

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Stjepan Križan

Zagreb, 2011.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

**IMPLEMENTACIJA KONSTRUKCIJSKOG
ZNANJA U RAČUNALNI 3D MODEL
PROIZVODA**

Mentor:

dr. sc. Nenad Bojčetić

Stjepan Križan

Zagreb, 2011.

Izjavljujem da sam ovaj rad izradio samostalno služeći se stečenim znanjem i navedenom literaturom.

Zahvaljujem se svome mentoru, dr. sc. Nenadu Bojčetiću na pomoći tokom izrade rada, te na ukazanom povjerenju da ću ovaj zadatak izvršiti na zadovoljavajući način i u roku.

Posebnu zahvalu dugujem svojoj obitelji i prijateljima na potpori koju su mi pružili tokom svih ovih godina mog školovanja.

Sažetak

Tema ovog rada je *Implementacija konstrukcijskog znanja u računalni 3D CAD model proizvoda*. Rad se bavi proučavanjem sustava temeljenim na znanju, metodologijama za izradu sustava temeljenim na znanju, mogućnosti implementacije inženjerskog znanja u model. Predmet rada je 3D model jedno-stupanjskog reduktora, na kojem će se integrirati konstrukcijsko znanje korišteno pri izradi 3D modela, određena ograničenja pri konstruiranju ovisna o: proračunskim veličinama, iskustvenim vrijednostima određenih parametara, dijelovima samog sklopa reduktora.

Detaljnije se analiziraju i opisuju mogućnosti modula „Knowledgeaware“ programskog paketa tvrtke Dessault Systems „CATIA V5“ sa ciljem implementacije inženjerskog znanja u model reduktora, radi brže i efikasnije izrade proizvoda, njegovog ponovnog korištenja, modificiranja modela, smanjivanje pogrešaka pri ponovnom konstruiranju.

Kazalo

Popis slika	6
1. Uvod	7
2. Inženjerstvo znanja	8
2.1 Osnove inženjerstva znanja	8
2.2 Sustavi temeljeni na znanju	9
2.2.1 Primjena sustava temeljenog na znanju (KBE sustava)	10
2.3 Metodologije za razvoj KBE sustava	11
2.3.1 Common KADS	11
2.3.2 MOKA	12
2.5 Komercijalni KBE sustavi	13
3. Softverski paket CATIA	14
3.1 Moduli za inženjerstvo temeljeno na znanju (Knowledgeware)	15
3.1.1 Parametri	15
3.1.2 Relacije	17
3.1.2.1 Formule	17
3.1.2.2 Konstrukcijske tablice (eng. design tables)	17
3.1.2.3 Pravila (eng. Rules)	18
3.1.2.4 Provjere (eng. check)	19
3.1.3 Reakcije (eng. reactions)	20
3.1.4 Upotreba funkcije „Set of Equations“	20
4. Upotreba modula „Knowledge“ na primjeru reduktora	22
4.1 Kućište	24
4.2 Poklopci	27
4.3 Distantni prsten	28
4.4 Zupčanici	28
4.5 Vratila	30
5. Zaključak	33
6. Literatura	34

Popis slika

Slika 1: Znanje organizacije	8
Slika 2: Sustav temeljen na znanju	9
Slika 3: Kvalitativna usporedba cijene razvoja	10
Slika 4: Šest koraka MOKA metodologije	12
Slika 5: Moduli softverskog paketa CATIA	14
Slika 6. Unutarnji i korisnički definirani parametri	16
Slika 7: Primjer formule	17
Slika 8: Primjer primjene pravila	18
Slika 9. Primjer primjene provjere	19
Slika 10: Primjer funkcije „Set of equations“	21
Slika 11: Model reduktora	23
Slika 12: Funkcija provjere pri promjeni debljine lima	24
Slika 13: Funkcija pravila pri promjeni promjera provrta (preporučeni vijak)	25
Slika 14: Informacijski prozor sa definiranim dimenzijama	26
Slika 15: Informacijski prozor sa definiranim dimenzijama poklopca	28
Slika 16: Korisnički definirani parametri	29
Slika 17: Upozorenje za prekoračen dozvoljeni maksimalni promjer vratila	30
Slika 18: Minimalni promjeri vratila integrirani u model iz tekstualne datoteke	30
Slika 19. Tekstualni prozor za definiranje novog promjera vratila	32

1. Uvod

Svaki proizvod tijekom svog životnog vijeka prolazi kroz mnogo faza; od samog planiranja proizvoda, koncipiranja, projektiranja sustava, konstruiranja, ispitivanja, poboljšavanja, te do samog lansiranja proizvoda. Tijekom tih faza sakupljaju se mnoga znanja, te je cilj svake tvrtke biti što efikasnija i samim time proizvoditi konkurentnije, kvalitetnije i jeftinije proizvode.

Inženjersko znanje je od velike važnosti pri razvoju proizvoda, te je bitno što korektnije prenijeti to znanje na sam proizvod i omogućiti njegovo ponovno korištenje.

Razvojem računalnih tehnologija otvorila su se vrata mnogim aplikacijama i programskim paketima sa ciljem jednostavnije izrade modela proizvoda, izrade tehničke dokumentacije, pohranjivanja u digitalni oblik, zapisivanja i integriranja znanja u sam 3D model proizvoda, radi referiranja, poboljšavanja i ponovnog korištenja. Samim time se smanjuje mogućnost pogreške radi ne stručnosti osobe koja radi modifikacije, daje se mogućnost automatizacije izrade proizvoda, te se dobiva na učinkovitosti pri izradi novog modela.

U ovom radu se razmatra pojam inženjerstva znanja, sustava temeljenih na znanju, te se detaljnije opisuju moduli bazirani na integriranju znanja u model proizvoda, programskog paketa tvrtke Dessault Systems „CATIA“. Kao primjer uzimamo 3D model jedno-stupanjskog reduktora na kojem ćemo integrirati inženjersko znanje.

2. Inženjerstvo znanja

2.1 Osnove inženjerstva znanja

Prema [1] inženjerstvo znanja se može definirati kao:

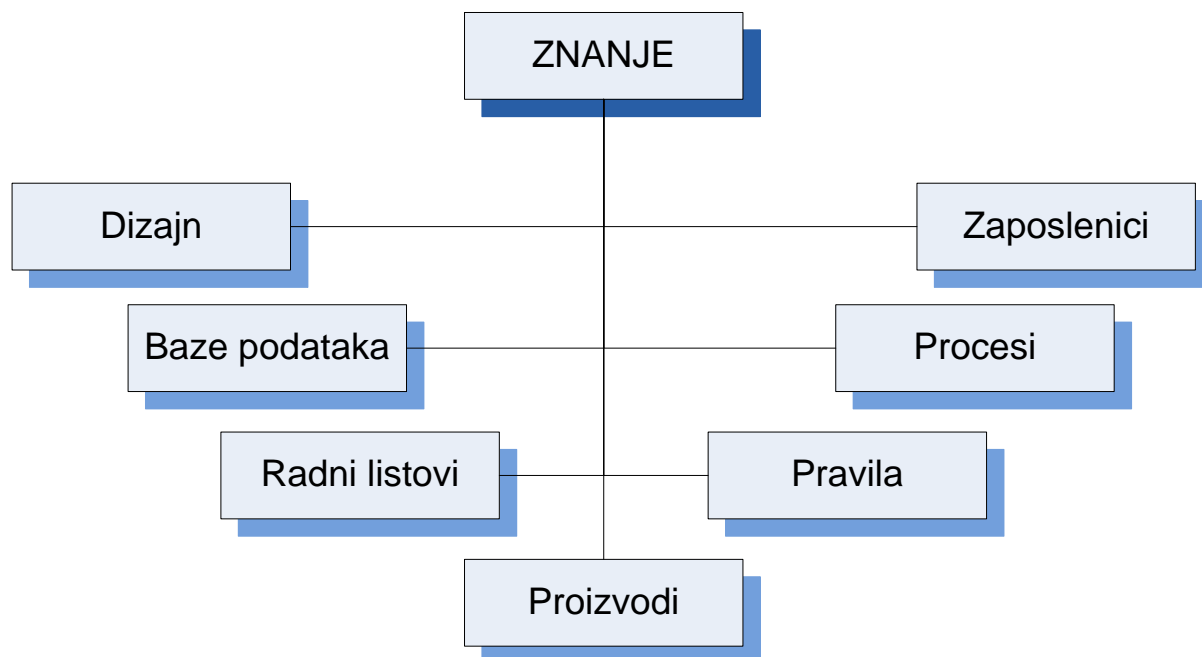
„Upotreba naprednih softverskih tehnologija za zapisivanje znanja u računalni sustav, te njegovo ponovno korištenje.“

Znanje je [2]:

- visoko-strukturirani oblik informacija
- potrebno za stručno razmišljanje
- segment raspodjele između ne-stručnjaka i stručnjaka
- potrebno za izvršavanje kompleksnih zadataka

Prema tom kontekstu znanje je aktivnost koja manipulira, transformira ili stvara rezultat, te djeluje kao mehanizam koji iz podataka i informacija stvara odluke i definirane postupke. Za shvaćanje mehanizma znanja svake osobe, moramo znati kako je to znanje strukturirano, tj. od kojih se komponenata sastoji i na koji način su one povezane zajedno.

U svakoj organizaciji, informacije u svakoj fazi razvoja proizvoda, cijelom životnom vijeku proizvoda, su zapisane u raznim formama, te se sustavima temeljenim na znanju prikupljaju i omogućuju ponovno korištenje.



Slika 1: Znanje organizacije

2.2 Sustavi temeljeni na znanju

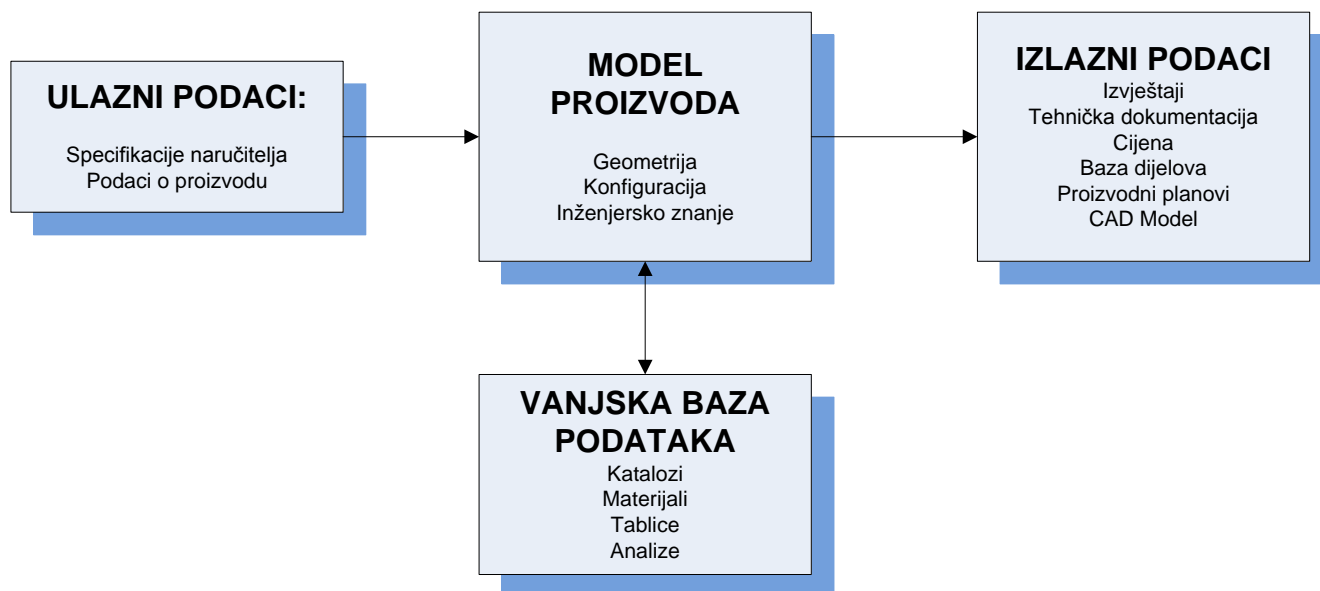
Sustav temeljen na znanju (eng. KBS - Knowledge based systems) se zasniva na korištenju umjetne inteligencije sa ciljem rješavanja kompleksnih problema, te sadrže široki spektar znanja i pravila, u svrhu rješavanja stvarnih problema, zamjenjujući ljudske stručnjake. Glavni cilj sustava temeljenih na znanju je sakupljanje znanja te omogućavanje njegovog ponovnog korištenja.

Jezgra sustava je model proizvoda gdje je sve to znanje zapisano, te se sastoji od znanja kako bi se određeni proizvod trebao razvijati, uzimajući u obzir razna pravila. (Slika 2.)

Dijeli se na:

- geometriju proizvoda: CAD sustavi omogućuju kreiranje parametarskih modela, što znači da sve ili samo neke dimenzije modela ovise o definiranim parametrima modela
- konfiguraciju proizvoda: omogućuje izbor dijelova od kojih se glavni sklop sastoji
- inženjerskog znanja: znanje koje moguće programirati u sam model

Ulazni podaci sustava su uglavnom specifikacije naručitelja, podaci o proizvodu, koje se unose kroz definirano sučelje, te nakon unosa u sustav daju razne izlazne podatke, koji su rezultati procesiranja ulaznih podataka.



Slika 2: Sustav temeljen na znanju

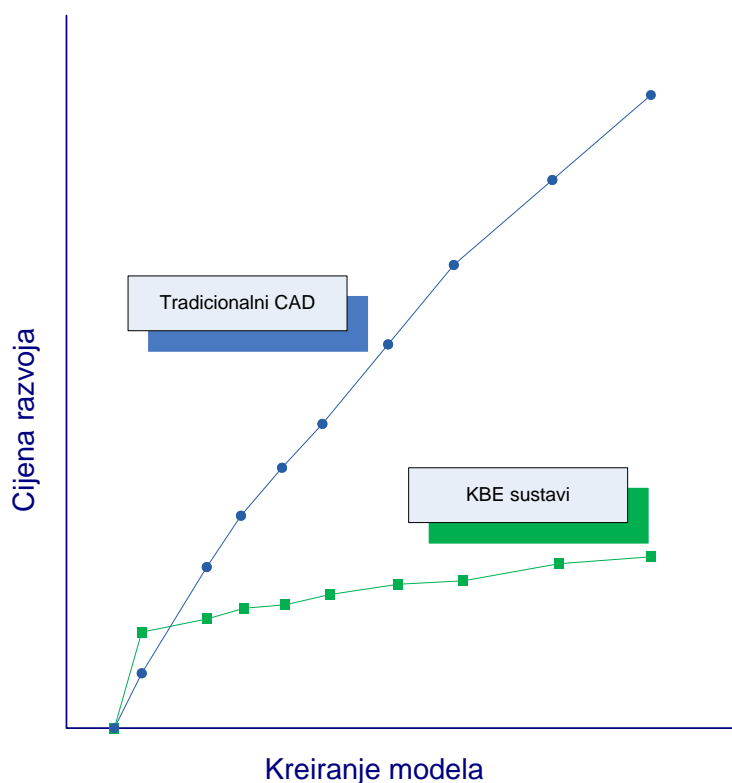
Vanjska baza podataka sadrži informacije o standardnim dijelovima, atributima materijala; potrebne za proizvod, te se ne smije zamijeniti sa sustavom temeljenom na znanju; koji je integriran u sam model proizvoda.

2.2.1 Primjena sustava temeljenog na znanju (KBE sustava)

KBE sustavi se mogu primijeniti za različite aplikacije vezane uz [2]:

- tumačenje
- dijagnosticiranje
- dizajniranje
- otklanjanje grešaka
- popravljavanje
- davanje uputa
- kontroliranje
- planiranje

KBE sustav za razliku od tradicionalnog CAD sustava ima veću cijenu samo u prvoj fazi kreiranja jer je potrebno više vremena za kreiranje i izradu, ali je tokom vremena pri konstantnom modificiranju i rekreiranju proizvoda, KBE sustav daleko učinkovitiji i brži, a samim time i jeftiniji od tradicionalnog CAD sustava (slika 3.)



Slika 3: Kvalitativna usporedba cijene razvoja prema [3]

Jedna od primjena KBE sustava je u području razvoja proizvoda, da automatski generira koncepte proizvoda ovisno o ulaznim podacima (izlaz pri ranoj fazi razvoja je uglavnom geometrija koncepta); koji se u daljnjem razvoju pripremaju za analizu konačnim elementima (geometrija koncepta je izložena iterativnom postupku promjene, ovisno o mogućnosti stvaranja mreže konačnih elemenata, te se primjenom KBE sustava ovaj proces automatizira i ubrzava)

Primjenom KBE sustava se zamjenjuje podrška stručnjaka pri rješavanju različitih problema (stručnjaci rješavaju kompleksne probleme upotrebom apstraktnih i simboličkih pristupa koji se mogu integrirati u KBE sustav)

Integracija KBS sustava u procesu proizvodnje (KBS sustav na osnovi ulaznih podataka daje različite proizvodne alternative koje bi bile vremenski previše zahtjevne za ljudske osobe)

2.3 Metodologije za razvoj KBE sustava

Osnovna problematika metodologije znanja je uvođenje ontologija. (Prema [4]: Ontologija je obrazac podataka koji predstavlja koncepte unutar neke domene i odnose između tih koncepta. Koristi se za razumijevanje objekata unutar te domene.)

Najpoznatije metodologije za razvoj KBE sustava su:

- Common KADS
- MOKA

2.3.1 Common KADS

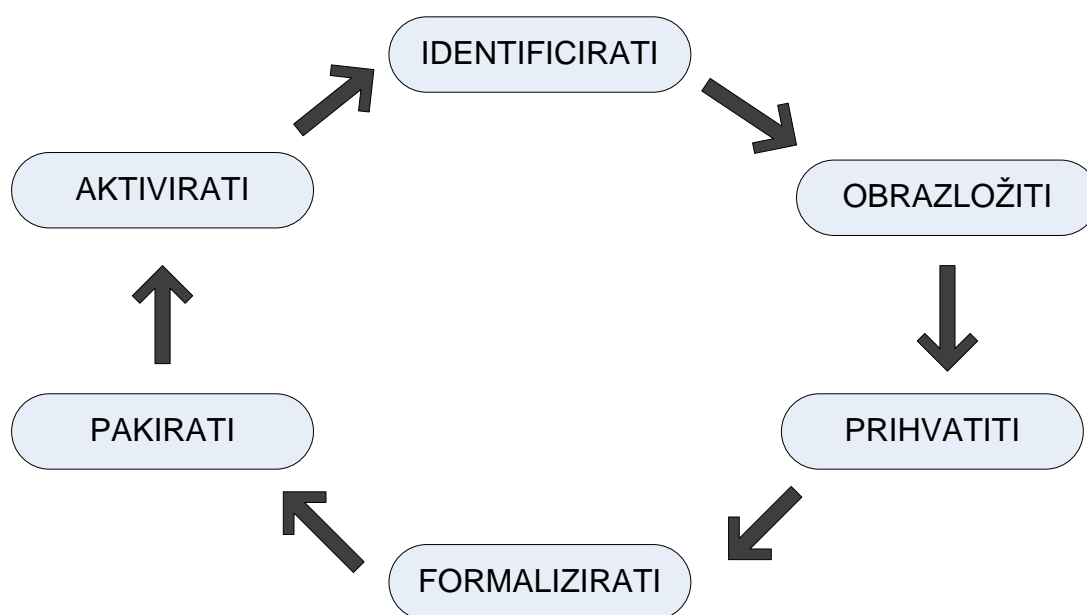
Najkorištenija metodologija u ovom području je trenutno „Common KADS“. Kao vodeća metodologija za podršku inženjerstvu znanja postupno se razvila i danas je potvrđena od mnoštva tvrtki i sveučilišta u kontekstu europskog „Espirit IT“ programa

Danas, jedna od glavnih metodologija, te standard za analizu znanja i razvoj sustava temeljenim znanjem. CommonKADS analizira razvoj organizacija, omogućuje distribuiranje i primjenjivanje njihovog znanja, te daje alate za korporativno upravljanje znanjem.

2.3.2 MOKA

MOKA (eng. Methodology and tools Oriented to Knowledge-Based Engineering Applications) je projekt koji je orijentiran razvoju metodologije koja će postati internacionalni standard za razvoj sustava temeljenih na znanju.

Ciljevi MOKA-e su smanjiti cijenu razvoja KBE sustava, omogućiti dosljedan pravac razvoja i održavanja KBE sustava, omogućiti upotrebu software alata upotrebu metodologije. Metodologija MOKA-e slijedi šest koraka (slika 4.)



Slika 4: Šest koraka MOKA metodologije

Korak 1: Identificirati: tijekom prvog koraka, definiraju se glavne značajke projekta, te se identificira tip KBE sustava.

Korak 2: Obrazložiti: tijekom ovog koraka procjenjuje se cijena i rizici projekta, na osnovu tih procjena definiraju se kriteriji za prihvaćanje projekta i generira se plan projekta

Korak 3: Prihvatiti: tijekom ovog koraka prikuplja se znanje stručnjaka i kreira se „neformalan“ model koji sadrži dijelove proizvoda, funkcijsku dekompoziciju modela, zahtjeve proizvoda, djelatnosti, pravila, ilustracije

Korak 4: Formalizirati: tijekom ovog koraka kreira se „formalni model“, jezikom MOKA metodologije (MML – MOKA Modelling Language), koji je grafički prikaz inženjerskog znanja. Formalni model se sastoji od dva glavna dijela: formalni model proizvoda (struktura, funkcija, ponašanje, tehnologija, geometrija); proces konstrukcije

modela (uzima u obzir ograničenja aplikacije, operativni sustav, izabrani programski jezik i tip stručnjaka uključenog u proces)

Korak 5: Pakirati: tijekom ovog koraka zapisuje se programski kod sustava, programski kod se zatim testira i otklanjaju se greške

Korak 6: Aktivirati: u ovom konačnom koraku, KBE sustav se distribuira i predstavlja krajnjim korisnicima

2.5 Komercijalni KBE sustavi

Danas na tržištu postoje mnogi softverski paketi koji nude integrirani KBE sustav u svojim CAD/CAM/CAE aplikacijama.

Neki od najpoznatijih softverskih paketa su:

- Dessault Systems – CATIA - Knowledgeware modul
- Dessault Systems – Solidworks - Sustainability
- PTC – Pro/Engineer (danas Creo Elements/Pro) - Behavioral Modeling Extension modul (BMX)
- Siemens - Unigraphics – NX Knowledge Fusion
- Autodesk - Inventor

U daljnjem tekstu, uzeti ćemo komercijalnu programsku aplikaciju CATIA – Knowledge modul, razmotriti detaljnije mogućnosti programskog paketa prema zapisivanju inženjerskog znanja u model proizvoda.

3. Softverski paket CATIA

CATIA (eng. Computer Aided Three-dimensional Interactive Application) je komercijalni više platformski CAD/CAM/CAE softverski paket. Razvila ga je francuska kompanija Dessault Systems, te je distribuiran na svjetskoj razini od strane IBM-a.

Softver je razvijen kasnih 70-tih, u cilju razvijanja Dessault-ovih borbenih zrakoplova „Mirage“, te je kasnije prilagođen za primjenu u brodogradnji, automobilističkoj, svemirskoj i ostalim industrijama. Prvotno nazvan CATI (franc. Conception Assistée Tridimensionnelle Interactive, eng. Interactive Aided Three-dimensional Design), bio je integriran u radne stanice, preimenovan u CATIA-u 1981., kada kompanija Dessault System otvara podružnicu za razvoj i prodaju softvera, te potpisuje ugovor o distribuciji sa IBM-om. Tokom godina mnoge velike kompanije su izabrale CATIA softverski paket kao njihovu glavnu 3D CAD aplikaciju, kao na primjer: „Boeing Company“ za razvoj svojih zrakoplova, „General Dynamics Electric Boat Corp.“ za razvoj američke podmornice klase „Virginia“, kao i mnoge druge. 1988. Godine prelazi na UNIX sustave, te je 1998. kompletno prepisan u programski jezik C++, dajući tako podršku za većinu operativnih sustava.

CATIA softverski paket se sastoji od mnogo različitih modula, neki od njih su:

- Moduli za konstruiranje
- Moduli za kinematičku analizu
- Moduli za analize i simulacije
- Moduli za inženjerstvo temeljeno na znanju
- Moduli za konstruiranje upotrebom površina
- Moduli za konstruiranje električnih sustava i električne opreme
- Moduli za NC strojeve i NC programiranje
- Moduli za ergonomsku analizu



Slika 5: Moduli softverskog paketa CATIA

3.1 Moduli za inženjerstvo temeljeno na znanju (Knowledgeware)

Glavni moduli za inženjerstvo temeljeno na znanju u softverskom paketu CATIA su:

- Knowledge advisor
- Knowledge expert
- Knowledge inspector

Moduli se baziraju na mogućnostima pohranjivanja znanja u 3D model proizvoda, te se dijele na tri glavne kategorije:

- Parametri
- Relacije
- Reakcije

3.1.1 Parametri

Kada kreiramo model, uobičajeno se počinje sa kreiranjem skeča gdje definiramo određena svojstva kao npr. udaljenosti, zavisnosti linija, različita ograničenja (okomitost, paralelnost, tangencijalnost, simetrije...); te se nakon tog postupka kreira značajka, korištenjem jedne od funkcija za kreiranje značajke (pad, pocket, shaft, rib...). Krajnji rezultat je model sačinjen od brojnih značajki koje su definirane „unutarnjim“ svojstvima modela. Mijenjanje ili uklanjanje značajki rezultira modifikacijama modela. Ta svojstva modela nazivamo parametrima i oni su od velikog značenja pri korištenju modula za pohranjivanje znanja u model.

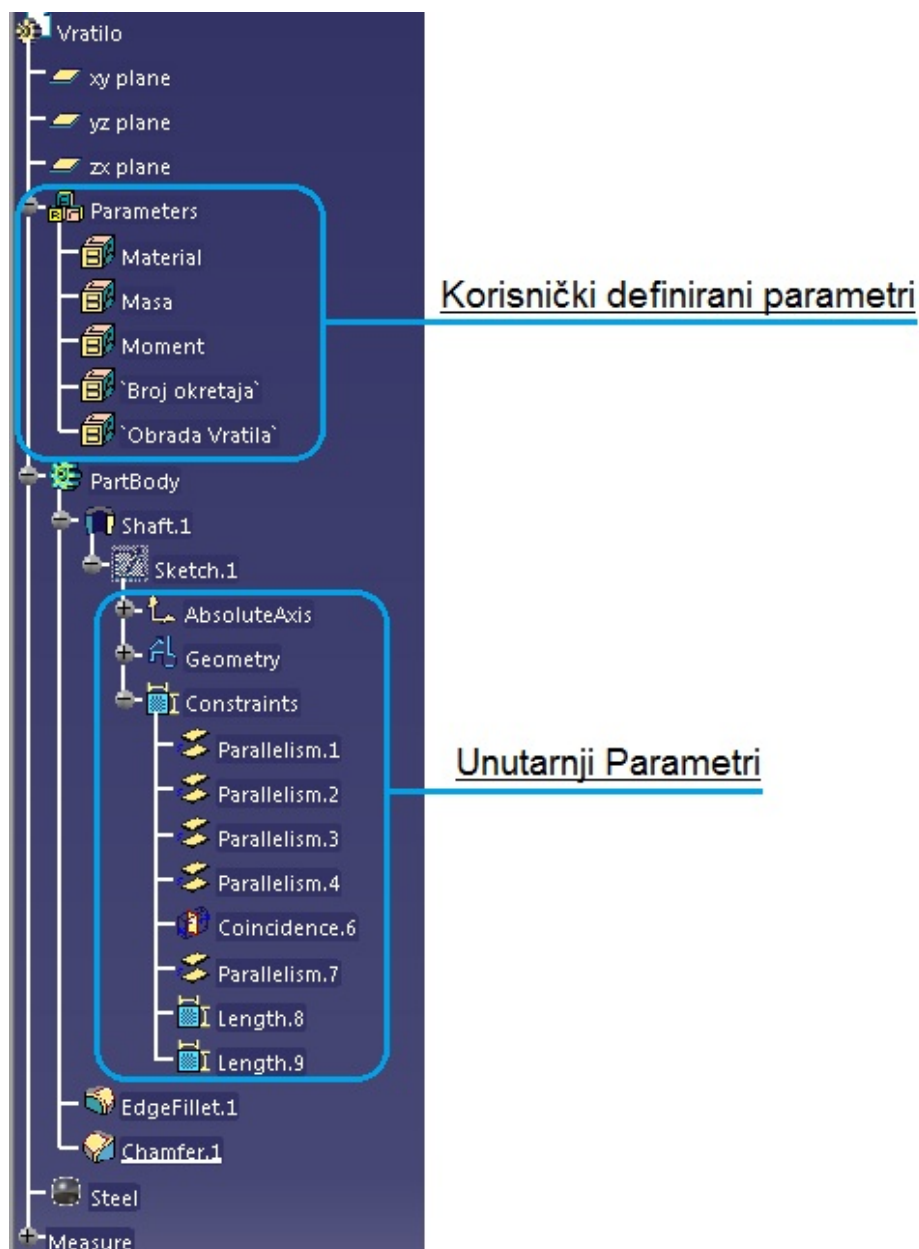
Parametre možemo podijeliti na: unutarnje i korisnički definirane parametre. Unutarnje parametre definira sam sustav na osnovu naših postupaka kreiranja modela kako je opisano u prijašnjem primjeru izrade modela.

Korisnički definirane parametre definira sam korisnik na temelju određenih specifikacija. Takvi parametri mogu ne moraju sadržavati samo brojčane vrijednosti, već se mogu davati određene informacije o proizvodu. Postoje različiti tipovi korisnički definiranih parametara kao npr.: real, integer, string, boolean, duljina, kut, vrijeme, masa, volumen, gustoća, površina, moment inercije, energija, sila, protočna masa, moment, tlak, temperatura, frekvencija, snaga, električni otpor...

Primjer razlike unutarnjih i korisnički definiranih parametara možemo vidjeti na slici 6., te njihov smještaj na samom stablu programskog paketa. Vidljivo je da je korisnik definirao parametre kao: materijal, masu, moment, broj okretaja, obrada vratila; te je također

indirektno definirao unutarnje parametre kreirajući model kao: duljina vratila, radijuse zaobljenja, skošenja, paralelnosti...

Parametri se zatim, upotrebom „Knowledge“ modula mogu povezati raznim relacijama, ovisnostima, ponašanjima, te tako povezani činiti modifikacije proizvoda mnogo jednostavnije i samim time učinkovitije.



Slika 6. Unutarnji i korisnički definirani parametri

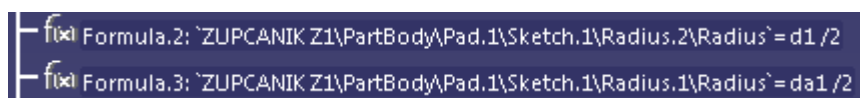
3.1.2 Relacije

Relacije u „Knowledge“ modulu možemo podijeliti u četiri vrste relacija:

- Formule
- Konstrukcijske tablice (eng. design tables)
- Pravila (eng. rules)
- Provjere (eng. check)

3.1.2.1 Formule

Formula definira na koji način se određeni parametar ponaša u odnosu na druge parametre. To su značajke koje nam koriste pri definiranju ili ograničavanju određenog parametra. Pri kreiranju formula omogućeno je korištenje aritmetičkih operatora (+, -, *, /), string funkcija (Search, Extract, Length, ToString...), matematičkih funkcija (log, sin, ln, exp...), jedinica (kg, N, N/mm, mm...), konstanti (PI, E, false, true)



Slika 7: Primjer formule

Prema slici 7. je vidljivo da je na lijevoj strani formule parametar koji je povezan, te sa desne relacija s kojom povezujemo parametar. U ovom primjeru se parametar „Radius.2“ povezuje sa parametrom „d1“ i dijeli sa 2, te bi se promjenom parametra „d1“ promijenio i „Radius.2“.

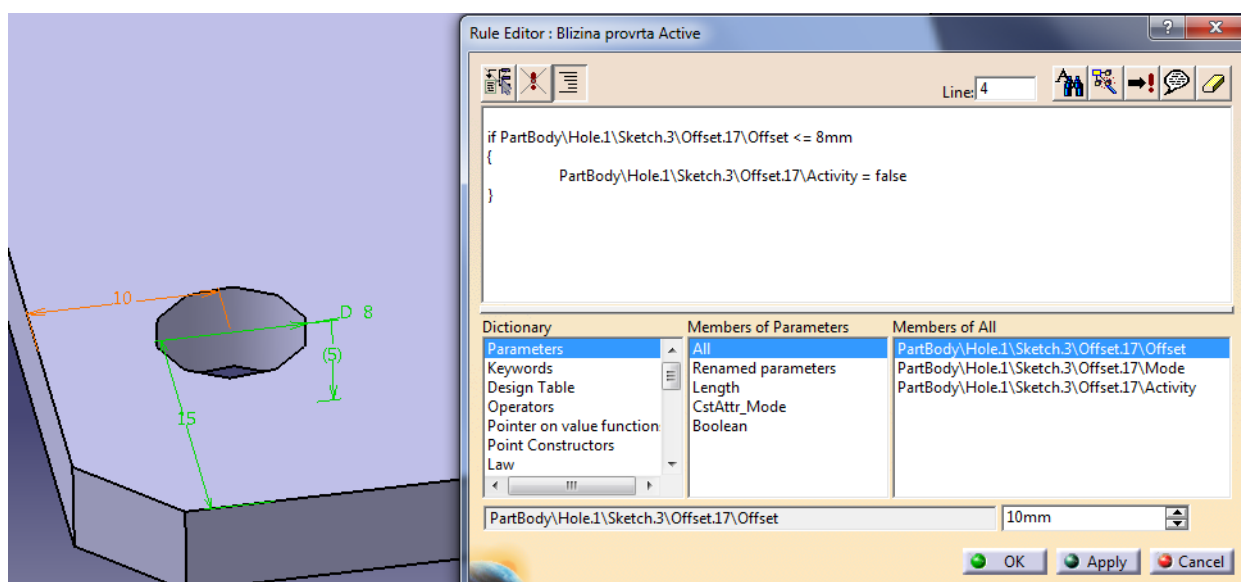
3.1.2.2 Konstrukcijske tablice (eng. design tables)

Konstrukcijske tablice nam omogućuju definiranje različitih parametara pomoću tablica (excel, tekstualna datoteka), te različite varijacije u odnosu na definirane parametre. Najbolji primjer za konstrukcijsku tablicu je vijak. Pojednostavljeno, vijak možemo definirati sa četiri parametra: visina glave, širina glave, visina tijela, navoj. Te parametre možemo zatim integrirati u model vijka, te kreirati tablicu sa različitim vrijednostima parametara (konfiguracijama). Na osnovi tih parametara, možemo brzo i efikasno kreirati određenu konfiguraciju vijka bez potrebe modificiranja prijašnjih parametara.

Konstrukcijske tablice možemo također koristiti za povezivanje određenih parametara sa vanjskim excel tablicama ili tekstualnim datotekama, koje po potrebi možemo povezati sa programskim paketima za proračune i time dobivamo ulazne podatke za parametre modela.

3.1.2.3 Pravila (eng. Rules)

Pravila možemo definirati kao set instrukcija preko kojih se kontrolira određeni parametar i događaj koji se dešava kao posljedica. Koriste Visual Basic sintaksu pri programiranju linija koda. Korištenjem ključnih riječi (if, else, else if, and, or), operatora ($=$, $<$, $>$, $=>$, ...), poruka i makro funkcija (Message, LaunchMacroFromFile) možemo definirati pravilo i posljedicu njegovog prekoračenja.



Slika 8: Primjer primjene pravila

Na slici 8 je prikazan jednostavan primjer korištenja pravila, gdje programski kod glasi:

```
if PartBody\Hole.1\Sketch.3\Offset.17\Offset <= 8mm
{
    PartBody\Hole.1\Sketch.3\Offset.17\Activity = false
}
```

U programskom kodu je ulazni podatak udaljenost provrta (slika 8.: narančasta kota), koju matematičkim operatorom „ $<=$ “ definiramo da ne smije biti manja od 8mm, te se u slučaju kršenja tog pravila, kao posljedica se deaktivira provrt.

3.1.2.4 Provjere (eng. check)

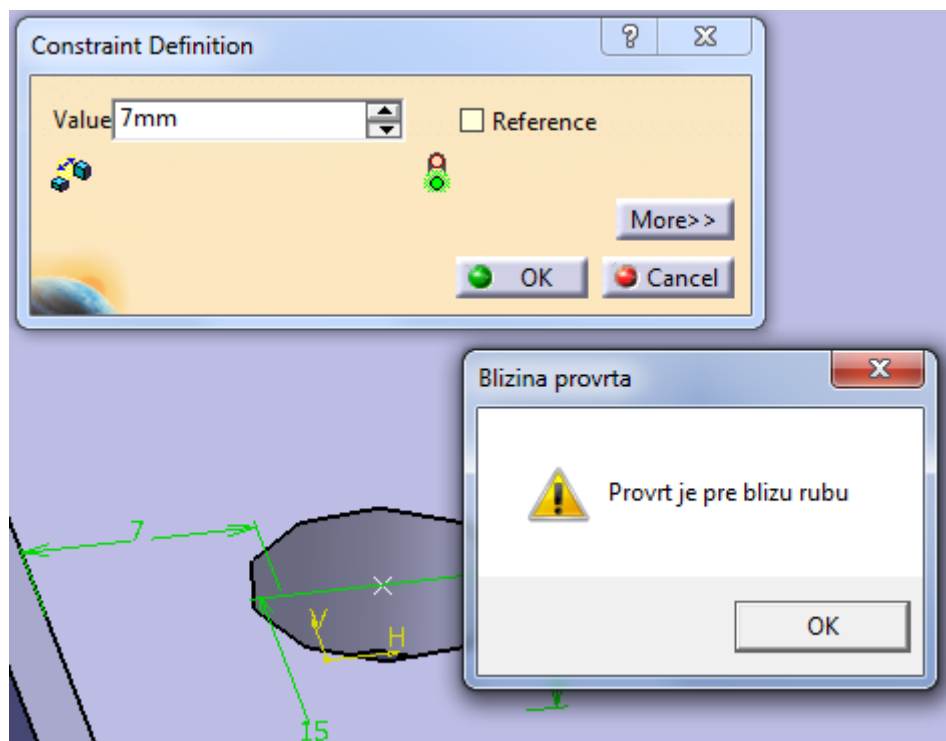
Slično kao i kod pravila, provjera je dio koda, ali za razliku od pravila, provjera nema nikakav utjecaj na vrijednosti parametara. Pri provođenju pravila nam daju samo informacije o provjerama određenih parametara. Postoje tri tipa provjera: tihi (ne prikazuje nikakvu poruku, u slučaju prekoračenja definiranih graničnih vrijednosti jedini način kako možemo vidjeti rezultat provjere je u stablu modela, simbolički kao semafor sa crvenim svjetlom), informacija (prikazuje se kao poruka informacije), upozorenje (prikazuje se kao poruka upozorenja da su prekoračene definirane vrijednosti provjere).

Kao i pravila, provjere koriste Visual Basic sintaksu koristeći ključne riječi (if, then, and, or)

Na slici 9 je prikazan jednostavan primjer korištenja pravila, gdje programski kod glasi:

```
PartBody\Hole.1\Sketch.3\Offset.17\Offset >= 8mm
```

Zadajemo relaciju koja govori da udaljenost provrta od ruba mora biti veća ili jednaka 8mm, u protivnom se javlja poruka upozorenja koja nam govori da je provrt preblizu.



Slika 9. Primjer primjene provjere

3.1.3 Reakcije (eng. reactions)

Reakcije su niz operacija povezane sa značajkama, te se provode kao odgovor na neki događaj. Za pisanje koda se također koristi Visual Basic sintaksa. Slično kao i relacije pravila, sprema se u model, reagira na promjene u modelu provodeći modifikacije, ali za razliku od pravila, daje nam bolju kontrolu nad akcijama, omogućuje izvođenje kompleksnijih reakcija, omogućuje korištenje kompletne Visual Basic sintakse.

3.1.4 Upotreba funkcije „Set of Equations“

Knowledge modul nam također omogućuje rješavanje niz jednadžbi te njihovo povezivanje sa određenim parametrima.

Za primjer ćemo uzeti dvije jednadžbe sa dvije nepoznanice:

$$A = 3$$

$$y - 2 \cdot \sqrt{x} = A$$

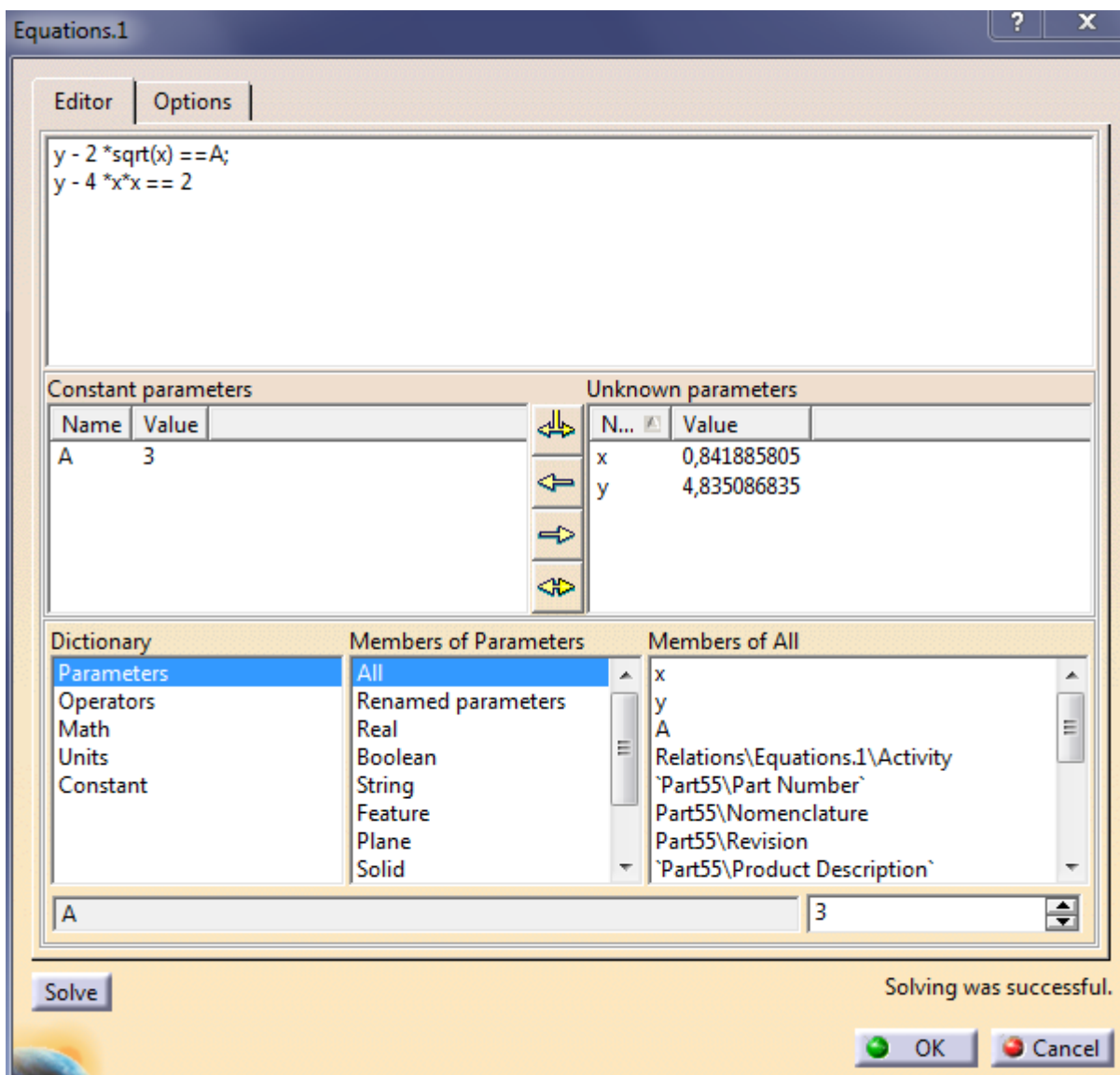
$$y - 4 \cdot x^2 = 2$$

Ove jednadžbe bi mogli riješiti ručno, te bi nam trebalo neko vrijeme da ih riješimo ili bi mogli koristiti neki od programskih paketa baziranih na matematičkim proračunima.

Definiramo parametre A, x i y u modelu, te funkcijom „Set of Equations“ definiramo njihove relacije, koje su našem slučaju dvije jednadžbe sa dvije nepoznanice. Softver automatski nalazi rješenje za parametre x i y, te ih pohranjuje u parametre modela kao brojčanu vrijednost.

Funkcija nam omogućuje da upotrijebimo i različite konstantne parametre, te se u slučaju izmjena tih parametara, automatski računaju nove vrijednosti, i ako su povezane sa modelom, mijenja se i sam model u zavisnosti o novim izračunatim vrijednostima. Ovom funkcijom izbjegavamo korištenje drugih komercijalnih programskih paketa, te njihovo povezivanje sa modelom koristeći programske jezike, daje nam mogućnost izvođenja proračuna unutar samog modela, te automatskog regeneriranja za različite vrijednosti ulaznih parametara.

Na slici 10. vidimo primjer rješavanja spomenutih jednadžbi gdje je parametar „A“ konstantna zadana vrijednost, te svakom promjenom tog parametra mijenja se i vrijednost parametara x i y.



Slika 10: Primjer funkcije „Set of equations“

4. Upotreba modula „Knowledge“ na primjeru reduktora

U daljnjem tekstu će se obraditi primjer pohranjivanja inženjerskog znanja na primjeru programskog zadatka kolegija „Elementi konstrukcija 3“, Reduktor.

Reduktori spadaju u zupčaničke prijenosnike snage. Od svih prijenosnika zupčanički i lančanički prijenosnici su danas najčešće u primjeni. Izrađuju se od najmanjih do najvećih snaga, te se upotrebljavaju za vrlo visoke brojeve okretaja u širokom rasponu brojeva okretaja. Imaju veliku pogonsku sigurnost i dug vijek trajanja uz mogućnost kratkotrajnog preopterećivanja.

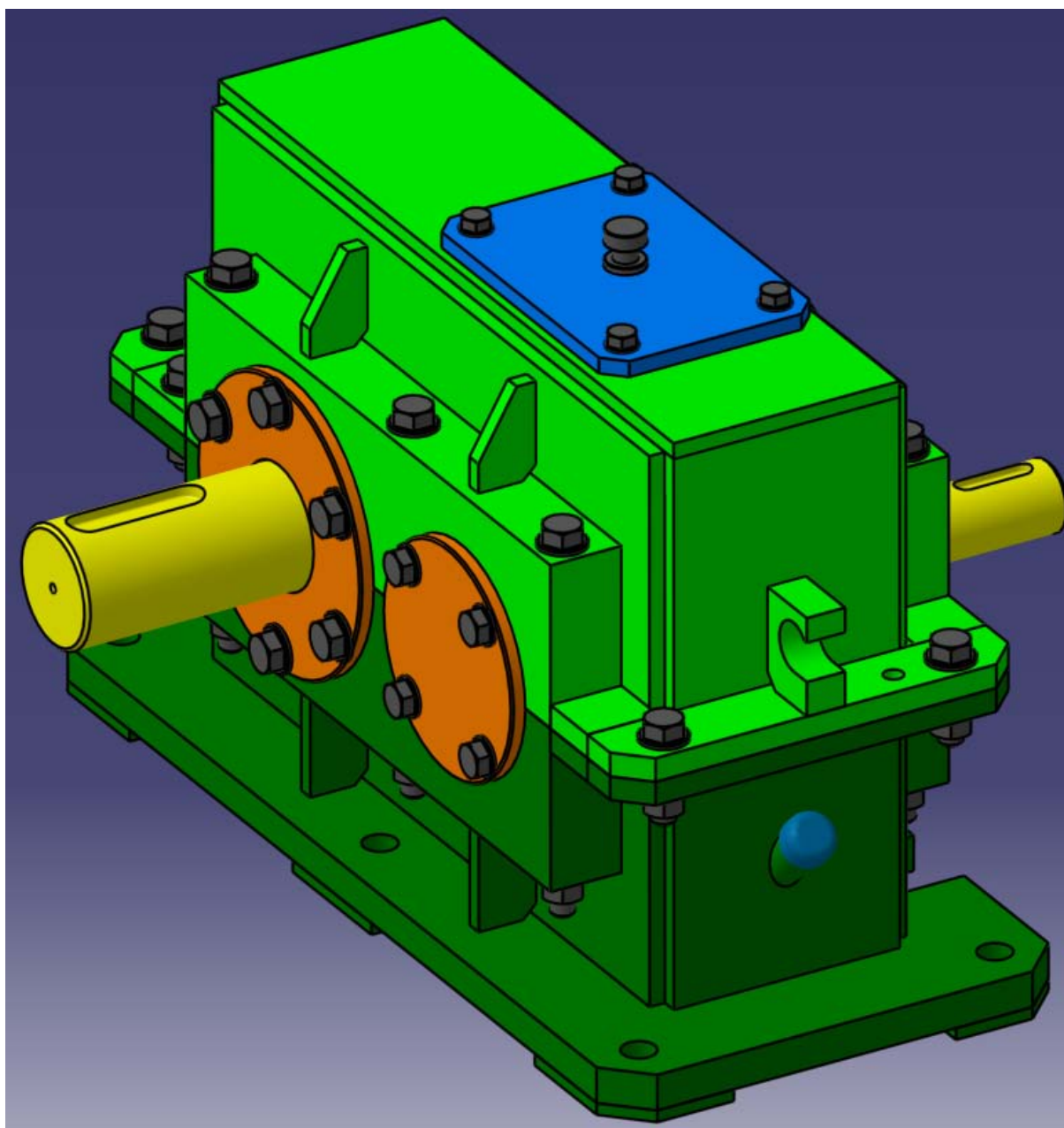
Prednosti su im još velika mogućnost prijenosa snage po jedinici volumena, te visok stupanj iskorištenja prenesene snage.

Glavni nedostaci su: relativno velika buka, velika krutost prijenosa (ugrađuju se sa elastičnim, hidrauličkim spojka)

Pohrana inženjerskog znanja na modelu reduktora može biti korisna sa aspekta rekreiranja modifikacija različitih varijanta reduktora, modifikacijama na određenim dijelovima reduktora, kojim se može ubrzati rad izrade modela reduktora, smanjiti mogućnost pogreške radi ne stručnosti korisnika.

Povezivanjem programskog paketa CATIA sa tekstualnim ili tabličnim datotekama otvara se mogućnost povezivanja vanjskih proračuna sa samim modelom, i integriranja parametara u konstrukciju modela, stvaranja njihovih relacija, pravila, uputa. Omogućujući tako vanjskom korisniku jednostavnu manipulaciju i modifikaciju određenih dijelova modela i sklopa.

Kroz pojedinačne dijelove ćemo zapisati inženjersko znanje na osnovu proračuna, određenih preporučenih vrijednosti prema [7].



Slika 11: Model reduktora

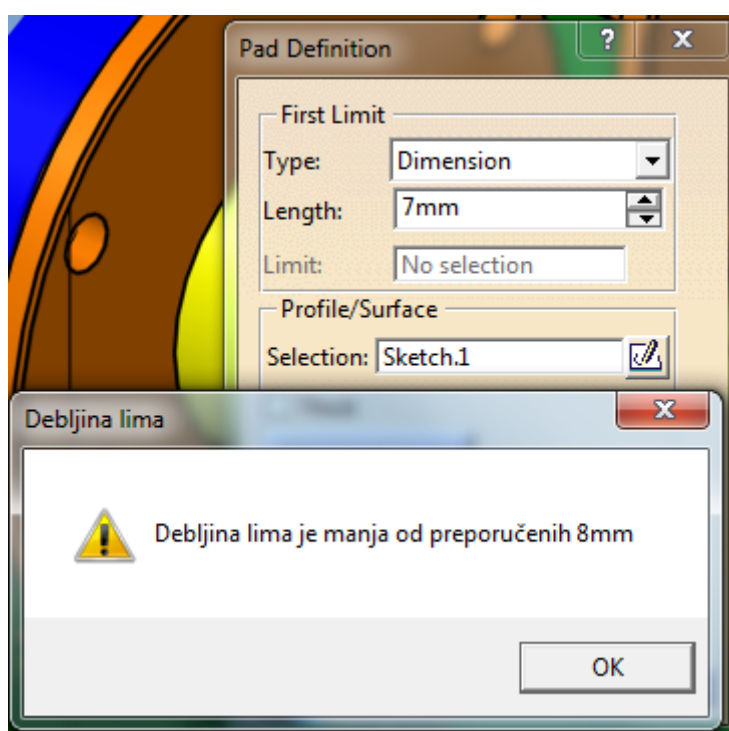
4.1 Kućište

Kod kućišta reduktora postoje određena ograničenja dobivena preporukama na odnosu izračunatog modula zupčanika.

Definirani parametri su:

- debljine limova koje ne smiju biti manje od izračunatih 8mm, te ćemo u tom slučaju koristiti funkciju provjere (eng. check), sa porukom upozorenja.

Prilikom promjene debljine lima ispod dozvoljene javlja se poruka upozorenja da je debljina lima manja od preporučenih 8mm, te smo time osigurali da će osoba koja nije sudjelovala u računanju proračuna i samom konstruiranju, te nije upoznata sa ograničenjima biti upozorena da je prešla dozvoljene vrijednosti.



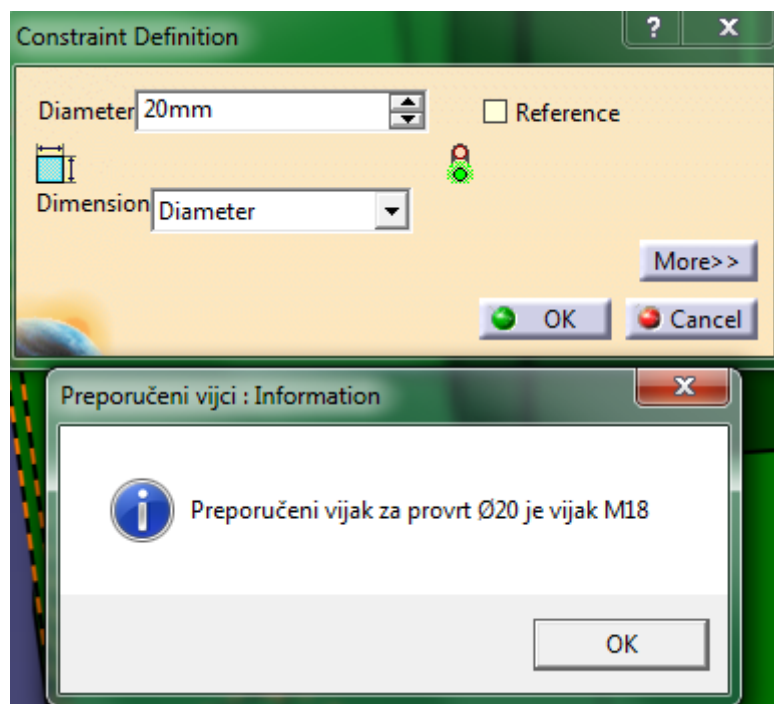
Slika 12: Funkcija provjere pri promjeni debljine lima

- debljina prirubnice stopala kućišta ne smije biti manja od preporučenih 18mm (funkcija provjere, sa porukom upozorenja)
- debljina prirubnice kućišta ne smije biti manja od 12mm (funkcija provjere, sa porukom upozorenja)
- provrt za vijke stopala prirubnice ne smije biti manji od M16 (funkcija provjere, sa porukom upozorenja);

Ako bi htjeli staviti veće vijke korištenjem funkcije pravila (eng. rule), korištenjem programskog koda koji glasi:


```
if PartBody\Pad.1\Sketch.1\Radius.28\Radius >= 10mm and  
PartBody\Pad.1\Sketch.1\Radius.28\Radius < 11mm  
{  
    Message("Preporučeni vijak za provrt Ø# je vijak M18",  
PartBody\Pad.1\Sketch.1\Radius.28\Radius*2000 )  
}  
else if  
....
```

Definiramo ulazni podatak: provrt, te ga matematičkim operatorima „ \geq “ i „ $<$ “ stavljamo u određene granice, kao izlazni podatak koristimo funkciju „Message“ koja javlja korisniku poruku o preporučenom vijku, na temelju promjera provrta koji je odabrao, prema dijelu programskog koda gore napisanog, ako korisnik promjeni promjer provrt u granicama od $\text{Ø}20$ do $\text{Ø}22$ javit će se poruka sa preporučenim vijkom M18 za izabrani promjer (slika 13.)



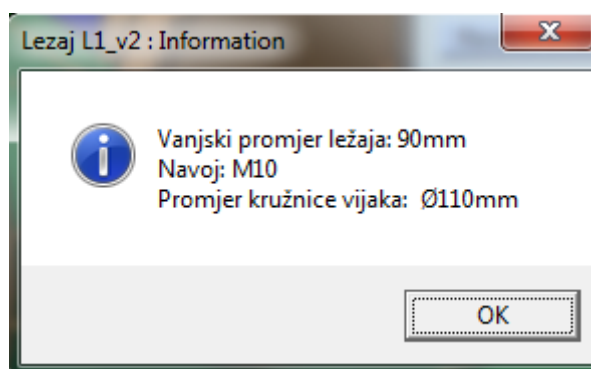
Slika 13: Funkcija pravila pri promjeni promjera provrta (preporučeni vijak)

- promjenom ležaja, tj. definiranjem vanjskog promjera ležaja na kućištu, prema preporučenim vrijednostima mijenja se broj provrta za vijke poklopaca, navoji vijaka, te se deaktivira utor za uskočnik (mijenjanjem ležaja mijenjaju se dimenzije uskočnika ili uskočnik više nije potrebno koristiti)

Koristimo funkciju pravila, dio programskog koda korištenog glasi:

```
...  
if PartBody\Pad.1\Sketch.1\Radius.24\Radius > 42.5mm and  
PartBody\Pad.1\Sketch.1\Radius.24\Radius <= 45mm  
{  
PartBody\Hole.1\Sketch.2\Radius.40\Radius = 55mm  
PartBody\Hole.1\ThreadDescription = "M10"  
Message(" Vanjski promjer ležaja: #mm| Navoj: #| Promjer kružnice vijaka: Ø#mm",  
PartBody\Pad.1\Sketch.1\Radius.24\Radius*2000, PartBody\Hole.1\ThreadDescription,  
PartBody\Hole.1\Sketch.2\Radius.40\Radius*2000)  
}  
...
```

Ulazni podatak koda je radijus rukavca (Radius.24), kojeg matematičkim operatorima „>“ i „<=“ stavljamo u određene definirane vrijednosti, izlazni podatci koda su promjena navoja vijka kućišta (Hole.1\ThreadDescription), radijus kružnice vijaka (Radius.40), te nas korištenjem funkcije „Message“, o svim novonastalim promjenama obavještava informativna poruka, modificirana da prikazuje promjere. (slika 14.)



Slika 14: Informativni prozor sa definiranim dimenzijama

- U slučaju povećanja dimenzija promjera rukavca, pri čemu za posljedicu imamo približavanje središnjeg provrta vijka kućišta rubu promjera rukavca, treba definirati dopušteni razmak vijka kućišta (funkcija provjere, sa porukom upozorenja)
- Skošenje rukavca ne smije biti manje od 1.5mm radi brtvenog prstena na poklopcu (funkcija provjere, sa porukom upozorenja)

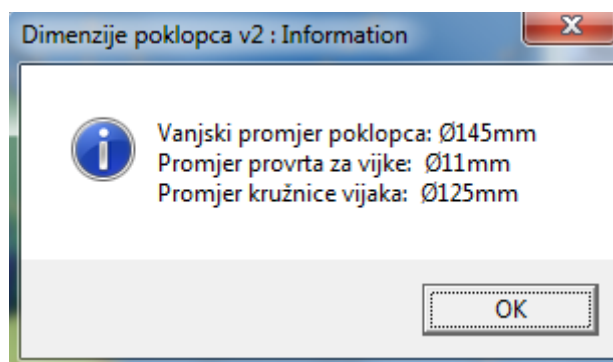
4.2 Poklopci

- Promjer poklopca je pomoću funkcije formula povezan sa parametrom promjera rukavca za ležaj, te izmjenom promjera rukavca se automatski mijenja i promjer poklopca
- Promjenom promjera poklopca na mjestu rukavca za ležaj, upotrebom funkcije pravila, te korištenjem programskog koda koji glasi:

```
if PartBody\Pad.1\Sketch.1\Radius.1\Radius > 42.5mm and
PartBody\Pad.1\Sketch.1\Radius.1\Radius <= 45mm
{
    PartBody\Pad.3\Sketch.6\Radius.11\Radius = 65mm
    PartBody\Hole.2\Sketch.7\Radius.20\Radius = 55mm
    PartBody\Hole.2\Diameter = 11mm
    Message(" Vanjski promjer poklopca: Ø#mm| Promjer provrta
za vijke: Ø#| Promjer kružnice vijaka: Ø#mm",
PartBody\Pad.3\Sketch.6\Radius.11\Radius*2000,
PartBody\Hole.2\Diameter,
PartBody\Hole.2\Sketch.7\Radius.20\Radius*2000)
}
```

Ulazni podatak koda je radijus rukavca (Radius.1), kojeg matematičkim operatorima „>“ i „<=“ stavljamo u određene definirane vrijednosti, izlazni podatci koda su promjena vanjskog radijusa poklopca (Radius.11), radijus kružnice vijaka (Radius.20), provrta za vijke (Hole.2\Diameter), te nas korištenjem funkcije „Message“, o svim novonastalim promjenama obavještava informativna poruka, modificirana da prikazuje promjere. (slika 14.)

Na osnovu promjera poklopca na mjestu rukavca za ležaj, mijenjaju se vanjske dimenzije poklopca, provrta za vijke, broj vijaka, također se korisnika informira o promjenama na poklopcu, dajući mu glavne informacije o vanjskom promjeru poklopca, promjeru provrta za vijke



Slika 15: Informacijski prozor sa definiranim dimenzijama poklopca

- Zaobljenje na mjestu gdje dolazi brtveni prsten ne smije biti manji od preporučenog
- Poklopci na izlaznim dijelovima vratila moraju biti minimalno: promjer vratila na mjestu izlaza poklopca uvećan za 1mm (funkcija provjere, povezana sa vanjskim promjerom vratila, sa porukom upozorenja)

4.3 Distantni prsten

- Unutarnji promjer distantnog prstena ne smije biti manji od promjera vratila (funkcija provjere, povezana sa promjerom vratila, sa porukom upozorenja)
- Unutarnji promjer distantnog prstena ne smije biti veći od promjera vratila uvećanog za 0.5mm, jer bi moglo doći do zvonjenja u radu, kada bi se sustav istrošio (funkcija provjere, povezana sa promjerom vratila, sa porukom upozorenja)

4.4 Zupčanici

Prilikom izrade modela zupčanika u glavnom sklopu definiramo parametre dobivene proračunom (korisnički definirani parametri): modul, standardni osni razmak, pomake profila, broj zubi zupčanika 1, broj zubi zupčanika 2, kut zahvatne linije. Pomoću funkcije „Set of equations“ rješavamo jednadžbe prema [8]:

$$d_1 = z_1 \cdot m$$

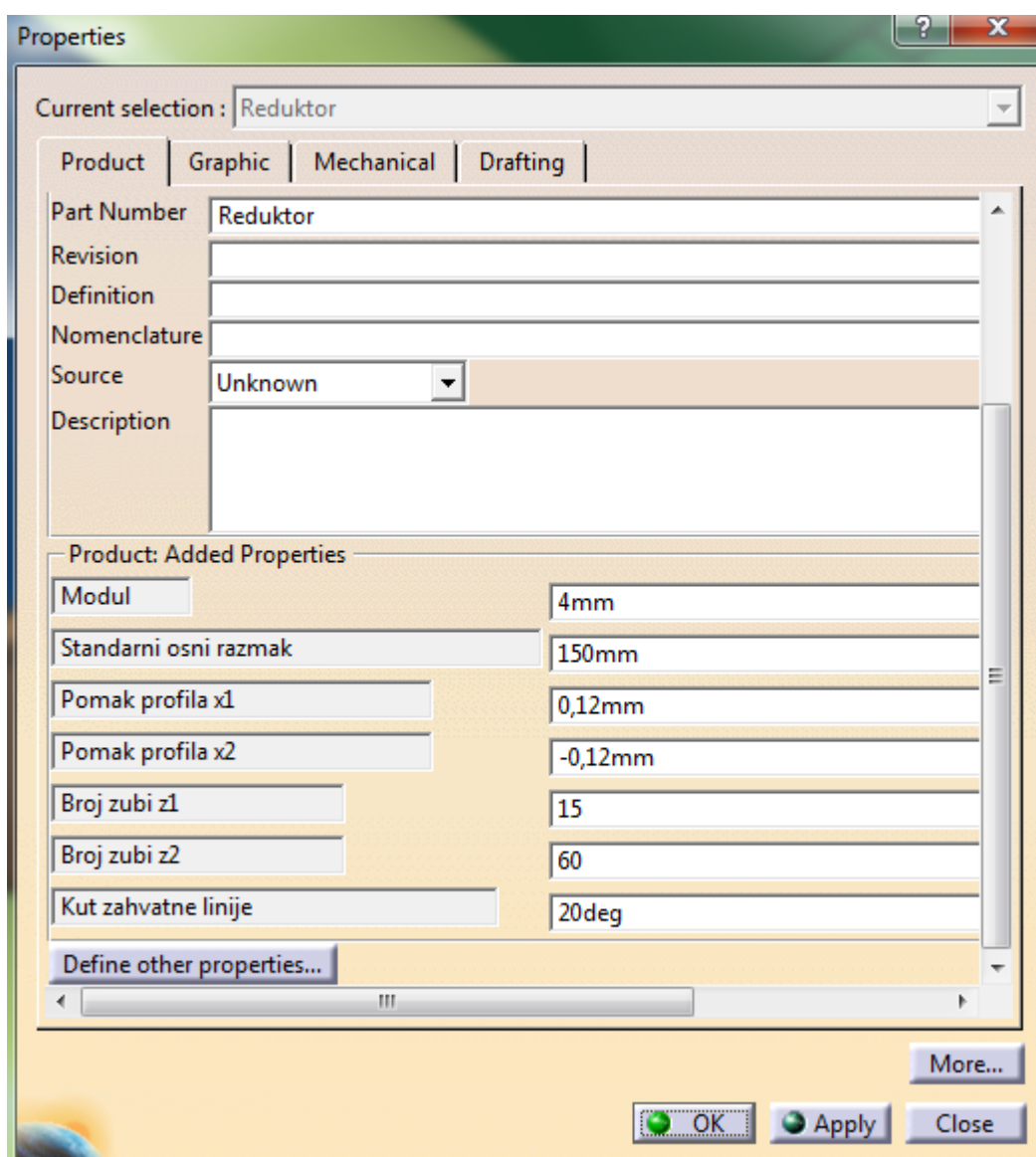
$$d_{a1} = d_1 + 2 \cdot m(1 + x_1)$$

$$d_{w1} = d_1 \cdot \frac{\cos \alpha}{\cos \alpha_w}$$

$$d_{f1} = d_1 - 2 \cdot m + 2 \cdot x_1 \cdot m - 2 \cdot c = d_1 - 2m(1,25 - x_1)$$

$$d_{b1} = d_{w1} \cdot \cos \alpha_w$$

Dobivamo nove parametre koji definiraju diobeni promjer, tjemeni promjer, kinematski promjer, podnožni promjer, temeljni promjer; koje zatim povezujemo sa geometrijom samog zupčanika te promjenom nekih od gore navedenih ulaznih parametara, mijenjamo geometriju samog zupčanika. Ovi izračunati izlazni parametri su nam od važnosti pri odabiru maksimalno dozvoljenog promjera vratila na mjestu uklinjenja sa zupčanikom.



Slika 16: Korisnički definirani parametri

4.5 Vratila

- Maksimalni radijusi zaobljenja na mjestu ležajeva su ograničeni samim ležajem (funkcija provjere, sa porukom upozorenja)
- Maksimalni radijusi zaobljenja na mjestu uskočnika su ovisni o uskočniku (funkcija provjere, sa porukom upozorenja)
- Maksimalni promjeri vratila na mjestu uklinjenja sa zupčanikom 2, te u slučaju vratila 1 koji je izrađen zajedno sa zupčanikom; vrijede relacije prema [7]:

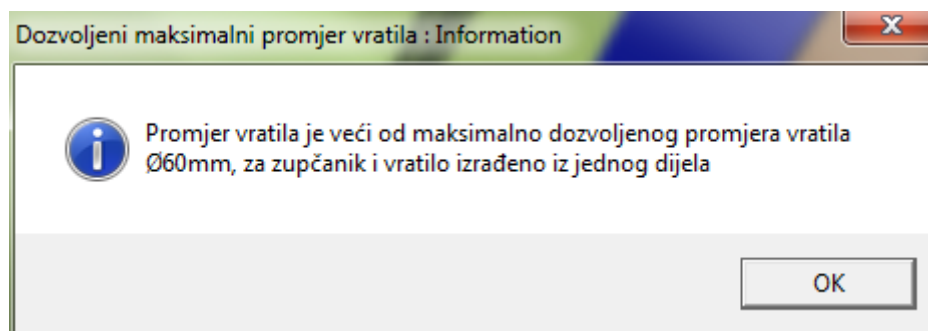
Zupčanik i vratilo iz jednog dijela:

$$d_1 \geq 1,2d_{vr}$$

Zupčanik uklinjen na vratilo:

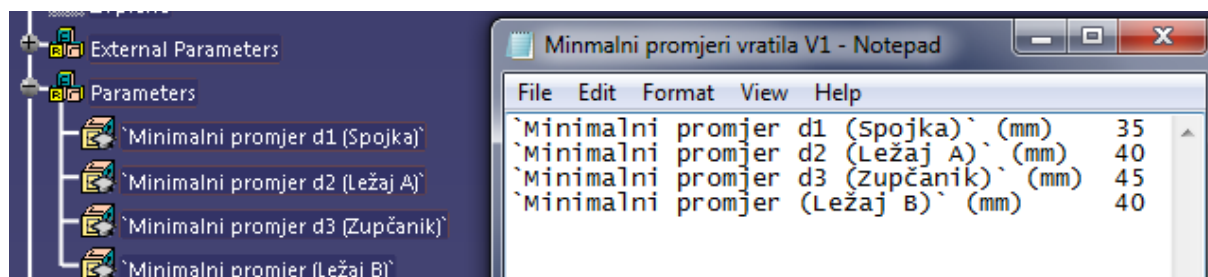
$$d_1 \geq 2d_{vr}$$

Prethodno su na osnovi ulaznih podataka izračunati diobeni promjeri zupčanika, te na odnosu tih promjera koje povezujemo sa maksimalnim promjerom vratila, postavljamo zadane relacije za iznos maksimalnog promjera vratila na mjestima zupčanika, koristimo funkciju provjere, sa porukom upozorenja.



Slika 17: Upozorenje za prekoračen dozvoljeni maksimalni promjer vratila

- Minimalni promjeri vratila su dobiveni proračunom, te ih možemo zapisati u tekstualnu datoteku, koju možemo po potrebi mijenjati. Tekstualnu datoteku zatim pomoću funkcije „design table“ unosimo u model, te definiramo kao ulazne parametre u modelu (slika 18.)



Slika 18: Minimalni promjeri vratila integrirani u model iz tekstualne datoteke

Promjenom tekstualne datoteke automatski se sinkronizira sustav, te nas obavještava da su se promijenili parametri povezani sa datotekom. Ovim postupkom se mogu povezati programski paketi za izvođenje proračuna, koji bi svoje izlazne podatke spremali u tekstualne datoteke ili excell tablice, koje bi se zatim povezale sa modelom, dajući tako nove uvjete i restrikcije modela.

Kod provjere minimalnog promjera vratila koristimo funkciju pravila, sa programskim kodom koji glasi:

```
if PartBody\Shaft.1\Sketch.3\Offset.38\Offset < (`Minimalni promjer d1
(Spojka)` /2)

{

    LaunchMacroFromDoc("Promjena promjera vratila.catvbs")

}
```

Ulazni parametri koda su: promjer vratila (Offset.38) i vanjski parametar učitani preko tekstualne datoteke (`Minimalni promjer d1 (Spojka) povezani matematičkim operatorom „<“ u relaciju, te se kao posljedica kršenja pravila pokreće Visual Basic makro datoteka (Promjena promjera vratila.catvbs) koja sadrži programski kod:

```
Language="VBSCRIPT"

Sub CATMain()

Set productDocument1 = CATIA.ActiveDocument

Set product1 = productDocument1.Product

Set product1 = product1.ReferenceProduct

Set documents1 = CATIA.Documents

Set partDocument1 = documents1.Item("ZUPCANIK Z1.CATPart")

Set part1 = partDocument1.Part

Set bodies1 = part1.Bodies

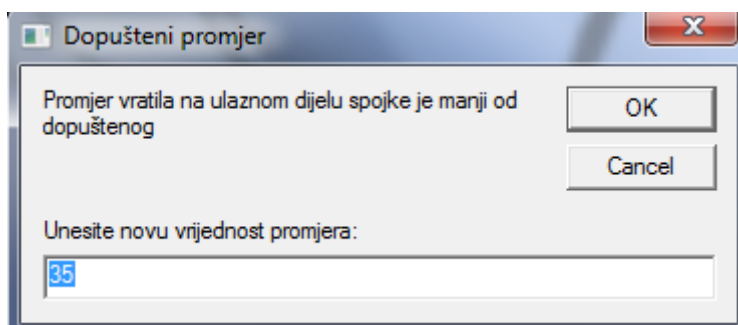
Set body1 = bodies1.Item("PartBody")

Set shapes1 = body1.Shapes

Set shaft1 = shapes1.Item("Shaft.1")
```

```
Set sketch1 = shaft1.Sketch  
Set constraints1 = sketch1.Constraints  
Set constraint1 = constraints1.Item("Offset.38")  
Set length1 = constraint1.Dimension  
Set parameters1 = part1.Parameters  
  
A = InputBox ("Promjer vratila na ulaznom dijelu spojke je manji od  
dopuštenog"  
+(Chr(13))++(Chr(13))++(Chr(13))++(Chr(13))+"Unesite novu  
vrijednost promjera:", "Dopušteni promjer","35")  
  
length1.Value = A/2  
  
End Sub
```

Definiraju se privremene varijable za izvođenje programa (unutarnji parametri referentnog modela), referentni model ("ZUPCANIK Z1.CATPart"). Sa funkcijom „InputBox“, koja pokreće tekstualni prozor sa definiranom porukom upozorenja, te omogućuje unos nove numeričke vrijednosti, definira se varijabla „A“. Varijabla „A“ je pridružena izlaznoj varijabli novog promjera vratila (length1)



Slika 19. Tekstualni prozor za definiranje novog promjera vratila

5. Zaključak

Implementacija konstrukcijskog znanja u 3D model nosi brojne prednosti pri izradi proizvoda, ali i neke nedostatke koje treba uzeti u obzir pri samoj implementaciji.

Glavne prednosti korištenja KBE sustava su povećanje efikasnosti, jednostavnija optimizacija proizvoda, problem promjene inženjera na projektima više ne bi trebao biti veliki problem, jer se njihovo stručno znanje integrira u sam model proizvoda. Automatizacijom monotonih, rutinskih procesa izrade modela, inženjer se može koncentrirati na nove inovativne solucije proizvoda.

Neki od glavnih nedostataka KBE sustava su povećano vrijeme izrade modela, tj. inženjer treba odvojiti vrijeme za izradu modela sa integriranim znanjem, što nekad i nije moguće radi rokova izrade proizvoda, te bi se u takvom slučaju trebalo razmisliti da li je isplativo i korisno integrirati znanje u proizvod, tj. koliko će taj proizvod biti modificiran, ponovno korišten. Veliki nedostatak KBE sustava je mogućnost osobe da bez razmišljanja mijenja proizvod, unoseći ulazne podatke, te dobivajući izlazne rezultate iz sustava, dajući nam lažnu sigurnost da je sve u redu. Takvo razmišljanje može dovesti do velikih problema ako sustav sadrži greške. Zato je jedna od važnih stvari pri implementaciji znanja u model, kvalitetno to znanje zapisati, tj. objasniti zašto je nešto promijenjeno i sa kojim povodom je nešto promijenjeno.

6. Literatura

- [1] Stokes M., Ed.: *Managing Engineering Knowledge – MOKA: Methodology for Knowledge Based Engineering Applications*, ASME Press, 2001.
- [2] Nick Milton: *Knowledge Technologies*, Polimetrica, 2008.
(http://books.google.it/books?id=qXwBQmMmMtQC&pg=PA5&dq=milton+polimetrica&sig=kL5UoF0U8Y7B_6H9NLwt9TcfOqE#v=onepage&q=milton%20polimetrica&f=false)
- [3] www.alycon.co.in
- [4] http://hr.wikipedia.org/wiki/Ontologija_%28informacijske_znanosti%29
- [5] Common KADS (<http://www.commonkads.uva.nl/frameset-commonkads.html>)
- [6] MOKA (<http://web1.eng.coventry.ac.uk/moka/default.htm>)
- [7] Opalić M., Rakamarić P.: *Reduktor*, Fakultet strojarstva i brodogradnje, 2001, Zagreb
- [8] Oberšmit E.: *Ozubljenja i zupčanici*, SNL, 1982. Zagreb
- [9] <http://www.3ds.com>
- [10] CATIA online documentation