

MILAN ZORIC
Zavod za telekomunikacije
ELEKTROTEHNIČKI FAKULTET
Zagreb, Unska 3

SDL RADNA STANICA I RAZVOJ PROGRAMSKE
PODRSKE KOMUTACIJSKIH SISTEMA

SDL TOOL AND DEVELOPMENT OF
SWITCHING SYSTEM SOFTWARE

SAHRZAJ - U radu se razmatraju mjesto i uloga SDL radne stanice u procesu razvoja i održavanja programske podrške komutacijskih sistema. Definiraju se uvjeti izgradnje takvog pomagala, te opisuje okolina na kojoj je bazirana realizacija prvog prototipa.

ABSTRACT - In this paper the role of SDL tool in the process of switching system software development and maintenance is elaborated. Principals upon which tool development is based are defined. The environment in which first prototype is being built is described.

1. UVOD

Prelaskom na ISDN povećavaju se obim i složenost funkcija komutacijskih sistema. Kako bi se unaprijedio proces razvoja i održavanja programske podrške, razvija se skup pomagala koje zajedničkim imenom nazivamo SDL radna stanica. Na ovome mjestu želimo istaci da su problemi razvoja takvog pomagala vrlo široki i da sežu od SDL jezika kao takvog (sintaksa, semantika), do sagledavanja i ovladavanja elemenata sklopovske i programske tehnologije (moguci računarski sistemi, operativni sistemi, grafika, metode leksičke i sintaksne analize itd.). No usprkos tome presudno za izgradnju takvog pomagala je poznavanje komutacijskih sistema, te iskustvo u njihovom razvoju.

2. MJESTO I ULOGA SDL RADNE STANICE

Razvoj i održavanje sistema možemo tretirati kao sekvencu transformacija nad reprezentacijom sistema u cilju dobijanja nove reprezentacije. Cilj svake pojedine transformacije može biti različit: jedna transformacija se provodi kako bi se promijenio jezik u kojem je reprezentacija dana, druga da bi se uključili novi aspekti (pouzdanost, implementacijski detalji). Globalni cilj svih transformacija je da se dobije reprezentacija sistema koja "u praksi" zadovoljava zahtjeve postavljene u specifikaciji. U tom smislu sistem kao takav može biti promatran kao reprezentacija sebe samog.

Inicijalna reprezentacija sistema su zahtjevi na sistem a, ako ovaj trenutak ne promatramo sklopovsku stranu sistema, konačnu reprezentaciju predstavljaju programi koji su ugrađeni u sistem. Između te dvije krajnosti postoji niz manje ili više formalno danih opisa sistema. Za opisivanje sistema na različitim nivoima apstrakcije koristi se SDL jezik [5,6]. Kako je SDL ipak samo jezik potrebno je formirati odgovarajuću metodologiju korištenja SDL-a. Pod time podrazumijevamo slijedeće:

- odabir podskupa elemenata jezika koji se koriste,
- nivo formalizma koji se koristi na različitim nivoima,
- metode i pomoćna sredstva koja se koriste za utvrđivanje ispravnosti specifikacije, za testiranje itd.

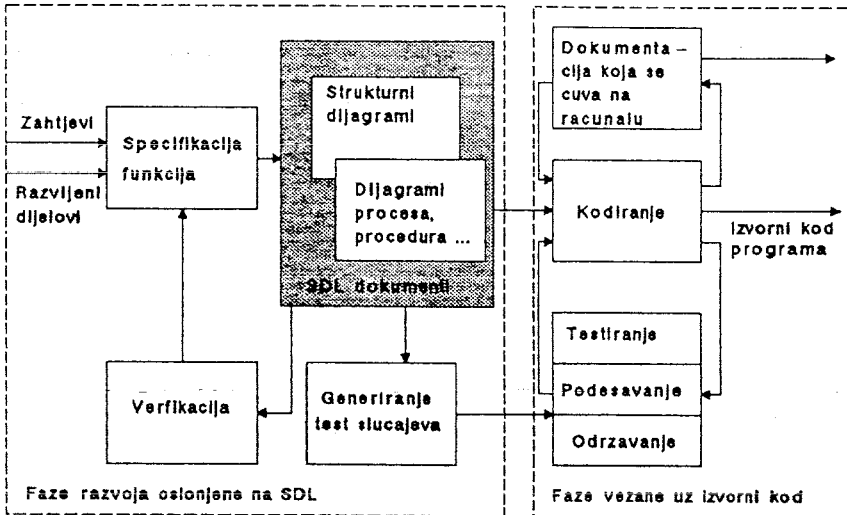
Prilikom razvoja komutacijskih sistema ETC familije [1,4], formirana je odgovarajuća metodologija primjene. Mogu se razlučiti tri faze u kojima je korišten SDL jezik:

1. prikupljanje i analiza podloga za razvoj (zahtjevi na sistem i njegove funkcije, -- postojeće preporuke i standardi, identifikacija razvijenih komponenta koje se mogu koristiti u novom razvoju)
2. identifikacija blokova sistema koji trebaju biti razvijeni i definiranje sučelja (dekompozicija funkcija)
3. detaljna razrada funkcija kroz razradu SDL dijagrama procesa i procedura uz dodavanje detalja specifičnih za konkretnu implementaciju

Dosadašnja iskustva posebno ukazuju na potrebu da se implementacija i kasnije testiranje mogu dovesti u vezu s

različitim (u najmanju ruku najnižim) nivoima dekompozicije sistema predstavljene odgovarajućim SDL dijagramima [13,14]. Ova je veza značajna iz dva razloga:

- da se proces testiranja može pratiti i razumjeti na različitim nivoima apstrakcije,
- da se eventualne odluke o izmjenama, koje su posljedica problema otkrivenih testiranjem, mogu locirati na određeni nivo apstrakcije, tako da se kod planiranja izmjena ima u vidu pravi kontekst i doseg izmjena



Slika 1.

Proces razvoja baziran na korištenju SDL-a prikazan je na slici 1. Faza rada sa SDL dokumentima nije bila podržana programskim pomagalima, a izvorni kod razvijanih programa bio je kreiran, čuvan i održavan na odgovarajućem pomagalu [2]. Stoga se može reći da je inicijalno postizana veza između SDL dokumenata i razvijanih programa, veza koja je nesumljivo doprinijela kvaliteti razvijanih programa, ali da organizacioni zahvati nisu dovoljni da ta veza ostane trajno prisutna. Najvažniji razlog za odstupanje SDL dokumenata od programa je što su dominantno korišteni dijagrami u grafičkom obliku, čije manuelno ažuriranje predstavlja objektivan problem. S druge strane trajna veza između SDL dokumenata i

razvijenih programa ključna je za fazu održavanja sistema, pri čemu pod održavanjem podrazumijevamo kako otklanjanje otkrivenih grešaka, tako i unapređenje funkcija sistema, što će u fazi prelaska na ISDN biti značajna aktivnost.

Pri razvoju ETC familije sistema razradivane su i korištene odgovarajuće metode verifikacije funkcija na nivou specifikacije [22,23]. Razvijeni postupci mogu se, međutim, primijenjivati u većoj mjeri samo uz automatizaciju, kako samih postupaka verifikacije, tako i pretvorbe iz SDL-a u odgovarajuće modele na kojima se bazira postupak verifikacije.

Navedena iskustva potakla su razvoj pomagala za rad sa SDL dokumentima [3,8]. Inicijalna funkcija takvog pomagala je da omogućiti unos, čuvanje i ažuriranje SDL dokumenata, na način i u obliku najpogodnijem za korisnika takvog pomagala. Iz toga proizlazi i planirani inicijalni skup komponenta SDL radne stanice:

- sintaksno vođeni grafički editor,
- sintaksno vođeni tekstualni editor,
- prevodioci iz tekstualog sintaksnog oblika u grafički i obratno

Automatizacijom ovog dijela procesa manipuliranje s SDL dokumentima dolazi u ravnopravan položaj s unosom i održavanjem izvornog koda programa, te se može osigurati tražena usklađenost svih oblika dokumentacije.

SDL radna stanica uvećava vrijednost metodologije bazirane na SDL-u time što podržava, olakšava i čini ugodnijim korištenje svih aspekata metodologije. Samo pomagalo osigurava korektnu sintaksu, pomaže u otkrivanju semantičkih nepravilnosti i daje značajnu pomoć da specifikacije sistema u cjelini budu konzistentne.

Cilj ovakvog pomagala je da se većina održavanja bazira na specifikacijskim dokumentima. Pri tome ne mislimo samo na aktivnosti otklanjanja grešaka, već i na unapređenje funkcija sistema. Kako troškovi održavanja čine do 80% ukupnih troškova koristi mogu biti znatne. Procjene i iskustva sličnih razvijenih i korištenih pomagala [15,18,20,21] pokazuju da se, pored povećanja kvalitete, može očekivati barem 30% povećanje produktivnosti u fazi razvoja, i do 80% smanjenje opterećenja i troškova u fazi održavanja.

U slijedećim fazama izgradnje SDL radne stanice moguće je realizirati niz pomagala koja imaju za cilj automatizaciju

verifikacije na nivou specifikacije [15,17], a nakon toga automatizirano generiranje izvornog koda visokog telekomunikacijskog jezika [15,16,19]. Ova pomagala moraju biti inteligentna pomagala bazirana na tehnikama umjetne inteligencije uz formiranje odgovarajućih baza znanja. Stoga će za njihovu realizaciju biti potreban značajan istraživački i razvojni napor.

3.UVJETI IZGRADNJE SDL RADNE STANICE

Definirano mjesto i uloga SDL radne stanice određuje i potencijalni krug korisnika, kao i zadatke u kojima se takvo pomagalo može dominantno koristiti. Potencijalni korisnici su proizvođači telekomunikacijske opreme, eksploatacijske organizacije i znanstvene i znanstveno-nastavne organizacije. Područja primjene obuhvaćaju:

- specifikaciju i razvoj funkcija,
- održavanje,
- školovanje kadrova,
- istraživanje metoda razvoja i održavanja,
- donošenje i održavanje standarda komutacijskih funkcija itd.

Na osnovu sagledavanja ovih elemenata realno je očekivati da se za različite primjene kod istog i kod različitih korisnika formiraju specifične konfiguracije pomagala, pri čemu pod time mislimo na različitu funkcijsku strukturu radne stanice implementiranu na različitim računalima s raznim operativnim sistema, različitom perifernom opremom i različitim načinom vezivanja u mrežu. Stoga se nameće zaključak da komponente SDL radne stanice moraju biti modularno građene, sa dobro definiranim sučeljima, te da programska podrška treba biti portabilna, u mjeri u kojoj je to, obzirom na karakter pomagala i današnje stanje programske tehnologije, moguće postići.

Polazeći od tih zahtjeva odlučeno je da za se za grafički dio programa koristi neka od implementacija GKS grafičkog standarda [7,12], te da se kao programski jezik u inicijalnoj fazi koristi C jezik. Također je odlučeno da se za generiranje rutina za leksičku odnosno sintaksnu analizu koriste LEX i YACC [10]. Ovako građeno pomagalo moguće je implementirati na nizu računarskih sistema, od velikih sistema s grafičkim terminalima, preko radnih stanica, vezanih u LAN ili samostalnih, do personalnih računala.

4. REALIZACIJA PRVOG PROTOTIPA

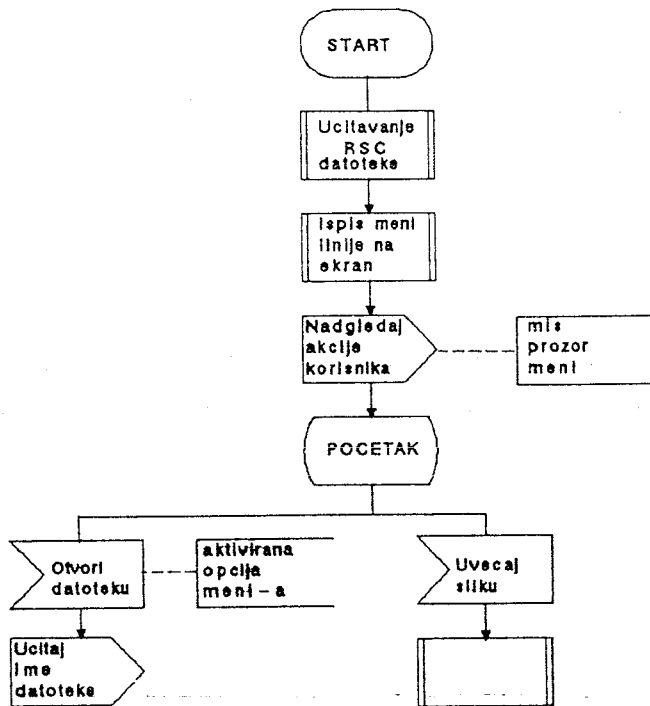
Prvi prototip SDL radne stanice razvija se na personalnom računalu ATARI ST520+ , uz oslonac na GEM [9] korisničko sučelje. Osnovni razlozi za to su:

- računalo je dovoljno brzo da zadovolji zahtjeve,
- rezolucija od 640/400 točaka zadovoljava,
- GEM korisničko sučelje pored ostalog sadrži odgovarajući nivo implementacije GKS standarda,
- operativni sistem je bitno manji ali ima dosta sličnosti s UNIX-om, a raspoložive C biblioteke su u velikoj mjeri UNIX kompatibilne,
- takava radna stanica relativno je jeftina,
- na raspolaganju su bila odgovarajuća razvojna pomagala.

Ovakva odluka donosi i svoje specifičnosti i na prvom mjestu to je oslonac na funkcije koje nudi GEM korisničko sučelje. GEM korisničko sučelje predstavlja u praksi standard komunikacije sa korisnikom na jačim personalnim računalima. GEM je razvijen za niz poznatih i široko rasprostranjenih personalnih računala (npr. IBM PC), te je razvijeno pomagalo direktno prenosivo na takve sisteme.

GEM sadrži grafički podsistem VDI koji progameru omogućava da bude neovisan o mašini i grafičkom ulaznom odnosno izlaznom uređaju, te dio nazvan AES (Application Environment Services), na kojem se želimo nešto duže zadržati.

AES se sastoji od niza rutina koje služe za kreiranje, prikazivanje i uopće rad sa prozorima, meni linijom, dijalozima i sl. Ove rutine odvajaju aplikaciju od operativnog sistema u užem smislu , bitno olakšavaju pisanje programa, te formiraju specifičnu, za našu primjenu vrlo poželjnu okolinu aplikacije. Vlastita aplikacija nije kontinuirano aktivna već neprestano predaje naloge AES rutinama od kojih biva aktivirana samo onda kada za to postoji potreba. Tako na primjer AES rutine nadgledaju interakciju korisnika sa elementima prozora, menija, akcijama miša odnosno korištenje tastature, te prikupljene podatke šalju aplikaciji kao poruku o događaju preko odgovarajućeg spremnika. Ovakav način rada stavlja aplikaciju u položaj koji se idealno opisuje SDL procesom. Na taj način dolazimo u situaciju da SDL radnu stanicu specificiramo, razvijamo i dokumentiramo uz pomoć samog SDL jezika. Primjer specifikacije funkcija grafičkog editora



Slika 2.

prikazan je na slici 2. Slika je napravljena uz pomoć grafičkog editora.

Način korištenja VDI rutina opisan je detaljnije u [11], a neki primjeri korištenja AES rutina navedeni su [12].

4. ZAKLJUČAK

Razvoj prvog prototipa SDL radne stanice uspješno se privodi kraju. Time se otvara prostor da se u praktičnoj primjeni verificira prilagodенost razvijenih pomagala korisniku, te da se pristupi istraživanjima složenijih dijelova radne stanice.

LITERATURA:

1. Jevtić, D. et al. "Modeli i metode u razvoju mikroprocesorski upravljanoj komutacijskog sistema ETC-K 28". ITA 5(1986)1-2
2. Strelar-Migotti, M., S. Zovko, "Programsko pomagalo APN 163". ITA 5(1986)1-2, pp. 197-209
3. Zorić, M., Z. Galović, "Elementi programskog pomagala za rad s SDL-om", Zbornik radova 28. Etan u pomorstvu, Zadar 1986
4. Kunšić, M., B. Mikac, M. Zorić, "Razvoj metoda specifikacije i modeliranja i primjena na sistem ETC960". ITA2(1983)1-2 pp67-82
5. XX, "Functional Specification and Description Language (SDL)", CCITT Red Book, Fascicle VI.10, Rec. Z101-104, Geneva 1985
6. XXX, "CCITT Specification and Description Language SDL - Draft Recommendation Z100", SDL Newsletter No 10, 1986
7. XXX, Information Processing - Graphical Kernel System (GKS) Draft International Standard ISO/DIS 7942
8. Zorić, M., "An Approach to SDL Tool Development", III SDL Users and Implementors Forum, Hague, April 1987
9. Szepanowski, G., "Das große GEM-Buch zum ATARI ST". Data Becker GmbH, Dueseldorf, 1985
10. Zorić, M., "Sintakсна analiza SDL/PR uz pomoć LEX-a i YACC-a", Zbornik radova 6. jugoslavensko savjetovanje o primjeni mikroprocesora u telekomunikacijama MIPRO-TE, Rijeka 1987
11. Zorić, M., Z. Galović, "Korištenje grafičkog standarda GKS za rad sa SDL dokumentima", Zbornik radova 6. MIPRO-TE, Rijeka 1987
12. Galović, Z., M. Zorić, "Realizacija SDL grafičkog editora", Zbornik radova ETAN, Bled 1987.
13. Marić, V., B. Mikac, K. Ruzić, M. Zorić, "Pristup testiranju signalizacijskog podsistema komutacijskog sistema ETC 960", Zbornik radova Jurema 28(1983) III svezak, pp. 123-126, 1983
14. Blašković, B., M. Zorić, "Osiguranje kvalitete programske podrške izvedene u telekomunikacijskom jeziku procedurnog tipa", Automatika 26 (1-2) :97-102 (1985)
15. Tinker, R., A. Howcroft, R. Lewis, H. Ichikawa, M. Itoh, M. Shibusaki, "A System design environment", 6. International Conference on SE for Telecommunication Switching Systems, 1986
16. Jackson, L.N., C.J. Fidge, R.S.V. Pascoe, P.H. Gerrand, "Computer-aided CHILL Program Generation from System Specifications - Design Experience from the MELBA Project, ISS '84, Florence, 1984
17. Lanzarone, G.A., "Automatic Specification Verification and Test Case Generation for Real-time Systems", 3. European Workshopp on Applications and Theory of Petri Nets, 1979
18. Modesti, M., R. Saracco, F. Serio, "Evaluation of the CCITT SDL in the Specification, Design and Documentation of Large Industrial Telecommunication Systems", 6. SETSS, 1986
19. Barra, S., O. Ghisio, M. Modesti, "Experience and Problems of Applications of Automatic Translations from SDL Specifications into CHILL Implementations", 6. SETSS, 1986
20. Vefsrmo, E.A.M., "DASOM - An integrated Software Engineering Tool for Telecommunication Software Systems", 6. SETSS, 1986
21. Bennett, R.L., J.A. Lindner, R.W. Michelsen, D.J. Rypka, "SDL in SESS™ Switching System Development", 6. SETSS, 1986
22. Čej, D., I. Lovrek, "Algoritmi obrade poziva u integriranoj mreži", Zbornik radova XXVIII Etan, Split, 1984
23. xx, ETC 960, ITA 2(1983)1-2, tematski dvobroj