

PROGRAMIRANJE ROBOTA ZA POSLOVE ZAVARIVANJA U EDUKACIJI STUDENATA

PROGRAMMING OF ROBOT IN WELDING PROCESS FOR THE STUDENTS EDUCATION

Tomislav ŠARIĆ¹⁾, Mario SUVAJAC²⁾, Goran KOLESARIĆ¹⁾

Ključne riječi: Robot, Programiranje, Zavarivanje, Generičko sučelje

Key words: Robot, Programming, Welding, Generic interface

Sažetak: U radu se daje pregled robotskih sustava instaliranih u laboratorijima Strojarskog fakulteta u Slavonskom Brodu. Dana je struktura robotskih sustava s programskom podrškom. Robotski sustavi se koriste za znanstveni rad te edukaciju studenata. U ovom radu se posebno opisuju primjeri programiranja robota za poslove zavarivanja. Uz prikaz problema kod programiranja na različitim programskim jezicima daje se i moguće rješenje kroz razvoj generičkog sučelja za programiranje robota. Predloženo rješenje je razvijeno na bazi prototipa te se pokazalo jednostavnim za korištenje te dobra osnova za nastavak istraživanja u ovom području.

Abstract: The article shows the review of robot systems installed in laboratories at Mechanical Engineering Faculty in Slavonski Brod. The structure of robot systems and program support is given. Robot systems are used for scientific work and student education. The paper describes examples of programming of robot in welding process. Beside programming problems in different programming languages, the solution based on generic interface for robot programming will be given. Given solution is developed on the prototype base and shows simplicity in used and possibility of further investigation in problem domain.

¹⁾ Strojarski fakultet, Trg I. B. Mažuranić 2, Slavonski Brod

²⁾ Trg Ante Starčevića 11, Donji Miholjac

1. UVOD

Industrijska primjena pokazala se izvrsnim područjem za razvoj robotike. U početku razvoja robota, primjena u industriji se odnosila na slične ponovljive zadatke. Kada bi se za industrijskog robota pripremio neki zadatak, isti bi izvjesno vrijeme ostajao nepromijenjen. Razvojem su roboti postajali sve moćniji, sa više senzora, inteligentniji i sa jeftinijim komponentama. Kao rezultat razvoja, roboti su se mogli maknuti iz kontroliranih industrijskih okruženja u nekontroliranu uslužnu okolinu kao što su domovi, bolnice i radna mjesta gdje obavljaju različite zadatke od usluga dostave do zabave. Industrijski roboti se prema ISO standardu definiraju kao "automatski upravljeni, reprogramabilni, višenamjenski manipulatori programabilni u tri ili više osi" [2]. Područje industrijske robotike se praktičnije može definirati kroz istraživanje, projektiranje i korištenje robotskih sustava za proizvodnju. Sa stajališta primjene industrijskih robota u proizvodnji mogu se podijeliti po različitim kriterijima a kao jedan od njih, navodi se podjela u četiri skupine i to: roboti za opsluživanje; tehnološki roboti, montažni roboti i merni roboti.

Od navedenih skupina industriojskih robota za opis u ovome radu posebno se ističu tehnološki roboti. Tehnološki roboti su opremljeni odgovarajućim alatkama kao što su klješta za zavarivanje, pištolj za zavarivanje, pištolj za raspršivanje boja ili premaza, brusilica i slično. Područje primjene tehnoloških robota su postupci zavarivanja (elektrootporno, elektrolučno); površinske zaštite (bojanje, nanošenje podloge, zaštiti od hrđe); brušenje, poliranje, skidanje srha; nanošenje ljepila, nanošenje brtvila i slično.

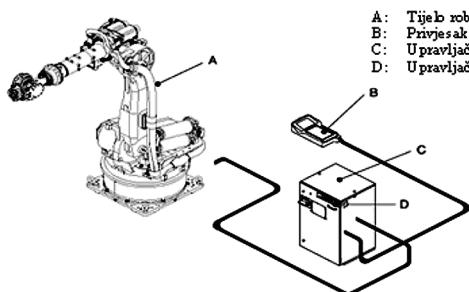
Danas se roboti najčešće koriste za elektrootporno, i to točkasto zavarivanje u automobilskoj industriji. Od robota se zahtjeva brzina i točnost pozicioniranja. Prednost robota u odnosu na čovjeka je kvaliteta i ponovljivost. Bitno drugačiji su zahtjevi pri lučnom zavarivanju i to pretežno MIG-MAG postupci. Tipična je primjena pri zavarivanju napr. segmenata brodskog trupa. Ovo je kvalificirani posao s visokim zahtjevima za čovjeka i robota. Radni uvjeti za ljudе su obično nehumani (položaj tijela, odbijesci, plinovi i slično), a zahtjeva se visoka točnost. Robot za zavarivanje ima pištolj a potreba je isti opremiti senzorima. U program mora biti uključeno upravljanje ne samo putanjom, već i brzinom pištolja, strujom i naponom zavarivanja te kontinuiranim dovođenjem žice (dodatnog materijala) uz osiguranje "njihanja" pištolja da bi se ostvarila dobra popuna žljeba materijalom.

Najpoznatiji proizvođači industrijskih robota su slijedeće tvrtke: Adept, Kuka, ABB, Yaskawa-Motoman, Fanuc, Nachi, Kawasaki, Mitsubishi, i drugi.

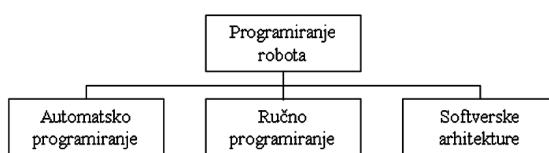
2. PROGRAMIRANJE ROBOTA

Jedan od osnovnih zahtjeva koji se postavljaju pred robote je programibilnost, tj. mogućnost da se robot za relativno kratko vrijeme reprogramira za izvršavanje željenog zadatka [9]. Važni dio robotskog sustava je njegova upravljačka jedinica. Upravljačka jedinica je digitalno računalo koje omogućuje upravljanje kretanjem manipulatora, obradu informacija dobivenih iz senzorskog sustava i upravljanje perifernim jedinicama. Rad upravljačke jedinice odvija se prema unaprijed pripremljenom robotskom programu, tj. nizu instrukcija kojima se opisuju operacije koje robot treba izvršiti tijekom rada, dok ih upravljačka jedinica interpretira i upravlja radom. Osnovna konfiguracija profesionalnog robotskog sustava dana je na slici 1. Većina robota ima kompleksna sučelja koja uključuju tekstualni programski jezik sa vrlo malo apstrakcija više razine. Programiranje robota na niskoj razini nije prihvatljivo za većinu prosječnih korisnika, stoga su potrebni sustavi koji korisniku omogućuju željenu razinu kontrole nad zadacima koje robot obavlja. Prema novijim istraživanjima [1] programiranje robota dijeli se na automatsko programiranje, ručno programiranje i softverske arhitekture (slika 2). Prva

dva načina programiranje čine razliku u metodi koja je korištena pri programiranju, što je vrlo važno za korisnike i programere. U automatskim sustavima za programiranje korisnik/programer ima vrlo malu ili nikakvu izravnu kontrolu nad robotskim kodom. U takve sustave se ubrajaju sustavi za podučavanje, programiranje demonstracijom i instruktivni sustavi. Ručni sustavi traže od korisnika/programera da direktno uneše željeno ponašanje robota, obično koristeći vizualni ili tekstualni programski jezik. Softverske arhitekture su važne za sve sustave za programiranje jer pružaju osnovu za npr. komunikaciju, kao i pristup samim robotima. Ovaj rad je usmjeren na sustave za ručno programiranje.

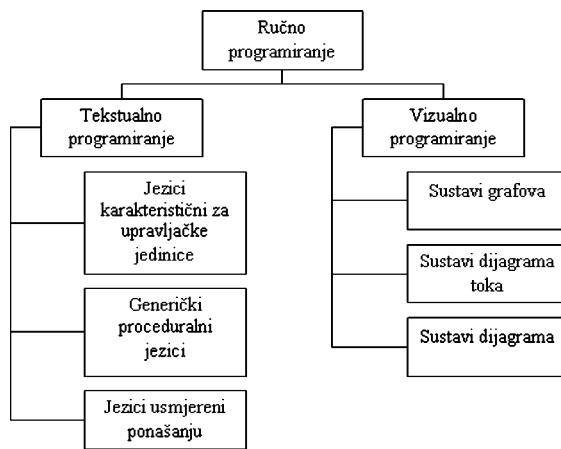


Slika 1. Osnovna konfiguracija robotskog sustava
 (Nachi AX) [6]

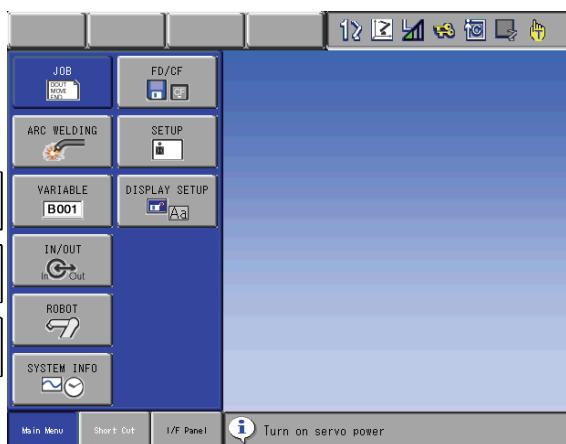


Slika 2. Podjela modernih sustava za programiranje
 robota

Korisnici ovakvih sustava moraju ručno stvarati robotske programe što se obično radi bez samog fizičkog robota. Završeni program se učitava u robot naknadno. Ovo su često neumreženi sustavi za programiranje gdje robot nije prisutan za vrijeme programiranja. Ručnim programiranjem ipak se može zamisliti umreženo upravljanje robotom koristeći interpretativni jezik ondje gdje ne postoje sigurnosne prepreke. Ručni sustavi za programiranje dodatno se mogu podijeliti na tekstualne i vizuelne sustave (slika 3). Jezici specifični za upravljačku jedinicu su početna metoda upravljanja industrijskim robotima, a do danas su ostali najčešće korištena metoda programiranja robota. Svaka robotska upravljačka jedinica ima nekakav oblik strojnog jezika i uz njega obično dolazi programski jezik u kojem se mogu oblikovati programi za željenog robota.



Slika 3. Podjela ručnog programiranja



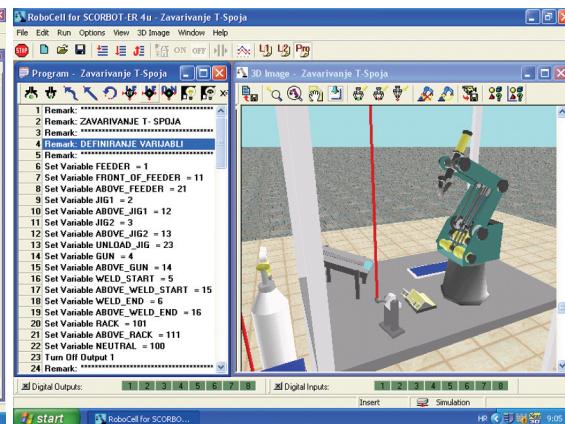
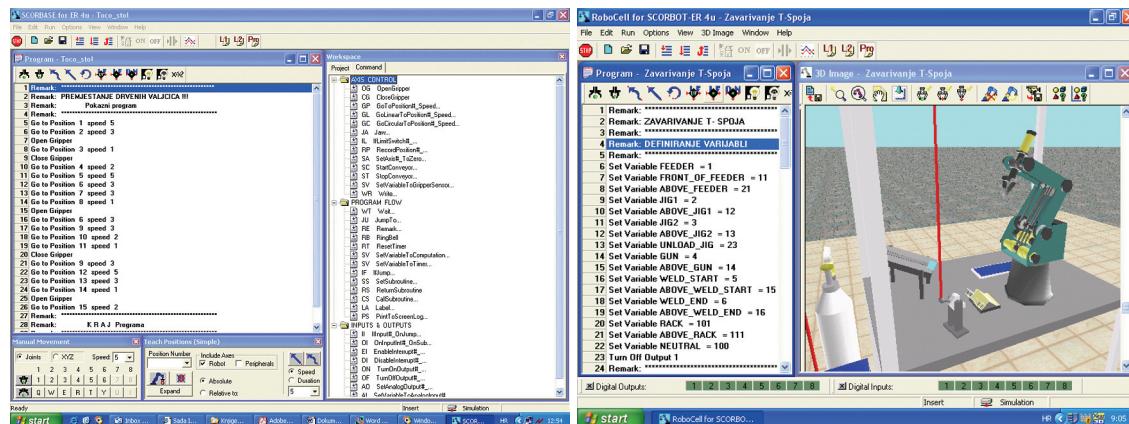
Slika 4. Sučelje za učenje robota-Motoman

Programski jezici obično su vrlo jednostavni, proceduralne paradigme, sintakse koja nalikuje na Basic i jednostavnih naredbi za upravljanje robotom i programske tokom. Dobar primjer je jezik koji se koristi u Nachi-jevim robotima - Slim, koji među ostalim dolazi i sa

dodatnim privjescima za učenje robota. Sučelje na privjescima za učenje je vrlo slično a na slici 4. prikazano je sučelje za učenje Motoman robota. Proizvođači robota često uz programski jezik nude i sustave za simulaciju (*ABB RobotStudio*, *Workspace5*, *Delmia*, *RoboCell*, *RobCAD* i drugi). To su napredni programski paketi u kojima se može simulirati kretanje robota kako bi se provjerilo postojanje mogućnosti sudara sa ostalim objektima u radnom prostoru robota i kako bi se uvjerili da robot obavlja zadani zadatak. Mnogi programi nude i mogućnost simulacije nekoliko različitih robota istovremeno ili čak čitavih postrojenja [2]. Da bi se omogućilo korištenje više jezika u jednom sustavu za simulaciju koriste se prevoditelji. Jezici za upravljačke jedinice imaju i neke nedostatke. Najveći problem je nedostatak univerzalnog standarda između robota različitih proizvođača. Ako tvornica koristi robote više različitih proizvođača tada će trebati uvježbati svoje programere za programiranje svakog pojedinog robota ili platiti proizvođaču za razvoj potrebnih programa. Obje metode znatno povećavaju vrijeme i trošak koji su potrebni za razvoj novih programa za robote.

3. KRATKI OPIS ROBOTSKIH SUSTAVA INSTALIRANIH NA STROJARSKOM FAKULTETU

U laboratorijima Fakulteta nalaze se instalirana dva robotska sustava: Robot SCORBOT ER 4u na Zavodu za industrijsko inženjerstvo te Robotska ćelija za MIG/MAG zavarivanje MOTOMAN na Zavodu za proizvodno strojarstvo.



3.1 Robot Scrbot ER 4u

Robot SCORBOT ER 4 sa upravljačkom jedinicom, privjeskom za učenje te programskom podrškom za programiranje predstavlja nadomjestak profesionalnih robotskih sustava namijenjenih za edukaciju studenata za programiranje i rad sa robotskim sustavima. Robot je vertikalne strukture sa šest stupnjeva slobode gibanja koji se može koristiti za rješavanje različitih tehnoških zadataka. Robot se putem upravljačke jedinice može povezivati sa različitom periferijom (senzorima, trakastim konvejerima, okretnim stolovima, edukacijskim CNC strojevima, drugim robotima i slično). Prihvatinica robota se također može pričvrstiti i koristiti za specifične tehnoške zadatke pa tako i za poslove zavarivanja. Tehnoške aktivnosti se oblikuju u algoritam te se putem naredbi definiranih u SCOREBASE-u definira program. ScrbBase je programsko sučelje koje je prilagođeno za sustavno i postupno usvajanje naredbi te njihovo učenje. Naredbe su organizirane u stablo naredbi (*Command Tree*),

prikazanom na slici 5. Na Uvodnoj razini (*Level 1*) pojavljuju se samo osnovne naredbe u stablu. Na naprednoj razini (*Level 2*) broj naredbi je povećan. Na profesionalnoj razini (*Pro level*) su dostupne (raspoložive) sve naredbe.

ScorBase naredbe su grupirane unutar tri mape: Grupa za kontrolu i upravljanje osima (*Axix & Control group*), Grupa za upravljanje tokom programa (*Program Flow group*), Grupa koja definira stanje ulaza & izlaza (*Inputs & Outputs group*). Na slici 6. prikazan je robotski sustav Scrbot ER4u (RoboCell) organiziran i prilagođen za edukaciju studenata na poslovima zavarivanja. Na istoj slici prikazan je i dio programskog koda za realizaciju konkretnog tehničkog zadatka.

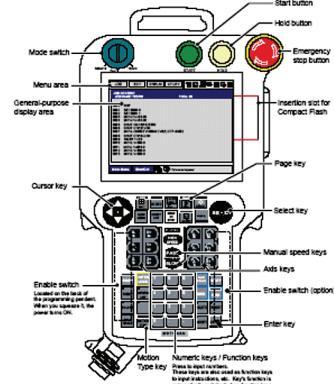
3.2 Robotska ćelija za MIG/MAG zavarivanje MOTOMAN

Robotska ćelija za MIG/MAG zavarivanje MOTOMAN (slika 7) sastoji se od sljedećih elemenata:

- industrijski robot MOTOMAN HP6,
- robotski kontroler NX100 sa jedinicom i privjeskom za programiranje (slika 8),
- rotacijsko nagibni pozicioner DK 22/15,
- robotski zračno hlađeni izvor TPS4000 firme FRONIUS,
- robotski gorionik Robacta MTG 4000,
- robotski pogon za žicu VR 15004R/F,
- nosač gorionika i naletna sklopka Kolisionbox XL i
- mehanički čistač gorionika – THIELMANN.



Slika 7. Industrijski robot MOTOMAN HP6



Slika 8. Privjesak za programiranje

Kao prikaz rada u programiranju i radu sa robotskom ćelijom za zavarivanje prikazan je primjer zavarivanja prstena na ploči. Pristup programiranju zasniva se na definiranju točaka kroz koje treba provesti pištolj za zavarivanje. Ručnim vođenjem robota uz korištenje privjeska i definiranjem položaja željenih točaka (koje se memoriraju u upravljačkoj jedinici) iste se poslije koriste u definiranju programa. Na slici 9. prikazan je položaj robota sa pištoljem za zavarivanje u jednoj od definiranih točaka (označene markerom na ploči uz prsten). Po definiranju i testiranju programa isti se može koristiti za izvršavanje željenih aktivnosti odnosno zavarivnaje (slika 10).



Slika 9. Definiranje položaja za programiranje



Slika 10. Testiran i izvršen program

Za prikazani zadatak oblikovan je programski kod koji je dan je u nastavku.

<pre> //JOB //NAME ZAVARIVANJE PRSTENA //POS ///NPOS 12,0,0,0,0,0 ///TOOL 0 ///POSTYPE PULSE ///PULSE C00000=1,0,0,-1,0,-1 C00001=66753,34900,36595,11593,-12752,-769 C00002=65750,28269,13413,11287,-3251,-2267 C00003=65750,28269,13413,11287,-3251,-2267 C00004=41320,35339,27050,-2070,-7144,-8948 C00005=30024,13731,-231,-8531,5860,-35916 C00006=57719,-18418,-34896,8343,25909,- 72499 C00007=69314,8788,-10002,14667,9762,-97786 C00008=58621,42367,32769,8653,-10870,- 117097 C00009=59235,43828,36004,8616,-11986,- 117129 C00010=59235,47802,53082,9122,-19541,- 115978 C00011=1,0,0,-1,0,-1 //INST </pre>	<pre> ///DATE 2008/12/17 14:43 ///ATTR SC,RW ///GROUP1 RB1 NOP MOVJ C00000 VJ=60.00 MOVJ C00001 VJ=60.00 MOVJ C00002 VJ=60.00 ARCON MOVC C00003 V=11.0 MOVC C00004 V=11.0 MOVC C00005 V=11.0 MOVC C00006 V=11.0 MOVC C00007 V=11.0 MOVC C00008 V=11.0 ARCOF TIMER T=2.00 MOVJ C00009 VJ=60.00 MOVJ C00010 VJ=60.00 MOVJ C00011 VJ=60.00 END </pre>
--	--

4. GENERIČKO SUČELJE ZA PROGRAMIRANJE ROBOTA

Pronlem koji se očituje kod programiranja različitih robotskih sustava odnosno robotskih celija sa različitim jezicima jest potreba za učenjem i znanjem velikog broja jezika. Moguće rješenje očituje se u razvoju generičkog sučelja za programiranje robota tj. programskom sustavu koji bi mogao vršiti pretvorbu programa napisanog u jednom programskom jeziku u drugi.

Na temelju razvijene arhitekture [8] razvijen je prototip programa za generičko prevođenje izvornog koda za dva odabrana izvorna programska jezika (*Rapid* i *Slim*) te sa jednim ciljnim programskim jezikom (*Slim*). Nakon razvoja jezgre za prevođenje, oko jezgre je razvijeno grafičko korisničko sučelje napisano u programskom jeziku C++ koristeći Qt tehnologiju. Kako bi omogućili daljnju analizu u korisničkom sučelju (značajka usporednog označavanja) potrebno je znati odakle pojedini prevedeni dio u ciljnem programu potiče, tj. iz kojeg dijela izvornog koda je nastao. Stoga je potrebno u unutarnjem prikazu prije analize svakom redu izvornog koda izvornog programa dodjeliti brojčanu oznaku na koju se može

pozvati u dalnjim prikazima. Ove aktivnosti se rade direktno iz programa grafičkog sučelja kako bi podaci o odgovarajućim redovima izvornog i ciljnog programa bili dostupni za obradu iz samog grafičkog sučelja. Na osnovi dostupnih specifikacija i referentne dokumentacije proizvodača za odabrane programske jezike napisane su gramatike za sintaksu svakog pojedinog jezika. Gramatika je napisana u obliku koji se može koristiti u FreeTXL implementaciji TXL programske jezike. Stvaranjem gramatike za ove jezike omogućena je provjera sintakse izvornog koda svakog ulaznog programa napisanog u jeziku za koji je napisana gramatika, a može se ostvariti pisanjem jednostavnog TXL programa. Slijedeći korak je ostvarenje pretvorbe čvorova AST-a u XML oznakama označenih čvorova najzanimljivijih dijelova programa. Koristeći tehniku agilnog parsiranja sa selektivnim označavanjem čvorova ovaj korak je ostvaren. Nakon ovoga jedino je preostalo ispravljanje grešaka u nastalom izlazu kako bi izlaz bio valjan po XML standardima za što su napisani dodatni programi za svaki izvorni jezik u TXL-u čija je svrha pronalaženje i ispravljanje nevaljanih znakova između označenih elemenata. Ove programe, iako se razlikuju za svaki jezik, jednostavno je izmijeniti kako bi zadovoljili potrebe ispravljanja bilo kojeg izvornog programa.

Procesor koji se koristi za interpretaciju XQuery programa je Java inačica program otvorenog koda Saxon [7]. Saxon je procesor otvorenog koda za XSLT i XQuery koji je oblikovao Michael Kay. Postoje inačice za Javu i .NET. Za potrebe ovoga rada koristi se inačica Saxonu napisana u Javi. Razlog korištenja ove inačice je portabilnost na različite platforme pošto je Java podržana na velikom broju računalnih platformi.

Sljedeća faza prevodenja započinje sa označenim izvornim kodom u obliku valjanog XML dokumenta. Pošto se radi o XML dokumentu omogućena je primjena ostalih XML tehnologija. Kao što je već u radu obrazloženo, kao odgovarajući alat za rješavanje ovog problema pokazala se XQuery tehnologija. Za prikaze specifične za svaki izvorni jezik razvijeni su alati za pretvorbu prikaza u GML. Ovi alati su napisani u programskom jeziku XQuery [4, 11]. Unutar svakog programa ugrađeni su inteligentni algoritmi koji ulazni XML oblik zapisa analiziraju i obogaćuju dodatnim informacijama za potrebe pretvorbe u GML.

Prilikom dodavanja novih značajki, kako bi se međudjelovanje između programa i korisnika što bolje odvijalo, potrebno je ponekad proširiti GML kako bi se pružile dodatne informacije potrebne za rad nove značajke. Prilikom dodavanja novih značajki pokazuje se sva mogućnost širenja i fleksibilnosti korištenja XML-a kao osnove za prikaz. Dodavanje novih značajki je iznimno lako i brzo. Npr. prilikom proširivanja GML-a za potrebe usporednog prikaza u grafičkom sučelju bilo je potrebno u pojedine naredbe jezika ugraditi samo jedan dodatni atribut broja reda. Samim dodavanjem novih značajki programu, tj. sučelju, sam jezik koji se koristi za neovisni prikaz programa, GML, evoluira te postaje sve robusniji i korisniji. Opremljen novim značajkama jezik je sve lakše primjenjivati na sve većem broju izvornih i ciljnih jezika. Postojanje specifičnosti u novim izvornim i ciljnim jezicima također je razlog da se GML, ukoliko poopćavanje nije moguće, proširi novim značajkama.

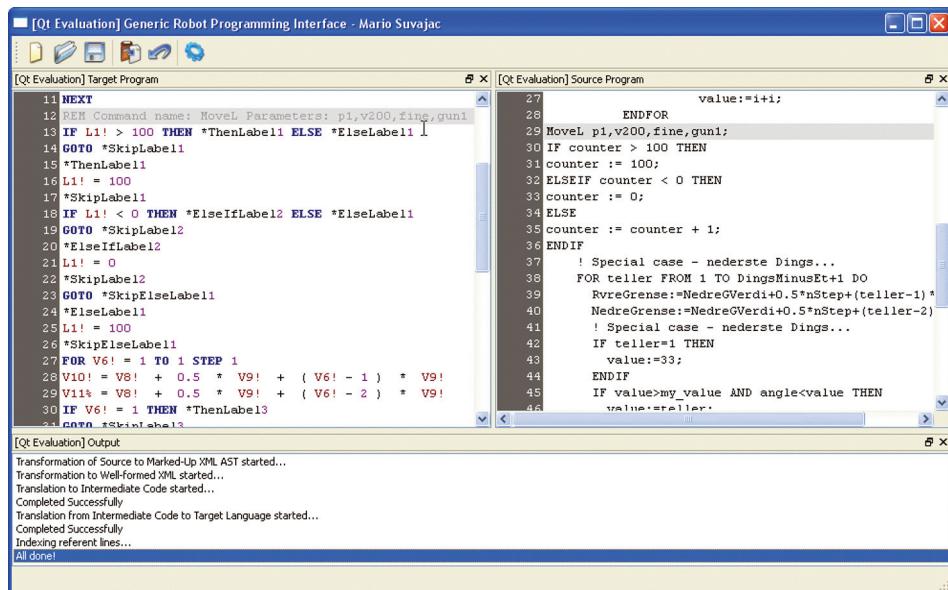
Postupak opisan prilikom sinteze izvornog programa u GML sličan je i prilikom sinteze ciljnog programa. Glavna razlika je postojanje jedinstvenog oblika prikaza programa u koji su u prethodnim koracima pretvoreni izvorni programi. Za svaki ciljni jezik potreban je jedan program za pretvorbu u ciljni jezik. To znači da svi izvorni programi koji su pretvoreni u GML mogu biti pretvoreni u ciljni jezik za koji postoji program za sintezu ciljnog programa.

Qt je premosni prevoditelj, grafička i aplikacijska skupina alata koja omogućava sintezu i pokretanje aplikacija na Windows-u, Mac OS X-u, Linux-u i različitim drugim inačicama Unix-a. Veliki dio Qt-a je posvećen pružanju platformski neovisnog sučelja. Počevši od načina prikaza znakova u memoriji do stvaranja višenitnih grafičkih aplikacija. Iako je Qt razvijen prvenstveno za korištenje u C++ aplikacijama dostupan je i za ostale programske jezike. Razvoj sučelja, kao i čitavog projekta, vođen je idejom jednostavnosti kako bi korisniku odnos sa

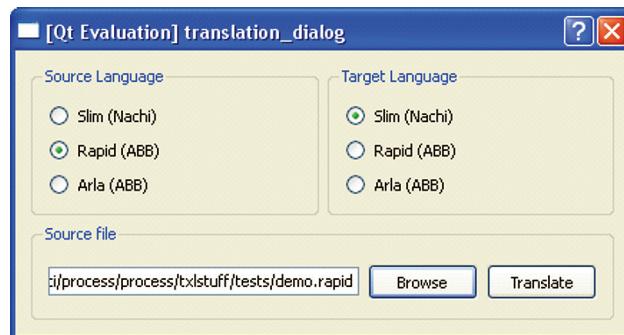
programom bio što jednostavniji i ugodniji.

Značajke korisničkog sučelja (slika 11):

- Automatizirano prevođenje velikog dijela koda. Razvojem intelligentnih algoritama postignut je veliki postotak automatskih prevodivih dijelova koda koji evolucijom GML-a sve više raste. Jednostavnim odabirom izvornog jezika i upućivanjem programa na putanju izvorne datoteke, te odabirom ciljnog programa obavlja se prijevod (*Translate*), slika 12;



Slika 11. Korisničko sučelje



Slika 12. Dijalog za prijevod

- Označavanje neprevedenih dijelova. Iako je velik dio koda prevodiv automatski još uvijek postoje dijelovi koji su specifični za pojedine jezike koje je potrebno prevesti ručno. Stoga je u sučelju omogućeno automatsko označavanje i naglašavanje neprevedenih dijelova;
- Istovremeno označavanje izvora i cilja. Kako bi se korisnik lakše snalazio u prevedenom kodu omogućeno je istovremeno označavanje izvornog i odgovarajućeg prevedenog koda;
- Naglašavanje i bojanje sintakse. Radi lakšeg i ugodnijeg rada sa prevedenim izvornim kodom omogućeno je naglašavanje sintakse;
- Označavanje redova brojevima;

- *Naglašavanje trenutne linije.* Radi bolje uočljivosti trenutnog reda rada, trenutni red naglašen je drugom bojom u odnosu na ostatak područja uređivanja;
- *Povećanje i smanjivanje teksta.*

Za detaljan opis predloženog rješenja i implementiranog modela pogledati u [8].

5. ZAKLJUČAK

U radu se daje uvod i karakteristike programiranja robotskih sustava. Kratkim opisom dan je pregled robotskih sustava instaliranih u laboratorijima Strojarskog fakulteta u Slavonskom Brodu. Dana je struktura robotskih sustava s programskom podrškom. Robotski sustavi se koriste za znanstveni rad te edukaciju studenata. Na opisanim robotskim sustavima daju se konkretni primjeri realizacije poslova zavarivanja, popraćeni prikazom programskog koda. Iz programiranja na različitim programskim jezicima očituje se problem u učenju i korištenju istih. Kroz analizu ovog problema daje se i moguće rješenje kroz razvoj generičkog sučelja za programiranje robota. Predloženo rješenje je razvijeno na bazi prototipa te se pokazalo jednostavnim za korištenje te čini dobru osnovu za nastavak istraživanja u ovom području.

6. LITERATURA

- [1] Biggs, Geoffrey; MacDonald, Bruce: *A Survey of Robot Programming Systems*. Proceedings of the 2003 Australasian Conference on Robotics and Automation (ACRA), 2003., URL: <http://www.araa.asn.au/acra/acra2003/papers/27.pdf>. (05.04.2008.).
- [2] Freund, Eckhard; Ludemann-Ravit, Bernd; Stern, Oliver; Koch, Thorsten: *Creating the architecture of a translator framework for robot programming languages*. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2001., 1, str. 187–192.
- [3] Freund, Eckhard; Ludemann-Ravit, Bernd; Stern, Oliver; Koch, Thorsten: *Creating the architecture of a translator framework for robot programming languages*. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2001., 1, str. 187–192.
- [4] Katz, Howard; Chamberlin, Don; Kay, Michael; Wadler, Philip; Draper, Senise: *XQuery from the Experts: A Guide to the W3C XML Query Language*. Boston; Addison-Wesley Longman Publishing Co., Inc., 2003., 484 str.
- [5] Kovačić, Z.; Bogdan, S.; Krajči, V.: *Osnove robotike*. Zagreb; Graphis, 2002., 202 str.
- [6] Nachi-Fujikoshi Corp.: *Nachi AX Controller Operating Manual: Robot Language (first edition)*. 2004., 87 str.
- [7] Saxonica: XSLT and XQuery Processing: Welcome. *Welcome to Saxon*. URL: <http://www.saxonica.com/documentation/>. (01.06.2008.).
- [8] Suvajac, Mario: Prilog razvoju generičkog sučelja za programiranje robota – Diplomski rad, Voditelj diplomskog rada prof.dr.Tomislav Šarić, Strojarski fakultet, Slavonski Brod, 2008.
- [9] Šurina, T.; Crneković, M.: *Industrijski roboti*. Zagreb; Školska knjiga, 1990., 199 str.
- [10] Vukobratović, Miomir; Stokić, Dragan; Kirčanski, Nenad; Kirčanski, Manja; Hristić, Dragan; Karan, Branko; Vujić, Dragan; Đurović, Milan: *Uvod u robotiku*. Beograd; Institut Mihajlo Pupin, 1986., 377 str.
- [11] Walmsley, Priscilla: *XQuery*. Sebastopol; O'Reilly Media, 2007., 510 str.