čime se hijerarhije nisu odmakle od apstraktne razine.

Uzevši u obzir učestale polovične rezultate primjenjivosti ovakvih sustava javlja se logična pretpostavka kako bi s ciljem izgradnje praktično primjenjivog sustava bilo razumno pokušati generirati hijerarhiju znanja od najnižih pojmova prema višima. Pri tome bi ovi najniži pojmovi kao i procedure koje ono definiraju bile razvijene u suradnji sa inženjerima konstruktorima.

Nakon inicijalnog oblikovanja u suradnji sa tvrtkom „KLEX“ formiran je popis kriterija za tri procedure zapisa. Ovi su se popisi pokazali dovoljno sažetima za prikazivanje na jednom sučelju te je stoga zaključeno kako će i popisi za ostale procedure biti iste veličine. Zahvaljujući ovakvoj strukturi procedura zapisa moguće je u kratkom vremenu generirati zapis objašnjenja odabira.

Za predloženi sustav utvrđene su setovi prednosti i mana pred ostalim sustavima. U odnosu na opisane sustave, prednosti ovog sustava su:

- Manje pisanja i razmišljanja o svojstvima zapisa – znatno manji utrošak radnog vremena konstruktora, stoga i veće šanse za efikasnu i praktičnu primjenu sustava
- Unaprijed definirana struktura zapisa omogućava stvaranje unaprijed definiranih upita za pretraživanje
- Mogućnost jednostavnog dodavanja novih procedura – fleksibilnost primjene u različitim okolinama
- Unificiranost i nedvosmislenost indeksiranja – pretpostavlja se da može djelomično riješiti osnovne probleme drugih razmatranih sustava
- Praktički automatizirano indeksiranje – dodatno oslobađa konstruktora u odnosu na korištenje ostalih sustava
- Konkretni sustav pretraživanja – očekuje se da će biti daleko lakše pronaći željeni zapis u odnosu na primjerice.. pretraživanje po nasumce odabranim ključnim riječima

Utvrđene mane u odnosu na druge sustave su:

- Nemogućnost obuhvaćanja cijelog područja inženjerskog znanja
- Sustav vrijedi samo u specifičnoj okolini za čiju domenu su razvijene konkretne procedure

Iako svaki konstrukcijski ured ima svoj set razloga za pojedini odabir, postoji niz razloga koji će se ponavljati, tj. postoje razlozi koji su isti za sve konstrukcijske urede. Moguće je usmjeriti istraživanje u pravcu definiranja ovakvih setova općevažećih obrazloženja za što bi bilo potrebno provesti eksperiment prikupljanja razloga iz barem tri konstrukcijska ureda koji bi e potom usporedili.

Neovisno o razlozima za pojedini odabir, same procedure zapisa obrazloženja su univerzalne. Razlike će se pojaviti u području djelatnosti pojedinog ureda kao i dostupnoj tehnologiji zbog čega svi uredi neće koristiti sve procedure. Neovisno o primjeni, znanstvena evaluacija ovih procedura leži u mogućnosti generiranja hijerarhijskog sustava pojmova. Kako je u trenutku pisanja ovog rada u izradi združena ontologija za koju je utvrđeno kako posjeduje pojmove koji se mogu proglasiti nadređenima procedurama iz ovog rada, predlaže se daljnji rad na povezivanju ovih dviju sustava. Povezivanjem bi sustav procedura dobio svoju znanstvenu podlogu, a združena ontologija mogućnost eksperimentalne verifikacije.

Kako je objašnjeno u poglavlju o planiranju eksperimentalne evaluacije sustava, biti će potrebne dvije godine za provjeru cijelog sustava zbog toga što će između zapisa i dohvata znanja proći onoliko vremena koliko je potrebno da se na bazi znanja zapisanog za prethodni projekt završi novi, tj. Biti će potrebna dva cijela ciklusa projekata, što, zbog ritma kojim se na tržište šalju nove generacije istih proizvoda, znači minimalno dvije godine.

Kako ovakvo vremensko razdoblje za provođenje eksperimenta značajno prelazi razdoblja koja su pogodna za suradnju sveučilišta i gospodarskih subjekata na temelju solidarne suradnje i održavanja dobrih odnosa, postavlja se potreba za formiranjem zasebnog projekta sa većim brojem učesnika kojim bi se mogli ostvariti valjani zaključci i napredovanje razvoja sustava do visokog stupnja primjenjivosti u realnim uvjetima.

14. ZAHVALE

Zahvaljujem svom mentoru, dr. sc. Nevenu Pavkoviću, na podršci i korisnim savjetima tokom izrade ovog rada.

Prof. dr. Saeemi Ahmed (Technical University Copenhagen) zahvaljujem na korisnim diskusijama, uvodu u područje njenog rada te savjetima o radovima drugih istraživača.

Dr. sc. Mariu Štorgi zahvaljujem na savjetima vezanim uz problematiku ontologije procesa konstruiranja.

Danijelu Rohdeu, dipl. ing., zahvaljujem na ustupljenoj literaturi te uvidu u njegovo područje istraživanja.

Svojim roditeljima dugujem zahvalnost za njihovu potporu tokom mog cjelokupnog školovanja.

15. POPIS LITERATURE

- [1] A feature-based database evolution approach in the design process, Raimundo R.M. da Cunha, A. Dias, Robotics and Computer Integrated Manufacturing, Volume 18, Number 3, pp. 275-281(7), Elsevier, June 2002
- [2] A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design, B.J.Hicks, S.J. Culley, R.D. Allen, G. Mullineux, International Journal of Management, Volume 22, Number 4, August 2002 , pp. 263-280(18)
- [3] An Engineering Design Knowledge Reuse Methodology Using Process Modelling, David Baxter, James Gao, Keith Case, Jenny Harding, Bob Young, Sean Cochrane, Shilpa Dani, Springer-Verlag London Limited, 2007
- [5] A Representation of Assembly and Process Planning Knowledge for Feature-based Design, K. Case & W. A. Wan Harun, 'Advances in Manufacturing Technology XI', (ed. D K Harrison), Glasgow Caledonian University, ISBN 1 9012 4811 9, pp 73-78, 1997
- [6] Analiza spojeva vratila s glavinom, Magistarski rad, Željko Ivandić, Zagreb, 1996.
- [7] Answering Engineers' Questions Using Semantic Annotations, Sanghee Kim, Rob Bracewell and Ken Wallace, Cambridge University Press, Volume 21, 02 April 2007
- [8] Behavior and Design Intent Based Product Modeling, László Horváth, Imre J. Rudas, Acta Polytechnica Hungarica Vol. 1, No. 2, 2004
- [9] Capture and Reuse of Experience, Dr Saeema Ahmed, Professor Ken Wallace, Prof Luciënne Blessing, Engineering Design Centre, University of Cambridge, 2002
- [10] Characterizing information acquisition in engineering design, Marco Aurisicchio, Darwin College, Cambridge, May 2005

-
- [11] Design Process Improvement: A Review of Current Practice, John Clarkson, Claudia Eckert, Chapter 13: Engineering Knowledge Management, Ken Wallace, Saeema Ahmed, Rob Bracewell, Springer, 2005, ISBN:185233701X
- [12] Elementi strojeva, Treće izmijenjeno i dopunjeno izdanje, Karl Heinz Decker, Golden Marketing, Tehnička knjiga, Zagreb, 2006.
- [13] Engineering Knowledge Management, prezentacija, Saeema Ahmed, Zagreb, 2008.
- [14] FAG valjni ležaji, katalog, 2004.
- [15] Grafičko sučelje programskog sučelja za prikaz i pohranu tijeka odlučivanja i rezoniranja, Završni rad, Ivan Vadla, Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu, 2008.
- [16] INDEXING DESIGN KNOWLEDGE IN RELATIONAL DATABASE, D. Rohde, N. Pavković, and M. Aurisicchio, INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008, Dubrovnik - Croatia, May 19 - 22, 2008.
- [17] Identifying and supporting the knowledge needs of novice designers within the aerospace industry, S. AHMED & K. M. WALLACE, Journal of Engineering Design, 2004
- [18] Identifying Design Errors and Human Errors Using Multiclass Classifications, Sanghee Kim, Ken M. Wallace, International Conference On Engineering Design, Iced'07, 28 - 31 August 2007, Cite Des Sciences Et De L'industrie, Paris, France
- [19] INTRODUCING THE CAPTURE OF ARGUMENTATION-BASED DESIGN RATIONALE INTO INDUSTRIAL PRACTISE, Rob Bracewell, Ken Wallace, pdf. prezentacija, Cambridge Engineering Design Centre, 2007
- [20] IPA: Concepts and Application in Engineering, Jerzy Pokojski, Springer, 2004

-
- [21] Merged Ontology for Engineering Design –contrasting an empirical and a theoretical approach to develop engineering ontology, Saeema Ahmed, Mario Štorga, 2008., članak u pripremi za objavljivanje u časopisu „AI EDAM“ (Artificial Intelligence for Engineering Design, Analysis and Manufacturing), Cambridge Journals
- [22] One-Dimensional Selections for Feature-Based Data Exchange, Ari Rappoporty, Steven Spitzz, Michal Etzionx, <http://www.cs.huji.ac.il/~arir/lds.pdf>
- [23] Prompting Designers to Design, Saeema Ahmed, INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006, Dubrovnik - Croatia, May 15 - 18, 2006.
- [24] Rastavljivi spojevi, Željko Cvirn, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb, 2000.
- [25] STRUKTURIRANJE BAZE ZNANJA U PROCESU KONSTRUIRANJA, disertacija, Zvonko Herold, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb, 1997.
- [26] Tehnički informacijski sustavi, Nenad Bojčetić, materijali za predavanja
- [27] Understanding the use and reuse of experience in engineering design, Saeema Ahmed, PhD-thesis, Cambridge University Engineering Department, 2001

16. SAŽETAK

Sustavi za upravljanje znanjem razvijaju se s ciljem ovladavanja velikom količinom znanja koje se generira tokom procesa razvoja proizvoda. Glavni problem više nije pohrana znanja, jer su resursi današnjih računala i više nego zadovoljavajući, već njegov dohvat. Kako bi se omogućilo pretraživanje zapisa, sustavi za upravljanje znanjem koriste određenu vrstu kategorizacije. S obzirom na dvosmislenost kategoriziranja apstraktnim pojmovima razvijenim razmatranjem područja inženjerskog znanja od najvišeg prema nižim razinama taksonomije, razmotrio se drugi pristup njezinom generiranju.

Predloženi pristup kreće od najniže razine hijerarhije stoga što su na toj razini pojmovi konkretniji i dobro definirani i time potpuno razumljivi inženjerima. Definiranjem najniže razine postavljen je temelj za izgradnju strukture hijerarhije grupiranjem definiranih pojmova u kategorije te definicijom viših razina.

Najniži pojmovi osnova su za definiciju paketa znanja (eng. Knowledge chunk) koji se sastoji od zapisa inženjerskog znanja vezanog za CAD model proizvoda. Sam postupak zapisivanja paketa znanja naziva se procedurom zapisa znanja čijim se provođenjem izgrađuje strukturirani zapis.

U odnosu na nestrukturirane zapise koje se naknadno moralo kategorizirati, ovakav način zapisivanja omogućuje visoku točnost pri pretraživanju jer je svaki dio strukture zapisa ujedno i indeks prema kojem se može pretraživati baza znanja u koju su spremljeni svi zapisi.

Ovaj rad predlaže 15 procedura zapisa inženjerskog znanja u obliku skupova mogućih (unaprijed definiranih) obrazloženja donesenih odluka te daje opis načina tehničke realizacije aplikacije koja će omogućiti visok stupanj automatizacije zapisa informacija čime se skraćuje vrijeme zapisa. Struktura baze znanja predlaže se u obliku jedne središnje baze sa zajedničkim elementima svih zapisa, te niza pojedinačnih manjih baza specifičnih za svaku vrstu „paketa“ znanja. Za tako predloženu strukturu baze znanja dane su smjernice sustava pretraživanja.

U konačnici iznesen je opći plan eksperimenta kojim bi se potvrdila primjenjivost opisanog predloženog sustava, a koji bi se provodio u dvije faze.

Ključne riječi: upravljanje znanjem, konstruiranje, zapis znanja, dohvaćanje znanja, indeksiranje znanja

17. SUMMARY

Knowledge management systems are being developed with the goal of managing large amounts of knowledge which is being generated during product development. Storing of knowledge is no longer a main issue, due to sufficient amount of storage that computers can provide, but rather its capture. In order to enable searching through stored data, knowledge management systems use some sort of categorization. Due to ambiguity of abstract categories developed through examination of engineering knowledge using top-down approach, a different approach towards taxonomy generation was taken.

Suggested approach starts at the lowest level of hierarchy due to elements on that level being more concrete and well defined and, by that, completely understandable to engineers. By defining the lowest level, a foundation was placed for creating hierarchy structure by grouping defined elements into categories and structuring higher levels.

Lowest elements are the basics for defining Knowledge chunks which consist of set of knowledge records connected to a CAD model of the product. The process of capturing knowledge chunk is called knowledge capture procedure, whose execution builds up a set of structured records.

In relation to non-structured records which have to be categorized afterwards, this way of recording enables high accuracy during search because each part of the record structure is at the same time an index by which knowledge base containing all records can be searched through.

This paper suggests 15 procedures for engineer knowledge capturing in the form of predefined possible decision explanation sets and gives a description of realization of application technical realization. This application will enable a high degree of knowledge capturing process automation which will gradually reduce the record generating time. Knowledge base structure is suggested in the form of one central base with common record elements and a certain number of individual smaller bases which are specific for each knowledge chunk. Guidelines of search engine for such knowledge base structure are given.

In the end, a general plan of experiment for confirming the applicability of the described suggested system, which is to be performed in two phases, is set out.

Key words: knowledge management, design, knowledge record, knowledge capture, knowledge indexing

18. PRILOG A - PROGRAMSKI KOD APLIKACIJE ZA DOHVAT IMENA IZ STABLA CAD MODELA

```
'#####  
'# ispunjavanje popisa odabira za ListBoxove      #  
'# i Comboboxove                                #  
'#####
```

```
Private Sub UserForm_Activate()      'upis vrijednosti u ListBox filtra odabira  
  
    ListBox3.Clear()                 'ocisti ListBox pri svakom pokretanju aplikacije  
    ListBox3.AddItem("Part")  
    ListBox3.AddItem("Znacajka")  
    ListBox3.AddItem("Ravnina")  
    ListBox3.AddItem("Točka")  
    ListBox3.AddItem("Pravac")  
  
    With ComboBox1                   'mogućnosti odabira iz ComboBoxa prefiksa  
        .Clear()                     'ocisti ComboBox pri svakom pokretanju  
        .AddItem("Prefiks_1_")  
        .AddItem("Prefiks_2_")  
        .AddItem("prefiks_3_")  
    End With  
  
    With ComboBox2                   'mogućnosti odabira iz ComboBoxa sufiksa  
        .Clear()                     'ocisti ComboBox pri svakom pokretanju  
        .AddItem("_Sufiks_1")  
        .AddItem("_Sufiks_2")  
        .AddItem("_Sufiks_3")  
    End With  
End Sub
```

```
'#####
'# Zamijeni naziv odabranog featurea sa prefiks + novi naziv + prefiks #
'# prebaci sve u Access i otvori Access bazu podataka i glavnu formu #
'#####
```

Private Sub Unos_u_bazu_Click()

Dim Tip_dok As String 'varijabla u koju se upisuje tip trenutno aktivnog dokumenta

Tip_dok = CATIA.GetWorkbenchId 'upisivanje trenutno aktivnog dokumenta

Select Case Tip_dok 'petlja mogućih slučajeva tip aktivnog dokumenta

'aktivni dokument je tipa Part *

Case "PrtCfg"

'Definiraj varijable za rad sa CATIA dokumentom

Dim partDocument1 As PartDocument 'definiraj varijablu partdokument

partDocument1 = CATIA.ActiveDocument 'zapisi otvoreni dokument u varijablu partdokument

Dim part1 As part 'definiraj varijablu part

part1 = partDocument1.part 'uvarijablu part1 zapisi trenutni otvoreni part

Dim bodies1 As Bodies 'definiraj varijablu bodya

bodies1 = part1.Bodies 'zapisi SVE bodye iz trenutnog parta

Dim body(1 To 50) As body 'definiraj array varijablu zasebnog bodya

Dim d As Integer 'definiraj varijablu brojaca

Dim hybridShapeFactory1 As HybridShapeFactory

hybridShapeFactory1 = part1.HybridShapeFactory

Dim originElements1 As OriginElements

originElements1 = part1.OriginElements

```
Dim hybridShapePlaneExplicit1 As HybridShapePlaneExplicit
hybridShapePlaneExplicit1 = originElements1.PlaneXY 'elementi globalnog
koordinatnog sustava
```

```
Dim reference1 As Reference
reference1 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit1)
```

```
Dim hybridShapePlaneOffset1 As HybridShapePlaneOffset
hybridShapePlaneOffset1 = hybridShapeFactory1.AddNewPlaneOffset(reference1,
50.0#, False)
'petlja za upis zasebnih bodya u varijablu
For d = 1 To bodies1.Count 'petlja ide od prvog do ukupnog broja bodya u
partu
    body(d) = bodies1.Item(d) 'svaki body je jedan item iz varijable svih bodya
Next d
```

```
'Otvori vezu sa bazom
Dim MyConn As ADODB.Connection
MyConn = New ADODB.Connection
Dim sConnect As String
sConnect = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data
Source=\\MARS\Home\Studenti\mhoic\DIPLOMSKI\Database\BAZA_2b.mdb"
'connection string
```

```
MyConn.Open(sConnect) 'konekcija
```

```
'upis u tablicu PARTS pri upisu znanja vezanom uz feature u partu
```

```
Dim Rs_1 As ADODB.Recordset 'rekord set za upis u tablicu PARTS
Rs_1 = New ADODB.Recordset 'to ti je objekt
```

```
Dim SQL_1 As String
SQL_1 = "Select * from PARTS" 'prebacivanje na tablicu partova
Rs_1.Open(SQL_1, MyConn, adOpenKeyset, adLockOptimistic, adCmdText)
```

```

Dim partID As String      'varijabla u koju se sprema automatski generirani partID,
                          'unos se u tablice PART i FEATURES

Dim kontrola As Boolean   'logička varijabla za provjeru da li part već postoji u
bazi podataka

kontrola = True           'početna vrijednost logičke varijable
Dim i As Integer          'cjelobrojna varijabla rezultata usporedbe stringova,
moguće vrijednosti
                          'su u donjoj tablici

Rs_1.MoveFirst()         'pozicioniraj na pocetak tablice PARTS

'petlja za provjeru da li je part vec unesen u tablicu
Do While Rs_1.EOF = False 'petlja se vrti do kraja tablice
    i = StrComp(part1.Name, Rs_1.Fields("part name"), vbBinaryCompare) 'usporedi
trenutacni part i zapis u tablici PARTS
    If i = 0 Then          'ako su zapisi jednaki
        kontrola = False  'promijeni logičku varijablu
        partID = Rs_1.Fields("part ID") 'i u partID upisi partID
postojećeg zapisa
    End If                'kraj provjere
    Rs_1.MoveNext()       'prelazak na slijedeći slog
Loop                      'zavrsetak petlje

#####
'#                                     #
'#   i = StrComp("A", "B")             -1 (string1 is less than string2)   #
'#   i = StrComp("A", "A")             0 (string1 is equal to string2)     #
'#   i = StrComp("B", "A")             1 (string1 is greater than string2)  #
'#   Option Compare Binary              #
'#   i = StrComp("ABC", "abc", vbTextCompare)  0                          #
'#   Option Compare Text                 #
'#   i = StrComp("ABC", "abc", vbBinaryCompare) -1                       #
'#   Option Compare Text                 #
'#   i = StrComp("abc", "ABC", vbBinaryCompare) 1                         #
'#                                     #
#####

```

```

        If kontrola = True Then                                'ako je log. var. ostala True - nije nađen isti
part u bazi
            Rs_1.MoveLast()                                  'pomakni se na zadnji slog tablice PARTS
            partID = Rs_1.Fields("part ID")                   'upisi zadnji ID u partID
            Rs_1.AddNew()                                     'dodaj novi slog u tablicu PARTS
            Rs_1.Fields("part ID") = (partID + 1)             'partID novog je za jedan veći od
prijasnjeg
            Rs_1.Fields("part name") = part1.Name            'upisi naziv parta
            Rs_1.Fields("file path") = partDocument1.FullName 'upisi put do datoteke
            Rs_1.Update()                                     'update tablice PARTS
        End If

        'upis u tablicu FEATURES
        Dim c As Integer      'brojac featurea u bodyu
        Dim featID As String   'varijabla u koju se upisuje featID, primary key vrijednost
FEATURE tablice

        Dim Rs_2 As ADODB.Recordset                            'otvori novi recordset za vezu sa
tablicom FEATURES
        Rs_2 = New ADODB.Recordset

        Dim SQL_2 As String
        SQL_2 = "Select * from FEATURES"
        Rs_2.Open(SQL_2, MyConn, adOpenKeyset, adLockOptimistic, adCmdText)

        Dim obj As AnyObject      'definiraj objekt u kojem će se spremiti naziv featurea koji
se upisuje
        Dim obj_name As String     'varijabla za ime bodya kojem feature pripada
        Dim obj_body_name As String

        obj_name = TextBox2.Text
        obj_body_name = TextBox7.Text

        MsgBox(obj_name & " " & obj_body_name)

        Rs_2.MoveFirst()

```

```

Do While Rs_2.EOF = False
    i = StrComp(obj_name, Rs_2.Fields("feature name"), vbTextCompare)
    j = StrComp(partID, Rs_2.Fields("part ID"), vbTextCompare)
    k = StrComp(obj_body_name, Rs_2.Fields("body name"), vbTextCompare)
    If (i = 0) And (j = 0) And (k = 0) Then
        featID = Rs_2.Fields("feature ID")
        kontrola = False
        MsgBox("false")
    End If
    Rs_2.MoveNext()
Loop

If kontrola = True Then
    Rs_2.MoveLast()                'pomakni se na zadnji slog
    featID = Rs_2.Fields("feature ID")    'upisi featID iz zadnjeg upisanog featurea
    Rs_2.AddNew()                    'dodaj novi slog u tablicu
    Rs_2.Fields("feature ID") = (featID + 1)    'featID novog je za 1 veći od prijašnjeg
    featID = Rs_2.Fields("feature ID")
    Rs_2.Fields("feature name") = obj_name    'unesi ime featurea
    Rs_2.Fields("part ID") = partID           'unesi partID koji definira kojem partu
    pripada feature
    Rs_2.Fields("body name") = obj_body_name
    Rs_2.Update()                    'update tablice
End If

'upis u tablicu knowledge chunkova kad se unosi zapis vezan uz feature u partu
Dim Rs_3 As ADODB.Recordset        'definiraj rekord set za komunikaciju
sa KNOWLEDGE_CHUNK
Rs_3 = New ADODB.Recordset          'to ti je objekt

Dim SQL_3 As String
SQL_3 = "Select * from KNOWLEDGE_CHUNKS"    'prebacivanje na tablicu
KNOWLEDGE_CHUNKS
Rs_3.Open(SQL_3, MyConn, adOpenKeyset, adLockOptimistic, adCmdText)

```

Dim chunkID As String

'varijabla za spremanje chunkID

Rs_3.MoveLast()

'pomakni se na zadnji slog

chunkID = Rs_3.Fields(0)

'spremi vrijednost zadnjeg upisanog chunkIDa

Rs_3.AddNew()

'dodaj novi slog

Rs_3.Fields(0) = (chunkID + 1)

'chunkID je za 1 veći od prošlog

Rs_3.Fields(1) = featID

'upisi featID featurea na koji se odnosi zapis

Rs_3.Fields(2) = partID

'upisi partID parta na koji se odnosi zapis

Rs_3.Update()

'update tablice KNOWLEDGE_CHUNKS

MsgBox("Podaci uneseni")

'javi korisniku da su podaci uneseni

MyConn.Close()

'zatvori vezu prema bazi

part1.Update()

'update aktivnog parta

* Aktivni dokument je tipa Product *

'Case "Assembly"

Case "PrsConfiguration"

'MsgBox ("Sklop")

'definiraj varijable za izmjenu podataka sa CATIA dokumentom

Dim productDocument1 As ProductDocument

'definiraj dokument sklopa

productDocument1 = CATIA.ActiveDocument

'dokument sklopa je aktivni

dokument

Dim product As product

'definicija varijable sklopa

product = productDocument1.product

'varijabla sklopa je sklop aktivnog

dokumenta

'Otvori vezu sa bazom

Dim MyConn_2 As ADODB.Connection

MyConn_2 = New ADODB.Connection

Dim sConnect_2 As String

sConnect_2 = "Provider=Microsoft.Jet.OLEDB.4.0;" & _

```

        "Data
Source=\\MARS\Home\Studenti\mhoic\DIPLOMSKI\Database\BAZA_2b.mdb"
        'connection string

MyConn_2.Open(sConnect_2)                'konekcija

        'Upis u tablicu PRODUCTS
Dim Rs_4 As ADODB.Recordset                'rekord set za upis u tablicu
PRODUCTS
Rs_4 = New ADODB.Recordset                'to ti je objekt

Dim SQL_4 As String
SQL_4 = "Select * from PRODUCTS"          'prebacivanje na tablicu
producta
Rs_4.Open(SQL_4, MyConn_2, adOpenKeyset, adLockOptimistic, adCmdText)

Dim productID As String                    'varijabla u koju se sprema automatski generirani
partID,
        'unosi se u tablice PART i FEATURES

Dim kontrola_2 As Boolean                    'logička varijabla za provjeru da li product već
postoji u bazi podataka
kontrola_2 = True                          'početna vrijednost logičke varijable
        'Dim j As Integer                    'cjelobrojna varijabla rezultata usporedbe stringova,
moguće vrijednosti
Dim ime_1, ime_2 As String                  'su u donjoj tablici

Rs_4.MoveFirst()                          'pozicioniraj na pocetak tablice PRODUCTS

        'petlja za provjeru da li je product vec unesen u tablicu
Do While Rs_4.EOF = False                  'petlja se vrti do kraja tablice
    j = StrComp(product.Name, Rs_4.Fields("product name").OriginalValue,
vbBinaryCompare) 'usporedi trenutacni part i zapis u tablici PARTS
    If j = 0 Then                          'ako su zapisi jednaki
        kontrola_2 = False                'promijeni logičku varijablu

```

```

        productID = Rs_4.Fields(0)                'i u partID upisi partID postojećeg
zapisa
    End If                                        'kraj provjere
    Rs_4.MoveNext()                             'prelazak na slijedeći slog
Loop                                             'zavrsetak petlje

    If kontrola_2 = True Then                    'ako je log. var. ostala True - nije nađen isti
part u bazi
        Rs_4.MoveLast()                        'pomakni se na zadnji slog tablice PARTS
        productID = Rs_4.Fields(0)             'upisi zadnji ID u partID
        Rs_4.AddNew()                          'dodaj novi slog u tablicu PARTS
        Rs_4.Fields(0) = (productID + 1)        'partID novog je za jedan veći od
prijašnjeg
        Rs_4.Fields(1) = product.Name          'upisi naziv producta
        Rs_4.Fields(2) = productDocument1.FullName 'upisi put do datoteke
        Rs_4.Update()                          'update tablice PARTS
    End If

    'upis u tablicu PARTS kada se zapisuje zapis vezan uz part iz sklopa

    Dim Rs_5 As ADODB.Recordset                 'rekord set za upis u tablicu PARTS
    Rs_5 = New ADODB.Recordset                  'to ti je objekt

    Dim SQL_5 As String
    SQL_5 = "Select * from PARTS"               'prebacivanje na tablicu partova
    Rs_5.Open(SQL_5, MyConn_2, adOpenKeyset, adLockOptimistic, adCmdText)

    Dim partID_2 As String                      'varijabla u koju se sprema automatski generirani
partID,
    'unos se u tablice PART i FEATURES

    Dim kontrola_3 As Boolean                   'logička varijabla za provjeru da li part već postoji u
bazi podataka
    kontrola_3 = True                          'početna vrijednost logičke varijable

```

'Dim k As Integer 'cjelobrojna varijabla rezultata usporedbe stringova,
 moguće vrijednosti
 'su u donjoj tablici

```
#####
```

#		#	
'#	i = StrComp("A", "B")	-1 (string1 is less than string2)	#
'#	i = StrComp("A", "A")	0 (string1 is equal to string2)	#
'#	i = StrComp("B", "A")	1 (string1 is greater than string2)	#
'#	Option Compare Binary		#
'#	i = StrComp("ABC", "abc", vbTextCompare)	0	#
'#	Option Compare Text		#
'#	i = StrComp("ABC", "abc", vbBinaryCompare)	-1	#
'#	Option Compare Text		#
'#	i = StrComp("abc", "ABC", vbBinaryCompare)	1	#

```
#####
```

```

Rs_5.Fields("part ID") = (partID_2 + 1)      'partID novog je za jedan veći od
prijašnjeg
Rs_5.Fields("part name") = TextBox2.Text    'upisi naziv parta
Dim q As Integer
'petlja prolazi kroz stablo i uspoređuje nazive partova sa odabito iz TextBox2.
Kad nade isto ime parta, u
'tablicu parts, polje file path, upisuje put do datoteke
For q = 1 To product.Products.Count
    i = StrComp(product.Products.Item(q).PartNumber, TextBox2.Text,
vbBinaryCompare)
    If i = 0 Then
        Rs_5.Fields("file path") =
product.Products.Item(q).ReferenceProduct.Parent.FullName
    End If
Next q
Rs_5.Update()      'update tablice PARTS
End If

'upis u tablicu KNOWLEDGE_CHUNKS kad se upisuje zapis vezan uz part u sklopu
Dim Rs_6 As ADODB.Recordset      'definiraj rekord set za komunikaciju
sa KNOWLEDGE_CHUNK
Rs_6 = New ADODB.Recordset      'to ti je objekt

Dim SQL_6 As String
SQL_6 = "Select * from KNOWLEDGE_CHUNKS"
Rs_6.Open(SQL_6, MyConn_2, adOpenKeyset, adLockOptimistic, adCmdText)

Dim chunkID_2 As String      'varijabla za spremanje chunkID

Rs_6.MoveLast()      'pomakni se na zadnji slog
chunkID_2 = Rs_6.Fields("chunk ID")      'spremi vrijednost zadnjeg upisanog
chunkIDa
Rs_6.AddNew()      'dodaj novi slog
Rs_6.Fields("chunk ID") = (chunkID_2 + 1)      'chunkID je za 1 veći od prošlog
Rs_6.Fields("part ID") = partID_2      'upisi partID parta na koji se odnosi zapis
Rs_6.Fields("product ID") = productID

```

```
Rs_6.Update() 'update tablice KNOWLEDGE_CHUNKS
```

MsgBox("Podaci uneseni") 'javi korisniku da su podaci uneseni

MyConn_2.Close() 'zatvori vezu prema bazi

```
product.Update()           'update aktivnog producta
```

* Aktivni dokument je tipa Drawing *

Case "Drw"

```
MsgBox("Crtez")
```

End Select

'Pokreni Access

```
Dim App As Access.Application
```

On Error Resume Next

```
App = GetObject(, "Access.Application")
```

```
If Err.Number <> 0 Then 'ako Access nije
    odignut
```

```
App = CreateObject("Access.Application")
```

App.Visible = True 'postavi ga kao vidljivog

```
App.OpenCurrentDatabase("\\MARS\Home\Studenti\mhoic\DILOMSKI\Database\BAZA_2b.mdb")
```

```
'App.DoCmd.Maximize
```

Else 'ako je Access otvoren

App.DoCmd.Close() 'zatvori ga

```
App = CreateObject("Access.Application")
```

```
App.Visible = True
```

```
App.OpenCurrentDatabase("\\MARS\Home\Studenti\mhoic\DIPLOMSKI\Database\BAZA_2b.mdb")
```

```
    'App.DoCmd.Maximize
```

```
End If
```

```
Err.Number = 0
```

```
Exit_Command3_Click:
```

```
Exit Sub          'izadi iz programa
```

```
Err_Command3_Click:
```

```
MsgBox(Err.Description)    'ispisi gresku
```

```
Resume Exit_Command3_Click    ' vrati na izlaz
```

```
End Sub
```

```
'#####
```

```
'# Prebaci odabrani feature iz popisa svih featurea #
```

```
'#####
```

```
Private Sub Prebaci_odabrani_Click()
```

```
Dim file_type As String
```

```
file_type = CATIA.GetWorkbenchId
```

```
Select Case file_type
```

```
    '*****
```

```
    'aktivni dokument je tipa Part *
```

```
    '*****
```

```
Case "PrtCfg"
```

```

If ListBox2.ListIndex = -1 Then Exit Sub 'ako nije nista odabrano, prekini program

'izdvajanje naziva featurea od naziva bodya
Dim ilen As Integer      'varijabla za dužinu riječi
Dim i As Integer
Dim body_name, feat_name As String    'varijabla za rijec sa promijenjenim prvim
slovom
Dim lasc(100)            'broj znaka po ascii kodu

ilen = Len(ListBox2.Text)    'duljina riječi

For i = 1 To ilen          'petlja za upis svakog slova u array
    lasc(i) = Asc(Mid(ListBox2.Text, i, 1))
Next i

body_name = ""            'brisanje sadržaja varijable nove riječi
feat_name = ""

Dim zagrada As Boolean    'logička varijabla kojom se određuje razmak koji dijeli
naziv bodya i featurea
zagrada = False          'za početak name razmaka

For i = 1 To ilen        'petlja prolazi kroz cijeli array u kojem se nalaze sva slova i
razmak

    If zagrada = False And lasc(i) <> 91 And lasc(i) <> 93 Then          'ako se
nije došlo do razmaka
        body_name = body_name & Chr(lasc(i))    'dodaj slovo u naziv bodya
    End If

    If zagrada = True Then          'ako se došlo do razmaka, dodaj slovo nazivu
featurea
        feature_name = feature_name & Chr(lasc(i)) 'osim ako je znak razmak
    End If

```

```

        If lasc(i) = 93 Then      'ako je znak jednak razmaku (ASCII kod 32)
            zagrada = True      'razmak postaje istinit, tj. došlo se do razmaka
        End If
    Next i

    TextBox1 = LTrim(feature_name)      'prebaci nazive u odgovarajuće
textboxove
    TextBox7 = body_name                'LTrim odreže razmak na početku naziva
featurea koji je
    TextBox2 = LTrim(feature_name)      'preuzet iz listboxa

If TextBox1.Text = "" Then
    MsgBox("odabran body")
    TextBox1 = "odabran body"
    TextBox2 = "odabran body"
End If

*****

'* Aktivni dokument je tipa Product *
*****

Case "PrsConfiguration"

    Dim part_name, numb_part As String    'varijabla za naziv i broj partova u sklopu
    ilen = Len(ListBox2.Text)             'duljina zapisa u listboxu

    For i = 1 To ilen                    'petlja za upis svakog slova u array
        lasc(i) = Asc(Mid(ListBox2.Text, i, 1))
    Next

    part_name = ""                      'brisanje sadržaja varijable nove riječi
    numb_part = ""

    'Dim razmak As Boolean              'logička varijabla kojom se određuje razmak koji dijeli
broj i naziv partova

```

```

razmak = False           'za početak nema razmaka

For i = 1 To ilen         'petlja prolazi kroz cijeli array u kojem se nalaze sva slova i
razmak
    If lasc(i) = 32 Then   'ako je znak jednak razmaku (ASCII kod 32)
        razmak = True     'razmak postaje istinit, tj. došlo se do razmaka
    End If

    If razmak = False And lasc(i) <> 91 And lasc(i) <> 93 Then 'ako se
nije došlo do razmaka
        numb_part = numb_part & Chr(lasc(i)) 'dodaj slovo u naziv bodya
    End If

    If razmak = True Then 'ako se došlo do razmaka, dodaj slovo nazivu featurea
        part_name = part_name & Chr(lasc(i)) 'osim ako je znak razmak
    End If
Next i

TextBox1 = LTrim(part_name) 'prebaci nazive u odgovarajuće textboxove
TextBox7 = numb_part
TextBox2 = LTrim(part_name)

*****
'* Aktivni dokument je tipa Drawing *
*****

Case "Drw"
    MsgBox("Crtez")
End Select
End Sub

#####
'# Izlaz iz aplikacije #
#####

Private Sub Izlaz_Click()
    End
End Sub

```

```
'#####
```

```
# Ulovi feature #
```

```
'#####
```

```
Private Sub odaberi_feature_Click()
```

```
Dim Tip_dok As String 'varijabla za upis tipa aktivnog dokumenta
```

```
Tip_dok = CATIA.GetWorkbenchId 'upisi tip aktivnog dokumenta
```

```
Select Case Tip_dok 'odredivanje koda za izvršavanje na prema tipu  
    'aktivnog dokumenta
```

```
*****
```

```
* Aktivni dokument je tipa Part *
```

```
*****
```

```
Case "PrtCfg"
```

```
    'Definiraj varijable za rad part dokumentom
```

```
    'JEL MI OVO UOPCE TREBA???
```

```
Dim partDocument1 As PartDocument
```

```
partDocument1 = CATIA.ActiveDocument 'aktivni dokument je part dokument
```

```
Dim part1 As part
```

```
part1 = partDocument1.part 'part u aktivnom dokumentu je part na kojem se  
radi
```

```
'Dim bodies1 As Bodies
```

```
'Set bodies1 = part1.Bodies 'bodyi aktivnog parta se uzimaju za razmatranje
```

```
'Dim body1 As body
```

```
'Set body1 = bodies1.Item("PartBody")
```

```

'Dim shapes1 As Shapes
'Set shapes1 = body1.Shapes

Dim LocSel 'As Selection
LocSel = partDocument1.Selection

Dim status As String
Dim InputObjectType(0)

Dim filter As String

If ListBox3.Text <> "" Then
    LocSel.Clear()
    'filtriranje mogućnosti odabira iz Catiaainog prozora prema odabiru korisnika
    'mogućnosti odabira su slijedeće
    'Part - Part
    'Znacajka - shape
    'Ploha - Face

    Select Case ListBox3.Text
        Case "Part"
            filter = "Part"
        Case "Znacajka"
            filter = "Shape"
        Case "Ravnina"
            'CATIA pod pojmom "Plane" podrazumijeva
            konstrukcijske elemente
            filter = "Plane" 'Mogu se odabrati iz stabla i iz modela
        Case "Točka"
            'pod tockom se podrazumijeva konstrukcijski element. Bira
            se na
            filter = "Point" 'modelu ili u stablu
        Case "Pravac"
            filter = "Line"
        Case "body"

    End Select

    InputObjectType(0) = filter

```

```

Me.Hide()

status = LocSel.SelectElement2(InputObjectType, "Select the face", False)
If (status = "cancel") Then Exit Sub

MsgBox(LocSel.Item(1).Value.Name)

TextBox1 = LocSel.Item(1).Value.Name
TextBox7 = LocSel.Item(1).Value.Name
Me.Show()
End If

part1.Update()

*****
* aktivni dokument je tipa Product *
*****

Case "PrsConfiguration"
'Case "Assembly"
MsgBox("Sklop")

'definiraj varijable za izmjenu podataka sa CATIA dokumentom
Dim productDocument1 As ProductDocument      'definiraj dokument sklopa
productDocument1 = CATIA.ActiveDocument      'dokument sklopa je aktivni
dokument

Dim product As product                        'definicija varijable sklopa
product = productDocument1.product           'varijabla sklopa je sklop aktivnog
dokumenta

Dim Sel_part 'As Selection
Sel_part = productDocument1.Selection

Dim InputPart(0)
Dim status1

```

```

    ListBox3.Clear()
    MsgBox("Choose part or subassembly")

    InputPart(0) = "Product"

    Me.Hide()

    status1 = Sel_part.SelectElement2(InputPart, "Select the face", False)
    If (status = "cancel") Then Exit Sub

    MsgBox(Sel_part.Item(1).Value.PartNumber)

    TextBox1 = Sel_part.Item(1).Value.PartNumber
    TextBox2 = Sel_part.Item(1).Value.PartNumber
    Dim broj, x, i, w As Integer

    broj = 0
    For x = 1 To product.Products.Count
        i = StrComp(product.Products.Item(x).PartNumber,
Sel_part.Item(1).Value.PartNumber, vbBinaryCompare)
        If i = 0 Then
            w = broj
            broj = w + 1
        End If
    Next x

    TextBox7 = broj

    Me.Show()

    Case "Drw"
        MsgBox("Crtez")
    End Select

End Sub
#####

```

'# Unos imena i prezimena u bazu #

'#####

Private Sub Unesi_u_bazu_Click()

Dim MyConn As ADODB.Connection

MyConn = New ADODB.Connection

Dim sConnect As String

sConnect = "Provider=Microsoft.Jet.OLEDB.4.0;" & _

"Data

Source=\\MARS\Home\Studenti\mhoic\DIPLOMSKI\Database\BAZA_2b.mdb"

'connection string

MyConn.Open(sConnect)

'konekcija

Dim Rs As ADODB.Recordset

'rekord set

Rs = New ADODB.Recordset

'to ti je objekt

Dim SQL As String

SQL = "Select * from DESIGNERS"

Rs.Open(SQL, MyConn, adOpenKeyset, adLockOptimistic, adCmdText)

'Rs.Open "Select * from Customers", objConn, adOpenKeyset, adLockOptimistic,
adCmdTable

Dim designerID As Integer

Rs.MoveLast()

'pomakni se na zadnji slog

designerID = Rs.Fields(0)

Rs.AddNew()

'dodaj novi slog

Rs.Fields(0) = (designerID + 1)

Rs.Fields(1) = TxBx_ime

'u prvi stupac upisi podatke iz Text boxa TxBx_ime

Rs.Fields(2) = TxBx_prezime

'u drugi stupac upisi podatke iz Text boxa TxBx_prezime

Rs.Update()

'update tablice

MyConn.Close()

'zatvori vezu prema bazi

MsgBox("Podaci uneseni")

'javi korisniku da su podaci uneseni

End Sub

```
'#####  
# Nadi podatke za dano ime #  
'#####
```

Private Sub Trazi_Click()

```
Dim MyConn As ADODB.Connection  
MyConn = New ADODB.Connection  
Dim sConnect As String  
sConnect = "Provider=Microsoft.Jet.OLEDB.4.0;" & _  
           "Data  
Source=\\MARS\Home\Studenti\mhoic\DIPLOMSKI\Database\BAZA_2b.mdb"  
'connection string  
  
MyConn.Open(sConnect)           'konekcija  
Dim Rs As ADODB.Recordset       'rekord set  
Rs = New ADODB.Recordset       'to ti je objekt  
  
Dim SQL As String  
SQL = "Select * from DESIGNERS"  
Rs.Open(SQL, MyConn, adOpenKeyset, adLockOptimistic, adCmdText)  
'Rs.Open "Select * from Customers", objConn, adOpenKeyset, adLockOptimistic,  
adCmdTable  
  
Dim flag As Boolean             'logicka varijabla za provjeru  
Dim ime As String              'varijabla za usporedbu  
  
flag = False                   'postavljanje default stanja logičke varijable  
  
Rs.MoveFirst()                 'pomak na pocetak tablice radi pretrazivanja  
ime = TxBx_trazi_ime           'unos imena trazene osobe u String varijablu  
'petlja za prolazak kroz bazu  
  
Dim i As Integer
```

```

'#####'
#
'# i = StrComp("A", "B") -1 (string1 is less than string2) #
'# i = StrComp("A", "A") 0 (string1 is equal to string2) #
'# i = StrComp("B", "A") 1 (string1 is greater than string2) #
'# Option Compare Binary #
'# i = StrComp("ABC", "abc", vbTextCompare) 0 #
'# Option Compare Text #
'# i = StrComp("ABC", "abc", vbBinaryCompare) -1 #
'# Option Compare Text #
'# i = StrComp("abc", "ABC", vbBinaryCompare) 1 #
'#####'

```

```

Do While Rs.EOF = False 'ponavlja petlju do zavrsetka tablice
    i = StrComp(ime, Rs.Fields("name"), vbTextCompare)
    If i = 0 Then
        'If ime = Rs.Fields("name") Then 'ako trazeno ime odgovara imenu u tom slogu
        flag = True 'kontrolna varijabla se mijenja u True
        GoTo van 'idi na prceduru van - izlazak iz petlje
    End If
    Rs.MoveNext() 'pomak na slijedeci slog
Loop 'kraj petlje

```

van: 'procedura za ispis rezultata

```

If flag Then 'ako je log. var. postala True, nadeno je ime
    Lbl_ime = Rs.Fields(1)
    Lbl_prezime = Rs.Fields(2)
    Lbl_redni_br = Rs.AbsolutePosition 'ispis podataka u Labele
Else 'ako je log. var. False (ostala ista), nije nadeno ime
    Lbl_ime = "Osoba"
    Lbl_prezime = "nije"
    Lbl_redni_br = "upisana." 'ispis poruke
End If

```

```

MyConn.Close() 'zatvori vezu prema bazi

```

End

Sub

```
'#####  
'# Ispisi cijelu Table1 - imena #  
'#####
```

```
Private Sub Ispisi_tablicu_Click()
```

```
    Dim MyConn As ADODB.Connection  
    MyConn = New ADODB.Connection  
    Dim sConnect As String  
    sConnect = "Provider=Microsoft.Jet.OLEDB.4.0;" & _  
        "Data  
Source=\\MARS\Home\Studenti\mhoic\DIPLOMSKI\Database\BAZA_2b.mdb"  
    'connection string  
  
    MyConn.Open(sConnect)                'konekcija  
    Dim Rs As ADODB.Recordset            'rekord set  
    Rs = New ADODB.Recordset             'to ti je objekt  
  
    Dim SQL As String  
    SQL = "Select * from DESIGNERS"  
    Rs.Open(SQL, MyConn, adOpenKeyset, adLockOptimistic, adCmdText)  
    'Rs.Open "Select * from Customers", objConn, adOpenKeyset, adLockOptimistic,  
adCmdTable  
  
    Rs.MoveFirst() 'pozicioniraj na pocetak tablice  
    ListBox1.Clear() 'izbrisi sve u ListBoxu  
  
    'petlja za prolazak kroz tablicu  
  
    Do While Rs.EOF = False 'prolazi kroz tablicu do zadjeg sloga  
        ListBox1.AddItem(Rs.Fields(1) & " " & Rs.Fields(2) & " " & Rs.AbsolutePosition) 'ispis  
sloga u ListBox  
        Rs.MoveNext() 'prijedi na slijedeci slog  
    Loop
```

End Sub

```
'#####  
'# Pokrece Access #  
'#####
```

Private Sub Pokreni_Access_Click()

'Pokreni Access

Dim App As Access.Application

On Error Resume Next

App = CreateObject("Access.Application")

'otvori Access

App.Visible = True

'postavi ga kao vidljivog

App.OpenCurrentDatabase("\\MARS\Home\Studenti\mhoic\DIPLOMSKI\Database\BAZA_2b.mdb")

App.DoCmd.Maximize()

Err.Number = 0

Exit_Command3_Click:

Exit Sub 'izadi iz programa

Err_Command3_Click:

MsgBox(Err.Description) 'ispisi gresku

Resume Exit_Command3_Click ' vrati na izlaz

End Sub

```
'#####  
' Za upisani dio imena ispisi sve odgovarajuće feature #  
'#####
```

Private Sub Pregledaj_imena_Click()

Dim Tip_dok As String
dokumenta

'varijabla u koju se upisuje tip trenutno aktivnog

Tip_dok = CATIA.GetWorkbenchId

'upisivanje trenutno aktivnog dokumenta

Select Case Tip_dok

'petlja mogućih slučajeva tip aktivnog dokumenta

'aktivni dokument je tipa Part *

Case "PrtCfg"

'definiraj varijable za izmjenu podataka sa CATIA dokumentom

Dim partDocument1 As PartDocument 'definiraj varijablu partdokument

partDocument1 = CATIA.ActiveDocument 'zapisi otvoreni dokument u varijablu partdokument

Dim part1 As part 'definiraj varijablu part

part1 = partDocument1.part 'uvarijablu part1 zapisi trenutni otvoreni part

Dim bodies1 As Bodies 'definiraj varijablu bodya

bodies1 = part1.Bodies 'zapisi SVE bodye iz trenutnog parta

Dim body(1 To 50) As body 'definiraj array varijablu zasebnog bodya

Dim d As Integer 'definiraj varijablu brojaca

Dim hybridShapeFactory1 As HybridShapeFactory

hybridShapeFactory1 = part1.HybridShapeFactory

Dim originElements1 As OriginElements

originElements1 = part1.OriginElements

Dim hybridShapePlaneExplicit1 As HybridShapePlaneExplicit

hybridShapePlaneExplicit1 = originElements1.PlaneXY 'elementi globalnog koordinatnog sustava

Dim reference1 As Reference

reference1 = part1.CreateReferenceFromObject(hybridShapePlaneExplicit1)

Dim hybridShapePlaneOffset1 As HybridShapePlaneOffset

```
hybridShapePlaneOffset1 = hybridShapeFactory1.AddNewPlaneOffset(reference1,  
50.0#, False)
```

```
For d = 1 To bodies1.Count      'petlja ide od prvog do ukupnog broja bodya u  
partu
```

```
    body(d) = bodies1.Item(d)    'svaki body je jedan item iz varijable svih bodya  
Next d
```

```
Dim obj As AnyObject    'definiraj objekt
```

```
Dim c As Integer        'brojac
```

```
Dim Trazi As String     'varijabla u koju se upisuje trazeni string
```

```
Dim flag As Boolean      'logicka varijabla za provjeru
```

```
flag = True              'postavljanje logičke varijable na početnu vrijednost
```

```
Trazi = Trazena_oznaka    'varijabla za pretrazivanje uzima string koji je upisao  
korisnik
```

```
'promjena malog u veliko slovo i obrnuto
```

```
Dim ilen As Integer      'varijabla za dužinu riječi
```

```
Dim i As Integer
```

```
Dim new_line As String   'varijabla za rijec sa promijenjenim prvim slovom
```

```
Dim lasc(100)            'broj znaka po ascii kodu
```

```
ilen = Len(Trazi)        'duljina riječi
```

```
For i = 1 To ilen        'petlja za upis svakog slova u array
```

```
    lasc(i) = Asc(Mid(Trazi, i, 1))
```

```
Next
```

```
new_line = ""            'brisanje sadržaja varijable nove riječi
```

```
Select Case lasc(1)
```

```
    Case 97 To 122        'ako riječ počinje velikim slovom
```

```
        lasc(1) = lasc(1) - 32    'promijeni u malo slovo
```

```
    Case 65 To 90        'ako riječ počinje malim slovom
```

```

        lasc(1) = lasc(1) + 32      'promijeni u veliko slovo
End Select

For i = 1 To ilen
    new_line = new_line & Chr(lasc(i))    'izradi novu riječ sa promijenjenim slovom
Next i

ListBox2.Clear()

'petlja za ispis svih bodya i featurea u tim bodyima
For d = 1 To bodies1.Count
    ListBox2.AddItem(body(d).Name)
    For c = 1 To body(d).Shapes.Count      'od 1 do ukupnog broja featurea
        obj = body(d).Shapes.Item(c)      'u obj spremi trenutacni feature

        If InStr(obj.Name, Trazi) Or InStr(obj.Name, new_line) Then      'ako u imenu
objekta ima dio jednak varijabli Trazi
            ListBox2.AddItem "[" & body(d).Name & "]" " & obj.Name)
            flag = False      'kontrolna varijabla se mijenja u False
        End If

        obj = body(d).HybridShapes.Item(c)    'u obj spremi geometrijske elemente
        If InStr(obj.Name, Trazi) Or InStr(obj.Name, new_line) Then      'ako u
imenu dijela ima dio jednak unesenom filtru
            ListBox2.AddItem "[" & body(d).Name & "]" " & obj.Name)      'dodaj
element na popis
            flag = False
        End If
    Next c
    ListBox2.AddItem(" ")
Next d

Label6.Caption = "Feature name"
Label7.Caption = "Body name"
Label8.Caption = partDocument1.Name
Label9.Caption = partDocument1.FullName

```

```

If flag Then                                'ako je log. var. postala True, nista nije nadeno
    MsgBox("Nema")
End If

*****

'* Aktivni dokument je tipa Product *
*****

Case "PrsConfiguration"
    'MsgBox ("Sklop")

    'definiraj varijable za izmjenu podataka sa CATIA dokumentom
    Dim productDocument1 As ProductDocument    'definiraj dokument sklopa
    productDocument1 = CATIA.ActiveDocument    'dokument sklopa je aktivni
dokument

    Dim product As product                    'definicija varijable sklopa
    product = productDocument1.product        'varijabla sklopa je sklop aktivnog
dokumenta

    Dim a As Integer
    'a = product.Products.Count                'broj partova i podsklopova

    Label6.Caption = "Naziv parta"
    Label7.Caption = "broj komada"
    Label8.Caption = productDocument1.Name
    Label9.Caption = product.Name
    Label10.Caption = "#"
    Label11.Caption = "part name"

    ListBox2.Clear()

Dim part(1 To 500) As PartComp
    Dim x, w, p, z As Integer
Dim part_list(1 To 500) As String
Dim part_numb(1 To 500) As Integer

```

Dim new_part As Boolean

'petlja za određivanje broja komada istog parta

For x = 1 To product.Products.Count 'prolazi kroz sve partove u sklopu

part(x) = product.Products.Item(x) 'postavlja trenutacni part u varijablu

new_part = True 'po defaultu to je novi part

For z = 1 To (x - 1) 'product.Products.Count 'petlja za prolazak kroz dosad
pregledane partove

i = StrComp(part(x).PartNumber, part(z).PartNumber, vbBinaryCompare) 'ako je
prije pregledan part sa istim imenom

If i = 0 Then

new_part = False 'ovo nije novi part

w = part_numb(z) 'dosad je bilo w ovakvih partova

part_numb(z) = w + 1 'odsad je još jedan više

End If

Next z

If new_part = True Then 'ako nije pronađen isti part

part_list(x) = part(x).PartNumber 'u popis partova dodati novi

part_numb(x) = 1 'ima zasd samo jedan

End If

Next x

For x = 1 To 500 'prolazak kroz sve partove

If part_list(x) <> "" Then 'ako u listi nije prazno mjesto (prazna mjesta su
tamo gdje se ponavljaju)

ListBox2.AddItem "[" & part_numb(x) & "]" & part_list(x) 'ispisi naziv parta

End If

Next x

```
#####
#
'# i = StrComp("A", "B") -1 (string1 is less than string2) #
'# i = StrComp("A", "A") 0 (string1 is equal to string2) #
'# i = StrComp("B", "A") 1 (string1 is greater than string2) #
'# Option Compare Binary #
'# i = StrComp("ABC", "abc", vbTextCompare) 0 #
'# Option Compare Text #
'# i = StrComp("ABC", "abc", vbBinaryCompare) -1 #
'# Option Compare Text #
'# i = StrComp("abc", "ABC", vbBinaryCompare) 1 #
#####
```

```
*****
```

```
* Aktivni dokument je tipa Drawing *
```

```
*****
```

```
Case "Drw"
```

```
MsgBox("Crtez")
```

```
End Select
```

```
End Sub
```