

PRIMER RAZVOJA PROCESNEGA SIMULATORJA Z OBJEKTNO USMERJENO LUPINO EKSPERTNEGA SISTEMA

Alenka Žnidaršič, *Juš Kocijan, **Andrej Skobe

Odsek za računalniško avtomatizacijo in regulacije, Institut Jožef Stefan, Ljubljana

*Fakulteta za elektrotehniko, Univerza v Ljubljani

**Iskra sistemi, d.o.o., Stegne 21, Ljubljana

E-pošta: alenka.znidarsic@ijs.si

Url: <http://www-e2.ijs.si/People/Alenka.Znidarsic.html>

Povzetek

Uspešno reševanje problemov vodenja tehnoloških procesov postaja eden ključnih dejavnikov, ki vplivajo na parametre, s katerimi se meri uspešnost podjetja. Kompleksnost računalniških sistemov, ki nastopajo v funkciji vodenja in nadzora tehnoloških procesov, se kaže v tesni prepletenosti analitičnih rešitev in heurističnega znanja, delovanja v realnem času in lastnosti tehnološkega postopka. Zato je vedno bolj prisotna zahteva po uporabi sodobnih metod in orodij, med katerimi je najbolj razširjena uporaba ekspertnih sistemov. V članku bomo predstavili objektno - orientirano okolje G2 (GenSym Co.), ki omogoča razvoj ekspertnih sistemov v realnem času. Njegove lastnosti lahko opišemo z uporabo naslednjih paradig: objektna orientiranost, sklepanje na osnovi pravil in postopkovnega programiranja. Osnovne gradnike okolja G2 bomo prikazali na primeru realizacije simulatorja procesa percipitacije amorfnega SiO_2 , ki je namenjen eksperimentiranju in testiranju s postopkom pridobivanja silikatov.

Abstract

Efficient solving of process control problems requires manipulating with a wide spectrum of knowledge representations ranging from purely analytical knowledge up to qualitative knowledge. These tendencies caused the growing interest of using methods that are flexible enough to accommodate the knowledge based paradigms. In the paper, the environment for development of real - time expert systems - G2 is presented. G2 represents a strong object - oriented environment, combining knowledge - based techniques with conventional programming techniques.. The basic concepts and characteristics of G2 are represented by the application of the simulation environment for the chemical SiO_2 process.



1. UVOD

Sistemi, kot celote prepletenih komponent, postajajo z razvojem novih tehnologij in znanj vse bolj kompleksni in zahtevnejši. Še toliko bolj velja to za računalniške sisteme, ki nastopajo v funkciji vodenja in nadzora tehnoloških procesov, katerih kompleksnost se kaže z delovanjem v realnem času in na področju, ki ga definira in omejuje tehnološki postopek. Zato se v praksi pogosto pojavlja vprašanje kako avtomatizirati določen proces glede na sodobne koncepte in filozofijo vodenja z uporabo sodobnih tehnik in metod tako, da se bomo najbolj približali definiranim ciljem. Pri realizaciji vodenja prihaja do integracije različnih disciplin [3], med drugim systemskega inženirstva, teorije vodenja, informacijske tehnologije ter v zadnjem času tudi

umetne inteligence. Zanimanje za metode umetne inteligence, predvsem ekspertne sisteme na področju vodenja temelji na dejstvu, da klasične metode niso primerne za reševanje celotnega spektra problemov avtomatizacije, predvsem na področjih, kjer človek s svojimi izkušnjami in sposobnostjo kvalitativnega sklepanja še vedno igra pomembno vlogo. Ta ugotovitev velja za višje nivoje vodenja, kjer gre predvsem pri razvoju nadzornih sistemov za tesno prepletanje analitičnega in heurističnega znanja. Prednost ekspertnih sistemov pred klasičnimi orodji se kaže v možnosti integracije analitičnega in kvalitativno opisljivega znanja.

Zaradi naraščajočega zanimanja za ekspertne sisteme so se na tržišču začela pojavljati orodja, ki podpirajo

razvoj ekspertnih sistemov. Taka orodja (t.i. *ekspertne lupine*) vsebujejo mehanizme za procesiranje znanja in uporabniški vmesnik ter omogočajo razvijalcu, da se osredotoči le na razvoj baze znanja. Prve komercialno dostopne ekspertne lupine, ki so temeljile na izhodiščih t.i. ekspertnih sistemov prve generacije, zaradi pretežne omejenosti na opis kvalitativnih izrazov v praksi niso bile uspešne (npr. Level 5). Zaradi vedno večjih zahtev po zanesljivosti industrijskih sistemov vodenja se je izrazilo pojavila potreba po uporabi orodij z močnimi uporabniškimi vmesniki, bogato izraznostjo jezika in arhitekturami, namenjenimi za delo v realnem času. Naštete zahteve so bile kasneje upoštevane pri razvoju ekspertnih lupin druge generacije, med katerimi naj omenimo samo nekatere [1]: RTAC, CogSys, RTWorks in druga. Tudi G2 (GenSym Co.) izhaja iz te skupine, vendar tako po funkcionalnosti kot tudi zmogljivosti presega omenjena orodja. V sedanjem času je eno najmočnejših orodij na področju razvoja ekspertnih sistemov v realnem času. V G2 so uspeli združiti koncepte razvoja ekspertnih sistemov, klasičnega programiranja in objektne tehnologije. Poleg tega je G2 tesno povezan tudi z usmeritvami razvoja mrežnih sistemov odjemalcev/strežnikov.

Taka zasnova omogoča razvoj široke palete aplikacij s področja vodenja. V praksi obstaja največ zanimanja za razvoj sistemov v G2, ki rešujejo probleme vodenja industrijskih procesov z vključevanjem sodobnih metod, spremljanja in analize alarmov, nadzora in odkrivanja napak v tehničnih sistemih ter sistemov za podporo logistiki in odločanju. O takih sistemih, ki omogočajo procesiranje analitičnega in kvalitativno opisljivega znanja, govorimo kot o inteligentnih sistemih.

Na odseku za računalniško avtomatizacijo in regulacije se ukvarjamo z načrtovanjem računalniških sistemov vodenja. Pri tem se velikokrat srečujemo s problemi, ki jih ni mogoče opisati le na osnovi analitičnih rešitev. Osnovna težava je v tem, da je mnoge procese težko predstaviti z ustreznim matematičnim modelom bodisi zaradi pomankljivega znanja ali zaradi njihove kompleksnosti. To spoznanje nas je pripeljalo do zahtev po dovolj zmogljivem orodju, ki bi omogočalo tudi vključevanje kvalitativnega znanja. Do sedaj smo G2 uspešno uporabili za razvoj inteligentnih sistemov za analizo alarmov in odkrivanje napak. Izkušnje pridobljene pri delu z orodjem pa so nas pripeljale do ugotovitve, da je mogoče njegovo uporabnost razširiti tudi na področje simuliranja zahtevnih procesov, katerih modeli združujejo tako zvezne kot tudi diskretne dogodke.

Namen članka je predstaviti objektno - orientirano orodje za razvoj inteligentnih sistemov - G2 (GenSym Corporation). Ker želimo poudariti predvsem njegovo uporabnost pri razvoju sistemov na področju vodenja,

bomo lastnosti tega okolja prikazali na primeru razvoja simulatorja procesa precipitacije sintetičnega amorfnega SiO₂.

2. OKOLJE ZA RAZVOJ INTELIGENTNIH SISTEMOV - G2

G2, GenSym Corporation [2] je grafično, objektno - orientirano razvojno okolje, ki podpira razvoj inteligentnih sistemov na osnovi množice objektov in njim pripadajočim razredom. Aplikacije razvite v G2 se lahko izvajajo v okoljih UNIX, VMS ali Windows, na delovnih postajah (Digital, HP, SUN, IBM, Silicon Graphics) ali zmogljivejših osebnih računalnikih. Razvoj okolja G2 izhaja iz potreb po učinkovitem reševanju problemov vodenja od tehnoloških procesov na najnižjih nivojih do celotnih proizvodnih obratov. Lastnosti G2 pridejo do izraza pri razvoju nadzornih sistemov, sistemov za spremljanje kvalitete in varnosti, za razporejanje opravil, pri optimiranju ter poslovnemu odločanju na najvišjih nivojih vodenja. Da se G2 učinkovito uporablja tudi v praksi pa kažejo tudi številne razvite aplikacije v industriji, predvsem na področjih kemijske in petrokemijske industrije, avtomobilske industrije, telekomunikacij, transporta, kosovne proizvodnje ter bančništva.

Prednosti G2 pred klasičnimi orodji, ki se uporabljajo za reševanje problemov vodenja na vseh področjih (npr. Factory link za razvoj nadzornih sistemov) se kažejo predvsem v moči izraznosti jezika, ki poleg analitičnih rešitev omogoča tudi vključevanje znanja, ki ga je mogoče opisati s pomočjo kvalitativnih pravil in izrazov. Struktura G2 temelji na treh med seboj povezanih paradigmah [1,2,7]:

- objektno orientirane predstavitve znanja,
- sklepanja na osnovi produkcijskih pravil in
- postopkovnega programiranja.

Mehanizmi, vgrajeni v G2 okolje, ki služijo uporabniku pri razvoju aplikacij, so naslednji [2]:

- baza znanja,
- mehanizmi, ki omogočajo sklepanje v realnem času,
- procedurni interpreter,
- dinamični simulator,
- objektno - orientirano grafično razvojno okolje, s podporo urejevalnika besedil,
- uporabniški vmesnik in vmesniki za neposreden dostop do zunanjih podatkovnih naprav. Okolje vključuje mrežno in integracijsko podporo za programibilne logične kontrolerje (PLC), porazdeljene sisteme vodenja (DCS) in podatkovne baze (Oracle, Sysbase, DecRdb).

Osnovna filozofija prisotna v okolju G2 izvira iz razvoja ekspertnih sistemov, kjer je znanje o specifičnem problemu (t.i. *baza znanja*) ločeno od mehanizmov, ki omogočajo procesiranje tega znanja (t.i. *mehanizmi sklepanja*).

Na osnovi tega dejstva, aplikacije razvite v okolju G2 poimenujemo tudi *baza znanja*. Razvoj baze znanja poteka na t.i. delovnih površinah (angl. *workspace*) v G2 jeziku (glej Slika 1) katerega sintaksa je podobna naravnemu jeziku. Celotna aplikacija je porazdeljena na posamezne delovne površine, ki so organizirane v hierarhično strukturo.

G2 podpira tudi modularnost. Celotno bazo znanja lahko razbijemo na posamezne dele (module), ki jih povezujemo v posamezne hierarhične strukture. Prednosti modularnega razvoja baze znanja izstopajo predvsem pri reševanju kompleksnih problemov, pri čemer je vsak modul neodvisna enota, ki ga razvijamo in testiramo.

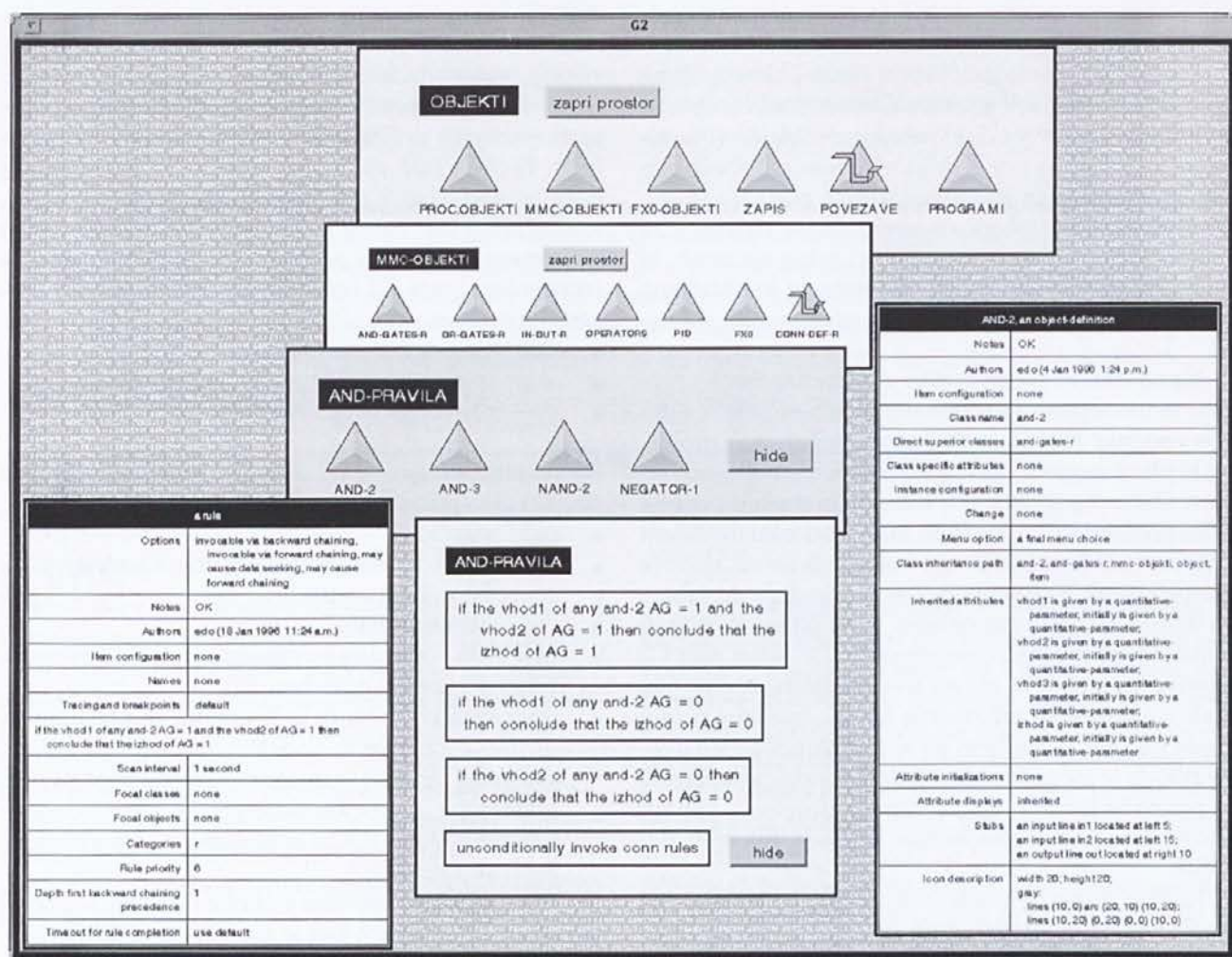
Objektna orientiranost G2

Osnovni gradniki vsake aplikacije v G2 so objekti. Vsak objekt je razpoznaven z množico *atributov* (internih spremenljivk objekta) in množico *metod*, ki operirajo nad objektovimi atributi.

Objekti, ki imajo podobne lastnosti, se združujejo v

razrede. Razred torej predstavlja osnovni vzorec množice podobnih si objektov, kjer se podobnost izraža v enaki množici atributov in metod. Vsak objekt tako predstavlja pojavitev določenega razreda. Definicija razreda (angl. *class*) v G2 je sestavljena iz definicije atributov in definicije metod, ki implementirajo karakteristične operacije za pripadajoči razred. G2 podpira vizualno programiranje, zato je vsak razred razpoznaven prek pripadajoče grafične podobe.

Razredi so organizirani v hierarhično strukturo, v kateri velja pravilo enojnega ali večkratnega dedovanja. Dedovanje omogoča, da se lastnosti (atributi in metode) nadrejenega razreda prenesejo na njegovega naslednika. V primeru večkratnega dedovanja lahko razred podeduje lastnosti vseh razredov iz katerih izhaja. Poleg principa dedovanja sta v okolju G2 prisotna tudi principa enkapsulacije (princip zapiranja podatkov in operacij) in polimorfizma (imensko enake operacije se lahko obnašajo različno v različnih razredih). Nekatere lastnosti G2, kot so definicija razredov, atributi in pravila, so prikazani na sliki 1.



Slika 1: Prikaz definicije razredov, atributov in pravil

Osnovni gradniki vsake aplikacije v G2 so grafično predstavljeni objekti. Objekte lahko razvrstimo v dve skupini:

- na množico objektov, katerih razredi so že vnaprej definirani v programskem okolju G2 (t.i. *sistemiški objekti*) in
- na objekte, za katere definicije razredov poskrbi razvijalec aplikacije (t.i. *uporabnikovi objekti*).

Med sistemske objekte sodijo splošni objekti, spremenljivke in parametri (G2-variable, parameter, G2-list, G2-array), gumbi, dialogi, objekti za prikaz (diagrami, tabele), definicije procedur, pravil in metod. Objekte lahko postavljamo v medsebojne relacije, katere definiramo grafično s t.i. *povezavami G2* ali negrafično preko t.i. *relacij G2*.

Osnovni podatkovni tipi, ki so znotraj G2 dovoljeni za opis atributov in G2 - izrazov, so numerični podatkovni tipi (G2-spremenljivke in G2-parametri), logični izrazi (Boolean ali fuzzy), simbolični podatkovni tipi, tekstovni nizi. Poleg trenutne vrednosti posamezne spremenljivke lahko spremljamo vrednosti spremenljivk v vnaprej določenem časovnem intervalu. G2 vsebuje tudi množico vnaprej definiranih funkcij, ki operirajo nad vrednostimi spremenljivk v določenem časovnem intervalu.

Manipulacija z objekti

Obnašanje posameznih objektov lahko opišemo s pomočjo

- pravil,
- procedur ali
- metod.

Na posamezne objekte pa se lahko sklicujemo v izrazih tudi preko definiranih povezav ali relacij.

Mehanizem sklepanja v G2

Kot ostali ekspertni sistemi ima tudi G2 bazo znanja ločeno od algoritmov, ki to znanje uporabljajo. Procesiranje baze znanja v okolju G2 opravlja poseben mehanizem sklepanja (t.i. *G2 Inference Engine*). Ta mehanizem skrbi za pravilno zaporedje izvajanja pravil, procedur in metod v enem ciklu, ki se lahko izvede v:

- v realnem času
- v simulacijskem času in
- "as fast as possible" - v najkrajšem možnem času. Pri uporabi tega načina se nov cikel izvajanja pravil in procedur začne takoj, ko se vsa pravila v prejšnjem ciklu izvršijo.

Pravila, ki so zapisana v bazi znanja se aktivirajo vsak cikel tako, kot smo določili v sistemskih atributih pravila. Najpogosteje se pravila izvršujejo od začetka do konca (t.i. *forward chaining* ali veriženje naprej), kjer najprej z določeno akcijo aktiviramo prvo pravilo, ki poleg ak-

cije vsebuje tudi aktiviranje naslednjega pravila in enako vse do zadnjega pravila. Na ta način se v enem ciklu izvršijo vsa pravila v verigi. Na podoben način se izvede zaporedno aktiviranje od zadnjega konca naprej (t.i. *backward chaining* ali veriženje nazaj) tako, da najprej aktiviramo zadnje pravilo, ki aktivira pravila, ki pogojujejo zadnje, rezultirajoče pravilo. Pravila se lahko izvedejo v vnaprej določenih časovnih intervalih ali asinhrono, ob pojavu določenega dogodka. Istočasno lahko zahtevamo tudi izvajanje skupine pravil, ki sodijo v kategorijo pravil določeno v sistemskem atributu pravila. V vsakem G2 ciklu lahko dobijo posamezne spremenljivke novo vrednost (t.i. *data seeking*) od podatkovnega strežnika, ki je določen v tabeli atributov kot:

- G2 mehanizem sklepanja,
- G2 simulator,
- GSI podatkovni strežnik ali
- kak zunanji podatkovni server (npr. fizična naprava ali podatkovna baza).

G2 simulator

Poleg ostalih že naštetih značilnosti, vsebuje okolje G2 tudi gradnike, ki podpirajo dinamično modeliranje in simuliranje procesov. Vgrajeni G2 simulator in povezljivi grafični objekti omogočajo hitro modeliranje in simuliranje zgrajenega modela. Delovanje simulatorja smo preizkusili pri razvoju simulacijskega okolja namenjenega študiji in razvoju postopkov za zaznavanje in ugotavljanje napak v industrijskih procesih [8].

Standardni vmesnik G2 za povezavo z zunanjimi sistem - GSI

GSI je splošno namenska knjižnica v G2 za razvoj procesnih vmesnikov iz G2 do podatkovnih baz, realnih sistemov vodenja (npr. tehnološki procesi) in drugih podatkovnih virov. Preko GSI lahko omogočimo tudi vključevanje že razvitih programov z uporabo programskih jezikov Pascal, C ali Fortran. Standardni vmesnik GSI je bil uporabljen tudi pri izvedbi povezave ekspertnega sistema za nadzor v G2 in simuliranega procesa, ki se odvija v okolju Matlab/Simulink [4,9]. Realizacija omenjene povezave je le vmesna faza pri postavitvi povezave med sistemom v G2 ter realnim fizičnim procesom.

Opisane gradnike in lastnosti okolja G2 smo uporabili pri razvoju simulatorja procesa za pridobivanje slikativ. V naslednjem poglavju sledi kratka predstavitev primera uporabe programskega orodja G2.

3. RAZVOJ SIMULATORJA V PROGRAMSKEM OKOLJU G2

3.1 UVOD

Opisane značilnosti okolja G2 za razvoj inteligentnih

sistemov bomo prikazali na primeru razvoja simulatorja procesa precipitacije amorfne SiO_2 , ki je namenjen pridobivanju silikatov. Silikati najrazličnejših vrst so zelo iskane industrijske surovine. Tehnologija tega procesa z avtomatiziranim vodenjem (glej Slika 2) je bila razvita na Kemijskem inštitutu ter testirana na laboratorijskem procesu. Preizkušanje in testiranje na samem procesu je ponavadi časovno zamudno in povezano z velikimi stroški. Zato smo se odločili, da z znanjem, pridobljenim pri razvijanju realnega procesa, zgradimo simulator, ki bo v mnogočem poenostavil in pocenil razvijanje podobnih procesov in odprl nadaljne možnosti raziskav proizvodnje najrazličnejših silikatov.

Proces je semi šaržen, zato so prisotni tako zvezni, kot tudi diskretni dogodki. Zaradi take narave procesa ga uvrščamo med zahtevne procese, ki jih je težko simulirati s klasičnimi simulacijskimi orodji (npr. Matlab/Simulink). Izrazna moč jezika G2 omogoča opis tako zveznih kot tudi diskretnih dogodkov, kar je prevladalo v odločitvi za razvoj simulatorja v okolju G2.

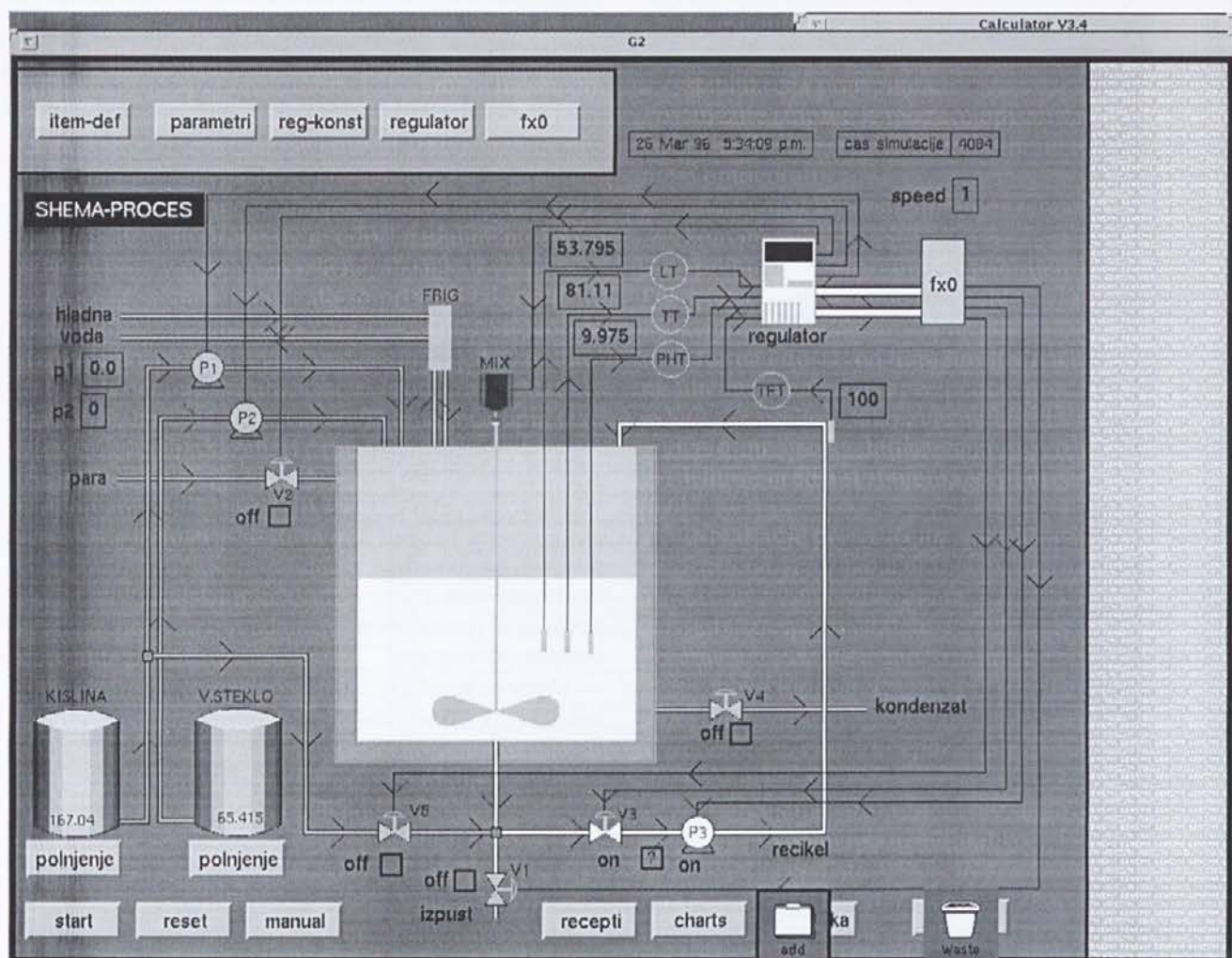
Celoten proces lahko časovno in funkcionalno razdelimo na štiri stopnje, ki so podrobneje opisane v [5,6]:

- ustvarjanje prenasajenja in nukleacije,
- formiranje in stabiliziranje nukleusov,
- rast in zorenje precipitata,
- stabilizacija precipitata.

Posamezne stopnje je mogoče opisati vsebinsko neodvisno, kar je tudi osnova razvoja simulatorja. Simulator mora omogočati:

- enostavno eksperimentiranje s tehnološkim postopkom in
- spreminjanje strukture in parametrov vodenja.

Poleg tega so pomembne tudi zahteve, ki se nanašajo na upravljanje simulatorja. Uporabnik želi uporabljati simulator na enostaven in razumljiv način. Nenazadnje ne smemo pozabiti na specifičnost procesa precipitacije SiO_2 , ki vsebuje tako zvezne kot tudi diskretne dogodke. Omenjene zahteve pogojujejo izbor takega programskega orodja, ki bo omogočal razvoj kvalitetnega



Slika 2: Grafični prikaz simulatorja

simulatorja, ki bo tako po funkciji, kot po ostalih kriterijih ustrežal postavljenim zahtevam. Na osnovi povečanega mora uporabljeno programsko orodje omogočati:

- razvoj uporabniško prijaznega vmesnika,
- modularnost (možnost razbitja simulatorja na posamezne module),
- fleksibilnost (enostavnost spreminjanja strukture simulatorja),
- integracijo analitičnega in hevrističnega (kvalitetno opisljivega) znanja o procesu.

Na osnovi definiranih zahtev smo ugotovili, da je simulator smiselno razviti v okolju G2, ki ponuja široko paleto možnosti pri razvoju uporabniško prijaznega grafičnega vmesnika in opis kvalitativnih pojavov.

3.2 REALIZACIJA SIMULATORJA

Simulator je zgrajen iz treh vsebinsko neodvisnih modulov (glej Slika 3). To so [6]:

- shema procesa,
- shema regulatorja MMC-90 in
- shema sekvenčnega krmilnika.

Shema procesa

Modul *Shema procesa* predstavlja jedro simulatorja skupaj z osnovnimi algoritmi za opis obnašanja procesa. Uporabniku je proces predstavljen grafično, tako kot je definiran prek tehnološke sheme, kjer so prikazane komponente procesa in elementi vodenja. Definicija osnovnih razredov je razvidna iz slike v prejšnjem poglavju (glej Slika 1).

Razvoj simulatorja temelji na objektno zasnovanem pristopu. Na sistem gledamo kot na množico objektov (fizikalne komponente procesa) in njihovih medsebojnih povezav. Fizikalne komponente procesa (delovne posode, črpalke, ventili, itd.) so predstavljene kot objekti, katerim smo kot attribute pripisali veličine, ki jih je mogoče meriti ali ocenjevati (nivo v posodi, napetost na

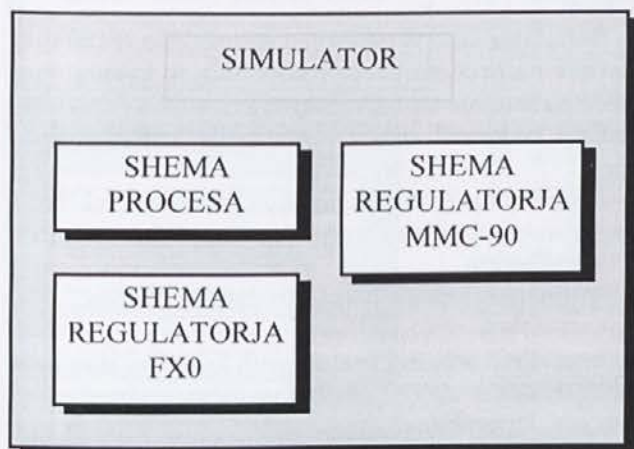
črpalki, pretok skozi ventil, itd.). Obnašanje posameznih objektov je opisano z uporabo pravil, metod ali procedur na osnovi poznavanja osnovnih fizikalnih zakonov in poznavanja topologije procesa (kako so komponente povezane med seboj). Simulator, kakor ga vidi uporabnik, je prikazan na sliki (glej Slika 2), diagram poteka algoritmov, ki so realizirani v tem modulu pa na sliki spodaj (glej Slika 4).

Uporabnik lahko vizualno na osnovi animacije in prek vrste grafičnih prikazov procesnih veličin sledi obnašanju procesa. Z dostopom do posebnega objekta *recept* spreminja parametre procesa, ki vplivajo na potek tehnološkega postopka. Ti parametri so naslednji: želene vrednosti temperature, pH vrednosti in vrednosti nivoja reakcijske mešanice v določenih časovnih intervalih.

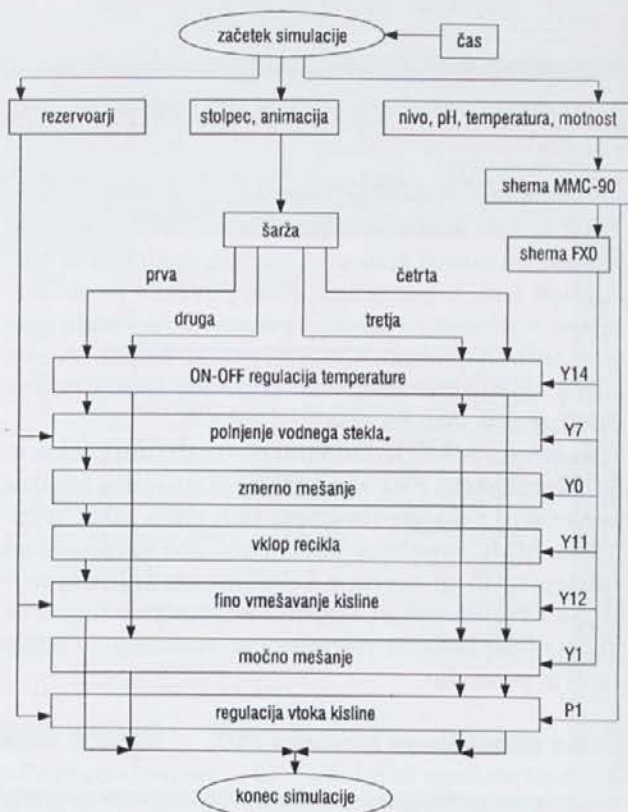
Medtem, ko je opisani modul jedro celotnega simulatorja, pa sta druga dva namenjena implementaciji vodenja, kjer je potrebno tako zvezno (regulator MMC-90) kot tudi sekvenčno vodenje (sekvenčni krmilnik). Struktura vodenja je prikazana z bločno shemo, kot je to v navadi v regulacijski tehniki.

Shema regulatorja MMC-90

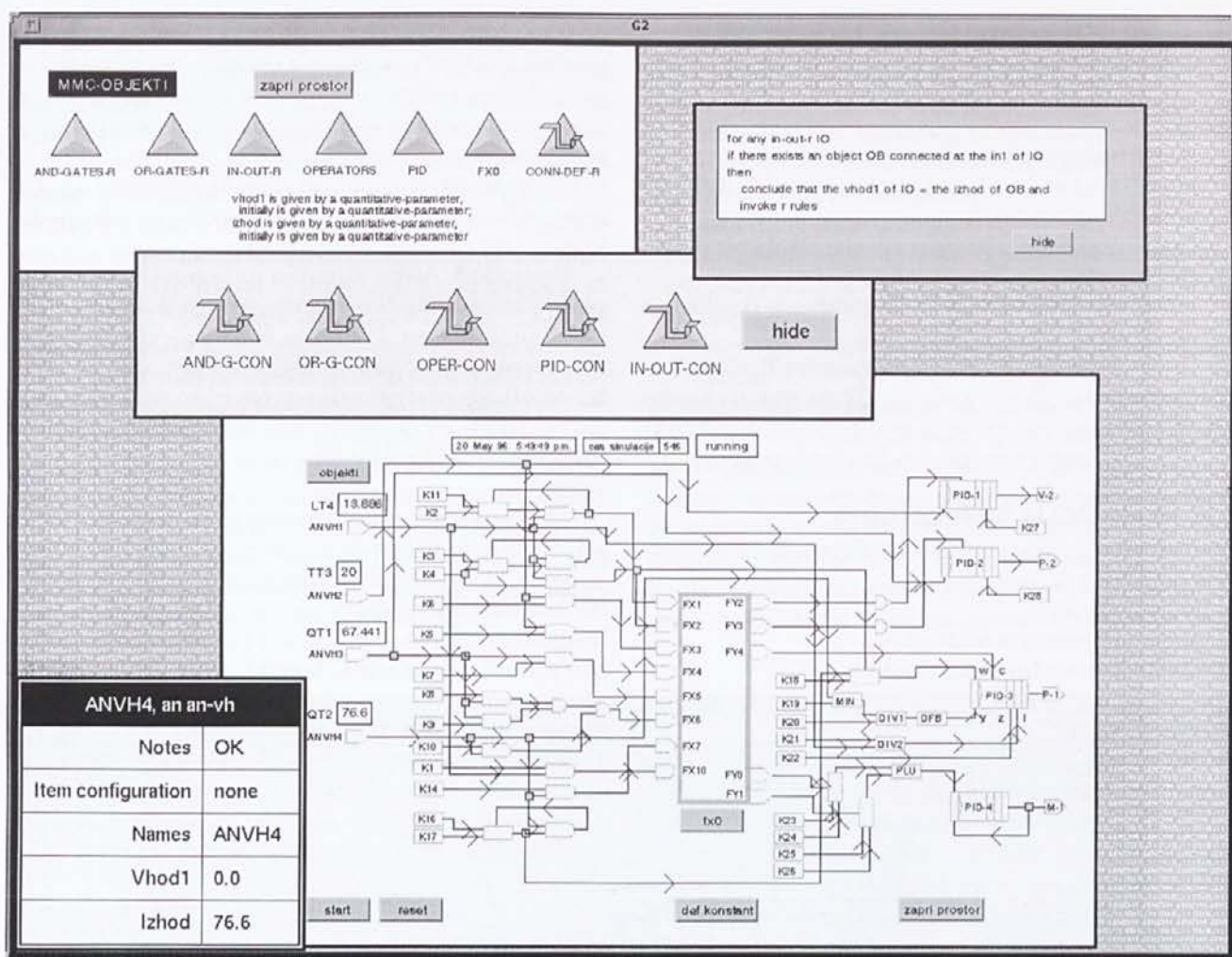
Osnovni logični elementi so definirani kot posamezni razredi. Delovanje regulatorja je tako opisano v obliki bločne sheme z uporabo objektov izpeljanih iz razredov logičnih elementov ter povezav med njimi.



Slika 3: Moduli simulatorja



Slika 4: Prikaz izvedbe celotnega simulatorja



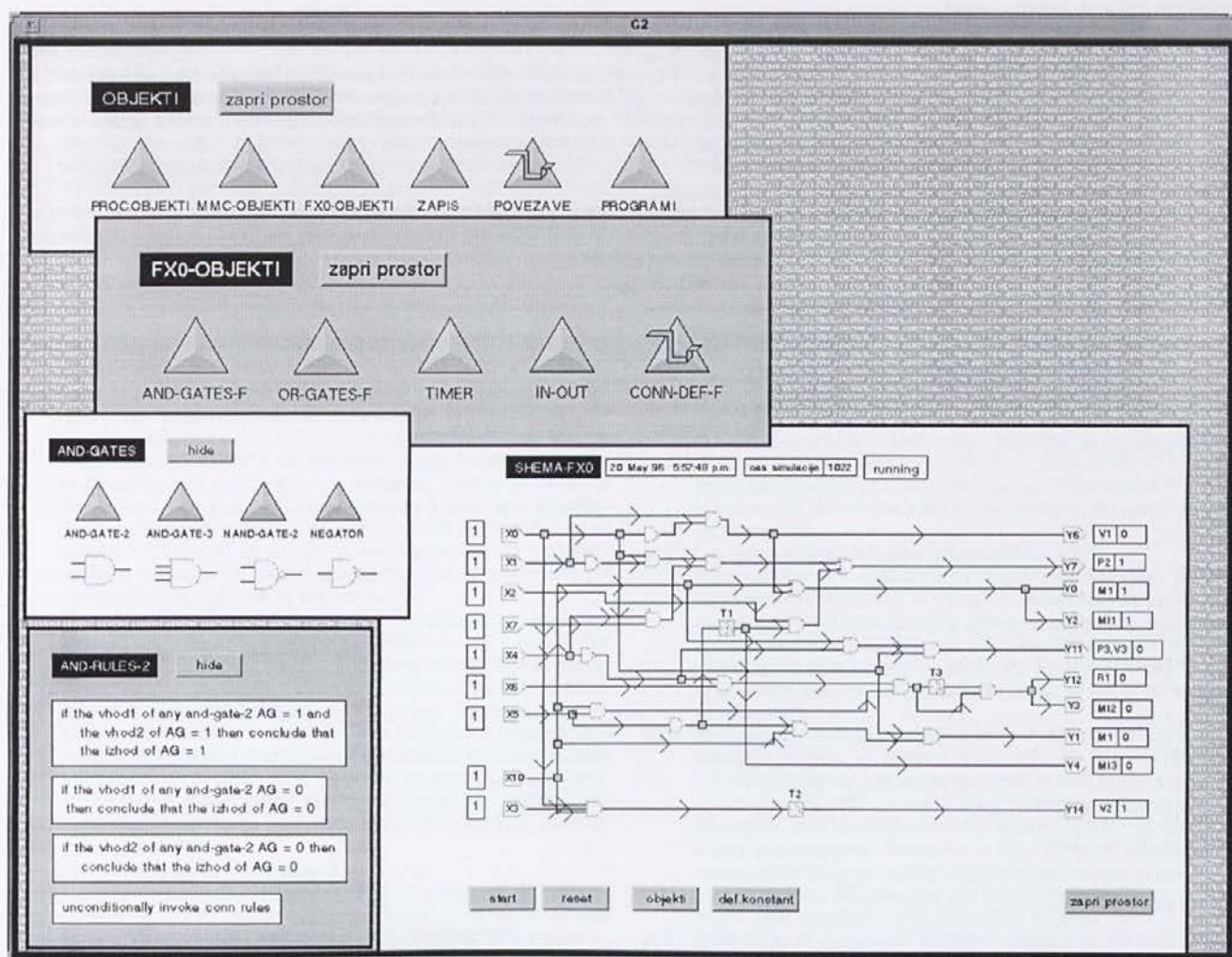
različnih struktur vodenja. Poleg tega je delo s simulatorjem enostavno, dogajanje v procesu pa prikazano s pomočjo grafične animacije in različnimi diagrami procesnih spremenljivk. Glede na rezultate vrednotenja lahko ocenimo, da simulator zadošča postavljenim zahtevam in da se lahko uporablja tudi v praksi, predvsem v fazi načrtovanja in testiranja proizvodnje novih vrst silikatov.

4. ZAKLJUČEK

Okolje G2 uvrščamo med zmogljivejša orodja za razvoj ekspertnih sistemov v realnem času. Filozofija pri razvoju orodja G2 je temeljila na ideji integriranega okolja, v katerem so združeni koncepti razvoja ekspertnih sistemov, klasičnega programiranja in objektne tehnologije. Poleg tega lahko G2 opišemo kot okolje z dobro razvitim uporabniškim vmesnikom, bogato izraznostjo jezika in arhitekturami, namenjenimi za delo v realnem času. Poleg tega sledi G2 tudi usmeritvam razvoja po-

razdeljenih mrežnih sistemov. Prednosti G2 pred klasičnimi orodji se kažejo predvsem v moči izraznosti jezika, ki poleg analitičnih rešitev omogoča tudi vključevanje znanja, ki ga je mogoče opisati s pomočjo kvalitativnih pravil in izrazov.

V članku opisane značilnosti okolja G2 ponujajo sredstvo za reševanje mnogih problemov s področja vodenja. Problematika vodenja ima namreč svoje značilnosti, ena izmed njih je tudi tesno prepletanje analitičnega in kvalitativno opisljivega znanja. Prav zaradi tega lahko govorimo o uporabi okolja G2 za razvoj nadzornih sistemov, sistemov za spremljanje kvalitete in varnosti, razporejanje opravil, optimiranju ter poslovnemu odločanju na najvišjih nivojih vodenja. Velikokrat govorimo o uporabnosti orodja G2 za razvoj inteligentnih sistemov. Na uporabnost G2 tudi v praksi pa kažejo številne razvite aplikacije v industriji, predvsem na področjih kemijske, petrokemijske in avtomobilske industrije, telekomunikacij, transporta, kosovne proizvodnje ter bančništva.



Slika 6: Bločna shema krmilnika FX0 s pripadajočimi razredi in pravili

V članku smo uporabnost G2, skupaj z njegovimi lastnostmi, prikazali na razvoju simulatorja procesa sintetičnega amorfne SiO_2 . Zaradi prisotnosti tako zveznih, kot tudi diskretnih dogodkov, ga je težko modelirati s klasičnimi simulacijskimi orodji. Z simulatorjem smo želeli poenostaviti in poceniti razvijanje silikatov, ki sodijo v skupino industrijskih surovin. Na primeru razvitega simulatorja smo pokazali, da se uporabnost G2 izraža predvsem pri reševanju problemov vodenja, kjer je prisotnost človeka s svojim odločanjem še vedno velika in zato poleg analitičnih rešitev obstaja veliko kvalitativno opisljivega znanja. V strukturi vodenja to velja predvsem za višje nivoje vodenja: nadzornih sistemov, optimiranja in odločanja.

LITERATURA

- [1] Arzen K.E., A survey of Commercial Real-time Expert Sistem Environments, Proc. IFAC/IFIP/IMACS International Symposium on AI in Real-Time Control, Delft University of Technology, Delft, pp. 611-618, 1992.
- [2] Gensym Corporation, Reference Manual, Gensym Corporation, Cambridge MA, 1995.
- [3] Mc. Ghee J., M. J. Grimbale in P. M. Mowforth. Knowledge-based systems for industrial control. Peter Peregrinus Ltd., London, 1990.
- [4] Kavčič M., Razvoj integriranega okolja za vodenje procesov, Diplomsko naloga, Fakulteta za računalništvo, Ljubljana, 1995.
- [5] Suša J., Načrtovanje procesa precipitacije s kemijsko reakcijo, Magistersko delo, Fakulteta za naravoslovje in tehnologijo, Ljubljana, 1993.
- [6] Skobe, A. Diplomsko naloga, Fakulteta za elektrotehniko, Ljubljana, 1995.
- [7] Vaesen N.H.J.G., A.R.M. De Feyter, G2 - The development environment for Real - Time expert systems. Proc. IFAC/IFIP/IMACS International Symposium on AI in Real-Time Control, Delft University of Technology, Delft, 1992.
- [8] Žnidaršič A., Zaznavanje in ugotavljanje napak v industrijskih procesih na osnovi večnivojskih modelov pretoka.. Magistrska naloga, Fakulteta za računalništvo, Ljubljana, 1994.
- [9] Žnidaršič, A. Đ.Juričić in M.Kavčič. Hybrid Environment for Prototyping of Intelligent Supervisory Systems. Elektrotehniški vestnik, 62 (3-4): 232 - 242, Ljubljana, 1995.

◆
Alenka Žnidaršič je diplomirala leta 1991 in magistrirala leta 1994 na fakulteti za elektrotehniko in računalništvo Univerze v Ljubljani. Trenutno je v pripravi doktorske naloge in je kot mlada raziskovalka zaposlena na Odseku za računalniško avtomatizacijo in regulacije Instituta Jožef Stefan v Ljubljani. Njeno raziskovalno področje obsega razvoj postopkov za zaznavanje in odkrivanje napak v tehničnih sistemih z uporabo metod umetne inteligence. Rezultate dela je predstavila v člankih domačih in tujih strokovnih revij ter konferenc.

◆
Juš Kocijan je diplomiral leta 1988, magistriral leta 1990 in doktoriral leta 1993, vse na fakulteti za elektrotehniko in računalništvo Univerze v Ljubljani. Je docent na fakulteti za elektrotehniko v Ljubljani. Objavlja številne članke v domačih in tujih strokovnih revijah in konferencah. Njegovo raziskovalno področje je načrtovanje vodenja multivariabilnih sistemov.

◆
Andrej Skobe je diplomiral leta 1996 na fakulteti za elektrotehniko Univerze v Ljubljani. Diplomsko nalogo s področja razvoja simulatorja procesa precipitacije amorfne SiO_2 je izdelal na Odseku za računalniško avtomatizacijo in regulacije Instituta Jožef Stefan v Ljubljani. Trenutno je zaposlen v podjetju Iskra sistemi, d.o.o. v Ljubljani.