

ZELJKO GALOVIC
MILAN ZORIC
Zavod za telekomunikacije
ELEKTROTEHNIČKI FAKULTET
Zagreb, Unska 3

REALIZACIJA SDL GRAFIČKOG EDITORA

SDL GRAPHICAL EDITOR IMPLEMENTATION

SADRŽAJ - U radu je dan prikaz SDL grafičkog editora kao elementa pomagala za specificiranje, razvoj i održavanje programske podrške komutacijskih sistema. Navedene su osnove na kojima je bazirana njegova realizacija. Opisane su funkcije editora i njihova realizacija.

ABSTRACT - In this paper SDL graphical editor which is part of a tool for specification, development and maintenance of switching system software is described. Editor functions are listed and their implementation described.

1. UVOD

Prelaskom na ISDN povećava se složenost programske podrške komutacijskih sistema, a raste i obim funkcija koje u što kraćem vremenu treba razviti i ugraditi u sisteme. Razvijene funkcije treba dalje održavati i dogradivati, što stavlja posebne zahtjeve na metode razvoja i održavanja programske podrške. Pri tome se poseban naglasak stavlja na metode specificiranja funkcija, te SDL jezik [5,6] koji se u toj fazi koristi. Iskustva stečena prilikom razvoja sistema ETC familije (ETC 960 i ETC-K28) [1,2,4] ukazuju na to da načine korištenja SDL jezika treba dalje unapređivati, kao i na potrebu korištenja odgovarajućih programskih pomagala za rad sa SDL dokumentima. Stoga se pristupilo razvoju SDL radne stanice [3,8,13]. Sastavni dio takve radne stanice predstavlja i grafički editor opisan u ovom radu.

2. POLAZNE OSNOVE

Rad na realizaciji SDL grafičkog editora započet je na osnovu CCITT preporuka iz 1984 godine [5]. Koncem 1986. godine objavljen je nacrt preporuka koje trebaju službeno biti usvojene 1988. godine [6]. Obzirom da radne grupe CCITT-a nemaju namjeru bitno mijenjati taj nacrt, odlučeno je da se u izgradnji SDL pomagala oslonimo na taj dokument. Na tu odluku navodi nas činjenica da u novim preporukama postoje novi elementi jezika koji su vrlo korisni (dvosmjerni kanal, prioritetni ulazni i izlazni signali, pojam servisa itd.), a još više način na koji su u prestrukturiranim preporukama definirani sintaksa i semantika SDL jezika, posebno grafička sintaksa.

Za definiranje tekstualne gramatike korišten je BNF. Za grafičku gramatiku definiran je novi metajezik koji za osnovu ima BNF, a dodani su slijedeći metasimboli: contains, is associated with, is followed by, is connected to, set. Tako je na primjer stanje definirano na slijedeći način:

```
<state area>::=
```

```
    <state symbol> contains <stated list> is connected to
    (<input association area>
    !<priority input association area>
    !<continuous signal association area>
    !<save association area> )*
```

```
<state symbol>::=
```



```
<input association area>::=
```

```
    <solid association symbol> is connected to <input area>
```

Ovako precizno definirana grafička sintaksa omogućava da se grafička gramatika formulira kao LALR(1) gramatika te da se niz rutina generira uz pomoć YACC-a [11]. Prvi prototip grafičkog

editora razvija se na računalu ATARI ST520+. U radu [13] navedeni su motivi za takav izbor, kao i opći principi izgradnje SDL radne stanice. Grafički editor je izgrađivan u skladu s postavljenim principima, te se u ovom radu posebno prikazuju načini realizacije postavljenih principa.

3.FUNKCIJE GRAFIČKOG EDITORA

Grafički editor treba u osnovi omogućiti unos i ažuriranje SDL dokumenata u grafičkom obliku. Kada govorimo o SDL dokumentima onda pod time mislimo na:

- SDL dijagrame procesa, procedura, servisa te makro dijagrame
- dijagrame koji definiraju strukturu sistema kao što su dijagram stabla blokova, blok interakcijski dijagram, te dijagrami dekompozicije na podstrukture.

Sve ovo treba biti napravljeno tako da slijedeci logiku odnosno sintaksu SDL-a vodi korisnika, ali da ga ni na koji način ne sputava u njegovim akcijama. Interakcija korisnika sa sistemom mora biti svedena na najnužniju mjeru, efikasna i što jednostavnija, tako da korisnik lako usvaja radne procedure i nikada nije u nedoumici u pogledu ishoda svojih akcija. Sve što može biti riješeno uz pomoć akcija miša treba biti riješeno na taj način, a tastatura se koristi praktički samo za unos teksta tamo gdje je to potrebno.

Funkcije editora možemo podijeliti u tri skupine:

- opće funkcije
- funkcije za inicijalni unos SDL dijagrama
- funkcije za editiranje postojećeg dijagrama

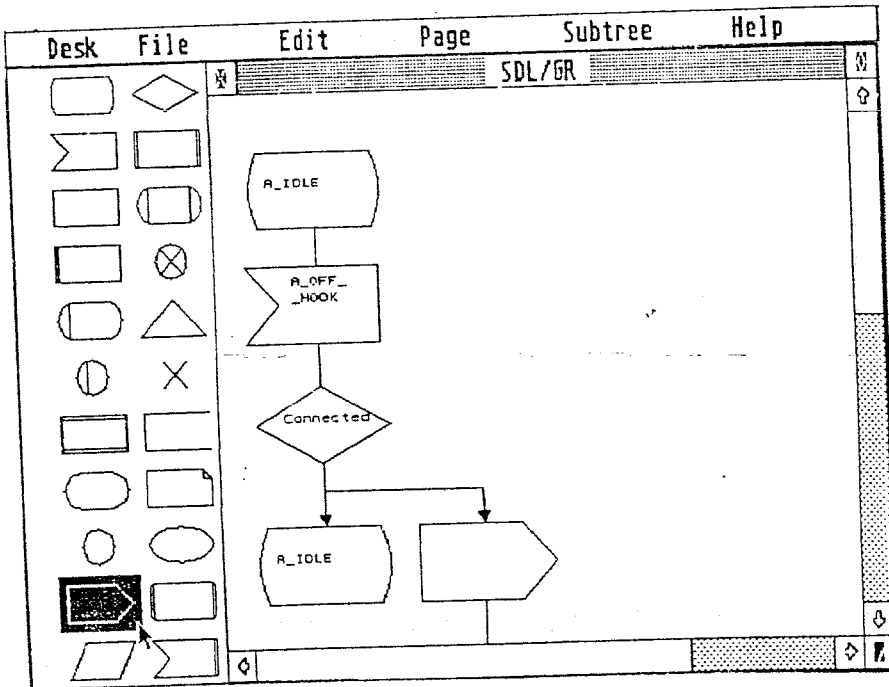
Opće funkcije omogućuju otvaranje postojećih dokumenata i odabir određene stranice, manipulaciju sa prozorima i komunikaciju sa korisnikom preko menija i dijaloga. Također je moguće birati stvarnu veličinu stranice i veličinu njenog prikaza unutar prozora na ekranu. Kako se radi sa stranicom koja cijela ne stane na ekran, nužne su i funkcije koje omogućuju prikaz

odredenog dijela stranice i njeno pomicanje u odnosu na prozor. Zahtjeve za pomicanjem korisnik daje preko, za tu svrhu, predviđenih elemenata na rubovima prozora. Nakon završenog unosa ili izmjena moguće je zatvoriti dokument, što rezultira zamjenom datoteke pod istim imenom, spremiti ga kao novu datoteku ili odbaciti sve što je promijenjeno. Moguć je rad sa više prozora od kojih svaki sadrži po jednu stranicu istog ili različitih dokumenata, što omogućuje dobar pregled i lako kopiranje dijelova dijagrama.

Kod inicijalnog unosa SDL dijagrama procesa najprije se odredi pozicija početnog simbola. Svi raspoloživi simboli nalaze se nacrtani na ekranu i jednosavno se pomoću miša odabiru redom kojim trebaju biti dodani u dijagram. Automatski se određuje pozicija simbola ispod ili desno ovisno o vrsti zadnjeg nacrtanog i odabranog sljedećeg simbola. Linija koja spaja simbole crta se također automatski. Povezivanje linijom koja predstavlja JOIN vrši se tako da se odaberu simbol kojeg treba povezati i linija ili drugi simbol sa kojim ga treba povezati, dok se putanja linije određuje automatski. Putanju linije ili poziciju simbola korisnik može po volji mijenjati. Ako neki simbol nije u određenom trenutku dozvoljen prema sintaksnim pravilima, onemogućen je njegov izbor, pa je isključena mogućnost kreiranja sintaktski neispravnih dokumenata. Unos teksta u pojedini simbol se vrši preko posebnog dijaloga koji se pojavi na ekranu na zahtjev korisnika za određeni simbol ili nakon svakog novog dodanog simbola. Odmah nakon upisa teksta vrši se sintakсна analiza i onemogućuje njegov prihvata ako je neispravan.

Editiranje postojećeg dijagrama omogućeno je funkcijama za selektiranje pojedinog simbola i njegovo brisanje ili promjenu tipa, kao i promjenu teksta preko istog dijaloga kao kod inicijalnog unosa. Simboli se mogu jednostavno pomoću miša premjestiti na novu poziciju unutar iste stranice, dok je rad sa dijelovima dijagrama baziran je na podstablu. Jedno podstablo određuje se odabirom jednog početnog simbola i jednog ili više završnih koji predstavljaju krajeve grana. Tako definirano podstablo se može preseliti ili kopirati na određenu poziciju na istoj ili drugoj stranici ili izbrisati. Funkcija pretraživanje omogućuje traženje određenog simbola i teksta na nivou cijelog

dokumenta. Funkcija kompletiranja omogućuje traženje nedovršenih grana što olakšava rad. Veličina SDL simbola i znakova za ispis teksta može se odrediti na nivou stranice ili posebno za svaki simbol. Ispis na papir se ostvaruje tako da se najprije dokument spremi u odgovarajuću meta-datoteku i nakon toga se SDL dijagram sadržan u toj datoteci nacrti pomoću posebnog programa. Program za crtanje može biti pozvan iz editora. Spremljeni u meta-datoteku i nacrtani na papiru mogu biti i nedovršeni dijagrami.



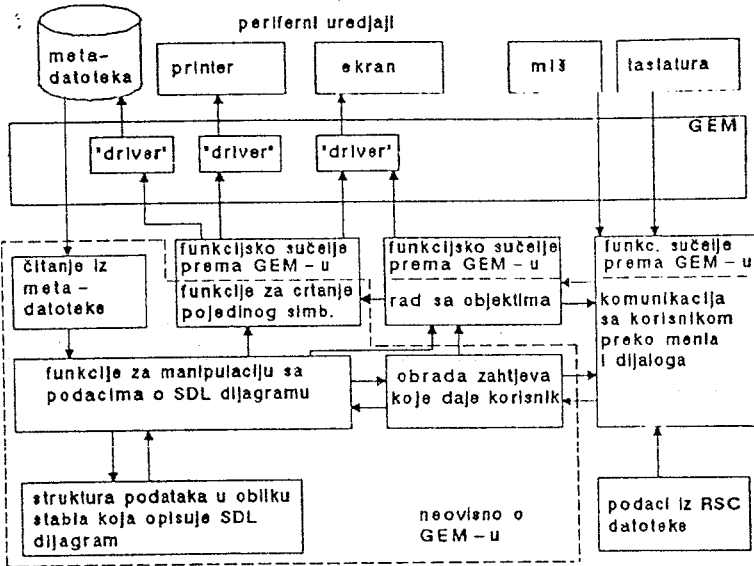
Slika 1. Izgled ekrana kod editiranja

4.NACIN REALIZACIJE

Sve stranice koje čine jedan dokument čuvaju se u jednoj meta-datoteci [12]. U programu se za pamćenje SDL dijagrama formira odgovarajuća struktura podataka u obliku stabla koja ga opisuje. Upis u meta-datoteku se vrši redoslijedom određenim sintaksnim pravilima, tako da je moguće jednostavno rekonstruiranje stabla prilikom učitavanja koristeći rutine generirane pomoću YACC-a [11]. Stranici koja se trenutno prikazuje na ekranu pridruženi su odgovarajući objekti. Objekti su grafički pravokutni elementi predstavljeni u programu određenom stablastom strukturom, a na ekranu predstavljaju niz znakova ili sličicu. AES raspolaze nizom funkcija za rad sa tako definiranim objektima npr. crtanje, detektiranje njihove prisutnosti na ekranu, selektiranje i označavanje itd.

Komunikacija sa korisnikom ostvarena je preko menia i dijaloga. Za definiranje njihovog izgleda koristi se razvojno pomagalo pod nazivom "Resource construction set". Njime je uz pomoć miša omogućeno jednostavno slaganje objekata koji čine dijalog ili meni. Tako definirani dijalozi i meni se pohranjuju u tzv. "resource" (RSC) datoteku, koju program pomoću odgovarajućih AES funkcija učita u memoriju računala. Činjenica da je sav tekst koji služi za komunikaciju sa korisnikom izdvojen u zasebnoj datoteci, omogućuje da se za različite govorne jezike napravi odgovarajuća RSC datoteka, dok se program koristi bez bilo kakve promjene.

Na slici 2. prikazana je funkcijska struktura grafičkog editora. Posebno je označen dio funkcija koje su realizirane neovisno o GEM [9] grafičkom sistemu. Funkcije koje čine sučelje između programa i GEM-a nalaze se u bibliotekama koje su sastavni dio razvojnog sistema odgovarajućeg programskog jezika. "Driver" u okviru GEM-a predstavlja pogonski program za određeni grafički uređaj. Rad sa podacima koji opisuju SDL dijagram ostvaren je preko minimalnog skupa funkcija i bilo kakve promjene nad podacima se vrše isključivo preko tih funkcija, što doprinosi većoj sigurnosti i preglednosti samog programa.



Slika 2. Funkcijska struktura grafičkog editora

Desk	File	Edit	Page	Subtree	Help
YAST	New	Delete	Full View	Mark root	Extra
	Open...		Normal Vi	Mark leaves	
Calcu	Merge...	Find...	Zoom in		
Clock	Read page	Replace..	Zoom out	Cut	
Acces				Paste	
Acces	Close	Find empty	Hide Grid		
Acces	Abandon c	Complete	Turn Snap	Copy	
Acces			Show Rule	Move	
	Save	Auto open	Ruler Spa		
	Save and	Go to pag	Page form	Delete subt.	
	Save as..		Symbol si	Hide subtree	
	Write page...				
	Write subtree...				
	To Output..				
	Quit				

Slika 3. Prikaz otvorenih meni-a

5. ZAKLJUČAK

Grafički editor predstavlja važnu komponentu SDL radne stanice. Obzirom na svoju maksimalnu prilagodенost korisniku editor treba doprinijeti povećanju kvalitete i produktivnosti rada na razvoju i održavanju programske podrške. Stupanj prilagodенosti korisniku treba verificirati u praktičnoj primjeni. Modularna struktura izgrađene programske podrške omogućit će lagano modificiranje postojećih odnosno uvođenje novih funkcija ako se ukaže potreba. Također, moguće je editor implementirati u drugoj okolini, jer su veze sa GEM korisničkim sučeljem dobro izolirane.

LITERATURA:

1. Jevtić, D., D.Kale, I.Lovrek, N.Noethig, M.Zorić, V.Kaštelan, D.Kršanac, V.Maričić, I.Spanić, B.Zalar, "Modeli i metode u razvoju mikroprocesorski upravljanoг komutacijskog sistema ETC-K 28", ITA 5(1986)1-2,43-62
2. Lovrek, I., M. Zorić, "Primjena SDL-a u razvoju programske podrške pisane u nižem programskom jeziku", Zbornik radova 27. ETAN u pomorstvu, pp. 494-498, Zadar 1985
3. Zorić, M., Z. Galović, "Elementi programskog pomagala za rad s SDL-om", Zbornik radova 28. Etan u pomorstvu, Zadar 1986
4. Kunštic, M., B. Mikac, M. Zorić, "Razvoj metoda specifikacije i modeliranja i primjena na sistem ETC 960", ITA, Vol. 2, No. 1-2, pp. 67-82, 1983
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 1987
9. Szepanovski, G., "Das große GEM-Buch zum ATARI ST", Data Becker GmbH, Dueseldorf, 1985
10. Galović, Z., "Realizacija prikaza elemenata SDL/GR na bazi grafičkog standarda GKS", Diplomski rad, ETF Zagreb, 1987
11. Zorić, M., "Sintaksna analiza SDL/PR uz pomoć LEX-a i YACC-a", Zbornik radova 6. jugoslavensko savjetovanje o primjeni mikroprocesora u telekomunikacijama MIPRO-TE, Rijeka 1987
12. Zorić, M., Z. Galović, "Korištenje grafičkog standarda GKS za rad sa SDL dokumentima", Zbornik radova 6. jugoslavensko savjetovanje o primjeni mikroprocesora u telekomunikacijama MIPRO-TE, Rijeka 1987
13. Zorić, M., "SDL radna stanica i razvoj programske podrške komutacijskih sistema", Zbornik radova ETAN, Bled 1987.