Sustav za pomoć osobama oštećena vida za potrebe programiranja grafičkih sučelja

Konecki, Mario

Doctoral thesis / Disertacija

2013

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics Varaždin / Sveučilište u Zagrebu, Fakultet organizacije i informatike Varaždin**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:754469

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI



Sveučilište u Zagrebu

Fakultet organizacije i informatike

Mario Konecki

SUSTAV ZA POMOĆ OSOBAMA OŠTEĆENA VIDA ZA POTREBE PROGRAMIRANJA GRAFIČKIH SUČELJA

DOKTORSKI RAD

Varaždin, 2013.



University of Zagreb

Faculty of Organization and Informatics

Mario Konecki

SYSTEM FOR AIDING VISUALLY IMPAIRED IN GRAPHICAL INTERFACES PROGRAMMING

DOCTORAL THESIS

Varaždin, 2013.

PODACI O DOKTORSKOM RADU

I. AUTOR

Ime i prezime	Mario Konecki
Datum i mjesto rođenja	3. kolovoza 1982., Virovitica
Naziv fakulteta i datum diplomiranja na VII stupnju	Fakultet organizacije i informatike, 26. studenog 2005.
Sadašnje zaposlenje	Fakultet organizacije i informatike

II. DOKTORSKI RAD

Naslov	Sustav za pomoć osobama oštećena vida za potrebe programiranja grafičkih sučelja
Broj stranica, slika, tabela, priloga, bibliografskih podataka	94 stranice, 40 slika, 5 tablica, 1 prilog, 104 bibliografska podatka
Znanstveno područje i polje iz kojeg je postignut doktorat znanosti	Društvene znanosti, Informacijske i komunikacijske znanosti
Mentori ili voditelji rada	Prof. dr. sc. Alen Lovrenčić Doc. dr. sc. Ante Bilić-Prcić
Fakultet na kojem je obranjen doktorski rad	Fakultet organizacije i informatike
Oznaka i redni broj rada	

III. OCJENA I OBRANA

Datum sjednice Fakultetskog vijeća na kojoj je prihvaćena tema	17. srpnja 2012.
Datum predaje rada	10. prosinca 2012.
Datum sjednice Fakultetskog vijeća na kojoj je prihvaćena pozitivna ocjena rada	22. listopada 2013.
Sastav povjerenstva koje je rad ocijenilo	Prof. dr. sc. Mirko Čubrilo, Prof. dr. sc. Alen Lovrenčić, Doc. dr. sc. Ante Bilić Prcić, Prof. dr. sc. Matjaž Gams, Prof. dr. sc. Ivica Crnković
Datum obrane doktorskog rada	22. studenog 2013.
Sastav povjerenstva pred kojim je rad obranjen	Prof. dr. sc. Mirko Čubrilo, Prof. dr. sc. Alen Lovrenčić, Doc. dr. sc. Ante Bilić Prcić, Prof. dr. sc. Matjaž Gams, Prof. dr. sc. Ivica Crnković
Datum promocije	



Sveučilište u Zagrebu

Fakultet organizacije i informatike

Mario Konecki

SUSTAV ZA POMOĆ OSOBAMA OŠTEĆENA VIDA ZA POTREBE PROGRAMIRANJA GRAFIČKIH SUČELJA

DOKTORSKI RAD

Mentori:

Prof. dr. sc. Alen Lovrenčić
Doc. dr. sc. Ante Bilić Prcić

Varaždin, 2013.



University of Zagreb

Faculty of Organization and Informatics

Mario Konecki

SYSTEM FOR AIDING VISUALLY IMPAIRED IN GRAPHICAL INTERFACES PROGRAMMING

DOCTORAL THESIS

Supervisors:

Prof. Alen Lovrenčić, Ph.D.
Asst. Prof. Ante Bilić Prcić, Ph.D.

Varaždin, 2013.

INFORMACIJE O MENTORIMA

Prof. dr. sc. Alen Lovrenčić je rođen u Čakovcu 8. rujna 1968. godine. Osnovnu školu i dva razreda opće srednje škole pohađao je u Čakovcu, a dva razreda usmjerene srednje škole u Varaždinu. Dodiplomski studij upisao je 1987. godine na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu, smjer profesor matematike i informatike, gdje je diplomirao 10. lipnja 1993. godine te stekao stručno zvanje profesor matematike i informatike. Nakon završenog dodiplomskog studija radio je u Informacijsko-projektantskom centru Međimurje u Čakovcu i tvrtki TMT d.o.o. u Čakovcu. Znanstveni poslijediplomski studij upisao je 1995. godine na Fakultetu organizacije i informatike u Varaždinu, a 1999. godine obranio je znanstveni magistarski rad. Doktorirao je 2003. godine na Fakultetu organizacije i informatike Sveučilišta u Zagrebu.

Od 1997. godine zaposlen je u zvanju mlađeg asistenta na katedri za teorijske i primijenjene osnove informacijskih znanosti na Fakultetu organizacije i informatike u Varaždinu, 1999. godine izabran je u zvanje asistenta na istoj katedri, a 2004. godine stječe zvanje docenta. 2008. godine je izabran u znanstveno zvanje višeg znanstvenog suradnika i znanstveno-nastavno zvanje izvanrednog profesora. Od 2007. godine je pročelnik Katedre za teorijske i primijenjene osnove informacijskih znanosti. Od 1998. godine sudjelovao je na znanstvenom projektu Integracija baza znanja, pod vodstvom prof. dr. sc. Mirka Malekovića, koji je financiran od strane MZOŠ. Od 2002-2007. godine sudjeluje na projektu Formalizacija sustava poslovnih pravila financiranom od strane MZOŠ.

Od 2007. godine sudjeluje na znanstvenom projektu financiranom od MZOŠ, Semantičko modeliranje višeagentnih sustava pod voditeljstvom prof. dr. sc. Mirka Malekovića. Od 2008. godine je voditelj projekta MZOŠ pod nazivom Automatizacija postupaka u projektiranju informacijskih sustava. Vodio je Eureka projekt IS4 - SME Innovation Navigator pod brojem E!3521. Osim toga, sudjelovao je na ALIS projektima Popularisation of Science and Dissemination of Scientific Information i na TEMPUS projektu Aspects of Organization and Information Systems: Curriculum Development pod vodstvom prof. dr. sc. Blaženke Divjak. Bio je član tima za izradu CARNET-ovog referalnog centra za prijavu znanstvenih projekata na daljinu. Suautor je sveučilišnog udžbenika Sveučilišta u Zagrebu pod naslovom Diskretna matematika s teorijom grafova.

Doc. dr. sc. Ante Bilić-Prcić je rođen 1. studenog 1965. godine u Studencima kod Imotskog gdje je završio i osnovnu školu. Nakon srednje škole, koju završava u Splitu, dolazi na studij u Zagreb gdje studira na Veterinarskom, Medicinskom i Edukacijsko-rehabilitacijskom fakultetu. Diplomirao je na Edukacijsko-rehabilitacijskom fakultetu i stekao zanimanje prof. defektolog - rehabilitator. Nakon završenog dodiplomskog studija upisuje poslijediplomski studij na Edukacijsko-rehabilitacijskom fakultetu, gdje je magistrirao 15. srpnja 2003. godine na Odsjeku za oštećenja vida. Doktorat znanosti je obranio 21. prosinca 2007. godine.

Zaposlenik je Edukacijsko-rehabilitacijskog fakulteta, a prije toga je radio u Poglavarstvu Grada Zagreba, Centru za autizam i Centru za odgoj i obrazovanje "Vinko Bek". Sudjeluje kao predavač na više stručnih skupova i seminara koje organizira MZOŠ. Koordinator je CARNet-ove ustanove i savjetnik u Agenciji za znanost i visoko obrazovanje. Aktivno je sudjelovao na više znanstvenih i stručnih skupova na kojima je prezentirao radove, posebice s područja oštećenja vida.

Nositelj je i izvoditelj nastave iz kolegija Nove tehnologije i pomagala u rehabilitaciji osoba oštećena vida, Adaptacija nastavnih pomagala za osobe oštećena vida i Procjena osoba oštećena vida na preddiplomskom studiju, te Funkcionalna procjena osoba oštećena vida i Elektronska pomagala i programi u rehabilitaciji osoba oštećena vida na diplomskom studiju.
U dva mandata obavljao je dužnost Voditelja za financijsko poslovanje i kontrole. Pročelnik je Odsjeka za oštećenja vida Edukacijsko-rehabilitacijskog fakulteta. Za docenta je izabran 12. studenog 2009. godine.

SAŽETAK

Otkad su računala ušla u masovnu uporabu, postojala su nastojanja da se osobe oštećena vida uvedu u svijet programiranja. Međutim, ubrzani razvoj grafičkih sučelja u ovom području doveo je programere oštećena vida do točke u kojoj se više nisu mogli nositi s vizualnim aspektima novih programskih koncepata. Analizirajući trenutno stanje, može se doći do zaključka da je upravo oblikovanje grafičkih sučelja jedan od dominantnih problema programera oštećena vida koji traži adekvatno rješenje. U okviru ovog doktorskog rada provedeno je istraživanje o potrebi i ključnim karakteristikama koncepta i sustava koji bi doveli do rješenja problema programera oštećena vida u području oblikovanja grafičkih sučelja u domeni klasičnog stolnog programiranja. Temeljem provedenog istraživanja izrađen je konceptualni model sustava za pomoć osobama oštećena vida u razvoju grafičkih sučelja.

Uz konceptualni model, razvijen je i funkcionalni sustav s novim deskriptivnim jezikom GUIDL (Graphical User Interface Description Language) kao centralnim dijelom koji omogućava programerima oštećena vida jednostavan i prilagođen način opisa grafičkih sučelja tekstualnim putem. Opisana sučelja je moguće u okviru razvijenog sustava prevesti u neku od aktualnih programskih tehnologija što u konačnici omogućava programerima oštećena vida da se uključe sa svojim idejama i radom u aktualne projekte. Nad razvijenim sustavom je provedeno testiranje koje je pokazalo da razvijeni konceptualni model i sustav predstavljaju uporabljivo rješenje za problem razvoja grafičkih sučelja za potrebe osoba oštećena vida.

Ključne riječi: osobe oštećena vida, programiranje, razvoj grafičkih sučelja, GUIDL jezik, GUIDL sustav, EBNF, vidne poteškoće, HAAT, CAT, pomoćna tehnologija

EXTENDED ABSTRACT

Since computers have entered into mass usage, there have been efforts to introduce visually impaired into the world of programming. These efforts have been successful and computers have brought a number of advantages for visually impaired, such as easier access to information, books and articles, easier communication, etc. Along with other advantages, computer usage has created new jobs for visually impaired. One of these jobs was programming and software design. Visually impaired programmers have been able to perform their jobs for years since all graphical user interfaces (GUIs) were based on text and developed assistive technology in the form of text to speech converters was sufficient for enabling visually impaired to produce computer code and develop programs. However, the rapid development of GUI in this area, has led the visually impaired programmers to the point where they could no longer cope with the visual aspects of new programming concepts. This transition did not come overnight, but is has soon become clear that existing assistive technology will not be able to cope with all of new graphical concepts.

Analyzing current situation, it can be concluded that a visual programming and GUI design are the dominant problems for visually impaired programmers that need an adequate solution. In this doctoral thesis, several different approaches for solution of the problems mentioned are stated along with the discussion about the most effective solution that is proposed as the general solution. Existing efforts in solving the problems mentioned are also stated and described. The research about the need for this kind of solution and about its desired characteristics is also presented. Considering a review of existing efforts to solve this problem a new model is proposed, whose central part consists of a new description language GUIDL (Graphical User Interface Description Language) that enables visually impaired to create visual interfaces in a suitable and simple way using pure textual form. A graphic interface described in GUIDL language is translatable into a form supported by newer software development environments and programming languages in the field of classic desktop programming. In this way proposed solution enables visually impaired to be involved in the design part of the overall software design process.

Keywords: visually impaired, programming, development of graphical interfaces, GUIDL language, GUIDL system, EBNF, visual difficulties, HAAT, CAT, assistive technology

SADRŽAJ

SADRŽAJ ..... **I**
POPIS SLIKA ..... **III**
POPIS TABLICA ..... **V**
1. UVOD **1**
1.1. Ciljevi rada 1
1.2. Hipoteze 2
1.3. Znanstveni doprinos 2
1.4. Postignuti rezultati 3
1.5. Struktura rada 4
2. OSOBE S INVALIDITETOM I POMOĆNE TEHNOLOGIJE **5**
2.1. Mjerenje učinkovitosti pomoćnih tehnologija 7
2.2. Definicije osnovnih pojmova 9
2.3. Poteškoće koje uzrokuju smetnje u komunikaciji 10
2.3.1. Govorne poteškoće 10
2.3.2. Vidne poteškoće 11
2.3.3. Slušne poteškoće 11
2.4. Opći sociodemografski podaci za Republiku Hrvatsku 11
2.5. Vidni sustav 13
2.5.1. Formiranje slike 14
2.5.2. Akomodacija oka 15
2.5.3. Binokularni vid i stereopsis 16
2.5.4. Nemogućnost prepoznavanja boja 16
2.5.5. Kortikalno oštećenje vida 18
2.6. Oštećenja vida i pripadajuća pomoćna tehnologija 18
2.6.1. Demografski podaci vezani za osobe oštećena vida 18
2.6.2. Prikaz nekih tipova oštećenja vida 20
2.7. Modeliranje sustava pomoćne tehnologije 25
2.7.1. HAAT model u modeliranju pomoćne tehnologije 26
2.7.2. CAT model u modeliranju pomoćne tehnologije 27
3. OSOBE OŠTEĆENA VIDA U SVIJETU RAČUNALA I PROGRAMIRANJA **29**
3.1. Programski alati za pomoć slijepima u radu s računalom 31

3.2. Programski alati za pomoć slijepima u edukaciji 34
4. AKTUALNA NASTOJANJA I PROBLEMI U PRIKAZU GRAFIČKIH SUČELJA 37
4.1. Web i stolno programiranje 39
4.2. Mogući pristupi problemu 40
4.3. Metodologija razvoja 42
5. GUIDL SUSTAV 43
5.1. Zahtjevi za GUIDL sustav 43
5.2. Model GUIDL sustava i njegov razvoj 49
6. GRAMATIKA PROTOTIPA GUIDL SUSTAVA 54
7. PRIMJER KORIŠTENJA GUIDL JEZIKA I SUSTAVA 63
8. TESTIRANJE PROTOTIPA GUIDL SUSTAVA 75
9. ZAKLJUČAK 83
LITERATURA 87
PRILOZI 95

POPIS SLIKA

Slika 2.1 Prikaz promatrane slike na mrežnici oka....14
Slika 2.2 Kratkovidnost i dalekovidnost bez i s korektivnom lećom....15
Slika 2.3 Akomodacija oka na dalji i bliži objekt....15
Slika 2.4 Stimulacija pojedinih skupina čunjića različitim bojama....17
Slika 2.5 Zdravo oko bez oštećenja....20
Slika 2.6 Gubitak oštrine centralnog vida....21
Slika 2.7 Manji gubitak oštrine vida....21
Slika 2.8 Veći gubitak oštrine vida....22
Slika 2.9 Gubitak vida zbog nistagmusa....22
Slika 2.10 Tunelski vid....23
Slika 2.11 Gubitak dijela vidnog polja....23
Slika 2.12 Površinska oštećenja oka....24
Slika 2.13 Oštećenje oka u obliku svjetlosnog prstena....24
Slika 2.14 HAAT model....26
Slika 2.15 CAT model....27
Slika 3.1 HAL govorne opcije....32
Slika 3.2 HAL Braille opcije....32
Slika 3.3 Alat JAWS....33
Slika 3.4 Virgo4 i Internet Explorer....33
Slika 3.5 APL alat....35
Slika 4.1 Taktilna ploča....38
Slika 4.2 Multimodalni pristup....39
Slika 5.1 Distribucija korištenja programskih jezika....46
Slika 5.2 Struktura odgovora na pitanje anketnog upitnika....48
Slika 5.3 Struktura odgovora na pitanje anketnog upitnika....48
Slika 5.4 Struktura odgovora na pitanje anketnog upitnika....48
Slika 5.5 Idejni model GUIDL sustava....49
Slika 5.6 Struktura i rad GUIDL sustava....53
Slika 6.1 Moguće lokacije forme/kontrole....60
Slika 7.1 Konačni izgled sučelja iz primjera 1....63
Slika 7.2 Poziv GUIDL kompajlera....67

Slika 7.3 Uspješno prevođenje GUIDL koda 67
Slika 7.4 Sintaktička pogreška prilikom prevođenja GUIDL koda 68
Slika 7.5 Proces prevođenja sučelja u željeni programski jezik 70
Slika 7.6 UML dijagram slučajeva korištenja – Izrada grafičkog sučelja.....70
Slika 7.7 UML dijagram aktivnosti – Izrada grafičkog sučelja 71
Slika 7.8 Konačni izgled sučelja iz primjera 2 72
Slika 8.1 Konačni izgled sučelja 3 76
Slika 8.2 Konačni izgled sučelja 4 78
Slika 8.3 Distribucija prihvaćanja GUIDL sustava s obzirom na godine iskustva u programiranju.....82

POPIS TABLICA

Tablica 2.1 Prikaz broja osoba s invaliditetom prema spolu, županijama prebivališta te dobnim skupinama 12
Tablica 2.2 Prikaz vrsta oštećenja koje uzrokuju invaliditet ili pridonose stupnju funkcionalnog oštećenja osobe.....13
Tablica 5.1 Rezultati istraživanja provedenog na programerima oštećena vida 45
Tablica 5.2 Odabrane grafičke kontrole i njihovi podržani atributi.....50
Tablica 8.1 Rezultati testiranja prototipa.....81

1. UVOD

Otkad su računala ušla u masovnu uporabu postojala su nastojanja da se osobe oštećena vida uvedu u svijet računala i programeri oštećena vida su godinama obavljali svoj posao s dobrim uspjesima. Međutim, ubrzani razvoj grafičkih sučelja i niza raznih tehnologija rastuće složenosti u ovom području doveli su programere oštećena vida do točke u kojoj se više nisu mogli nositi s grafičkim aspektima novih programskih koncepata. Iako ovaj problem još uvijek nije toliko izražen u domeni web tehnologija, vrlo je aktualan i jasan u području klasičnog stolnog programiranja. Postojeći alati za pomoć osobama oštećena vida u korištenju računala su se pokazali nedostatni da bi u potpunosti riješili novonastali jaz između programera oštećena vida i novonastalih tehnologija s izraženom grafičkom komponentom. Pojavili su se neki napori usmjereni prema pokušaju rješavanja ovih problema u nekim specifičnim slučajevima i područjima, ali se nije pojavilo ništa što bi dovelo do općeprihvaćenog rješenja na globalnoj razini.

Problem sličan onome pri korištenju programskih alata i jezika javlja se i u području edukacije osoba oštećena vida na području novijih programskih jezika i alata. Da bi se pokušalo riješiti dio ovih problema, potrebno je napraviti detaljan pregled svih dostupnih metoda, tehnika i tehnoloških rješenja koja postoje na području pomoći osobama oštećena vida u korištenju računala i donijeti zaključak o adekvatnosti pojedinih napora u dolasku do globalnog rješenja. Konačno, potrebno je definirati područje u kojem je problem programera oštećena vida najveći te predložiti adekvatno i uporabljivo rješenje na tom području kao području od posebnog interesa. U ovom radu će se prikazati navedena problematika i postojeći pokušaji rješavanja pojedinih dijelova problematike. Rad daje osvrt na neka značajnija rješenja u području definirane problematike i daje pregled mogućih koraka i pristupa u njenom rješavanju. U radu je također opisan i vlastiti pristup rješavanja spomenute problematike koji uključuje konceptualni model i razvijeno prototipno rješenje.

1.1. Ciljevi rada

U okviru rada definirano je nekoliko ciljeva:

1. Definirati temeljne probleme i zahtjeve osoba oštećena vida u domeni klasičnog stolnog programiranja.

2. Istražiti aktualna nastojanja u domeni prikaza grafičkog sadržaja osobama oštećena vida i rješenja problema programera oštećena vida u domeni razvoja grafičkih sučelja.
3. Istražiti i definirati moguće pristupe rješavanja problema razvoja grafičkih sučelja za osobe oštećena vida.
4. Izraditi konceptualni model temeljem kojeg bi bilo moguće razviti rješenje koje bi omogućilo programerima oštećena vida razvoj GUI sučelja u domeni klasičnog stolnog programiranja na jednostavan i prihvatljiv način.
5. Razviti novi deskriptivni jezik GUIDL kao centralni dio osmišljenog modela.
6. Razviti cjelokupni GUIDL sustav temeljem osmišljenog modela koji bi omogućio dobivanje izvornog koda ciljanih programskih jezika.
7. Testirati sustav i provjeriti stupanj njegove uporabljivosti na pojedinim zahtjevima iz prve točke te mjeru u kojoj sustav rješava cjelokupni zadani problem razvoja GUI sučelja za programere oštećena vida u domeni klasičnog stolnog programiranja.

1.2. Hipoteze

U okviru rada definirane su sljedeće hipoteze:

H0. Postoji potreba programera oštećena vida za rješenjem problema razvoja grafičkih sučelja.

H1. Razvit će se adekvatno rješenje koje će osobama oštećena vida omogućiti razvoj grafičkih sučelja.

1.3. Znanstveni doprinos

U radu je definirana problematika razvoja grafičkih sučelja u domeni klasičnog stolnog programiranja u smislu nemogućnosti osoba oštećena vida da razviju grafička sučelja zbog grafičkih razvojnih okolina baziranih na uporabi miša i nedostatka alternativnih mogućnosti opisa sučelja na način koji bi bio prihvatljiv i dovoljno jednostavan za praktičnu uporabu za osobe oštećena vida. Temeljni znanstveni doprinos rada je razvoj konceptualnog modela temeljem kojeg su razvijeni novi deskriptivni jezik i sustav koji omogućava osobama oštećena vida da na prihvatljiv i njima dovoljno jednostavan i zoran način razviju grafička sučelja koja će moći uključiti u aktualne programske tehnologije i razvojne okoline te se na taj način

aktualizirati sa svojim idejama kao ravnopravni članovi razvojnog tima i u području oblikovanja grafičkih sučelja što s postojećih pristupima i rješenjima nisu bili u stanju učiniti na njima dovoljno prihvatljiv način. U radu je dan pregled cjelokupnog područja koje se odnosi na navedenu problematiku, kao i pregled postojećih pristupa u rješavanju problema vizualizacije grafičkih objekata za osobe oštećena vida. Temeljem vlastitog istraživanja je u samom početku identificirana realna potreba programera oštećena vida za ovakvim rješenjem, kao i svi aspekti koje osobe oštećena vida smatraju važnim za rješavanje u spomenutoj problemskoj domeni. Navedeno istraživanje potreba programera oštećena vida u domeni razvoja grafičkih sučelja također predstavlja dio znanstvenog doprinosa rada. Razvijeni deskriptivni jezik i sustav su testirani na populaciji programera oštećena vida te je u radu dan zaključak o potvrdi uporabljivosti ovakvog rješenja.

Znanstveni doprinos rada se može identificirati kroz nekoliko stavki:

1. Identificiranoj potrebi, problemima i zahtjevima programera oštećena vida u području razvoja grafičkih sučelja.
2. Definiranim mogućim pristupima rješavanju problema razvoja grafičkih sučelja za osobe oštećena vida i odabiru adekvatnog pristupa kao generalnog rješenja.
3. Izrađenom konceptualnom modelu temeljem kojeg je moguće razviti rješenje koje omogućava programerima oštećena vida razvoj GUI sučelja u klasičnom stolnom programiranju na jednostavan i prihvatljiv način.
4. Razvijenom deskriptivnom jeziku pod nazivom GUIDL kao centralnom dijelu osmišljenog modela i sustava te razvijenom cjelokupnom GUIDL sustavu temeljem osmišljenog modela koji omogućava dobivanje izvornog koda ciljanih programskih jezika.
5. Potvrdi uporabljivosti GUIDL sustava u populaciji programera oštećena vida za potrebe razvoja grafičkih sučelja u domeni klasičnog stolnog programiranja.

1.4. Postignuti rezultati

U okviru ovog rada provedena su istraživanja s ciljem definiranja potrebe i zahtjeva programera oštećena vida za novim konceptom rješenja problema razvoja grafičkih sučelja. Istraženi su postojeći pristupi i napori na području prikaza grafičkih elemenata osobama

oštećena vida i definirani su mogući pristupi koji bi doveli do adekvatnog rješenja problema razvoja grafičkih sučelja za programere oštećena vida. Temeljem dobivenih spoznaja o mogućim pristupima, odabran je onaj koji bi mogao dovesti do adekvatnog rješenja te je izrađen konceptualni model takvog rješenja. Temeljem konceptualnog modela razvijen je novi deskriptivni jezik pod nazivom GUIDL (Graphical User Interface Description Language) kao centralni dio sustava za dobivanje izvornog koda ciljanih programskih jezika kojim su opisana grafička sučelja u pojedinoj tehnologiji. Provedeno je testiranje razvijenog prototipa rješenja kako interno, tako i na odabranoj populaciji osoba oštećena vida. Rezultati testiranja su pokazali da je novorazvijeni koncept i prototip adekvatan za osobe oštećena vida za potrebe razvoja grafičkih sučelja koja su uporabljiva u okviru aktualnih programskih tehnologija.

1.5. Struktura rada

Rad je podijeljen na devet poglavlja, uključujući uvod i zaključak. U uvodnom poglavlju definirani su ciljevi rada, hipoteze, postignuti rezultati i ostvareni znanstveni doprinosi te struktura rada. Drugo poglavlje opisuje terminologiju, vidne poteškoće i metode koje se primjenjuju kod izgradnje i evaluacije pomoćne tehnologije za osobe oštećena vida. Treće poglavlje opisuje aktualne probleme s kojima se osobe oštećena vida susreću prilikom rada s računalima i u programiranju te pomoćne (asistivne) tehnologije koje osobama oštećena vida pomažu u korištenju računalnih tehnologija. U četvrtom poglavlju je dan pregled problematike razvoja grafičkih sučelja od strane osoba oštećena vida te analiza pojedinih područja programiranja i dizajna u kojima je navedeni problem aktualan u manjoj ili većoj mjeri a navedena je i domena rada u kojoj će biti ponuđeno adekvatno rješenje. U ovom poglavlju se također identificiraju mogući pristupi rješavanju problema i odabire se najprikladniji pristup koji će biti razrađen u nastavku rada. Peto, šesto, sedmo i osmo poglavlje opisuju predloženo rješenje problema razvoja grafičkih sučelja od strane programera oštećena vida, daju uvid u istraživanje provedeno u cilju identificiranja aktualnosti navedenog problema, kao i u ključne aspekte koje je potrebno riješiti. U ovim poglavljima se također daje prikaz modela i funkcionalnosti osmišljenog sustava te rezultata testiranja prototipa sustava. Na kraju, zaključak sumira sve rezultate provedenih istraživanja te razvoja i testiranja novog sustava za pomoć osobama oštećena vida u razvoju grafičkih sučelja.

2. OSOBE S INVALIDITETOM I POMOĆNE TEHNOLOGIJE

Moderna znanost definira dva osnovna modela zajedno s pripadajućim pojmovima koji definiraju osobe i rad s osobama s invaliditetom:

- Medicinski model
- Socijalni model

Medicinski model se definira pojmovima poput: defekt, hendikep, oštećenje i nesposobnost, dok se socijalni model definira pojmovima kao što su: koncepti oštećenja i koncepti invalidnosti. U ovom slučaju na oštećenje se gleda kao na funkcionalno oštećenje koje ima podlogu tj. koje je uzrokovano fizičkim, senzorskim (osjetilnim) ili mentalnim oštećenje, dok se na invalidnost gleda kao na gubitak ili smanjenje mogućnosti sudjelovanja u svakodnevnim aktivnostima (Hersh & Johnson, 2008; Mršić, 1995; Zovko, 1994; Zovko, 1998).

Medicinski model se fokusira na fizička oštećenja i nemogućnost obavljanja određenih radnji ili nemogućnost percipiranja određenih dijelova objektivne stvarnosti putem vlastitih osjetila. Fokus medicinskog modela je na rehabilitaciji i unaprjeđenju funkcije navedenih ograničenja. Socijalni model se s druge strane fokusira na fizičke i društvene barijere s kojima se osobe s invaliditetom susreću (Swain et al., 2003). Socijalni model smatra da je glavni problem u društvu i društvenim barijerama, a ne u samoj osobi s invaliditetom i njezinim sposobnostima. Socijalni model je usporediv s pristupom koji se bazira na osnaživanju osoba s invaliditetom i pristupima koji uključuju korisnika kao centar modela i koji uključuju korisnika u pojedine aktivnosti društva (Damodaran, 1996; Rowley, 1998). Socijalni model je izradio Sindikat osoba s invaliditetom protiv segregacije (Fundamental Principles of Disability, 1976). Ovaj model je kasnije izmijenila Međunarodna organizacija osoba s invaliditetom (DPI – Disabled Peoples International) (Barnes, 1994).

Ukoliko se za primjer uzmu osobe oštećena vida, tada medicinski model identificira invaliditet osobe oštećena vida kao njezinu nemogućnost da čita slova standardne veličine u dnevnim novinama, dok socijalni model identificira invaliditet osobe oštećena vida kao društvenu neravnopravnost tj. postojanje samo malog broja knjiga u inačicama s velikim slovima. Drugim riječima, socijalni model smatra da je problem u dizajnu automobila a ne činjenici da su osobi noge nepokretne. Može se zaključiti da u svijetu djeluju dvije vrste

organizacija koje promoviraju i štite osobe s invaliditetom. Jedne se baziraju na medicinskom modelu i nastoje novim tehnologijama postići što veći stupanj rehabilitacije i funkcionalnosti osoba s invaliditetom, dok se druge bave promjenama stavova društva prema osobama s invaliditetom i osiguravanjem novih tehnologija i mogućnosti koje bi osobama s invaliditetom omogućile što ravnopravniju ulogu u društvu. Važnost socijalnog modela je priznala i Svjetska zdravstvena organizacija (WHO), na način da je prepoznala njegovu važnost i uvrstila ga kao termin u svoj sustav klasifikacije, a napravljene su i određene izmjene u terminima u inačici ove klasifikacije pod nazivom ICIHD2 - International Classification of Functioning Disability and Health (International Classification of Functioning, Disability and Health, 2001).

Socijalni model je pogodniji za područje istraživanja, dizajna i razvoja pomoćne tehnologije (Manduchi & Kurniawan, 2012) jer je cilj pomoćne tehnologije upravo rušenje barijera i omogućavanje uključivanja osoba s invaliditetom u što veći spektar društvenih i poslovnih aktivnosti u smislu postizanja jednakih šansi i jednakih mogućnosti za sve osobe. Socijalni model ima dvije temeljne zadaće a to su dizajniranje pomoćne tehnologije za što širi krug osoba koje su u potrebi za ovakvim tehnologijama te pomoću pomoćnih tehnologija otkloniti prepreke uključivanja osoba s invaliditetom u što veći spektar aktivnosti koje su omogućene osobama bez invaliditeta u smislu otvaranja što većeg broja mogućnosti i opcija za sve osobe s invaliditetom koje bi ih postupno dovele do ravnopravnosti s ostalima. Socijalni model nastoji putem pomoćne tehnologije premostiti jaz između onoga što bi osobe s invaliditetom željele moći raditi i onoga što postojeća društvena infrastruktura trenutno omogućava. Ova nastojanja socijalni model ostvaruje tehnologijom tj. raznim uređajima i programskim sustavima.

U skladu sa socijalnim modelom je razvijena i ideja o konceptima GUIDL jezika i sustava. Zadaća GUIDL sustava je premostiti barijeru na koju nailaze programeri oštećena vida u razvoju programskih proizvoda a to je upravo razvoj grafičkih sučelja u domeni klasičnog stolnog programiranja. GUIDL sustav je oblikovan na način da predstavlja svojevrsnu pomoćnu tehnologiju koju programeri oštećena vida koriste u okviru razvoja programa u određenoj programskoj tehnologiji i pripadajućoj razvojnoj okolini u kojoj stvaraju programski kod koristeći standardnu pomoćnu tehnologiju za rad s računalima tj. sintetizatore govora a u trenutku kad žele kreirati grafičko sučelje koriste pomoćnu tehnologiju u obliku GUIDL sustava kojom razvijaju sučelje u obliku datoteka koje je moguće pomoću razvojne

okoline odabranog programskog jezika uključiti u aktualni projekt te nastaviti pisati kod za pojedinu grafičku kontrolu sučelja u samoj programskoj razvojnoj okolini.

2.1. Mjerenje učinkovitosti pomoćnih tehnologija

Učinkovitost pomoćnih tehnologija se može mjeriti kroz poboljšanje korisnosti pojedinog uređaja ili sustava krajnjem korisniku ili kroz poboljšanje kvalitete pružanja usluga. Ova mjerenja su naročito važna zbog podataka o prestanku korištenja niza pomoćnih uređaja (Phillips & Zhao, 1993). Mjerenja učinkovitosti pomoćnih tehnologija bi također mogla imati pozitivan utjecaj na osiguranje odgovornosti za njihovu funkcionalnost (DeRuyter, 1997). Međutim, u praksi se relativno malo pozornosti posvećuje mjerenju učinkovitosti pomoćne tehnologije zbog nekoliko razloga, uključujući mišljenje da je učinak pomoćnih tehnologija očit i također zbog većeg fokusa na tehnologiju a manje na zadovoljstvo korisnika korištenjem iste (Fuhrer et al., 2003). Da bi se efikasno izmjerila učinkovitost pomoćne tehnologije, potrebno je razmotriti ne samo funkcioniranje tehnologije već se tehnologija i njeno korištenje te zadovoljstvo korisnika moraju promatrati zajedno. Također je potrebno uzeti u obzir da pojedina osoba s invaliditetom vrlo često koristi više od jedne pomoćne tehnologije te da je potrebno uzeti u obzir kontekst u kojem se one koriste a također je potrebno u obzir uzeti i osobne karakteristike korisnika pomoćne tehnologije (Gelderblom & de Witte, 2002).

Drugim riječima, potrebno je uzeti u obzir objektivno i subjektivno mjerenje. Objektivno mjerenje se odnosi na mjerenje performansi određene tehnologije, primjerice brzine i jasnoće kojom čitač teksta izgovara tekst, dok subjektivno mjerenje predstavlja stupanj ergonomije i zadovoljstva sa stanovišta krajnjeg korisnika koji koristi određenu pomoćnu tehnologiju (Smith, 1996). Subjektivna mjerenja se u praksi provode putem anketnih upitnika u kojima je uobičajeno da pojedini upitnici nisu ispunjeni u potpunosti što dovodi do potrebe za korekcijama što ponekad može značajno smanjiti veličinu uzorka (Kim & Curry, 1977). Prilikom definiranja pojedinih metrika za mjerenje subjektivnih kriterija zadovoljstva pojedinom pomoćnom tehnologijom izuzetno je važno u obzir uzeti i osobe s invaliditetom jer kriteriji tj. indikatori koje istraživač smatra važnim u pojedinim slučajevima ne moraju biti isti kao indikatori koje osoba s invaliditetom smatra važnim (Hughes et al., 1995). Uključenje osoba s invaliditetom u kreiranje metrike je važno kako bi indikatori bili potpuni i

sveobuhvatni s obzirom na ciljanu populaciju. Mjerenje učinkovitosti pomoćnih tehnologija je dio cjelokupnog mjerenja kvalitete života osoba s invaliditetom. U ovakvim mjerenjima se kao kriteriji kvalitete života često uzimaju kriteriji definirani Maslowljevom hijerarhijom ljudskih potreba (Sawicki, 2002; Maslow, 1968).

Kvaliteta života se kao pojam često spominje u medicinskoj literaturi i sve je češća komponenta ciljeva raznih kliničkih studija (Bowling, 1995; Bowling & Windsor, 1999). Posljedica ovog pristupa je određeni skup pretpostavki koje govore kako osobe s invaliditetom ne mogu imati u potpunosti zadovoljavajuću i dobru kvalitetu života što u praksi nije točno (Doward & McKenna, 2004). Također je važno napomenuti da još uvijek ne postoji koncenzus o tome što točno kvaliteta života predstavlja (Beckie & Hayduk, 1997). Bez obzira na to kvaliteta života je nešto o čemu se neprestano sve više piše i pojavljuje se na tisuće citata svake godine koji govore o toj temi (Fuhrer, 2000). Jedna od definicija kvalitete života govori kako je kvaliteta života razlika između želja i nada pojedinca i njegovih stvarnih iskustava (Calman, 1984). Temeljem ove definicije razvijene su pripadajuće mjerne skale i to u širem (O'Boyle et al., 1992; O'Boyle et al., 1993) i užem obliku (Browne et al., 1994; Hickey et al., 1996). U svojoj kraćoj formi ova mjerna skala se definira zasebno za svakog pojedinca a temelji se na 5 domena kvalitete života koje se smatraju najvažnijima (Mountain et al., 2004).

U praksi postoji više metoda mjerenja učinkovitosti pomoćne tehnologije a najpoznatije su: Life-H (Life Habits) kojom se mjeri stupanj društvene uključenosti tj. lakoća u obavljanju svakodnevnih društvenih i poslovnih aktivnosti (Noreau et al., 2002; Fouygerollas et al., 1998), OT (Occupational Therapy) Fact je metoda koja mjeri funkcionalnost osobe s invaliditetom u pojedinim radnjama kako objektivno tako i subjektivno (Smith, 2002), PIADS (Psychosocial Impacts of Assistive Devices Scale) metoda kojom se putem upitnika od 26 stavki podijeljenih u 3 grupe procjenjuje utjecaj pomoćne tehnologije na funkcijsku neovisnost, blagostanje i kvalitetu života osobe s invaliditetom (Day & Jutai, 1996; Day et al., 2002; Jutai & Day, 2002), QUEST 2.0 (Quebec User Evaluation of Satisfaction with Assistive Technology) metoda kojom se procjenjuje zadovoljstvo osobe s invaliditetom nekom od niza mnogih pomoćnih tehnologija putem pripadajućeg upitnika (Demers et al., 2002; Simon & Patrick, 1997), MPT (Matching Person and Technology) metoda kojom se procjenjuje prikladna pomoćna tehnologija za pojedinog korisnika u određenom kontekstu tj. određenoj okolini i uvjetima (Scherer & Craddock, 2002) a bazira se na modelu spajanja osobe s

tehnologijom (Scherer, 2000; Scherer & Cushman, 2000) i IPPA (Individually Prioritised Problem Assessment) metoda kojom se procjenjuje učinkovitost tehnologije kroz stupanj u kojem su se određene prepreke i barijere, s kojima se osoba s invaliditetom susretala, smanjile (Wessels et al., 2002) na način da se osobi postavi zadatak da navede 7 prepreka koje je imala prije korištenja pojedine pomoćne tehnologije te da navede za svaku prepreku način na koji je ona smanjena korištenjem određene pomoćne tehnologije.

GUIDL sustav se može promatrati kroz sve navedene metode mjerenja učinkovitosti pomoćne tehnologije. GUIDL sustav kao pomoćna tehnologija omogućava osobama oštećena vida da se uključe u obavljanje posla koji im je do sada bio relativno nedostupan tj. omogućava im kreiranje grafičkih sučelja (Life-H) i time im povećava funkcionalnost u poslovnom svijetu (OT). Povećanjem mogućnosti osobama oštećena vida raste i kvaliteta života jer mogu izraziti svoje ideje u području koje im je bilo relativno nedostupno (PIADS) a programeri oštećena vida su u mogućnosti izraziti svoje zadovoljstvo GUIDL pomoćnom tehnologijom koja se temeljem dobivenih povratnih informacija može dodatno razviti i unaprijediti (QUEST 2.0). GUIDL sustav se kao pomoćna tehnologija može upariti s korisnicima koji žele i imaju potrebu za ovakvom tehnologijom (MPT) a korisnici tj. programeri oštećena vida mogu nakon korištena ove pomoćne tehnologije identificirati prepreke koje su uklonjene njenim korištenjem a koje su bile prisutne prije pojave navedene pomoćne tehnologije (IPPA). Većina ovih aspekata je obuhvaćena kroz istraživanje potreba programera oštećena vida, kao i kroz testiranje uporabljivosti prototipa GUIDL sustava koje je prikazano u kasnijim poglavljima rada.

2.2. Definicije osnovnih pojmova

Pojam defekt se godinama najčešće koristio za označavanje nedostatka ili oštećenja no s vremenom se kao termin uvriježio u svakodnevnom govornom jeziku sa značenjem vrijeđanja i podcjenjivanja osobe te se danas zbog praktičnih razloga pojam defektne osobe izbjegava a umjesto njega se koristi pojam osobe s invaliditetom. Isto tako se u slučaju osoba s poteškoćama u vidu manje često koristi pojam slijepe osobe jer s praktične strane obuhvaća samo osobe koje su praktički u potpunosti izgubile vid, već se koristi pojam osobe oštećena vida koji pokriva sva moguća oštećenja i sve razine i vrste gubitaka vida. U nastavku su dane

definicije osnovnih pojmova koji se koriste u radu s osobama s invaliditetom tj. u radu s osobama oštećena vida.

Oštećenje označava bilo kakvo odstupanje u odnosu na normalnu fiziološku ili psihičku funkciju.

Nesposobnost označava bilo kakvu vrstu ograničenja sposobnosti osobe za pojedine akcije koje se podrazumijevaju kao normalne za sve ljude.

Hendikep označava poteškoću koja ograničava ili onemogućava osobi određenu aktivnost a proizlazi iz oštećenja ili invaliditeta.

Posljedica hendikepa je određeni stupanj izoliranosti osobe iz svoje okoline zbog nemogućnosti sudjelovanja u svim aktivnostima koje su na raspolaganju ostalim pripadnicima okoline.

2.3. Poteškoće koje uzrokuju smetnje u komunikaciji

Postoji puno fizioloških i psiholoških poteškoća koje mogu uzrokovati invaliditet no najčešće poteškoće koje uzrokuju najviše smetnji u komunikaciji osobe sa svojom okolinom su govorne, vidne i slušne poteškoće.

2.3.1. Govorne poteškoće

Govor je zvučna komunikacija između ljudskih bića koja se ostvaruje putem glasovnog kanala pomoću rečenica, riječi i slogova. Govorne poteškoće mogu biti uzrokovane različitim fiziološkim i psihološkim poremećajima a u razvoju osobe veliku ulogu u razvoju govora ima upravo sluh te vrsta i stupanj njegovog možebitnog oštećenja koje može dovesti do zaostajanja u razvoju sposobnosti pravilnog govora.

2.3.2. Vidne poteškoće

Kada se govori o vidnim poteškoćama, koristi se kao što je već spomenuto termin osobe oštećena vida. Ovaj pojam u sebi uključuje dvije skupine osoba s poteškoćama u vidu a to su slijepe i slabovidne osobe. Razlika je u tome što se slabovidnim osobama korištenjem određenih pomagala vid može u određenoj mjeri poboljšati (Manduchi & Kurniawan, 2012).

Sljepoća označava oštrinu vida na oku s boljom funkcijom uz korekcijsko staklo od 0,10 (10%) ili manje te centralni vid uz korekciju od 0,25 (25%) ili manji, uz suženo vidno polje na 20 stupnjeva ili manje.

Slabovidnost označava oštrinu vida na oku s boljom funkcijom uz korekcijsko staklo od 0,4 (40%) ili manje. Oštećenja vida se kategoriziraju prema puno kriterija kao što su vrijeme nastanka, utjecaj na funkciju oka, mjesto oštećenja i intenzitet oštećenja.

2.3.3. Slušne poteškoće

Od svih podjela osoba s oštećenim sluhom najjednostavnija je ona na gluhe i nagluhe osobe. Gluhe osobe imaju gubitak sluha od 80 dB (decibela) ili veći te nisu u stanju ni uz pomoć najmodernijih slušnih pomagala percipirati ljudski govor na dovoljno funkcionalan način za svakodnevno korištenje. Gubitak sluha može nastati samim rođenjem ili u kasnijoj životnoj dobi te se znatno odražava na sposobnost usvajanja govora.

Nagluhe osobe imaju gubitak sluha od 25 – 80 dB na uhu s boljom funkcijom sluha. Ove osobe najčešće uz pomoć odgovarajućih slušnih pomagala mogu percipirati ljudski govor u dovoljnoj mjeri za svakodnevno funkcioniranje a također imaju i dovoljno razvijene govorne sposobnosti.

2.4. Opći sociodemografski podaci za Republiku Hrvatsku

Stanje u Republici Hrvatskoj zaključno s danom 17.1.2013. govori o brojci od 520.437 osoba s invaliditetom na području Republike Hrvatske što čini oko 12% cjelokupnog stanovništva. Od ukupnog broja osoba s invaliditetom 313.217 (60%) osoba su muškarci a 207.220 (40%) su žene. Čak 278.564 (53,6%) osoba s invaliditetom je u radno aktivnoj skupini od 19 do 64

godine, dok je u radno neaktivnoj skupini tj. starosnoj skupini od 65 i više godina starosti 202.607 (38,9%) osoba. Invaliditet je prisutan također i u dječjoj dobi od 0 do 19 godina i to u opsegu od 7,5%. Najveći broj osoba s invaliditetom se nalazi u Gradu Zagrebu i Splitsko dalmatinskoj županiji u kojima ima prebivalište više od 30% osoba s invaliditetom (Benjak, 2013). Ukoliko bi se broj osoba s invaliditetom stavio u omjer s ukupnom populacijom županije, tada se najveći udio osoba s invaliditetom nalazi u Krapinsko zagorskoj županiji. Prikaz broja osoba s invaliditetom prema spolu, županijama prebivališta te dobnim skupinama je dan u tablici 2.1.

Tablica 2.1 Prikaz broja osoba s invaliditetom prema spolu, županijama prebivališta te dobnim skupinama (Benjak, 2013)

ŽUPANIJA PREBIVALIŠTA	Dobne skupine						Ukupno
	0-19		20-64		65+		
	m	ž	m	ž	m	ž	
BJELOVARSKO-BILOGORSKA	902	595	4949	2511	2720	3764	**15441**
BRODSKO-POSAVSKA	897	566	8316	2392	3161	3255	**18587**
DUBROVAČKO-NERETVANSKA	563	386	4750	2005	2122	1638	**11464**
GRAD ZAGREB	5691	3403	26416	18993	16413	20345	**91261**
ISTARSKA	832	502	4888	2684	2760	2551	**14217**
KARLOVAČKA	430	278	5745	2268	3067	4210	**15998**
KOPRIVNIČKO-KRIŽEVAČKA	1005	653	3942	2062	1780	2126	**11568**
KRAPINSKO-ZAGORSKA	1004	580	6588	4139	3946	4229	**20486**
LIČKO-SENJSKA	207	132	2557	812	1615	1199	**6522**
MEĐIMURSKA	1067	764	3465	2296	1831	2315	**11738**
OSJEČKO-BARANJSKA	1685	1163	14558	5483	5664	5837	**34390**
POŽEŠKO-SLAVONSKA	389	274	4310	1609	2103	1863	**10548**
PRIMORSKO-GORANSKA	542	369	7836	4633	5479	6787	**25646**
SISAČKO-MOSLAVAČKA	771	447	8411	2879	4033	3694	**20235**
SPLITSKO-DALMATINSKA	2934	1850	22710	11954	12077	11663	**63188**
ŠIBENSKO-KNINSKA	289	183	5927	2054	3524	3382	**15359**
VARAŽDINSKA	820	541	6675	4172	3907	5162	**21277**
VIROVITIČKO-PODRAVSKA	341	249	4457	1529	1666	1945	**10187**
VUKOVARSKO-SRIJEMSKA	889	607	8836	3005	3426	3122	**19885**
ZADARSKA	741	398	6623	2238	3567	2678	**16245**
ZAGREBAČKA	1974	1132	11627	6508	5839	4932	**32012**
Nespecificirano	138	83	14040	4712	10780	4430	**34183**
Ukupno	**24111**	**15155**	**187626**	**90938**	**101480**	**101127**	**520437**

3,5% tj. 17.979 osoba s invaliditetom u Republici Hrvatskoj ima oštećeni vid dok u gradu Zagrebu ima 2,2% osoba s invaliditetom tj. 1.985 osoba. Također, 3,7% tj. 1.245 djece s invaliditetom ima oštećen vid. U Hrvatskom savezu slijepih i Udruzi slijepih grada Zagreba je registrirano preko 170 osoba oštećena vida u radnom odnosu (Benjak, 2013). Podaci Svjetske zdravstvene organizacije izvještavaju o preko 285 milijuna osoba oštećena vida u svijeta, od

kojih je 39 milijuna slijepih i 246 milijuna slabovidnih osoba (10 Facts About Blindness and Visual Impairement, 2013). 90% osoba oštećena vida živi u zemljama u razvoju.

U Republici Hrvatskoj višestruka oštećenja koja utječu na cjelokupno funkcioniranje osobe s invaliditetom ima 28,2% osoba s invaliditetom. Broj osoba s invaliditetom prema vrstama oštećenja je prikazan u tablici 2.2.

Tablica 2.2 Prikaz vrsta oštećenja koje uzrokuju invaliditet ili pridonose stupnju funkcionalnog oštećenja osobe (Benjak, 2013)

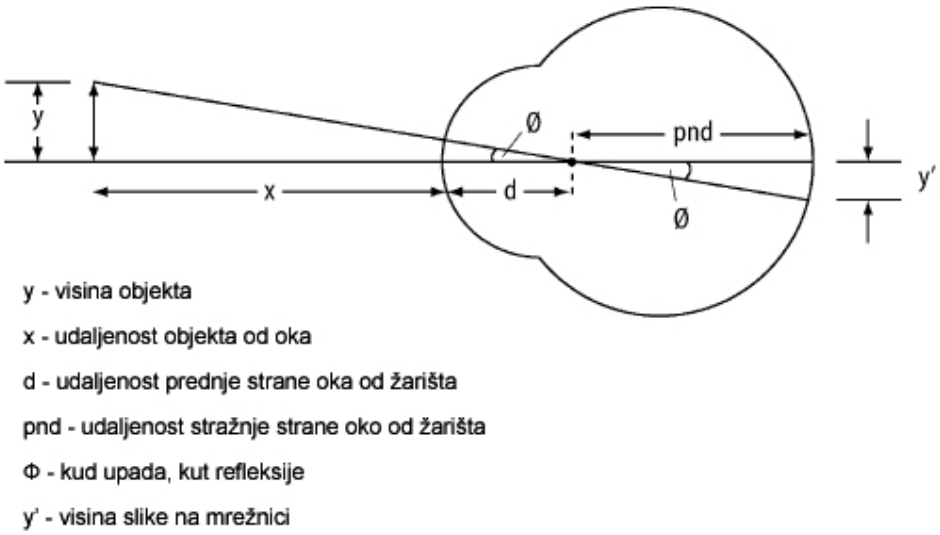
Vrste oštećenja	Ukupan broj	% od ukupnog broja osoba s invaliditetom	Prevalencija / 1000 stanovnika
oštećenje lokomotornog sustava	152005	29,2	35
duševni poremećaji	126277	24,3	29
oštećenje drugih organa	120934	23,2	28
oštećenje središnjeg živčanog sustava	95295	18,3	22
mentalna retardacija	22535	4,3	5
oštećenje vida	17979	3,5	4
oštećenje glasovno govorne komunikacije	16389	3,1	4
oštećenje sluha	13206	2,5	3
oštećenje perifernog živčanog sustava	12262	2,4	3
prirođene anomalije i kromosomopatije	8791	1,7	2
autizam	1198	0,2	0,3
višestruka oštećenja	146683	28,2	34

2.5. Vidni sustav

Poznato je da je upravo osjetilo vida najvažnije osjetilo zaduženo za pravilno funkcioniranje osobe u vlastitoj okolini. Putem osjetila vida osoba dobiva sliku koja opisuje njegovu okolinu a ovaj proces uključuje niz izuzetno složenih koraka obrade niza informacije putem mozga. Za osjetilo vida su zaduženi oko, očni živac i vidni put koji se nalazi u samom mozgu. Dokaz složenosti i važnosti ovog osjetila se može potražiti upravo u mogućnostima koje ono pruža. Čovjek je sposoban vidjeti okom i ono što je blizu i ono što je daleko, osoba je u stanju vidjeti ne samo ono u što joj je pogled usmjeren već sadržaj cjelokupnog vidnog polja a vid je također sredstvo putem kojeg osoba dobiva osjećaj orijentacije i prostorne dubine za što su zadužena oba oka u paru. Oko također ima veliku sposobnost prilagođavanja različitim intenzitetima svjetla širenjem ili sužavanjem zjenice a u stanju je vrlo precizno razlikovati cijeli spektar različitih boja.

2.5.1. Formiranje slike

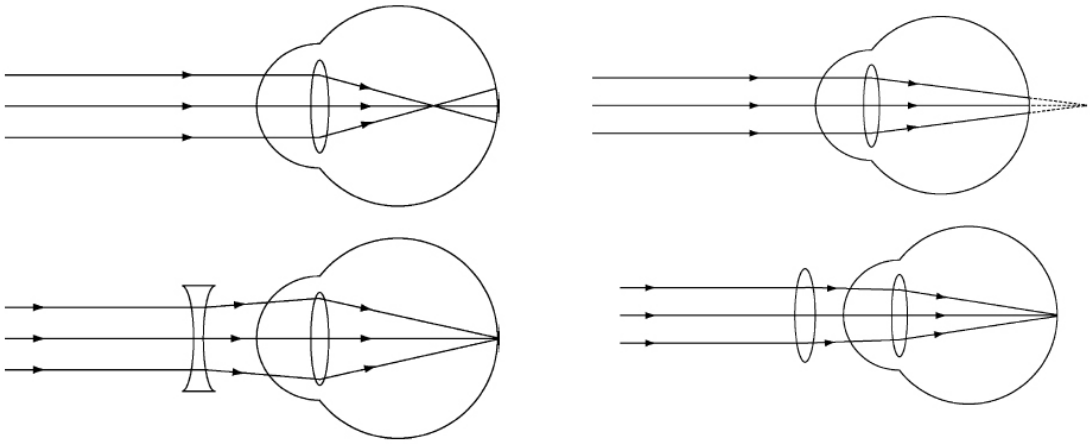
Oko se ne razlikuje u svojoj građi puno od klasične kamere ili fotoaparata. Oko ima svoj sustav leća, šarenicu koja je odgovorna za širenje i sužavanje zjenice tj. za propuštanje veće ili manje količine svjetlosti u oko, i stražnji segment tj. mrežnicu koja reagira na svjetlost i ima mogućnost izoštravanja slike. Svjetlost se u okolini osobe reflektira od raznih objekata i ulazi u oko gdje se kroz sustav leća lomi na odgovarajući način i pada na mrežnicu na kojoj nastaje obrnuta i umanjena slika onoga što se nalazi u vidnom polju (slika 2.1). Mozak unatoč ovoj činjenici percipira promatranu sliku na ispravno okrenuti način (Hersh & Johnson, 2008; Guyton & Hall, 2012).



Slika 2.1 Prikaz promatrane slike na mrežnici oka (Hersh & Johnson, 2008)

Velik broj osoba međutim ne vidi jasno sliku bez prilagodbe svjetla koje ulazi u oko odgovarajućom lećom. Ukoliko osoba ne vidi dobro na daljinu tada se radi o kratkovidnosti a ukoliko osoba ne vidi dobro na blizinu tada se radi o dalekovidnosti.

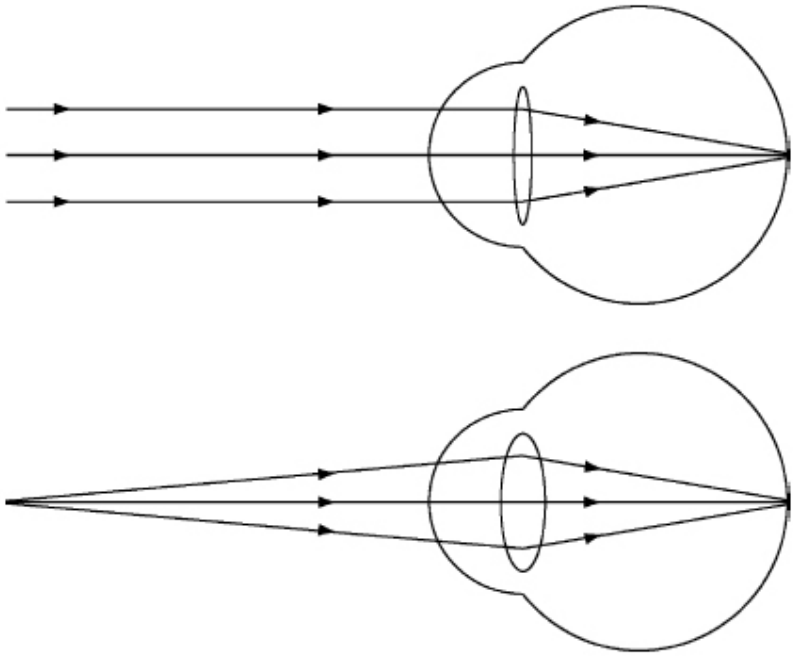
Kratkovidnost se definira kao refrakcijska pogreška oka kod koje se zrake svjetla lome ispred mrežnice, umjesto na njoj, dok se dalekovidnost definira kao refrakcijska pogreška oka kod koje se zrake svjetla lome iza mrežnice, umjesto na njoj. Ilustracija opisanih stanja oka je prikazana na slici 2.2.



Slika 2.2 Kratkovidnost i dalekovidnost bez i s korektivnom lećom (Hersh & Johnson, 2008)

2.5.2. Akomodacija oka

Akomodacija oka je sposobnost oka da se prilagodi gledanju bliskih odnosno dalekih predmeta. Oko se prilagođava ovim promjenama na način da smanjuje ili povećava žarišnu duljinu leće kako bi slika promatranog objekta pala na mrežnicu u mjesto koje će osigurati dovoljnu oštrinu. Slika 2.3 daje prikaz akomodacije oka na objekt koji se nalazi na većoj i manjoj udaljenosti.



Slika 2.3 Akomodacija oka na dalji i bliži objekt (Hersh & Johnson, 2008)

2.5.3. Binokularni vid i stereopsis

Iako jedno oko ima mogućnost prikazati sliku ljudi svijet promatraju s dva oka, no unatoč tome u mozgu dobivaju jednu jedinstvenu sliku o tome što se događa uslijed sposobnosti mozga da senzorički i motorički koordinira oba oka na besprijekoran način. Da bi se ovo postiglo, objekt koji promatramo pojedinim okom je zapravo pomaknut u sam centar vidnog polja, tj. da bismo vidjeli objekt od interesa on mora biti oslikan na oba oka tj. njihovim žutim pjegama. Svi ostali objekti koji nisu fiksirani se vide jednostruko s okom na čiju mrežnicu padaju. Ovaj proces naravno podrazumijeva normalnu fiziološku i psihološku funkciju. Da bi se ovaj efekt mogao u potpunosti razumjeti, moguće je pogledati u neki predmet ali pri tome u potpunosti opustiti očne mišiće, pri čemu će do malo prije fiksirani objekt koji se prikazivao jednostruko, biti nefiksiran i prikazivati se dvostruko tj. moći će se zamijetiti dvije slike istog objekta jer slike objekta više ne padaju na iste točke mrežnice.

Binokularni vid se kod ljudi može podijeliti na:

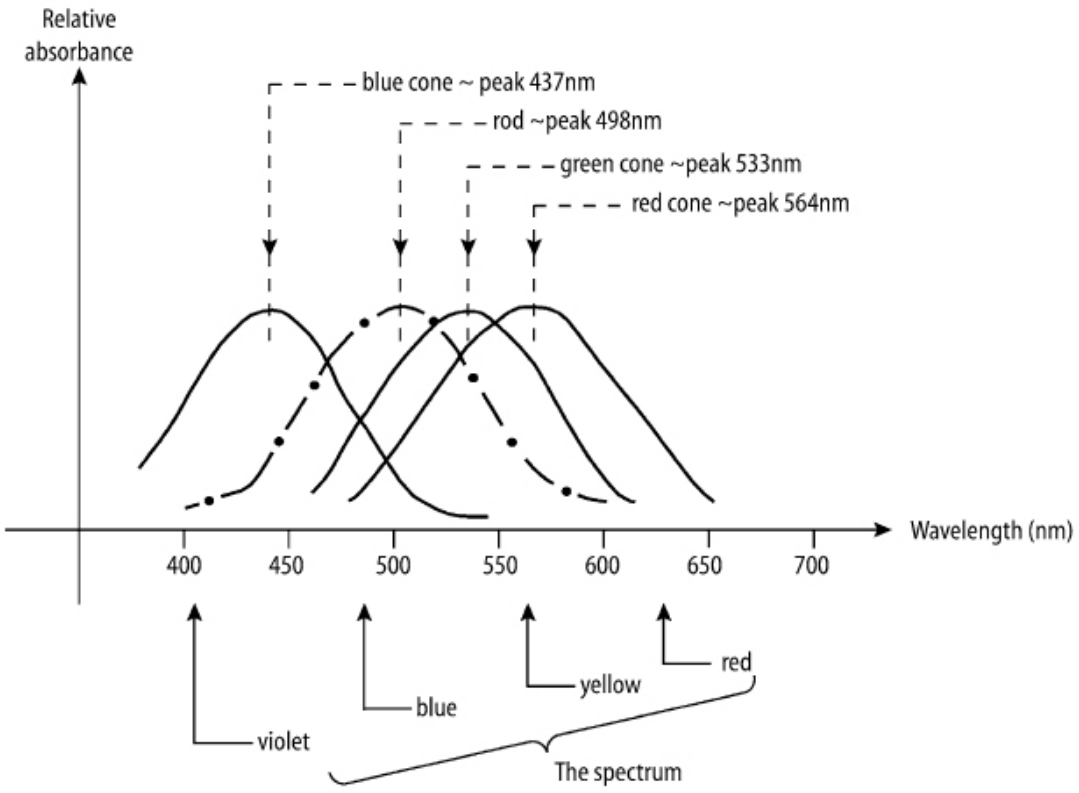
- **Simultanu percepciju** koja označava mogućnost da oba oka gledaju isti objekt.
- **Fuziju** koja označava mogućnost spajanje dvije slike u jednu.
- **Stereopsis** koji označava mogućnost gledanja objekata u tri dimenzije. Ovaj efekt je moguć zbog sličnog razloga kao što je moguće dobiti 3D sliku pomoću 3D kino naočala. Dakle, postoji mala razlika između dvije slike koje svako pojedino oko dobiva zbog nekoliko centimetara razmaka između očiju što dovodi do različitog kuta gledanja svakog oka. Proces fuzije zanemaruje ovu razliku a stereopsis je koristi kako bi osoba dobila dojam dubine prostora. Ukoliko osoba izgubi vid na jednom oku gubi i mogućnost percepcije dubine.

2.5.4. Nemogućnost prepoznavanja boja

Anatomski gledano, mrežnica uz vidne stanice sadrži i čunjiće i štapiće koji imaju ulogu u potpomaganju osjeta svjetla i u raspoznavanju boja. Čunjići su zaduženi za dnevni vid i raspoznavanje boja, dok su štapići zaduženi za noćni crnobijeli vid.

Ukoliko su opisane fiziološke strukture oštećene kod pojedinca, on može imati problema s prepoznavanjem boja. Prva osoba koja je postavila teoriju o raspoznavanju boja i oštećenju

ove sposobnosti (daltonizam) je bio Young a njegovom teorijom se eksperimentalno kasnije koristio i Helmholtz, te je zbog toga teorija nazvana Young-Helmholtzovom teorijom (Guyton & Hall, 2012). Prema ovoj teoriji u oku postoje tri vrste čunjića koji su različito osjetljivi na pojedinu boju, slično kao što kod slušnih struktura razlikujemo slušne niti koje su različito osjetljive na niske, srednje ili visoke frekvencije. Dakle, kao što nagluhost uzrokuje oštećenje ili potpuni nedostatak slušnih niti za pojedinu skupinu frekvencije, isto tako kod oka oštećenje ili pak potpuni nedostatak čunjića za pojedinu boju uzrokuje probleme s percepcijom i raspoznavanjem boja. Na slici 2.4 je prikazana osjetljivost pojedinih čunjića na nekoliko različitih boja.



Slika 2.4 Stimulacija pojedinih skupina čunjića različitim bojama (Hersh & Johnson, 2008)

2.5.5. Kortikalno oštećenje vida

Kortikalno oštećenje vida ili CIV (engl. Cortical Visual Impairment) označava privremeno ili trajno oštećenje koje je uzrokovano fiziološkim smetnjama u posteriornim dijelovima vidnih putova ili u okcipitalnom režnju mozga što dovodi do toga da mozak nije u stanju konzistentno razumjeti tj. interpretirati ono što oko vidi (Pedijatrijski materijal za vizualnu dijagnostiku i istraživanje, 2013).

2.6. Oštećenja vida i pripadajuća pomoćna tehnologija

Prema trenutnom pristupu dizajniranju pomoćne tehnologije za osobe oštećena vida, ne uzima se u obzir tip oštećenja već se pomoćna tehnologija razvija generalno za sve osobe koje imaju funkcionalno oštećenje vida u obliku potpune sljepoće ili djelomičnog oštećenja. Statistički gledano, samo 3% osoba oštećena vida je u potpunosti slijepo, svi ostali imaju neki oblik tj. stupanj vida što nameće pitanje o načinu dizajniranja pomoćne tehnologije tj. postavlja se pitanje bi li osobe oštećena vida bile u boljoj poziciji ukoliko bi se prilikom dizajniranja pomoćne tehnologije u obzir uzeo tip njihovog oštećenja vida (Hersh & Johnson, 2008). Ideja ovakvog pristupa koji uzima u obzir tip oštećenja vida je u mogućnosti iskorištavanja preostalog stupnja vida pojedine osobe u sklopu nove pomoćne tehnologije. Naravno, problem kod ovakvog pristupa je velik broj individualnih pomagala te ostvarivost ovakve ideja s praktičnog i financijskog aspekta. Sve ovo je trenutno samo ideja ali vrlo je izvjesno da će se ovakvi uređaju moći vidjeti u skoroj budućnosti.

GUIDL sustav je kao pomoćna tehnologija primjenjiv kako u slučaju potpuno slijepih osoba, tako i u slučaju osoba s oštećenim vidom u mjeri koja je tolika da ne dozvoljava jasno očitavanje kontura i oblika koje bi dozvolilo dizajniranje grafičkih sučelja putem grafičkih oblika i uporabe miša.

2.6.1. Demografski podaci vezani za osobe oštećena vida

Svjetska zdravstvena organizacija (WHO - World Health Organisation) procjenjuje da na svijetu trenutno postoji preko 45 milijuna slijepih odnosno 135 milijuna osoba s djelomično oštećenim vidom (Global Initiative for the Prevention of Avoidable Blindness, 1997; Blindness: Vision 2020 - The Global Initiative for the Elimination of Avoidable Blindness,

1999). Širom svijeta se broj osoba oštećena vida kao i uzrok oštećenja znatno razlikuju. Primjerice bakterija Chlamydia trachomatis je odgovorna za stvaranje ožiljaka na rožnici ili stanja koje se u medicini naziva trahom, što na svjetskoj razini predstavlja uzrok sljepoće u 18 – 24% slučajeva, no isto tako je poznato da je trahom uzročnik sljepila u industrijskih zemljama u samo 2% slučajeva (Zhang et al., 2004). Razlog ove činjenica leži upravo i nedostatku higijenskih i zdravstvenih uvjeta u zemljama koje su u razvoju tj. u siromašnijih zemljama. Prema svjetskoj zdravstvenoj organizaciji (WHO) oko 90% slijepih osoba živi u tzv. zemljama u razvoju (Global Initiative for the Prevention of Avoidable Blindness, 1997). Još jedan bitan faktor u oštećenju vida predstavlja i starenje. Osobe oštećena vida iz ove skupine predstavljaju značajan dio ukupne populacije osoba oštećena vida. U Ujedinjenom Kraljevstvu je gotovo 73% osoba oštećena vida starije od 80 godina (Bruce et al., 1991; Walker et al., 1992). U industrijski razvijenim zemljama se postotak osoba oštećena vida povećava s faktorom 3 svakih 10 godina nakon što osoba navrši 40 godina (Taylor & Keeffe, 2001). Neke bolesti su također bitno učestalije kao razlog oštećenja vida kod starije populacije. Primjerice makularna degeneracija koja napada makulu tj. žutu pjegu uzrokuje preko 39% oštećenja vida u svijetu a naročito je vjerojatna kod starijih osoba kod kojih se njezina vjerojatnost povećava za 8% nakon što navrše 65 godina života.

Katarakta ili siva mrena je jedan od najčešćih uzroka oštećenja vida kod starijih osoba. Katarakta uzrokuje zamućenje leće oka i postepeni gubitak vida sve do slijepila, ukoliko se ne liječi. Katarakta više pogađa žene i odgovorna je za preko 45% slučajeva oštećenja vida u svijetu (World Health report – life in the 21st century: a vision for all, 1998). Vjerojatnost katarakte se povećava za 40% nakon što osoba doživi 70 godina života. Bolest koja je druga po učestalosti oštećenja vida (u postotku od otprilike 15%) je glaukom (kod kojeg dolazi do oštećenja stanica mrežničnih ganglija) koji također postaje sve vjerojatniji s povećanjem godina starosti, i to za oko 1% nakon navršene 65. tj. za oko 7% nakon navršenih 80 godina života. Jedan od najčešćih uzročnika glaukoma je previsoki očni tlak koji ne uzrokuje nikakve posebne simptome i samim tim dovodi do činjenice da većina ljudi s glaukomom nisu ni svjesni da ga imaju (Swain et al., 2003). Razne bolesti mrežnice dovode do oštećenja oka u oko 9% slučajeva s povećanjem vjerojatnosti od 2% nakon 65. tj. od 7% nakon 75. godine života.

2.6.2. Prikaz nekih tipova oštećenja vida

Na sljedećih nekoliko slika bit će prikazane ilustracije nekoliko tipova oštećenja vida. Slika 2.5 prikazuje čistu slike, bez ikakvih oštećenja vida, slika 2.6 prikazuje oštećenje oka u području centralnog vidnog polja koje je zamućeno, dok je periferni vid oštriji. Kod ovakvog tipa oštećenja osoba će biti svjesna objekta ispred sebe ali će teško procijeniti detalje objekta ili njegovu veličinu. Na slici 2.7 prikazano je blago oštećenje vidnog polja koje i dalje omogućava razmjerno dobro snalaženje i kretanje no opet postoji problem u raspoznavanju detalja. Na slici 2.8 je prikazan veći gubitak oštrine vida kod kojeg je kretanje jako otežano baš kao i prepoznavanje objekata. Slika 2.9 prikazuje stanje poznato kao nistagmus (engl. nystagmus) kod kojeg se kod osobe javlja nesvjesno pomicanje očiju u svim smjerovima što može dovesti do gubitka oštrine vida. Gubitak perifernog vida, tzv. vizija tunela, je prikazan na slici 2.10 kod koje je centralni dio slike jasan, ali periferni dio slike ne postoji. Smanjenje vidnog polja je prikazano u nešto drugačijem obliku i na slici 2.11. Slika 2.12 prikazuje oštećenje površine oka koje dovodi do pojave manjeg ili većeg broja slijepih točaka koje u manjoj ili većoj mjeri ograničavaju vid. Na slici 2.13 je prikazano oštećenje u obliku svjetlosnog prstena koji postaje sve veći što može dovesti do dekoncentracije prilikom dinamičnih situacija i kretanja.



Slika 2.5 Zdravo oko bez oštećenja (Hersh & Johnson, 2008)



Slika 2.6 Gubitak oštrine centralnog vida (Hersh & Johnson, 2008)



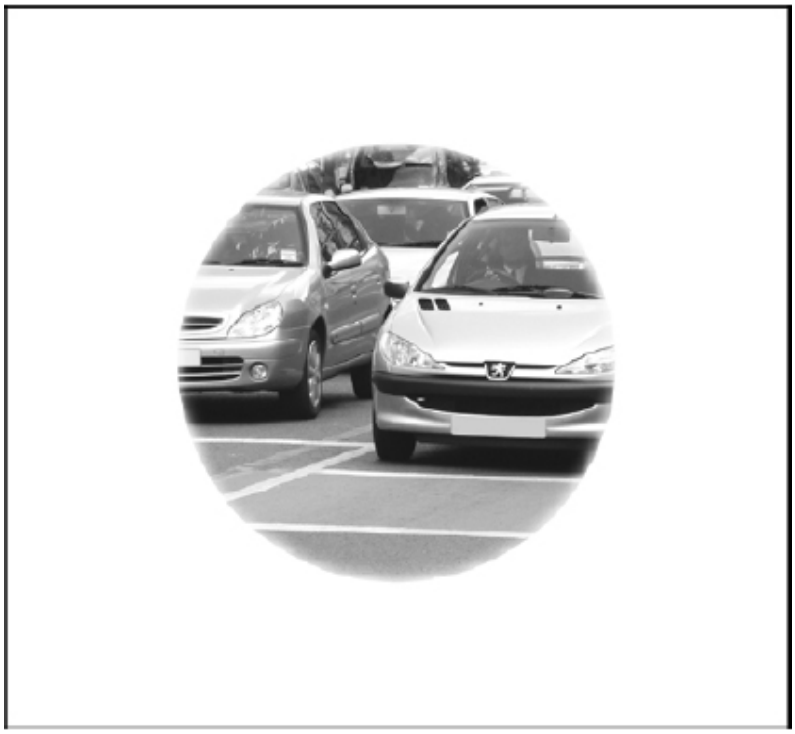
Slika 2.7 Manji gubitak oštrine vida (Hersh & Johnson, 2008)



Slika 2.8 Veći gubitak oštrine vida (Hersh & Johnson, 2008)



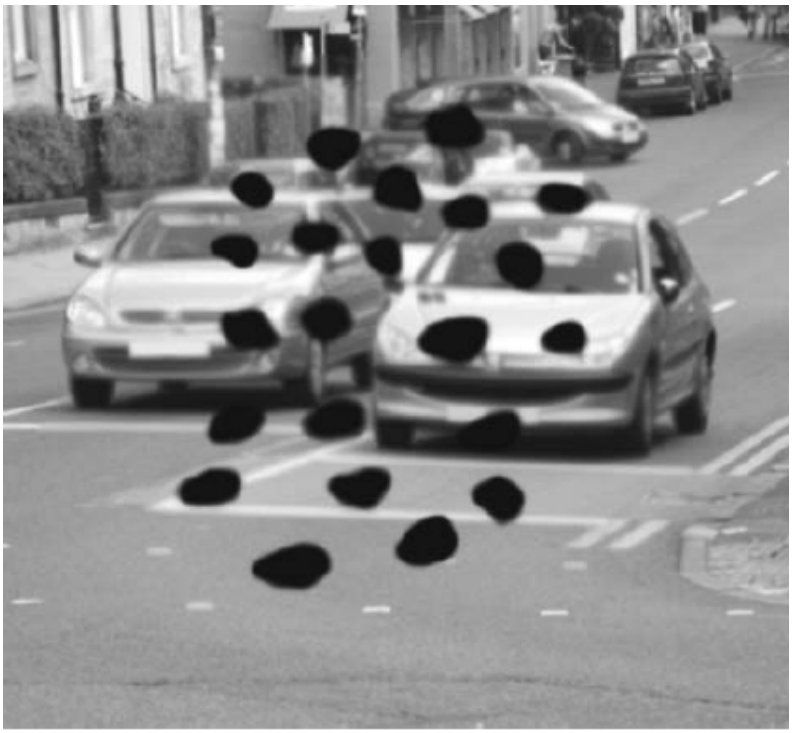
Slika 2.9 Gubitak vida zbog nistagmusa (Hersh & Johnson, 2008)



Slika 2.10 Tunelski vid (Hersh & Johnson, 2008)



Slika 2.11 Gubitak dijela vidnog polja (Hersh & Johnson, 2008)



Slika 2.12 Površinska oštećenja oka (Hersh & Johnson, 2008)



Slika 2.13 Oštećenje oka u obliku svjetlosnog prstena (Hersh & Johnson, 2008)

2.7. Modeliranje sustava pomoćne tehnologije

Pomoćna tehnologija označava tehnologiju, uređaje i sustave koji pomažu osobama s invaliditetom da prevladaju svakodnevne barijere kako društvene tako i poslovne i druge kako bi se u što većoj mjeri uključili u normalni društveni život. Cilj pomoćne tehnologije je premostiti jaz između postojeće infrastrukture koja je prilagođena osobama bez poteškoća i potreba osoba s invaliditetom. Pomoćna tehnologija dakle predstavlja most između postojeće infrastrukture i mogućnosti osobe s invaliditetom te joj omogućava korištenje te infrastrukture.

Korisnici pomoćnih tehnologija se jako razlikuju u svojim karakteristikama, tipu invaliditeta, potrebama i željama, znanju i vještinama te se samim tim nameće potreba za stvaranjem odgovarajućih modelirajućih okvira unutar kojih bi se potrebne pomoćne tehnologije razvijale.

Osnovni ciljevi modelirajućeg okvira su (Hersh & Johnson, 2008):

- da bude primjenjiv na sve tipove pomoćnih tehnologija i sustava
- da bude osnova za izradu klasifikacijskog okvira pomoćnih tehnologija i sustava
- da pruži temeljnu strukturu pomoćnih tehnologija te da se može upotrijebiti u specifikacijama specijaliziranih uređaja i tehnologija
- da pruži okvir za razvoj novih pomoćnih tehnologija i sustava na način da isti budu izrađeni po mjeri i potrebama krajnjih korisnika
- da pruži okvir koji će osigurati razvoj pomoćne tehnologije na način da ista bude prihvatljiva krajnjem korisniku
- da omogući znanstvenicima dublji uvid u funkcioniranje pomoćne tehnologije unutar određenog društvenog konteksta

S obzirom da su situacije u realnom svijetu jako složene i individualne s obzirom na situaciju i stanje krajnjeg korisnika, prilikom razvoja pomoćne tehnologije često se ne mogu zadovoljiti svi principi modelirajućeg okvira te se zbog toga nameću dva moguća pristupa:

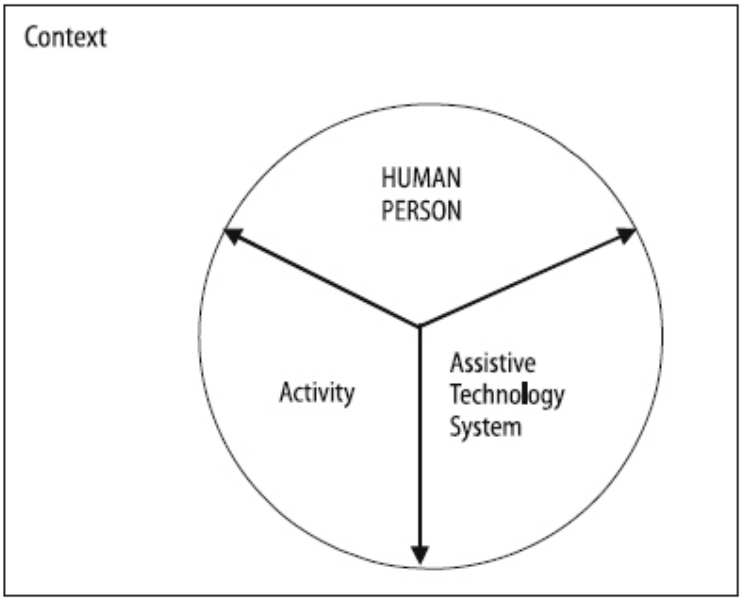
- razvoj pomoćne tehnologije uzimajući u obzir korisnika i njegove krajnje potrebe te mjerenje stupnja zadovoljavanja tih potreba razvijenom pomoćnom tehnologijom
- razvoj generičkog tj. općeg modela za potrebe analize pojedinog uređaja ili tehnologije

Prilikom modeliranja pomoćne tehnologije potrebno je uzeti u obzir mnoge aspekte i upravo u tome pomažu temeljni modeli poput HAAT (engl. Human Activity Assistive Technology) i CAT (engl. Comprehensive Assistive Technology) modela (Hersh & Johnson, 2008).

2.7.1. HAAT model u modeliranju pomoćne tehnologije

HAAT (engl. Human Activity Assistive Technology) model prema Cook-u i Husse-u (2002.) daje podlogu za razvoj opće strukture koja se koristi za analizu, sintezu i razvoj pomoćne tehnologije, ali ne i za uparivanje korisnika s tehnologijom (Hersh & Johnson, 2008). Ovaj model počinje s definicijom sustava pomoćne tehnologije i nastavlja s osobom s invaliditetom koja uz pomoć određene pomoćne tehnologije obavlja određene aktivnosti. Ovaj sustav se sastoji od četiri promatrane komponente koje su prikazane i na slici 2.14 (Hersh & Johnson, 2008):

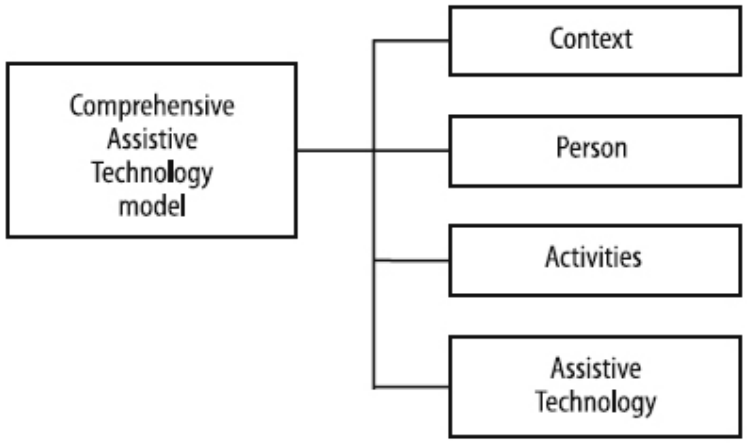
- **Kontekst** predstavlja društveni okvir i okruženje u kojem korisnik i pomoćna tehnologija funkcioniraju.
- **Korisnik** predstavlja osobu koja se nalazi u samom centru modela.
- **Aktivnosti** predstavljaju radnje koje korisnik izvršava i koje utječu na cijeli model.
- **Pomoćna tehnologija** predstavlja tehnološko rješenje za premošćivanje prepreka koje se javljaju u određenom okruženju i kontekstu.



Slika 2.14 HAAT model (Hersh & Johnson, 2008)

2.7.2. CAT model u modeliranju pomoćne tehnologije

CAT (engl. Comprehensive Assistive Technology) model je nastao iz HAAT modela (Hersh & Johnson, 2008). Cilj ovog modela je prikazati nedostatke u korištenju određene pomoćne tehnologije. Posebno je pogodan za razvoj pomoćnih tehnologija u područjima u kojima ne postoje adekvatna rješenja ili za unaprjeđenje postojećih sustava s novim mogućnostima, no također je pogodan za uparivanje korisnika s određenom pomoćnom tehnologijom. Prikaz CAT modela je dan na slici 2.15.



Slika 2.15 CAT model (Hersh & Johnson, 2008)

Iz slike 3.2 je vidljivo da se CAT model sastoji od četiri komponente koje definiraju sustav konačne pomoćne tehnologije:

- **Kontekst** – okruženje u kojem će se pomoćna tehnologija koristiti
- **Korisnik** – konačni korisnik koji se nalazi u samom centru modela
- **Aktivnosti** – skup aktivnosti u kojima će se koristiti nova pomoćna tehnologija
- **Pomoćna tehnologija** koja će se koristiti

Svaka od 4 glavne grane CAT modela se dalje dekomponira na grane sljedeće razine, ovisno o konkretnom slučaju i potrebi. CAT model donosi nekoliko prednosti a to su prvenstveno mogućnost kontroliranja razine složenosti modela. Drugim riječima, s obzirom da se CAT model prikazuje u stablastoj hijerarhijskoj strukturi, vrlo je jednostavno dodati nove faktore i učiniti model preciznijim, ali kompleksnijim, ili oduzeti tj. izostaviti faktore koji za pojedini

slučaj nisu nužni te na taj način učiniti model jednostavnijim te lakše shvatljivim i primjenjivim.

U razvoju GUIDL sustava su uzete u obzir sve 4 komponente CAT modela. Kontekst GUIDL sustava je radna okolina koja se sastoji od timova i pojedinaca koji razvijaju programske proizvode, cjelokupno tržište rada, stav društva prema osobama oštećena vida kao programerima i dizajnerima, državne poticajne mjere za razvoj pomoćnih tehnologija koje bi pomogle osobama oštećena vida prilikom zapošljavanja i obavljanja određenih poslova, itd. Korisnik GUIDL sustava je osoba oštećena vida koja se želi baviti programiranjem tj. dizajniranjem sučelja u profesionalne ili edukacijske svrhe. Aktivnosti koje GUIDL sustav podržava su aktivnosti razvoja grafičkih sučelja u domeni klasičnog stolnog programiranja a GUIDL sustav sam po sebi predstavlja pomoćnu tehnologiju koja osobama oštećena vida omogućava lakši i brži razvoj grafičkih sučelja kao i uključivanje sa svojim idejama i konkretnim dizajnerskim doprinosima u cjelokupni proces razvoja programskog proizvoda kao ravnopravnih članova razvojnog tima.

3. OSOBE OŠTEĆENA VIDA U SVIJETU RAČUNALA I PROGRAMIRANJA

Otkad su računala postala prepoznatom i općeprihvaćenom tehnologijom i otkad su programske aplikacije započele svoj intenzivniji razvoj, javila su se nastojanja da se uključi osobe oštećena vida u korištenje ove tehnologije (Hodson, 2004). Najbrži rast popularnosti računala i njihove praktične primjene dogodio se 1970-ih i 1980-ih godina. Mogućnosti primjene računala u različitim područjima života, poslovnog svijeta i edukacije su bile jako velike. Od tog vremena, bili su prisutni napori da se osobe oštećena vida uvedu u ovu novu i brzorastuću tehnologiju te da im se omogući praćenje njezinog brzog trenda rasta.

Masovna uporaba osobnih računala kod kuće i unutar raznih postrojenja i institucija je donijela veliku korist osobama oštećena vida. Nekoliko tehnologija je uvedeno u svrhu pomoći osobama oštećena vida u korištenju računala, poput tipkovnice bazirane na Braillovom pismu na kojoj su osobe oštećena vida mogle tipkati pomoću Braillovih znakova umjesto korištenja klasične tipkovnice koja pruža osobi oštećena vida malo informacija o tome je li željeni unos unesen ispravno ili ne. Druga tehnologija koja se pojavila uključivala je konverziju teksta u govor, tzv. tekst-u-govor konverzija, korištenjem sintetizatora govora pomoću kojeg su osobe oštećena vida mogle čuti slova i riječi koje su se pojavljivale na ekranu kao ulazni ili izlazni podaci.

Navedeni napori su rezultirali velikim uspjehom u edukaciji i zapošljavanju osoba oštećena vida. Edukativni materijali koji su bili većinom grafičkog karaktera su postali dostupni osobama oštećena vida preko novih tehnologija. Ista tehnologija, namijenjena ostvarivanju pomoći osobama oštećena vida u edukaciji, omogućila je osobama oštećena vida da postanu profesionalni programeri s obzirom da je sav softver bio baziran na tekstu, što nije predstavljalo problem za postojeća tehnološka rješenja u obliku softvera namijenjenog pomaganju osobama oštećena vida. Sintetizatori govora su bili u mogućnosti pokriti uporabu svog softvera na tržištu i osobe oštećena vida su dobile priliku postati profesionalni programeri. Uključivanje osoba oštećena vida u programerski svijet je bilo uspješno, a interes osoba oštećena vida za ovo područje nije nikada nestao (Alexander, 1998). Uključivanje osoba oštećena vida u svijet programiranja je tema koja je aktualna i danas (Blind Programming Project in NetBeans IDE, 2010).

U trenutku pisanja ovog rada, preko 130 slijepih programera profesionalaca je registrirano u Američkoj udruzi slijepih s uspješnim karijerama, a ako se uzme u obzir situacija u nekim drugim zemljama, može se vidjeti da je interes za ovu profesiju također prisutan. Primjerice u Italiji postoji više od 200 slijepih osoba i osoba koje imaju poteškoće s vidom, a zaposlene su kao programeri unutar zadnjih nekoliko godina (Cattani, 2009). Programiranje je navedeno kao jedna od popularnih profesija među slijepima u Češkoj Republici, Finskoj, Poljskoj i Španjolskoj (bfi Steiermark's European Labour Market Report, 2004). Ako se pogleda lista tipičnih profesija za slijepe u Europi, može se vidjeti da je programiranje navedeno kao jedna od vrlo obećavajućih profesija za slijepe (bfi Steiermark's European Labour Market Report, 2004). Google, vlasnik poznate svjetske Internet tražilice ima zaposlena dva slijepa programera, Peter Lundblada i T.V. Ramana. Udruženja slijepih postoje u mnogim zemljama pa tako i u Hrvatskoj gdje je registrirano preko 6000 osoba oštećena vida, a porazna je činjenica da prema statističkim podacima samo 50-ak od njih aktivno koristi računalo iako ono može uvelike pomoći osobama oštećena vida u mnogim segmentima razvoja vlastitih kompetencija i uključivanja u društvo.

Međutim, ovaj trend koji je pogodovao programerima oštećena vida se uskoro promijenio. 1980-ih godina razvoj grafičkih korisničkih sučelja (GUI – Graphical User Interface) je postao stvarnost i iako prijelaz na grafička sučelja nije bio trenutačan, postalo je očito da postojeća tehnologija za pomoć osobama oštećena vida neće biti dostatna u vrlo bliskoj budućnosti i da će polako, ali sigurno, postati praktički neupotrebljiva u nekim segmentima novih trendova razvoja računalnog softvera. Ove pretpostavke su se ubrzo pokazale točnima s obzirom da je tekstualan oblik sučelja pojedinih operativnih sustava i softvera polako napušten i da su grafičke forme zamijenile tekst. Sve ovo je ostavilo osobe oštećena vida u vrlo nezavidnom položaju.

Nekoliko problema se pojavilo s obzirom na aktualne trendove. Brz razvoj novih tehnologija nije bio popraćen prikladnom dokumentacijom za osobe oštećena vida a postojeće tehnologije za pomoć osobama oštećena vida nisu bile u stanju pratiti tako brz razvoj različitih softverskih rješenja i novih koncepata. Nove programske paradigme i nova programska okruženja postajali su sve zahtjevniji i kompleksniji. Uz rastuću kompleksnost, nove tehnologije su bile bazirane na grafičkim sučeljima što je predstavljalo dodatni problem za postojeće sintetizatore govora koji su pokušavali čitati tekst s gumbi, izbornika i ostalih grafičkih elemenata, no tim

načinom rada je kontekst ekrana i njegovih elemenata bilo gotovo nemoguće specificirati i predočiti.

Najveći problem je predstavljala nemogućnost osoba oštećena vida da izrađuju grafička sučelja koja su postala sastavni dio procesa razvoja softvera. Grafička sučelja bazirana na pokazivaču miša i akciji klika mišem (point-and-click) nisu bila od puno koristi profesionalnim programerima oštećena vida, a tekstualni opis određenih kontrola je postao praktički nemoguć (Franqueiro & Siegfried, 2006). Sve ove okolnosti su ostavile gorak okus unutar zajednice programera oštećena vida. Navedeni problemi su nametnuli potrebu za analizom postojećih rješenja u području pomoći osobama oštećena vida u korištenju računala i programiranju te detaljnom analizom problema koje treba riješiti zajedno s prijedlozima i konceptima koji bi omogućili njihovo rješavanje.

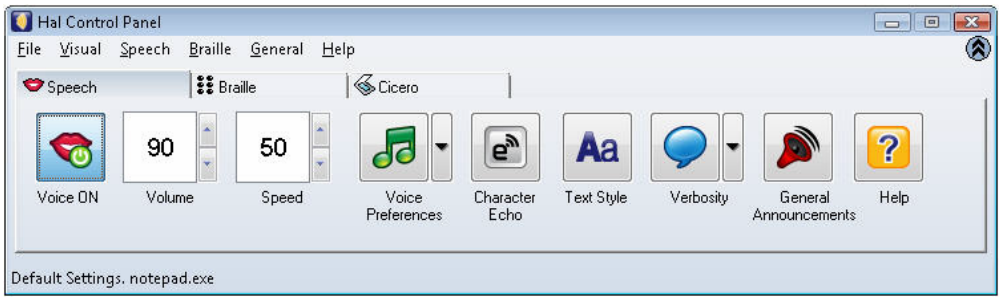
3.1. Programski alati za pomoć slijepima u radu s računalom

Programski alati za pomoć osobama oštećena vida u području svakodnevnog korištenja računala su isti kao alati korišteni za pomoć programerima profesionalcima oštećena vida. Svi alati ove vrste su bazirani na tekst-u-govor konverterima tj. sintetizatorima govora koji interpretiraju tekst na ekranu koji je ispisan kao izlazni rezultat nekog programa ili koji je unesen kao ulaz. Tekst se konvertira u govor ili ponekad u Braillovo pismo pomoću posebne ploče za tu namjenu. Sintetizatori govora mogu interpretirati cijele riječi u slučaju poznatih fraza i poznatih govornih jezika. Programski alati za pomoć osobama oštećena vida se nazivaju još i pomoćna tehnologija (assistive technology) (Francioni & Smith, 2002).

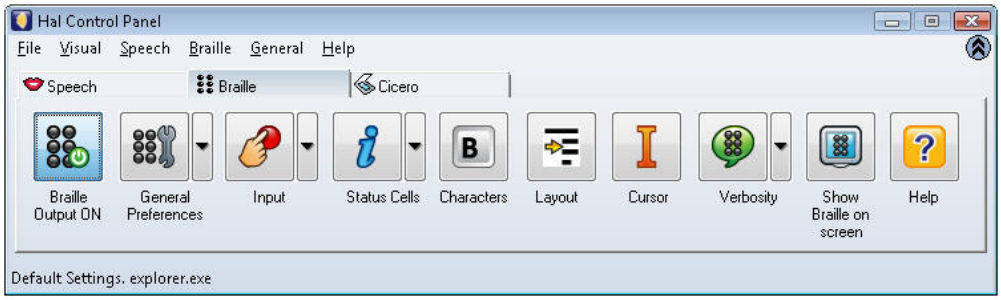
Neki od najpopularnijih i najpoznatijih programskih alata za pomoć osobama oštećena vida danas su:

- JAWS (Rosmaita, 2006)
- HAL Screen Reader (Pitt & Edwards, 1996)
- COBRA (Screen Reader COBRA, 2011)
- Window Eyes (GW Micro - Window-Eyes, 2011)
- Easy Web Browsing (Easy Web Browsing, 2011)

Sučelje alata HAL je prikazano na slikama 3.1 i 3.2.



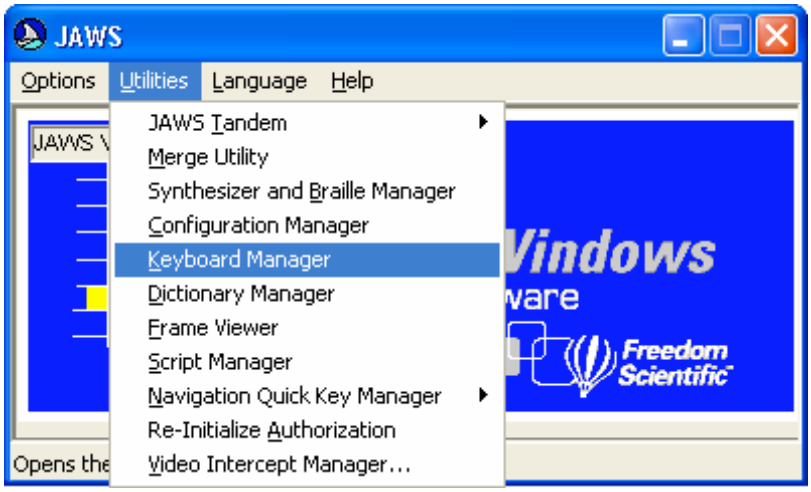
Slika 3.1 HAL govorne opcije (SuperNova Screen Reader, 2011)



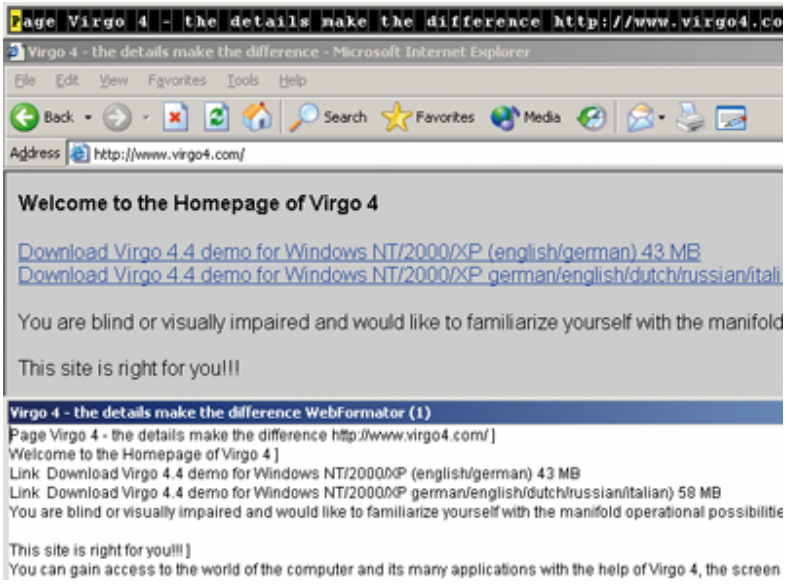
Slika 3.2 HAL Braille opcije (SuperNova Screen Reader, 2011)

JAWS je najpoznatiji i najkorišteniji čitač ekrana u svijetu. Podržava mnoge programske aplikacije, a ima i podršku za pretraživanje Interneta. JAWS ima vlastitu zvučnu karticu i interni sintetizator govora koji podržava tekst-u-govor konverziju, ali i mogućnost prevođenja teksta u Braillovo pismo. Sučelje alata JAWS je prikazano na slici 3.3. HAL Screen Reader prevodi tekst u govor koristeći zvučnu karticu računala, podržava različite Braillove uređaje i prepoznaje standardne Windows elemente (gumbe, ikone, dijaloške okvire, itd.). COBRA predstavlja zamjenu za poznati alat Virgo4 koji je imao specijalnu značajku koja je omogućavala korisniku odabir informacija koje će se prikazati putem audio kanala, a koje putem Braillovog pisma. Alat Virgo 4 je prikazan na slici 3.4. COBRA donosi, zajedno s već spomenutim značajkama alata Virgo4, još više mogućnosti poput podrške za još više tipova Braillovih ekrana, poboljšanu podršku za Microsoft Office alate, povećalo sadržaja na ekranu i mogućnost prikaza preko punog ekrana ili podijeljeni prikaz ekrana. Window Eyes podržava Braillovo pismo i tekst-u-govor konverziju. Sve Windows kontrole su podržane inicijalno, a postoji i mogućnost modifikacije postavki u svrhu prepoznavanja drugih nestandardnih kontrola. Easy Web Browsing je zamjena za alat Home Page Reader koji je imao

specijalizirane značajke za podršku prikaza web stranica. Postojala je podrška za prepoznavanje raznih elemenata web stranica, poput slika, okvira, elemenata formi, itd. Home Page Reader nije podržan od 2005. godine. Njegova zamjena, Easy Web Browsing ima sve značajke svog prethodnika, uz neke nove, još sofisticiranije mogućnosti, poput opcije povećavanja bilo kojeg dijela ekrana, opcije promjene pozadine web stranice u željenu boju što rezultira čitljivijim tekstom i opcije dodavanja zvukova raznim događajima, kao što je dovršetak preuzimanja datoteka s Interneta.



Slika 3.3 Alat JAWS (Configuring Your Machine For Testing With A Screen Reader, 2008)



Slika 3.4 Virgo4 i Internet Explorer

Bez obzira na to što čitači ekrana imaju prisutnu razvojnu podršku, postoji nekoliko bitnih ograničenja (Konecki, et al., 2011):

- Interpretacija slika – Čitači ekrana ne mogu interpretirati slike. Jedini elementi slike koji mogu biti interpretirani su naziv slike i eventualno opisi slike definirani uz sliku. Primjer je <alt> oznaka unutar HTML koda.
- Grafički raspored – Čitači ekrana čitaju informacije linearno i samim tim kompleksne grafičke konstrukcije na ekranu ne mogu biti adekvatno predočene. Elementi ekrana ne mogu biti preskočeni i kontekst rasporeda ekrana je teško opisati i zamisliti na linearan način.
- Interpretacija podatkovnih tablica – S obzirom da su tablice također čitane i interpretirane linearno, može se zaključiti da je takav pristup zadovoljavajuć kada se radi o relativno malim tablicama, no u slučaju velikih tablica s većim brojem redova i stupaca vizualizacija tablice postaje vrlo teška jer pamćenje većeg broja podataka postaje problematično.
- Grafički dijagrami – Grafički dijagrami se sastoje od raznih grafičkih oblika i vrijednosti koje je teško opisati i percipirati čak i za jednostavnije, a naročito za kompleksnije dijagrame.
- Boje – Boje su sastavni dio mnogih grafičkih formi i raznih grafičkih elemenata te mogu biti vrlo bitan dio konteksta, a njihov opis predstavlja poseban izazov.
- Robusnost – Osjetljivost alata na nove tehnologije i nemogućnost praćenja novih softverskih rješenja.

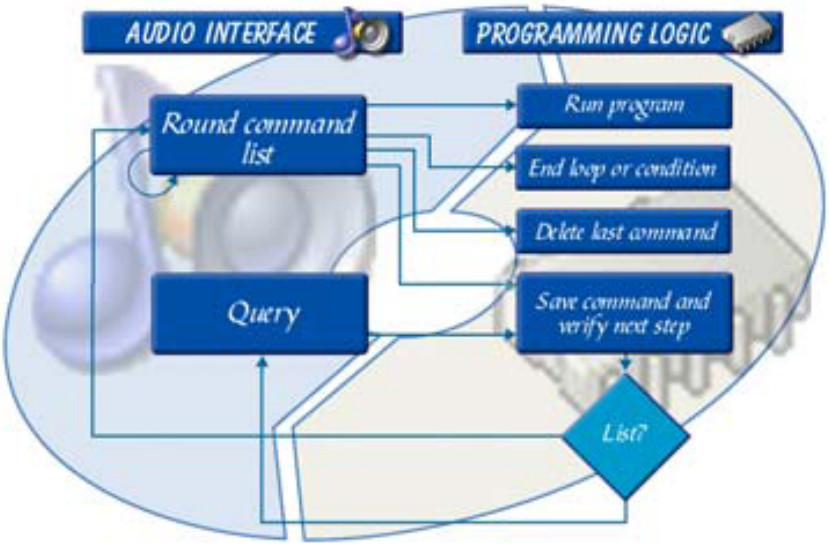
Svi ovi nedostaci su naročito vezani za GUI programske razvojne okoline i objektno orijentirane programske koncepte što predstavlja veliki izazov za tehnologije pomoći osobama oštećena vida u programiranju.

3.2. Programski alati za pomoć slijepima u edukaciji

Razvojem modernih GUI programskih razvojnih okolina i grafičkih sučelja javlja se i problem u edukaciji osoba oštećena vida, koje moraju shvatiti nove koncepte programiranja na jednostavan i razumljiv način, kao jedno od aktualnih pitanja (Riccobono, 2004). Problemi

i potrebna rješenja u edukaciji osoba oštećena vida su identični onima profesionalnih programera oštećena vida i svakodnevnih korisnika računala.

Da bi se osobama oštećena vida predočio algoritamski način razmišljanja i da bi ih se naučilo kako rješavati probleme, nastao je koncept audio programskog jezika APL (Audio Programming Language) (Sanchez & Aguayo, 2005) zajedno s APL alatom koji je prikazan na slici 3.5. APL je osmišljen da premosti razliku između studenata oštećena vida i onih koji vide u smislu približavanja i predočavanja koncepata programiranja osobama oštećena vida na način na koji to klasični tekst-u-govor konverteri ne mogu. Osobe oštećena vida ne mogu pratiti programsku logiku na isti način kao oni koji vide tj. na način da jednostavno prate govornu interpretaciju pojedinih programskih redaka. Primjerice, varijable ne mogu biti inicijalizirane prije nego li su deklarirane. Ovo i slična pravila formiraju strukturu programskog jezika koju osobe oštećena vida s poteškoćama shvaćaju i upravo je ovdje glavna svrha APL alata. APL zapravo ne predstavlja novi programski jezik već sredstvo motivacije studenata oštećena vida za programiranje na način da im se omogući lakše predočavanje određenih programskih koncepata. APL koristi audio kao ulazni i izlazni medij umjesto tipkovnice. S obzirom da cilj APL alata nije učenje specifične programske sintakse, nego učenje programskih koncepata u alatu, nema nikakvog pisanja koda, već se radi jednostavan odabir željenih elemenata poput varijabli i funkcija iz ponuđenih listi elemenata.



Slika 3.5 APL alat (Sanchez & Aguayo, 2005)

APL se sastoji od dvije glavne razine: Audio sučelje (Audio Interface) i programske logike (Programming Logic). Audio sučelje uključuje cirkularnu listu naredbi koja daje popis naredbi koje korisnik može odabrati, poput petlje, uvjeta, upisa/ispisa i varijabli te dio upita koji se koristi za definiranje naziva i vrijednosti varijabli a također za unos/ispis u audio formatu. Programska logika se sastoji od četiri funkcije:

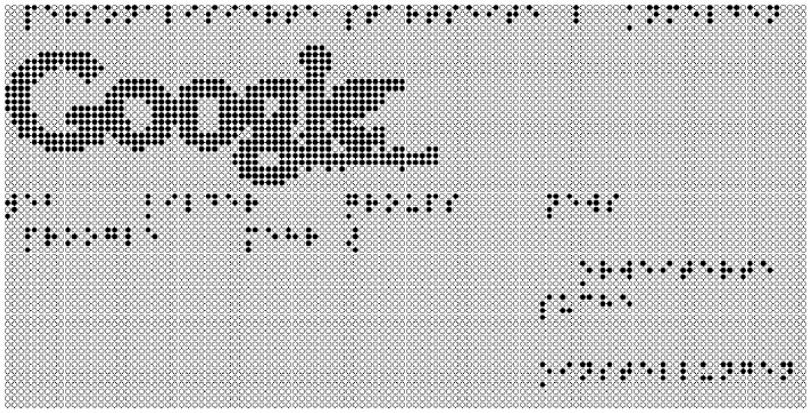
- Pokretanje programa (Run program)
- Završetak petlje ili uvjeta (End loop or condition)
- Brisanje zadnje naredbe (Delete last command)
- Spremanje naredbi i verificiranje sljedećeg koraka (Save command and verify next step)

Još jedan pristup boljem razumijevanju programiranja za osobe oštećena vida je nastojanje da se programska sintaksa i semantika opišu na detaljan, ali razumljiv način korištenjem sintetizatora govora. Ovaj opis uključuje semantički opis određene linije koda gdje je to potrebno i detaljan prijevod sintakse koda u oblik koji je što bliže standardnom govornom jeziku. Primjer ovakvog rješenja je JavaSpeak (Smith et al., 2000) program razvijen za učenje Java programskog jezika.

4. AKTUALNA NASTOJANJA I PROBLEMI U PRIKAZU GRAFIČKIH SUČELJA

Jedan od problema koji se javlja među osobama oštećena vida je razumijevanje web sučelja. World Wide Web je napravio veliku revoluciju u načinu prezentacije sadržaja krajnjem korisniku. Nažalost, kako su web sučelja postajala sve kompleksnija i kako su se nove tehnologije pojavljivale, osobama oštećena vida je postajalo sve teže percipirati kontekst cijele web stranice, a parsiranje takve stranice u svrhu ekstrakcije važnih informacija je postao novi izazov. Problem je u tome da iako postoji mnogo standarda i smjernica koje sugeriraju adekvatnu pristupačnost prilikom razvoja web stranica, većina web stranica je i dalje slabo pristupačna (Lazar et al., 2004). Istraživanje 50 najpopularnijih web stranica je rezultiralo postotkom od preko 50% djelomično pristupačnih ili nepristupačnih web stranica (Sullivan & Matson, 2000). Razlog ovakvog rezultata najvjerojatnije leži u tome da mnogi web programeri ne znaju mnogo o pristupačnosti dizajna ili misle da taj dio nije toliko bitan. Programski kod ispod web sučelja može također biti nepristupačan. Primjerice, određene radnje mogu biti pokrenute akcijama miša bez adekvatne alternative putem tipkovnice.

Jedan od mogućih pristupa interpretaciji ovakvih sučelja je razvoj taktilnih ploča za taktilno web pretraživanje (Rotard et al., 2008). Taktilna ploča je prikazana na slici 4.1. Taktilno web pretraživanje predstavlja korak naprijed u grafičkoj percepciji u usporedbi s jednostavnom audio interpretacijom web sadržaja. Razlika između taktilnih zaslona i Braillovih linija je u tome što taktilni zasloni imaju iglice koje nisu grupirane. Jednostavno rečeno taktilne ploče ili zasloni su višelinijski Braillovi zasloni. Informacije s web stranica se prikupljaju, izdvajaju i prikazuju putem taktilnog zaslona. Podizanjem određenih iglica moguće je prikazati kompleksne grafičke slike koje korisnici i programeri oštećena vida mogu osjetiti.



Slika 4.1 Taktilna ploča (Rotard et al., 2008)

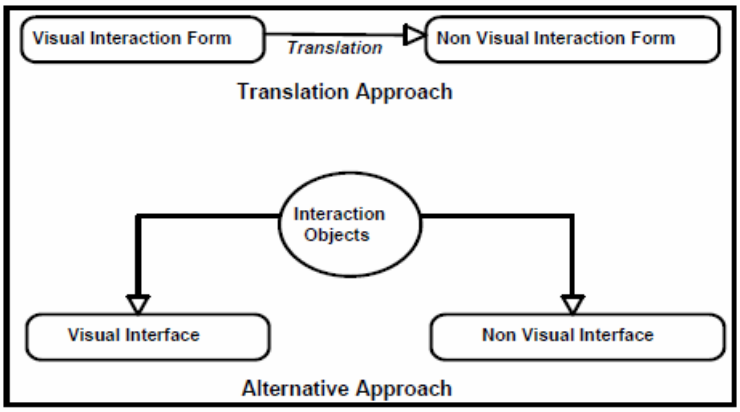
Drugi pokušaj rješavanja problema interpretacije i mogućnosti grafičkog dizajniranja je pokušaj oponašanja procesa crtanja na način na koji bi to učinila osoba koja vidi. Osobe koje vide, konstantno prilagođavaju poziciju olovke bazirano na vezi između do tad nacrtanog i modela koji se crta iz stvarnog života (prilagođavanje je bazirano na povratnoj vezi). Na ovaj način osoba koja vidi prilagođava linije, krivulje, oblike i pripadajuće kutove da bi što vjernije prikazala neki objekt. Da bi se ovaj proces mogao oponašati potrebno je osigurati osobama oštećena vida mogućnost lociranja i relociranja određene točke u crtežu, ali na način da mogu također odrediti i odnos određenih točaka jednih prema drugima ili u odnosu na sliku u cjelini. Da bi se ovo postiglo, predloženo je rješenje u kojem se crtača površina dijeli na 9 jednakih pravokutnih područja s točno prepoznatljivim centrom (Kamel & Landay, 2000). Na ovaj način se postiže bolja orijentacija i razumijevanje prostora. Pojedini pravokutnici mogu na isti način biti dalje podijeljeni u svrhu postizanja veće preciznosti.

Još jedan pristup ovom problemu je pokušaj prezentiranja grafičkih informacija putem glazbe. U okviru ovog rješenja razvijen je alat Audiograph (Alty & Rigas, 1998) da bi se pokušalo prikazati jednostavne grafove i oblike osobama oštećena vida. Audiograph sustav prezentira korisniku sljedeće informacije:

- trenutnu poziciju pokazivača ili poziciju grafičkog objekta,
- prirodu grafičkog objekta (tip, veličinu i oblik),
- globalnu poziciju grafičkih objekata korištenjem raznih tehnika skeniranja.

Da bi se prikazalo grafičke oblike, koriste se sljedeći koncepti. X i Y koordinate su predstavljene različitom bojom zvuka. Koordinata X je predstavljena orguljama, a koordinata Y klavirom. Pozicija točke je opisana glazbenim mapiranjem u kojem viša frekvencija zvuka predstavlja veću X odnosno Y koordinatu.

Pristup koji svojevrsno objedinjava navedene napore nosi naziv multimodalni pristup (Bellik & Burger, 1994). Multimodalni pristup koristi više načina interakcije da bi stvorio korisnički prihvatljivo okruženje koje osobe oštećena vida mogu koristiti. Drugim riječima, ovaj pristup koristi kombinaciju govora, taktilnih ploča, Braillovog pisma te drugih medija i koncepata da bi stvorio sinergijski efekt i pomogao osobama oštećena vida u percipiranju grafičkih informacija na što je moguće prirodniji način. Multimodalni koncept je prikazan na slici 4.2.



Slika 4.2 Multimodalni pristup (Bellik & Burger, 1994)

Svi navedeni pristupi nude smislena i eksperimentalno dobra rješenja, no ona imaju ograničenu praktičnu uporabu u širem smislu, ali daju dobre temelje za buduća istraživanja. Opisana rješenja isto tako ne nude opće i uporabljivo rješenje problema razvoja grafičkih sučelja s kojim se susreću programeri oštećena vida.

4.1. Web i stolno programiranje

Promatrajući dva glavna područja programiranja: web i klasično stolno programiranje, može se zaključiti da problem dizajniranja i programiranja još uvijek nije toliko izražen u području web programiranja. Iako postoje neki grafički uređivači za razvoj web sučelja i web stranica, većina koda i dizajna se još uvijek radi tekstualnim putem. Postoje neka otvorena pitanja

poput odabira i prepoznavanja boje i pitanja vezana za određene elemente pristupačnosti (Spencer, 2001), no to su problemi na nižoj razini. Generalno se može reći da osobe oštećena vida još uvijek mogu razvijati grafička sučelja u području web programiranja uz postojeću programsku podršku za pomoć osobama oštećena vida. Također, postoji mnogo različitih tehnologija, baziranih većinom na XML (EXtensible Markup Language) jeziku za opis podataka i drugih elemenata, koje bi mogle koristiti osobama oštećena vida u vizualizaciji sadržaja na način da se određeni prikazi elemenata realiziraju tekstualnim opisima. Primjerice, jezici bazirani na XML-u, omogućavaju opis objekata i njihovih karakteristika i u tom smislu mogu pružiti osobama oštećena vida određeni način na koji mogu opisati i razumjeti određene grafičke elemente. Pokušaj da se grafički elementi i njihovo ponašanje definiraju na ovakav način su jezici UIML (User Interface Markup Language) (Abrams et al., 1999) i UsiXML (User Interface eXtensible Markup Language) (Pascual et al., 2005).

U području klasičnog stolnog programiranja situacija je dosta drugačija. Kompleksni programski koncepti i sve razvijenije programske razvojne okoline bazirane na grafičkim uređivačima čine klasično stolno programiranje i dizajniranje grafičkih formi korištenjem miša i pokazivača praktički nemogućim za osobe oštećena vida. Terminološki gledano, termin GUI označava forme u stolnom programiranju dok WWWUI (World Wide Web User Interface) označava grafički prikaz i raspored elemenata web stranica (Galitz, 2007). Da bi se specificiralo općeprihvatljivo i uporabljivo rješenje u područjima u kojima programeri oštećena vida imaju najviše poteškoća, potrebno je fokusirati napore u prvom redu na rješavanje pitanja GUI razvoja.

4.2. Mogući pristupi problemu

Da bi se riješili spomenuti problemi i pitanja, postoji potreba za pronalaskom novog pristupa koji bi pružio čvrsto i adekvatno rješenje. Iako spomenuti pristupi rješavanju problema prikaza grafičkih sučelja nude obećavajuće rezultate u smislu budućih istraživanja, nisu dostatni da bi riješili aktualna pitanja s kojima su suočeni programeri oštećena vida. Razmišljajući o mogućim pristupima rješavanju navedenih problema, može biti definirano nekoliko mogućnosti. Mogli bi biti izvedeni interpreteri koji bi interpretirali forme i njihove elemente zajedno s pripadajućim atributima. Ovi bi interpreteri morali podržavati Windows kontrole za svaki pojedini programski jezik i razvojnu okolinu, ali bi morali imati i

mogućnost podrške za ostale kontrole. Navedeni proces bi se mogao i obrnuti. Programske razvojne okoline bi mogle podržati mogućnost stvaranja određenih grafičkih elemenata putem audio sučelja. Mogli bi se razviti i posebni skriptni jezici za pojedine programske razvojne okoline i njihove inačice, koji bi omogućili tekstualni opis GUI elemenata unutar pojedinog programskog jezika (Konecki, et al., 2010; Konecki, et al., 2011). Ovakav skriptni jezik je razvijen u svrhu stvaranja formi podržanih Visual Basic programskom razvojnom okolinom (Siegfried, 2006).

Konačno, da bi se došlo do općeg rješenja, predlaže se opisni jezik GUIDL (Graphical User Interface Description Language) koji bi bio razvijen na način da omogući jednostavan opis korisničkih sučelja i njihovih elemenata putem jezika koji je što bliži govornom jeziku, kako bi bio prikladan kako za programere tako i za dizajnere. Opisni jezik bi bio centralni dio sustava koji bi omogućio prevođenje opisanih sučelja u konkretan GUI pojedinog programskog jezika korištenjem pripadajućeg GUI generatora. Na ovaj način sustav bi ponudio opće rješenje koje je jednostavno, adekvatno i proširivo za nove tehnologije (Konecki, 2012). Važno je napomenuti da je GUIDL zamišljen kao svojevrsna pomoćna tehnologija osobama oštećena vida koja omogućava izradu grafičkih sučelja u odabranoj programskoj tehnologiji. Programer oštećena vida započinje razvoj programa kreiranjem praznog projekta u kojem prije izrade bilo kakvog grafičkog sučelja može napisati kod po želji, primjerice skup potrebnih funkcija i procedura. Nakon toga, programer oštećena vida umjesto korištenja grafičkih kontrola i kreiranja grafičkog sučelja uporabom miša, koristi GUIDL sustav i u GUIDL jeziku koristeći jednostavni tekstualni opis, s kojim se programeri oštećena vida susreću od samog početka korištenja računala, i ugrađene pomoćne koncepte opisuje željeno sučelje i prevodi ga u željenu programsku tehnologiju te se nakon toga vraća u programsku razvojnu okolinu gdje uključuje datoteke kreirane putem GUIDL sustava u postojeći projekt te na taj način dobiva gotovo grafičko sučelje.

Nakon uključivanja grafičkog sučelja programer oštećena vida nastavlja pisanje koda za pojedine grafičke kontrole koje se nalaze na sučelju. Ovaj dio je programer oštećena vida bez poteškoća u mogućnosti izraditi u programskoj razvojnoj okolini uz uporabu sintetizatora govora s obzirom da se radi o pisanju koda tj. tekstualnom režimu rada koji je oduvijek bio podržan postojećim pomoćnim tehnologijama za rad na računalu. Do koda za pojedinu grafičku kontrolu programer oštećena vida može također doći bez poteškoća s obzirom da su sintetizatori govora u mogućnosti pročitati atribut naziva pojedine grafičke kontrole kada se

programer oštećena vida koristeći tipkovnicu pozicionira na odgovarajuću grafičku kontrolu. Dakle, programeri oštećena vida razvijaju programsko rješenje u odabranoj programskoj razvojnoj okolini a GUIDL sustav im služi kao pomoćna tehnologija za razvoj grafičkih sučelja koja su pojavom grafičkih programskih razvojnih okolina postala praktički nemoguća za opisati tekstualnim putem u okviru same programske razvojne okoline (Franqueiro & Siegfried, 2006)

4.3. Metodologija razvoja

Prilikom razvoja navedenog sustava i njegovih dijelova korišteni su instrumenti koji su prepoznati kao prikladni i najčešće korišteni u ove svrhe. EBNF (Extended Backus–Naur Form) (Information technology – Syntactic metalanguage – Extended BNF. ISO/IEC 14977, 1996) je formalna notacija koja se koristi za definiranje sintakse putem niza pravila a nastala je proširenjem BNF notacije (Gabbrielli & Martini, 2010). Pomoću EBNF notacije moguće je formalno definirati ne samo sintaksu programskog jezika, već i niz drugih formalnih definicija poput formata za citiranje, formata fusnota, itd. EBNF se koristi prilikom razvoja i prikaza gramatike deskriptivnog jezika GUIDL sustava. Sintaktički dijagrami se koriste prilikom detaljnog i zornog prikaza kontekstno neovisne gramatike kakva je i razvijena gramatika GUIDL jezika. Osim sintaktičkih dijagrama (Braz, 1990) koriste se i UML dijagramske tehnike modeliranja (Fowler & Scott, 2003). Od UML dijagrama koriste se UML dijagram aktivnosti (Activity diagram) i dijagram slučaja uporabe (Use case diagram) za prikaz slijeda aktivnosti koje je potrebno učiniti da bi korisnik došao do konačnog rezultata u razvijenom sustavu tj. do konačnog i konkretnog koda grafičkog sučelja vezanog za neku od podržanih aktualnih tehnologija.

5. GUIDL SUSTAV

Da bi se došlo do općeg rješenja, razvijen je novi deskriptivni (opisni) jezik GUIDL (Graphical User Interface Description Language) na način da omogući osobama oštećena vida jednostavan i adekvatan način opisa grafičkih sučelja i njihovih elemenata putem jezika koji je što bliži govornom jeziku kako bi bio prikladan kako za programere tako i za dizajnere i ostale informatičare oštećena vida. GUIDL jezik predstavlja centralni dio sustava koji omogućava prevođenje opisanih sučelja u konkretan GUI kod pojedinog programskog jezika korištenjem pripadajućeg medijatora tj. GUI generatora. Na ovaj način sustav nudi adekvatno rješenje koje je također proširivo za nove tehnologije.

5.1. Zahtjevi za GUIDL sustav

Prvi korak u razvoju novog GUIDL sustava je bio definiranje zahtjeva koje takav sustav mora zadovoljiti. Temeljem analize literature i promišljanja o problemu moguće je definirati nekoliko karakteristika koje bi ovaj sustav morao zadovoljavati kako bi bio primjenjiv i održiv.

Spomenute karakteristike su (Konecki, 2012):

- Lakoća uporabe: sustav bi trebao biti lak za uporabu kako programerima tako i ostalim zainteresiranim osobama iz informatičke struke, primjerice dizajnerima oštećena vida.
- Intuitivna, jednostavna i lako razumljiva sintaksa: deskriptivni jezik koji će se koristiti za opis grafičkih korisničkih sučelja mora biti intuitivan i lako primjenjiv svima koji se bave programiranjem u području klasičnih stolnih programa, ali i ostalim programerima oštećena vida.
- Neovisnost deskriptivnog jezika: deskriptivni jezik mora biti općenit, a ne specifičan ili usko vezan samo za jednu programsku razvojnu okolinu tj. jezik.
- Proširivost deskriptivnog jezika: deskriptivni jezik mora biti proširiv na način da omogućava uključivanje podrške za nove programske razvojne okoline i jezike, kao i za nove grafičke elemente (kontrole).

Da bi se ostvario kontakt s programerima oštećena vida i ocijenila njihova zainteresiranost za opisani sustav te provjerile navedene potrebne karakteristike sustava, provedeno je odgovarajuće istraživanje. Istraživanje je provedeno temeljem kreiranog anketnog upitnika koji je sadržavao izjave vezane za navedene karakteristike, ali također i za niz drugih aspekata i elemenata koji su uočeni kao potencijalno bitni prilikom kreiranja samog upitnika. U okviru upitnika ispitanici su imali mogućnost samostalnog upisivanja komentara i dodatnih aspekata koji nisu spomenuti, a iste smatraju bitnima.

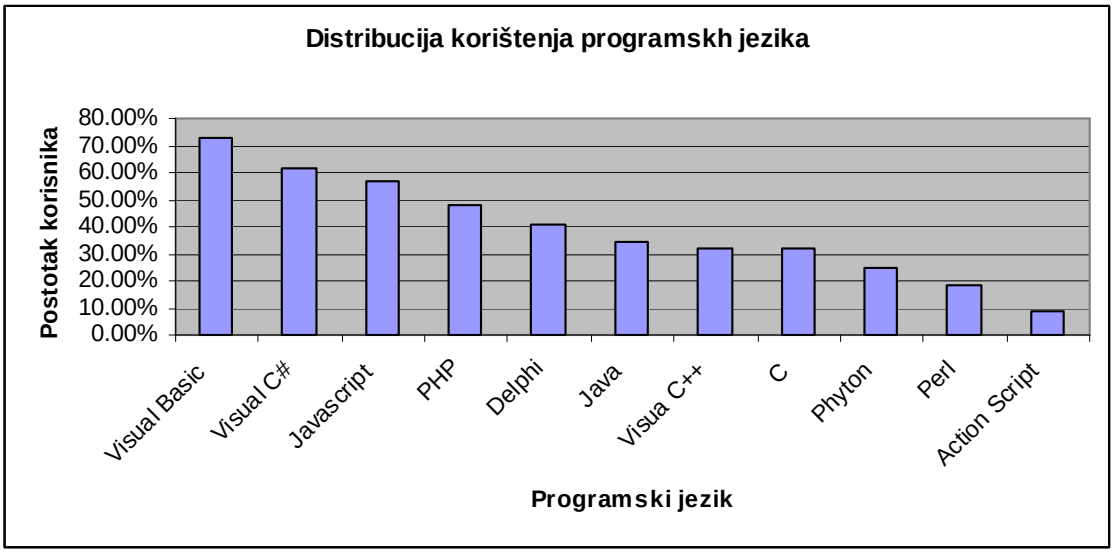
Da bi se istraživanje provelo na odgovarajući način, bilo je potrebno pronaći adekvatna mjesta na kojima je moguće doći do željene populacije. Istraživanjem su pronađene grupe koje predstavljaju prepoznata mjesta na kojima se okupljaju programeri oštećena vida. U istraživanje su uključene kako poznatije grupe (program-l: V.I. Programmers Discussion List, 2011; programmingblind: Programming Blind, 2011; Blind Geeks, 2011; JAWSScripts, 2011), tako i one nešto manje poznate zbog uključivanja što većeg broja ispitanika u istraživanje. Navedena populacija je predstavljala prigodni uzorak kojem je upućena molba za sudjelovanje u istraživanju. U trenutku analize rezultata istraživanje je obuhvatilo 44 dobivena odgovora. Ono što je također bilo značajno za istraživanje i cjelokupni rad su osobni komentari potpore i interesa koji su zaprimljeni od ispitanika.

Ispitanicima je dana mogućnost da se prijave osobnim e-mailom kao zainteresirani za testiranje prototipa sustava te je najavljeno da će funkcionalni prototip biti dan na raspolaganje i testiranje te da će se u kasnijoj fazi vršiti dodatno istraživanje putem kojeg će se ispitanici moći izjasniti u kojoj mjeri je razvijeni prototip ispunio očekivanja i doprinjeo rješavanju problema oblikovanja grafičkih sučelja osobama oštećena vida. Rezultati istraživanja su prikazani u nastavku. Gotovo su svi odgovori imali mogućnost odabira vrijednosti na Likertovoj skali od 1 do 5 (1 – u potpunosti se ne slažem, 5 – u potpunosti se slažem) radi što realnije distribucije vrijednosti odgovora. Pitanja i rezultati anketnog upitnika su prikazani u tablici 5.1.

Tablica 5.1 Rezultati istraživanja provedenog na programerima oštećena vida

Izjava	**Prosječna vrijednost odgovora**	**Standardna devijacija**
Koliko imate godina?	28,68	6,11
Koliko godina programirate?	7,36	4,50
Stvaranje grafičkih korisničkih sučelja je jedan od najvećih izazova za mene kao programera.	4,52	0,79
Problem stvaranja grafičkih korisničkih sučelja je izraženiji u području stolnog nego u području web programiranja.	4,02	0,95
Važno mi je da imam rješenje koje omogućava stvaranje sučelja koja se mogu uključiti u aktualna programska okruženja (Visual Basic, Visual C#, Visual C++, Delphi, itd.).	4,61	0,69
Važno mi je da mogu opisati grafička sučelja na što jednostavniji način, uporabom što manjeg broja naredbi i parametara radi što lakšeg korištenja.	4,43	0,73
Želio/la bih novi programski jezik i okolinu radije nego mogućnost uključivanja mojih sučelja u postojeća programska okruženja.	1,52	0,73
Nije mi važno koliko je složeno opisati grafička sučelja, ukoliko je to moguće.	1,68	0,80
Želio/la bih da sustav i opisni jezik budu nezavisni od konkretne programske okoline i da se mogu prevesti u oblik pogodan za programske okoline aktualnih programskih jezika.	4,57	0,62
Želio/la bih da sustav i jezik za opis grafičkih sučelja bude nadogradiv i da ima mogućnost dodavanja podrške za nove jezike i okoline prema potrebi.	4,11	0,81
U mogućnosti sam bez poteškoća izraditi grafičko sučelje u aktualnim programskim okolinama kao što su Visual Basic, Visual C#, Visual C++, Delphi, itd.	1,82	0,76

Postojeće pomoćne tehnologije i alati su u potpunosti dovoljni da pomoću njih izradim grafička sučelja na meni adekvatan i prihvatljiv način.	1,95	0,78
S kojim programskim jezicima ste se do sada susretali u svom radu?	Distribucija odgovora je prikazana na slici 5.1.	



Slika 5.1 Distribucija korištenja programskih jezika

Osim navedenih izjava, upitnik je sadržavao pitanje o spolu, pitanje o iskustvu u programiranju tj. je li ispitanik imao više iskustva u web ili stolnom programiranju ili je njegovo iskustvo podjednako u oba područja, te mogućnost osobnog komentara putem kojeg je bilo moguće također navesti još neki aspekt koji ispitanik smatra bitnim. Iz provedenog istraživanja se može zaključiti da programeri oštećena vida imaju poteškoća s izradom grafičkih sučelja i da im postojeće pomoćne tehnologije ne pružaju adekvatno rješenje te da samim time imaju potrebu za razvojem novog rješenja za ovaj problem. Programerima oštećena vida je također važno da budu uključeni u aktualne projekte i programske tehnologije te da pomoćna tehnologija bude neovisna o pojedinom programskom jeziku i da ima mogućnost proširenja novim programskim jezicima.

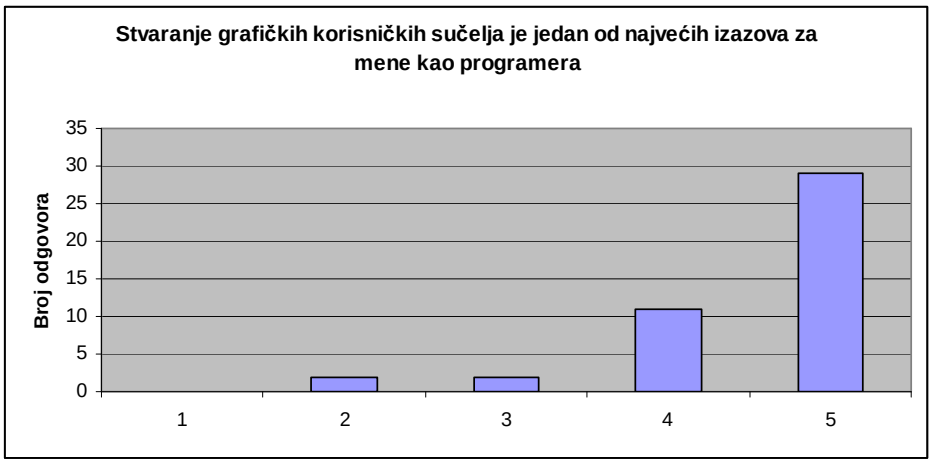
Od ukupnog broja ispitanika koji su odgovorili na upitnik, bilo je 33 muška i 11 ženskih ispitanika što pokazuje da oba spola pokazuju interes za isto područje rada. 23 ispitanika se izjasnilo da se njihovo iskustvo većinom veže za stolno programiranje, 9 ispitanika se

izjasnilo za područje web programiranja, a 12 ispitanika se izjasnilo da imaju podjednako iskustvo u oba područja programiranja. Iz komentara ispitanika se uz sve navedeno moglo zaključiti da smatraju bitnim u prvom redu jednostavnost sintakse i sučelja te da smatraju jako bitnim dobru dokumentaciju. Svi komentari i rezultati su uzeti u obzir kao zahtjevi prilikom razvoja GUIDL sustava.

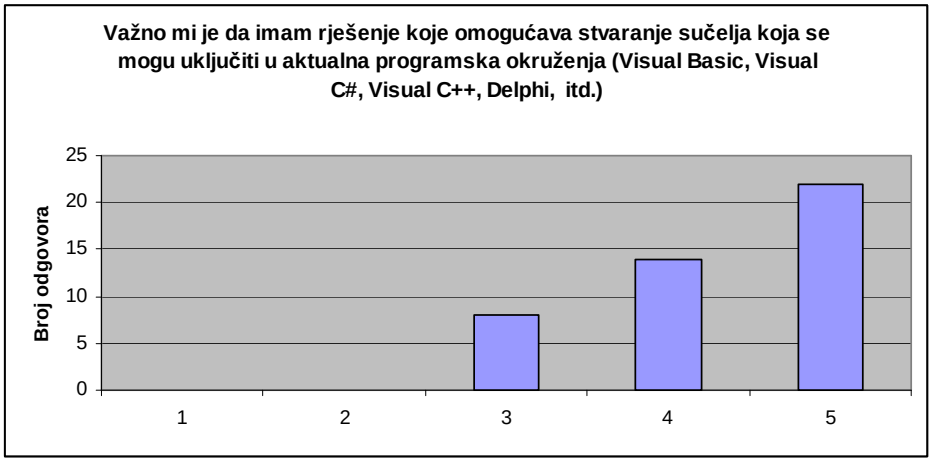
U nastavku su dani neki komentari ispitanika i distribucija odgovora za neke izjave:

- Komentar 1: Best of luck. Maybe good documentation combined with very simple syntax would be desirable.
- Komentar 2: Easy to use interface.
- Komentar 3: This is an interesting concept, and one I've been thinking about for years. Thank you; your project is an interesting one, and it would be great to see new systems become available.
- Komentar 4: Good idea, my opinion is that blind programmers should be included in the mainstream technologies, not to have special languages, so go ahead. Good luck!!
- Komentar 5: Proper documentation.

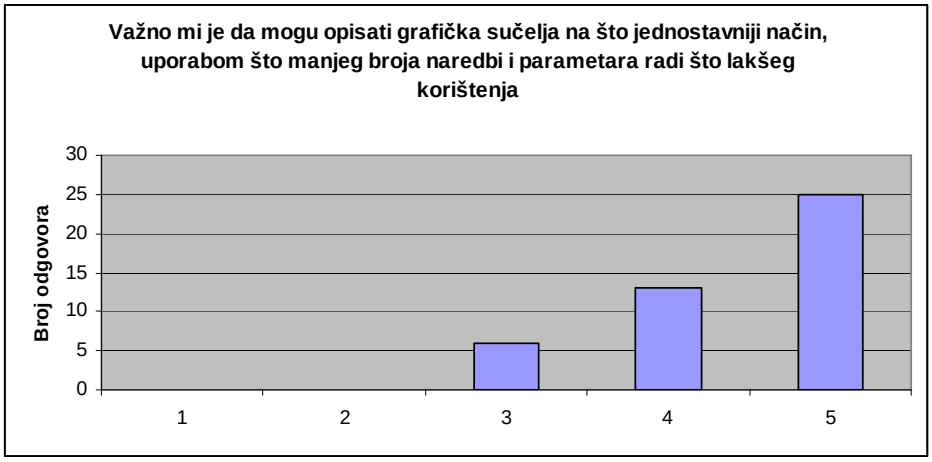
S obzirom na rezultate provedenog istraživanja, proširena je lista zahtjeva za karakteristikama sustava GUIDL. Kao što je već spomenuto, jedan od bitnih aspekata koje su programeri oštećena vida naveli kao važan je dobra i prilagođena dokumentacija u kombinaciji sa što jednostavnijim sučeljem. Na slikama 5.2, 5.3 i 5.4 je dana struktura odgovora na neka od temeljnih pitanja u istraživanju.



Slika 5.2 Struktura odgovora na pitanje anketnog upitnika



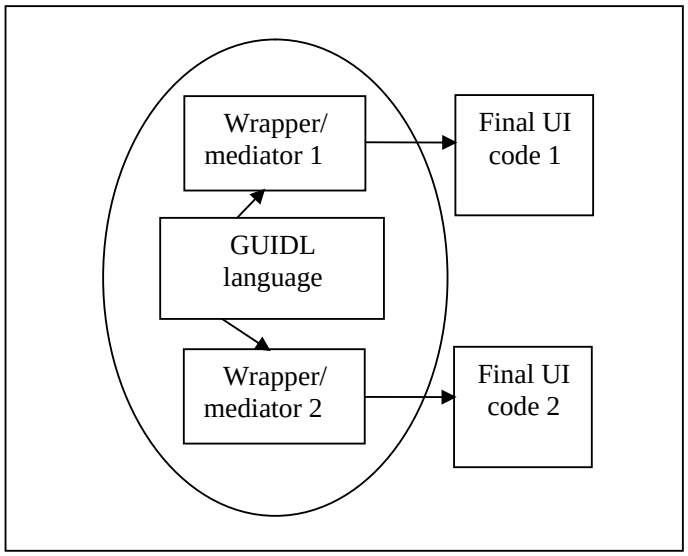
Slika 5.3 Struktura odgovora na pitanje anketnog upitnika



Slika 5.4 Struktura odgovora na pitanje anketnog upitnika

5.2. Model GUIDL sustava i njegov razvoj

Nakon definiranja zahtjeva, definiran je model i osnovni parametri GUIDL sustava te sam proces njegova razvoja. Na slici 5.5 prikazan je idejni model funkcioniranja GUIDL sustava.



Slika 5.5 Idejni model GUIDL sustava

GUIDL sustav se sastoji od nekoliko elemenata. Glavni dio sustava je GUIDL jezik. GUIDL jezik služi za opis grafičkog sučelja, a izvorni kod u GUIDL jeziku se pohranjuje za daljnju obradu. GUIDL opis sučelja se nakon pohrane obrađuje jednim od pripadajućih medijatora. Svaki pojedini medijator služi kao prevoditelj GUIDL koda u kod koji je specifičan za pojedini programski jezik tj. pojedino programsko razvojno okruženje. Na ovaj način postiže se proširivost sustava.

Prvi korak u procesu razvoja GUIDL jezika je bio definirati parametre jezika u smislu opsega i sintakse te nakon toga definirati sam jezik. U prototipu jezika napravljene su restrikcije u smislu podržanih grafičkih kontrola. Odabrane su kontrole koje su najčešće u uporabi prilikom razvoja najvećeg broja programa.

Odabrane kontrole su:

- Form (Forma/Ekran)
- Button (Gumb)
- Label (Natpis)
- TextBox (Tekstualno polje)
- Checkbox (Polje za odabir)
- ComboBox (Padajući popis)
- PictureBox (Okvir sa slikom)
- RadioButton (Radio gumb)
- MenuStrip (Padajući izbornik)

Na isti je način napravljen odgovarajući odabir atributa pojedinih kontrola. U tablici 5.2 slijedi popis atributa za svaku odabranu kontrolu.

Tablica 5.2 Odabrane grafičke kontrole i njihovi podržani atributi

Naziv grafičke kontrole	Podržani atributi	Opis
Form	*Name*	Naziv grafičke kontrole
	BackColor	Boja pozadine kontrole
	ForeColor	Boja fonta kontrole
	Text	Naslov kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
	WindowState	Prikaz kontrole u jednom od mogućih formata: Normal, Minimized, Maximized
Button	*Name*	Naziv grafičke kontrole
	BackColor	Boja pozadine kontrole
	ForeColor	Boja fonta kontrole
	Text	Naslov kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
Label	*Name*	Naziv grafičke kontrole
	BackColor	Boja pozadine kontrole

	ForeColor	Boja fonta kontrole
	Text	Naslov kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
TextBox	*Name*	Naziv grafičke kontrole
	BackColor	Boja pozadine kontrole
	ForeColor	Boja fonta kontrole
	Text	Tekst kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
	PasswordChar	Oznaka za sakrivanje lozinke u slučaju da se u kontrolu unosi lozinka
CheckBox	*Name*	Naziv grafičke kontrole
	BackColor	Boja pozadine kontrole
	ForeColor	Boja fonta kontrole
	Text	Naslov kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
	Value (Checked)	Vrijednost kontrole: 0-Unchecked, 1-Checked, 2-Grayed
ComboBox	*Name*	Naziv grafičke kontrole
	BackColor	Boja pozadine kontrole
	ForeColor	Boja fonta kontrole
	Text	Tekst kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
	List (Items)	Popis elemenata kontrole
PictureBox	*Name*	Naziv grafičke kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
	Picture (Image)	Slika smještena u kontrolu
RadioButton	*Name*	Naziv grafičke kontrole
	BackColor	Boja pozadine kontrole
	ForeColor	Boja fonta kontrole

	Text	Naslov kontrole
	Width	Širina kontrole
	Height	Visina kontrole
	Top	Udaljenost kontrole od vrha ekrana
	Left	Udaljenost kontrole od lijevog ruba ekrana
	Value (Checked)	Vrijednost kontrole: True, False
MenuStrip	*Name*	Naziv grafičke kontrole
	Items	Popis elemenata izbornika u dvije razine

Sintaksa GUIDL jezika je odabrana tako da bude jednostavna i da bude poznatog oblika tj. da podsjeća na govorni jezik. Jezik koji zadovoljava navedene kriterije i dobro je poznat je upravo BASIC (Beginner's All-purpose Symbolic Instruction Code). GUIDL jezik je stoga, inspiriran BASIC-om, iako nema direktne veze s njim i BASIC interpreter/kompajler ne može razumjeti GUIDL kod, što osigurava njegovu lakoću uporabe i učenja. S obzirom na jednostavnu sintaksu blisku engleskom jeziku, GUIDL jezik će biti razumljiviji kako programerima tako i ostalim informatičarima, primjerice dizajnerima. Dakle, razlog odabira tekstualnog načina opisa sučelja koji podsjeća na govorni jezik je upravo identificirana potreba i želja programera oštećena vida za jednostavnim rješenjem i sintaksom koju će s lakoćom koristiti i oni manje i oni više iskusni programeri ali također i osobe oštećena vida koje su primarno dizajneri kao i oni koji se žele iz edukacijskih razloga okušati u kreiranju grafičkih sučelja.

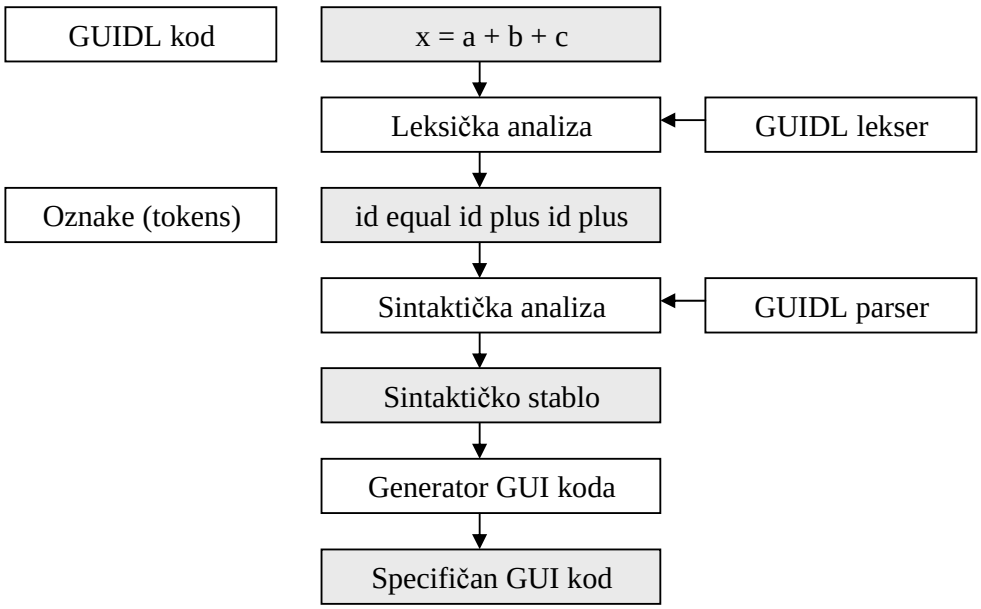
Da bi se definirao GUIDL jezik, potrebno je definirati gramatiku jezika. Gramatika jezika predstavlja skup pravila koja tvore riječi formalnog tj. programskog jezika koji se u teoriji naziva još i umjetni jezik (Kiš, 2000). Za tu svrhu koristi se već spomenuta EBNF notacija koja omogućava formalni opis sintakse programskih jezika. EBNF je notacija kojom se mogu prikazati kontekstno neovisne gramatike. Kontekstno neovisne gramatike (gramatika tipa 2 prema hijerarhiji Noama Chomskog (Searle, 1972; Chomsky, 1961; Chomsky, 1965; Chomsky, 1966; Chomsky, 2002)) su gramatike koje su u stanju prikazati gramatiku većine programskih jezika. Kontekstno neovisne gramatike generiraju kontekstno neovisne jezike, a to su svi oni koji mogu biti prepoznati putem pushdown automata (Lewis & Papadimitriou, 1981).

EBNF notacija opisuje gramatiku kojom se definira pripadajući kontekstno neovisan jezik, u ovom slučaju GUIDL. Gramatika se može definirati kao $G = (V, \Sigma, R, S)$, gdje je V abeceda, Σ skup završnih simbola, R je skup produkcijskih pravila i S je početni simbol. Također

možemo reći da vrijedi $\Sigma \subset V$. Jezik nad zadanom gramatikom G se označava kao $L(G) = \{w \in \Sigma^* : S \rightarrow_G^* w\}$.

Izvorni kod GUIDL jezika se prosljeđuje GUIDL lekseru koji vrši leksičku analizu koda. Sastavni dio leksera je skener. Skener čita datoteku s izvornim kodom, znak po znak i prosljeđuje lekseru. Lekser prima znakove i kada primi znakovni niz koji odgovara nekom od definiranih uzoraka znakova, prepoznaje i zapisuje određenu oznaku. Set mogućih oznaka je definiran unutar GUIDL parsera odakle je uključen u lekser. Rezultat leksičke analize je niz oznaka koje GUIDL parser čita prilikom sintaktičke analize. Sintaktička analiza se vrši temeljen niza pravila, koja su definirana putem gramatike jezika, uporabom dobivenih oznaka. Parser provjerava ispravnost unesenog koda te generira odgovarajuće sintaktičko stablo koje služi kao podloga generatoru GUI koda koji temeljem sintaktičkog stabla vrši pretvorbu izvornog koda u specifičan GUI kod konkretnog programskog jezika i razvojnog okruženja.

Rad GUIDL sustava je prikazan po fazama na slici 5.6. (Konecki, 2012)



Slika 5.6 Struktura i rad GUIDL sustava

6. GRAMATIKA PROTOTIPA GUIDL SUSTAVA

Da bi se izradio prototip cjelokupnog GUIDL sustava nužno je krenuti od njegovog centralnog dijela tj. samog GUIDL jezika. S obzirom na već definirane zahtjeve za GUIDL sustav, može se zaključiti da i centralni GUIDL jezik mora biti lak i intuitivan zbog potrebe da se osobama oštećena vida pruži metoda pomoću koje će na njima prihvatljiv način moći oblikovati sučelja koja će moći podijeliti kao svoje vlastite dizajnerske ideje s ostalim članovima tima kao ravnopravni sudionici. Smisao prototipa je da prikaže smjer i mogućnosti cjelokupnog sustava i da dokaže smislenost ovakvog pristupa u uključivanju osoba oštećena vida u cjelokupni proces razvoja informacijskih sustava.

Da bi se ostvarili navedeni ciljevi GUIDL jezika i sustava, potrebno je definirati gramatiku GUIDL jezika na način da zadovoljava postavljene ciljeve. GUIDL jezik je definiran tako da bude jednostavan i razumljiv te da bude što bliži osobama oštećena vida u smislu što lakšeg definiranja veličina i pozicija sučelja i njegovih elemenata tj. grafičkih kontrola, kao i ostalih karakteristika elemenata sučelja. Sintaksa GUIDL jezika je zbog svoje jednostavnosti inspirirana sintaksom BASIC jezika, kao jezika koji je posebno razvijen za edukaciju programera početnika te kao takav podsjeća na govorni engleski jezik. U sintaksu su ugrađeni i posebni koncepti za osobe oštećena vida poput stupnjevitog definiranja prototipnih veličina elemenata, kao i određivanje pozicija elemenata sučelja prema ekranskim kvadrantima.

EBNF gramatika GUIDL jezika je dana u nastavku:

project = projectcode, controlname, form;
projectcode = 'Project ' | 'project ';
form = formcode, controlname, formattributes, [controldeclarations], formend;
formcode = 'Frm ' | 'frm ';
formend = ('End' | 'end'), [eol];
controlname = qoute, word, qoute, eol;
word = alphabeticcharacter, {alphabeticcharacter | digit};
qoute = """ | "';
alphabeticcharacter = "A" | "B" | "C" | ... | "X" | "Y" | "Z" | "a" | "b" | "c" | … | "x" | "y" | "z" ;

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;

eol = ['\r']'\n' | ['\r']'\n', ['\r']'\n';

formattributes = frmcommonattributes, windowstateattribute, {colorattribute};

commonattributes = textattribute, restcommonattributes;

frmcommonattributes = textattribute, frmrestcommonattributes;

restcommonattributes = sizeattribute, locationattribute;

frmrestcommonattributes = frmsizeattribute, locationattribute;

textattribute = textcode, qoute, sentence, qoute, eol;

sentence = word, {ws, word} | ws;

textcode = 'Text = ' | 'Text=' | 'text = ' | 'text=';

sizeattribute = sizecode, (size | width, ws ,height), eol;

frmsizeattribute = sizecode, (frmsize | frmwidth, ws ,frmheight), eol;

sizecode = 'Size = ' | 'Size=' | 'size = ' | 'size=';

size = 'small' | 'medium' | 'big';

frmsize = 'frmsize1' | 'frmsize2' | 'frmsize3'

width = 'width1' | 'width2' | 'width3';

height = 'height1' | 'height2' | 'height3';

frmwidth = 'frmwidth1' | 'frmwidth2' | 'frmwidth3';

frmheight = 'frmheight1' | 'frmheight2' | 'frmheight3';

ws = {' ' | '\t'}-;

locationattribute = locationcode, xposition, ws, yposition, eol;

locationcode = 'Location = ' | 'Location=' | 'location = ' | 'location=';

xposition = 'left' | 'center' | 'right';

yposition = 'top' | 'middle' | 'bottom';

windowstateattribute = windowstatecode, windowsize, eol;

windowstatecode = 'Windowstate = ' | 'Windowstate=' | 'windowstate = ' | 'windowstate=';

windowsize = 'normal' | 'maximized' | 'minimized';

colorattribute = colorcode, color, ws, color, eol;

colorcode = 'Color = ' | 'Color=' | 'color = ' | 'color=';

color = 'red' | 'green' | 'blue' | 'black' | 'white';

offsetvalue = offsetcode, {digit}-, eol, [offsetvalue];

offsetcode = 'xoffsetplus=' | 'xoffsetminus=' | 'yoffsetplus=' | 'yoffsetminus=' | 'xoffsetplus = ' |
'xoffsetminus = ' | 'yoffsetplus = ' | 'yoffsetminus = '

controldeclarations = controldeclaration, anothercontroldeclaration, eol;

anothercontroldeclaration = controldeclaration, anothercontroldeclaration | / nothing */;*

controldeclaration = button | label | textbox | checkbox | combobox | picturebox | radiobutton | ddmenu;

button = buttoncode, controlname, commonattributes, {colorattribute}, {offsetvalue}, endcode;

buttoncode = 'Btn ' | 'btn ';

endcode = ('Cend' | 'cend'), eol;

label = labelcode, controlname, commonattributes, {colorattribute}, {offsetvalue}, endcode;

labelcode = 'Lbl ' | 'lbl ';

textbox = txtcode, controlname, commonattributes, passwordcharattribute, {colorattribute}, {offsetvalue}, endcode;

txtcode = 'Txt ' | 'txt ';

passwordcharattribute = passwordcode, alphabeticcharacter | digit | specialsign, eol;

passwordcode = 'Passwordchar = ' | 'Passwordchar=' | 'passwordchar = ' | 'passwordchar=';

specialsign = "#" | "$" | "%" | "&" | "?" | "" | "@";*

checkbox = checkboxcode, controlname, commonattributes, valueattribute, {colorattribute}, {offsetvalue}, endcode;

checkboxcode = 'Chk ' | 'chk ';

valueattribute = valuecode, 'uncheck' | 'check', eol;

valuecode = 'Value = ' | 'Value=' | 'value = ' | 'value=';

combobox = comboboxcode, controlname, commonattributes, listattribute, {colorattribute}, {offsetvalue}, endcode;

comboboxcode = 'Cmb ' | 'cmb ';

listattribute = listcode, controlname, {anothername};

listcode = 'List = ' | 'List=' | 'list = ' | 'list=';

anothername = ws, controlname;

picturebox = pictureboxcode, controlname, restcommonattributes, pictureattribute, {colorattribute}, {offsetvalue}, endcode;

pictureboxcode = 'Pic ' | 'pic ';

pictureattribute = picturecode, path, eol;

picturecode = 'Picture = ' | 'Picture=' | 'picture = ' | 'picture=';

*path = alphabeticcharacter, ':\', {alphabeticcharacter | digit | ws | "\"}, ".", 5 * (alphabeticcharacter | digit);*

radiobutton = radiobuttoncode, controlname, commonattributes, stateattribute, {colorattribute}, {offsetvalue}, endcode;
radiobuttoncode = 'Rab ' | 'rab ';
stateattribute = statecode, 'true' | 'false', eol;
statecode = 'State = ' | 'State=' | 'state = ' | 'state=';
ddmenu = ddmenucode, controlname, ddmenuattributes, endcode;
ddmenucode = 'DDMenu ' | 'ddmenu ';
ddmenuattributes = ddmenuattributescode, item, {anotheritem};
ddmenuattributescode = 'Menuitems = ' | 'menuitems = ' | 'Menuitems=' | 'menuitems=';
item = itemname, {subitem}
itemname = {alphabeticcharacter | digit}-;
subitem = itemname, "~", itemname, {anothersubitem};
anothersubitem = "~", itemname;
anotheritem = ws, item;

Definiranje sučelja započinje definiranjem naziva projekta. Nakon toga se pristupa definiciji naziva forme tj. ekrana i njezinih atributa. Atributi forme su, uz naziv, pozadinska i prednja boja (boja fonta), veličina i lokacija, naslov forme i inicijalno stanje forme (normalno, minimizirano, maksimizirano). Forma počinje ključnom riječi *Frm* ili *frm*, a završava ključnom riječi *End* ili *end*. Između ovih ključnih riječi dolaze definicije svih ostalih grafičkih kontrola tj. elemenata. Svaka grafička kontrola započinje s ključnom riječi koja označava tip kontrole koji se definira. Nakon tipa kontrole dolazi naziv kontrole, atributi kontrole i kraj kontrole koji je označen ključnom riječi *Cend* ili *cend*. Atributi svake pojedine kontrole, uz naziv, su tekst, veličina, lokacija, pozadinska i prednja boja, lokacijski pomak te pojedini specifični atributi, poput oznake za sakrivanje znakova zaporke kod kontrole tekstualnog polja, elemenata liste u padajućem popisu, elemenata padajućeg izbornika, vrijednost inicijalnog odabira kućice ili gumba za odabir, putanje do slike kod kontrole za prikaz slike te inicijalnog načina pojavljivanja forme (*normal*, *minimized*, *maximized*).

Naziv projekta (*Project*) atribut je potreban kao parametar zbog formiranja klasa i naziva okvira nazivlja koji se koriste kao koncepti u novijim objektno orijentiranim jezicima i njihovim razvojnim okolinama. Svaki naziv (projekta, forme ili kontrole) mora početi sa znakom, nakon čega može slijediti još znakova ili brojeva. Naziv projekta i forme se navodi u dvostrukim navodnicima dok se nazivi pojedinih tipova kontrola navode u jednostrukim

navodnicima. U jednostrukim navodnicima navodi se i atribut teksta svake pojedine kontrole za koji vrijede ista pravila kreiranja kao i za naziv projekta. Svi ostali atributi se navode bez navodnika. Atribut naziva projekta se navodi na samom početku, a nakon njega slijedi ključna riječ za početak forme, iza koje slijede definicije pojedinih kontrola te ključna riječ *End* koja označava kraj forme. Svaka kontrola započinje ključnom riječi koja označava tip kontrole, nakon koje slijedi naziv kontrole u navodnicima i u nastavku definicija svih ostalih atributa kontrole čije vrijednosti se navode bez navodnika.

Svaka definicija pojedinog atributa kontrole se nalazi u zasebnom retku. Na kraju definicije svake pojedine kontrole i njenih atributa dolazi ključna riječ *Cend* ili *cend* koja označava kraj definicije pojedine kontrole na formi. Između definicija pojedinih kontrola i između definicija pojedinih atributa može doći jedan ili nijedan prazan red. Pojava više od jednog praznog reda će dovesti do sintaktičke pogreške. Ovo pravilo je uvedeno kako ne bi došlo do prevelike nepreglednosti koda uslijed slučajnog ili namjernog unošenja više praznih redova. Pojedina vrijednost atributa se atributu pridružuje pomoću znaka jednakosti koji može biti u formatu ' = ' ili '=', dakle koji može i ne mora imati prazninu prije i poslije znaka jednakosti.

Atributi zajednički svim kontrolama, kao i formi su tekst, veličina, lokacija, boja i lokacijski pomak. Tekst označava sadržaj koji je ispisan na kontroli (primjerice gumbu) ili u slučaju primjerice forme, sadržaj koji se nalazi na traci naslova forme i navodi se u jednostrukim navodnicima. Veličina i lokacija označavaju dimenzije forme tj. pojedine kontrole kao i njihovu poziciju na ekranu tj. na formi u slučaju grafičkih kontrola i navode se bez navodnika. Boja predstavlja boju pozadine tj. prednjeg dijela forme tj. kontrole (boja fonta) dok lokacijski pomak predstavlja pozitivan ili negativan horizontalni ili vertikalni pomak kontrole u određenom lokacijskom kvadrantu u koji kontrola može biti smještena. U prototip GUIDL jezika su ugrađeni određeni koncepti koji osobama oštećena vida olakšavaju postavljanje veličine i pozicije forme tj. pojedine grafičke kontrole na formi. Umjesto definiranja visine i širine u broju piksela, forma ima tri definirane veličine:

- *frm1size* – 420 piksela * 280 piksela
- *frm2size* – 540 piksela * 370 piksela
- *frm3size* – 690 piksela * 470 piksela

Potrebno je dakle jednostavno definirati atribut veličine forme (*Size*) kao jednu od tri navedene veličine što olakšava dizajniranje jer se osoba oštećena vida mora odlučiti za samo jednu od tri moguće veličine forme: mala, srednja i velika. Na taj način se postiže veća jednostavnost, pogotovo za programere i dizajnere početnike. Daljnjim razvojem ovog jezika i sustava između ostalog, omogućit će se detaljnije stupnjevanje veličina formi a bit će uvedena i mogućnost navođenja veličine forme u pikselima po visini i širini za naprednije korisnike. Ista funkcionalnost će biti razvijena i za grafičke kontrole. Po istom principu, za svaku grafičku kontrolu moguće je odrediti jednu od tri ponuđene veličine:

- *small* – 75 piksela * 40 piksela
- *medium* – 120 piksela * 70 piksela
- *big* – 170 piksela * 90 piksela

Ukoliko postoji potreba za različitim odnosom između visine i širine od odnosa koji se nude kroz jednu od tri ponuđene veličine forme tj. pojedine kontrole, moguće je zasebno za visinu i širinu odabrati jednu od tri ponuđene veličine koje odgovaraju veličinama koje su ugrađene u tri navedene veličine forme/kontrole. Navedene veličine visine tj. širine su (za formu/kontrole):

- *frmwidth1* – 420 piksela | frm*height1* – 280 piksela
- *frmwidth2* – 540 piksela | frm*height2* – 370 piksela
- *frmwidth3* – 690 piksela | frm*height3* – 470 piksela

- *width1* – 75 piksela | *height1* – 40 piksela
- *width2* – 120 piksela | *height2* – 70 piksela
- *width3* – 170 piksela | *height3* – 90 piksela

Da bi se osobama oštećena vida određivanje lokacije, kako forme, tako i grafičkih kontrola olakšalo, ekran računala tj. forma su podijeljeni na 9 kvadranata (slika 6.1).

lijevo gore (*left top*)	centar gore (*center top*)	desno gore (*right top*)
lijevo sredina (*left medium*)	centar sredina (*center middle*)	desno sredina (*right middle*)
lijevo dolje (*left bottom*)	centar dolje (*center bottom*)	desno dolje (*right bottom*)

Slika 6.1 Moguće lokacije forme/kontrole

Postavljanjem atributa *location* (u pikselima) na jednu od željenih vrijednosti, forma tj. pojedina kontrola se pozicionira u određeni kvadrant ekrana tj. forme. Ukoliko se dvije ili više kontrola postave u pojedini kvadrant forme i time ukupna širina kontrola u kvadrantu prijeđe samu širinu kvadranta, tada će kontrola koja prelazi u drugi kvadrant prelaziti u prvi susjedni kvadrant s desne strane. Ukoliko ukupna širina tj. visina kontrola premaši ukupnu širinu tj. visinu forme, sustav po završetku prevođenja koda ispisuje poruku upozorenja uz poruku o uspješnom ili neuspješnom prevođenju koda. Forma se u pojedini kvadrant postavlja tako da lijevi gornji kut kvadranta označava početnu točku tj. lijevi gornji kut same forme. Ukoliko se dogodi da forma prelazi van ekrana s donje ili desne strane, tada će se forma automatski pomaknuti gore tj. lijevo za toliko piksela koliko je potrebno da bi cijela stala na ekran. Ukoliko sama veličina forme prijeđe veličinu samog ekrana, tada će se također prilikom završetka procesa prevođenja koda ispisati odgovarajuća poruka upozorenja. Unutar pojedinog kvadranta je kontroli moguće odrediti odgovarajući pozitivan ili negativan horizontalni tj. vertikalni pomak kako bi se kontrola dodatno pozicionirala unutar samog lokacijskog kvadranta. Atribut lokacijskog pomaka (*offset*) se definira u pikselima kao jedna od četiri moguće komponente s tim da je unutar atributa pojedine kontrole moguće odrediti i više lokacijskih pomaka koji će se potom zbrojiti tj. oduzeti. Četiri moguća lokacijska pomaka su navedena u nastavku:

- xoffsetplus – horizontalni pozitivan pomak u pikselima
- xoffsetminus – horizontalni negativan pomak u pikselima
- yoffsetplus – vertikalni pozitivan pomak u pikselima
- yoffsetminus – vertikalni negativan pomak u pikselima

Nakon definiranja sučelja u GUIDL jeziku, pripadajući kod se provjerava sa sintaktičkog aspekta te se nakon utvrđene sintaktičke ispravnosti prevodi u jednu od odabranih tehnologija tj. programskih jezika. U prototipu GUIDL sustava podržani jezici u koje se definirani opis sučelja može prevesti su Visual Basic.NET i Visual C#.NET.

Atribut inicijalnog stanja forme se navodi bez navodnika i može se postaviti na jednu od tri moguće vrijednosti: *normal, maximized* ili *minimized* koje definiraju hoće li se forma pojaviti u svojim stvarnim dimenzijama, maksimizirana preko cijelog ekrana ili minimizirana u Windows traci. Atribut boje se definira kao dvokomponentni atribut u kojem se navode dvije boje. Prva boja označava prednju boju tj. boju fontova, a druga boja označava boju pozadine.

U prototipu su podržane sljedeće boje:

- Red (Crvena)
- Green (Zelena)
- Blue (Plava)
- Black (Crna)
- White (Bijela)

Iako se radi o osobama oštećena vida, boje su bitan detalj koji je potrebno definirati što osobe oštećena vida čine temeljem iskustvenog opisa pojedinih dijelova okoline oko sebe i time procjenjuju prikladnost pojedine boje za pojedinu priliku iako u principu ne znaju kako pojedina boja izgleda. Kod tekstualnog polja moguće je odrediti atribut zaporke koji označava znak koji će služiti za sakrivanje znakova koji se upisuju u tekstualno polje. Podržani znakovi u prototipu GUIDL jezika kojima je moguće sakriti sadržaj polja sa zaporkom su:

- Abecedni znakovi
- Brojevi
- #
- $
- %
- &
- ?

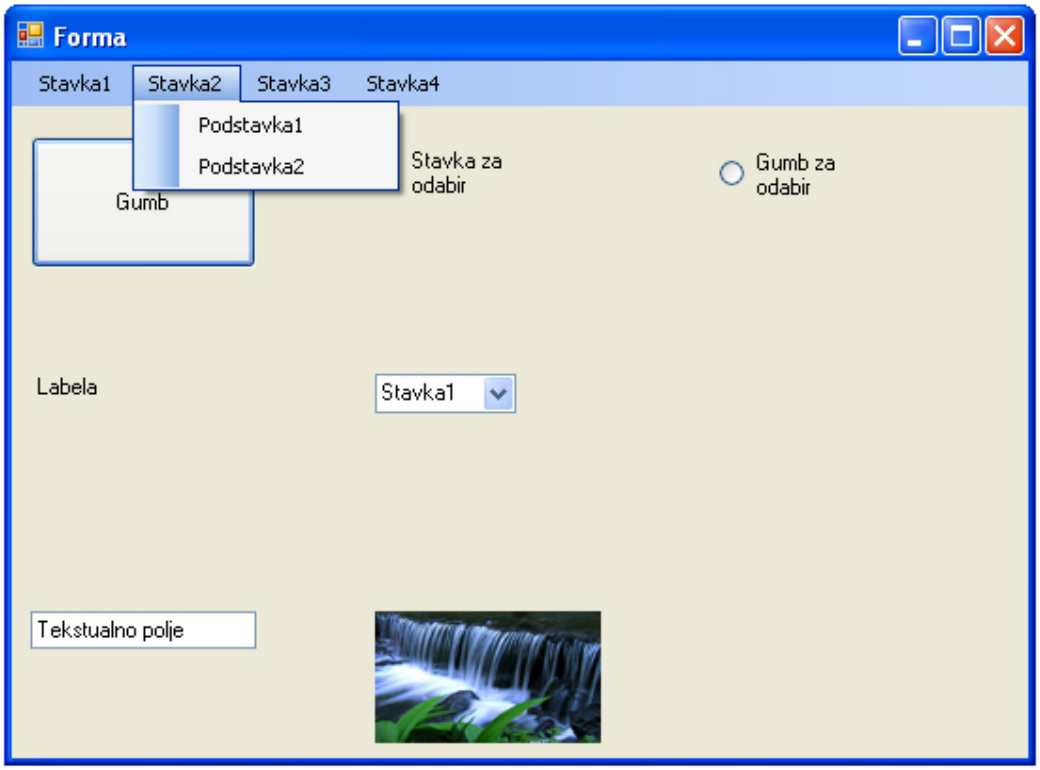
- *
- @

Atribut ***value*** CheckBox (polja za odabir) kontrole može biti definiran kao ***check*** (odabran) i ***uncheck*** (neodabran). Isto tako atribut ***state*** radio button (radio gumba) kontrole može biti definiran kao ***true*** (odabran) ili ***false*** (neodabran). Lista se u kontroli ComboBox (padajući popis) definira navođenjem svih stavki kontrole odvojenih razmakom, bez navodnika. Isto tako se bez navodnika navodi i atribut ***picture*** kontrole PictureBox (okvir sa slikom) koji definira putanju na disku do slike koja će se prikazati unutar slikovne kontrole. Stavke i podstavke kontrole MenuStrip (padajućeg izbornika) se definiraju putem atributa ***menuitems*** navođenjem svih stavki i podstavki izbornika odvojenih razmakom, bez navodnika. Podstavke se međusobno razdvajaju znakom ~.

Sintaksa GUIDL jezika je prikazana putem sintaktičkih dijagrama u prilogu A na kraju rada, slijedom navedenim u EBNF prikazu gramatike.

7. PRIMJER KORIŠTENJA GUIDL JEZIKA I SUSTAVA

Kao primjer korištenja GUIDL sustava bit će prikazan proces izrade sučelja koje će na sebi jednostavno sadržavati sve kontrole ugrađene u prototip GUIDL sustava a također će biti dan i primjer izrade sučelja za prijavu korisnika u neki zamišljeni sustav. Prvi korak u izradi prvog primjera sučelja je pisanje samog GUIDL koda za prvi primjer. GUIDL kod se može pisati u bilo kojem uređivaču teksta i sprema se pod ekstenzijom .guidl, iako kompajler može primiti i datoteke s drugim ekstenzijama ukoliko se iste definiraju kao ulazne. Radi boljeg razumijevanja primjera, na slici 7.1 je prikazan konačan rezultat tj. konačan izgled sučelja iz prvog primjera.



Slika 7.1 Konačni izgled sučelja iz primjera 1

Kao što je već spomenuto u nastavku slijedi prvi korak izrade prikazanog sučelja tj. pisanje samog GUIDL koda.

project "vbgui"
frm "frmForma"

Text = 'Forma'

Size = frmsize2

Location = center middle

Windowstate = normal

DDMenu 'izbornik'

Menuitems = Stavka1 Stavka2 Podstavka1~Podstavka2 Stavka3 Stavka4

cend

btn 'btnGumb'

Text = 'Gumb'

Size = medium

Location = left top

yoffsetplus = 30

cend

lbl 'lblLabela'

Text = 'Labela'

Size = small

location = left middle

yoffsetplus = 30

Cend

Txt 'txtTekst'

Text = 'Tekstualno polje'

Size = medium

Location = left bottom

yoffsetplus = 30

cend

Chk 'chkOdabir'

Text = 'Stavka za odabir'

Size = small

Location = center top

Value = uncheck
yoffsetplus = 30
cend

Cmb 'cmbPopis'
Text = 'Stavka1'
Size = small
Location = center middle
List = Stavka1 Stavka2 Stavka3 Stavka4
yoffsetplus = 30
cend

Pic 'picSlika'
size=medium
Location = center bottom
Picture = C:\Projects\vbgui\vbgui\waterfall.jpg
yoffsetplus = 30
cend

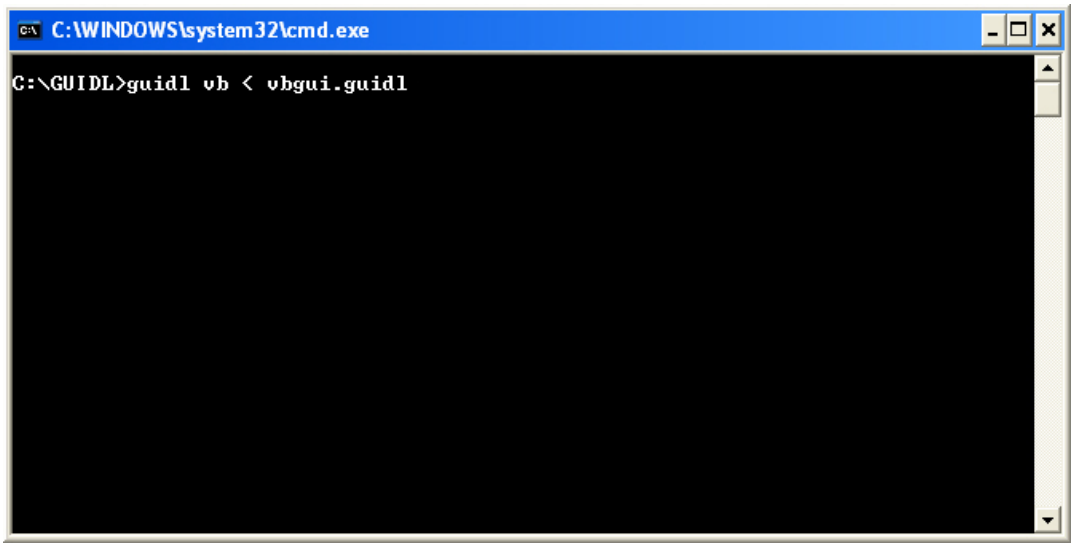
Rab 'radiobut'
Text = 'Gumb za odabir'
Size = small
Location = right top
State = false
yoffsetplus = 30
cend
End

Iz navedenog koda se može vidjeti način konstruiranja samog sučelja. Uz navođenje naziva projekta, navodi se forma s pripadajućim atributima te unutar nje definicije svih željenih kontrola. Samo dizajniranje sučelja je jednostavno i svodi se na navođenje ključnih riječi za pojedine kontrole tj. formu te pripadajuće atribute pojedinih kontrola tj. forme pomoću kojih se oblikuje izgled, veličina i pozicija forme tj. pojedinih kontrola na formi. Određivanje veličine i pozicije je također pojednostavljeno i svodi se na mogućnost biranja između tri

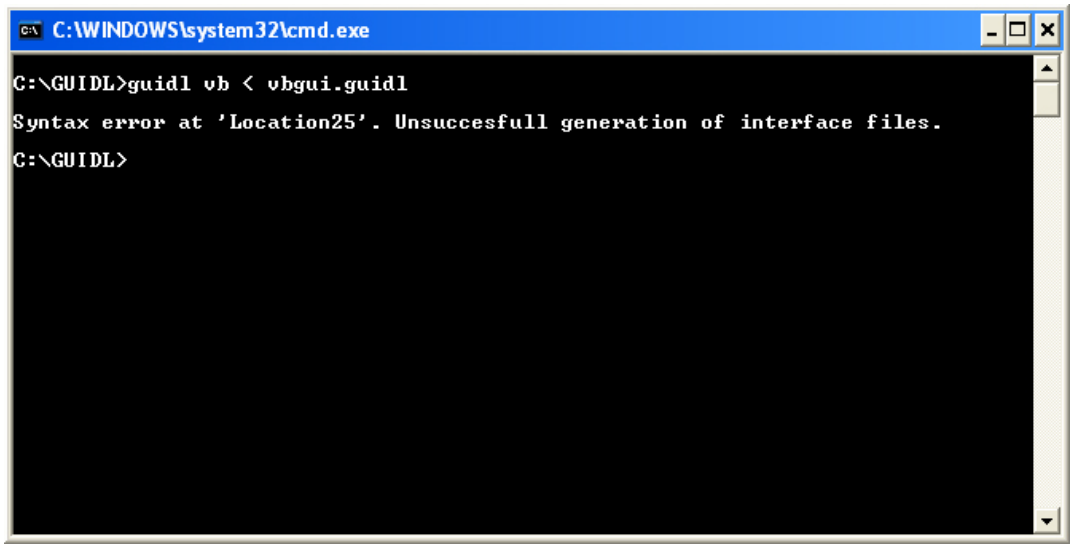
standardne veličine koje određuju ujedno i širinu i visinu forme tj. kontrole ili biranje između tri standardne veličine koje određuju zasebno širinu tj. visinu forme tj. kontrole.

U navedenom primjeru koda se može vidjeti da je između definicija pojedine kontrole moguće ostaviti red razmaka, da je ključne riječi koje označavaju tip pojedine kontrole moguće navesti s velikim ili malim početnim slovom, da je ispred i iza znaka jednakosti moguće ostaviti po jedan razmak ili ga je moguće navesti bez ikakvog razmaka, da je ključnu riječ *cend*, koja označava kraj definicije kontrole, moguće napisati također s početnim velikim ili malim slovom, a isto vrijedi i za ključnu riječ *end* koja označava kraj definicije forme. Sva ostala pravila sintakse je moguće vidjeti u navedenoj EBNF gramatici kao i u odlomcima teksta koji slijede nakon same gramatike i opisuju gramatiku te pravila formiranja sučelja. U navedenom sučelju su navedene sve kontrole podržane u prototipu s većinom atributa koje je moguće definirati. Od podržanih atributa nisu navedeni atributi *passwordchar* koji određuje znak za sakrivanje unosa kod tekstualnog polja u koje se unosi zaporka te atribut *color* koji definira prednju i pozadinsku boju forme tj. kontrole. Navedeni atributi će biti prikazani u drugom primjeru.

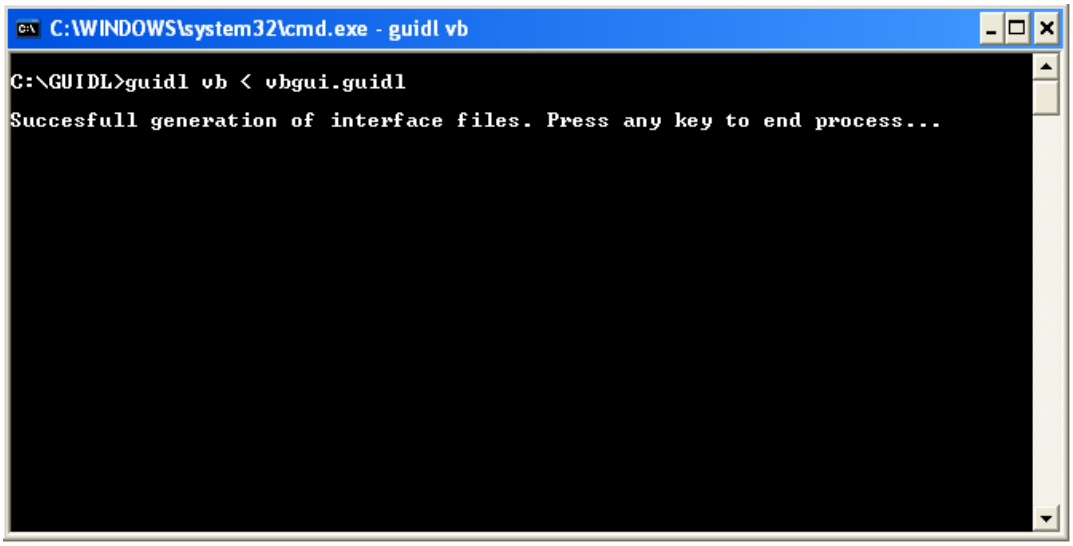
Nakon definiranja sučelja pisanjem odgovarajućeg GUIDL koda, napisani kod se sprema u datoteku oblika naziv.guidl, tj. u navedenom primjeru vbgui.guidl te se koristeći navedenu datoteku kao ulaz, poziva GUIDL kompajler (slika 7.2). GUIDL kompajler će u slučaju sintaktičke pogreške javiti da je došlo do pogreške te će javiti na kojem mjestu je došlo do pogreške (slika 7.3), dok će u slučaju potpune ispravnosti koda ispisati poruku o uspješnom prevođenju koda (slika 7.4). Prilikom poziva GUIDL kompajlera potrebno je naznačiti u koju od programskih tehnologija tj. jezika se GUIDL sučelje prevodi. U prototipu GUIDL sustava su podržani jezici Visual Basic.NET i Visual C#.NET. Za prevođenje GUIDL koda u Visual Basic.NET format potrebno je pozvati GUIDL kompajler naredbom guidl vb < vbgui.guild gdje je vbgui.guidl datoteka s GUIDL kodom iz primjera 1, a ukoliko se GUIDL kod želi prevesti u Visual C#.NET, potrebno je pozvati GUIDL kompajler naredbom guidl c# < vbgui.guidl. Ukoliko se parametar jezika izostavi tj. ukoliko se GUIDL kompajler pozove naredbom guidl < vbgui.guidl, tada će se prema inicijalnim postavkama izvršiti prevođenje koda u Visual Basic.NET format.



Slika 7.2 Poziv GUIDL kompajlera



Slika 7.3 Sintaktička pogreška prilikom prevođenja GUIDL koda



Slika 7.4 Uspješno prevođenje GUIDL koda

Ukoliko je prevođenje GUIDL koda uspješno izvršeno, automatski će se generirati datoteka s parametrima koji su nađeni unutar GUIDL definicije sučelja. Ovi parametri će se koristiti prilikom generiranja sučelja u odabranom programskom jeziku. Ukoliko su svi parametri definirani u skladu s pravilima definiranja parametara, kompajler će generirati odgovarajuće datoteke koje su kompatibilne s odabranom programskom tehnologijom tj. programskim jezikom i njegovom razvojnom okolinom. Tako će se za Visual Basic.NET generirati tri datoteke s odgovarajućim kodom, koje je potrebno uključiti u prazan Visual Basic.NET projekt. Uključivanje datoteka u projekt se provodi putem razvojne okoline odabranog jezika i u potpunosti je podržano postojećom pomoćnom tehnologijom koju osobe oštećena vida koriste pri radu s računalom. Alternativno je moguće također i kopirati kreirane datoteke u odgovarajući direktorij projekta. Navedeni postupci su detaljnije opisani u nastavku.

Navedene datoteke koje GUIDL sustav kreira su:

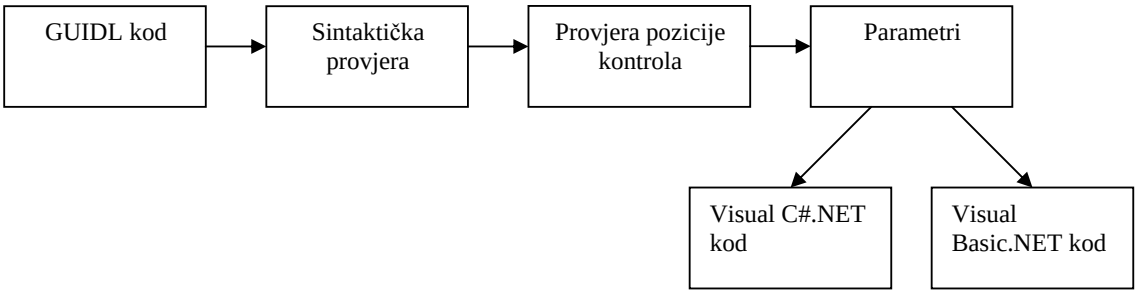
- Application.Designer.vb
- Form1.Designer.vb
- Form1.vb

Ukoliko su navedene datoteke uspješno generirane, bit će spremljene u istom direktoriju u kojem se nalazi i GUIDL kompajler. Sljedeći korak je stvaranje novog praznog Visual Basic.NET projekta kojeg treba snimiti pod istim nazivom koji je određen kao naziv projekta

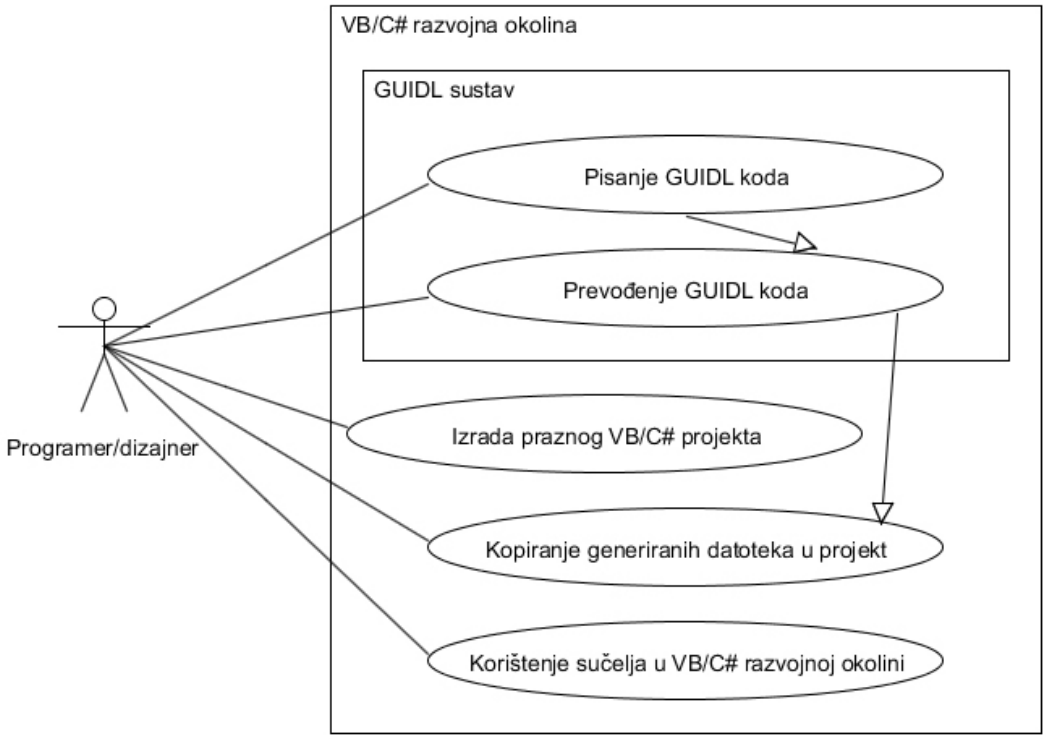
(atribut *project*) u GUIDL kodu ili je također moguće otvoriti postojeći projekt koji također mora nositi isti naziv kao atribut *project*. Nakon snimanja praznog Visual Basic.NET projekta pod nazivom projekta definiranom u GUIDL kodu ili otvaranja postojećeg projekta pod istim nazivom, potrebno je u okviru razvojne okoline odabranog jezika uključiti sve tri datoteke u otvoreni projekt. Ova radnja je u potpunosti podržana postojećom pomoćnom tehnologijom tj. sintetizatorima govora. Alternativno je također moguće kopirati sve tri navedene datoteke u pripadajući Visual Basic.NET projekt (preko postojećih datoteka ukoliko one postoje). Datoteke Form1.Designer.vb i Form1.vb se nalaze u direktoriju koji nosi naziv samog projekta, u kojem se nalazi i direktorij My Project u koji je potrebno kopirati datoteku Application.Designer.vb. Ukoliko je Visual Studio bio otvoren prilikom kopiranja navedenih datoteka, povratkom u njega će se automatski postaviti pitanje o ponovnom učitavanju izmijenjenih datoteka na koje je potrebno odgovoriti potvrdno da bi se izmjene vidjele u razvojnoj okolini. Ukoliko je Visual Studio bio zatvoren prilikom kopiranja navedenih datoteka, samim otvaranjem projekta će sve izmjene biti vidljive tj. u razvojnoj okolini će se pojaviti definirano sučelje. U slučaju prevođenja u Visual C#.NET format rezultat prevođenja će biti sljedeće datoteke:

- Form1.cs
- Form1.Designer.cs
- Program.cs

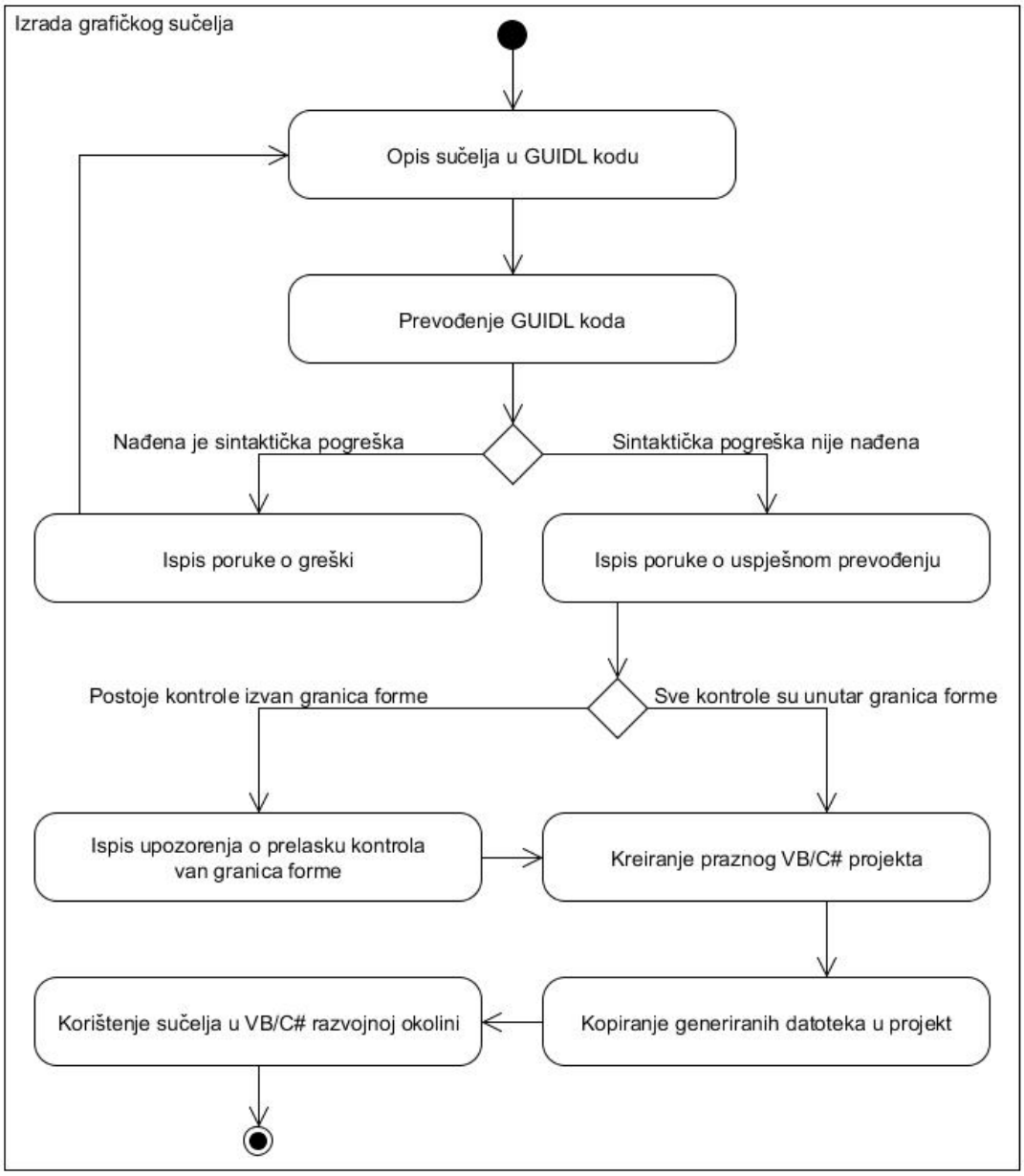
Nakon što se napravi novi prazan Visual C#.NET projekt i spremi pod nazivom projekta definiranom u GUIDL kodu ili se otvori postojeći projekt pod istim nazivom, potrebno je u okviru razvojne okoline uključiti sve tri datoteke u otvoreni projekt. Kao što je već spomenuto, ova radnja je u potpunosti podržana postojećom pomoćnom tehnologijom. I u ovom slučaju je moguće navedene datoteke kopirati u direktorij koji nosi naziv samog projekta, preko postojećih datoteka i zatim se vratiti u Visual Studio i odgovoriti potvrdno na pitanje o ponovnom učitavanju izmijenjenih datoteka ili ponovo otvoriti cjelokupni projekt. U oba slučaja će se u razvojnoj okolini prikazati sučelje definirano u GUIDL kodu. Proces prevođenja je prikazan na slici 7.5 a na slikama 7.6 i 7.7 prikazani su UML dijagram slučajeva korištenja (Use Case Diagram) i dijagram aktivnosti (Activity Diagram).



Slika 7.5 Proces prevođenja sučelja u željeni programski jezik

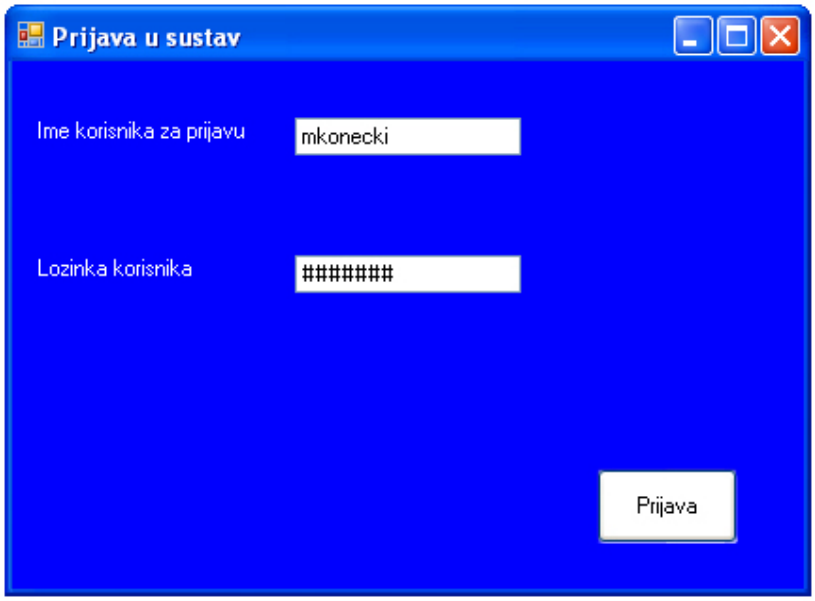


Slika 7.6 UML dijagram slučajeva korištenja – Izrada grafičkog sučelja



Slika 7.7 UML dijagram aktivnosti – Izrada grafičkog sučelja

U primjeru 2 bit će prikazana izrada sučelja za prijavu korisnika u neki zamišljeni sustav. Sučelje će se sastojati od forme koja će imati plavu pozadinu i bijela slova, odgovarajući natpis, te pripadajuće nazive i tekstualna polja za korisničko ime i lozinku čiji će znakovi biti sakriveni znakom # te gumb za prijavu. Prije samog prikaza GUIDL koda za navedeno sučelje bit će prikazan konačan izgled sučelja (slika 7.8). Sučelje će biti prevedeno u Visual C#.NET format.



Slika 7.8 Konačni izgled sučelja iz primjera 2

U nastavku slijedi izrada prikazanog sučelja tj. pisanje samog GUIDL koda:

project "c#gui"
frm "frmForma"
Text = 'Prijava u sustav'
Size = frmwidth1 frmheight1
Location = center middle
Windowstate = normal
color = white blue

Lbl 'lblImeKor'
Text = 'Ime korisnika za prijavu:'
Size = medium
location = left top
yoffsetplus = 20
Cend

Lbl 'lblLozinka'
Text = 'Lozinka korisnika:'
Size = medium

```
location = left middle
Cend

Txt 'txtImeKor'
Text = ' '
Size = medium
Location = center top
yoffsetplus = 20
cend

Txt 'txtLozinka'
Text = 'lozinka'
Size = medium
Location = center middle
passwordchar = #
cend

Btn 'btnPrijava'
Text = 'Prijava'
Size = small
Location = right bottom
color = black white
xoffsetplus = 20
yoffsetplus = 20
cend
End
```

U prikazanom GUIDL kodu su definirani elementi sučelja spomenuti u ranijem odlomku, dakle: forma, dva natpisa (label), dva tekstualna polja i jedan gumb. U odnosu na primjer 1, u primjeru 2 su uvedena tri nova atributa. To su *color* koji definira prednju tj. pozadinsku boju forme tj. kontrole, atribut *passwordchar* koji definira znak kojim se sakriva unos u tekstualno polje koje ima svrhu unosa lozinke i atribut lokacijskog pomaka (*xoffsetplus* i *yoffsetplus*) kojim se pojedina kontrola pomiče horizontalno tj. vertikalno u pozitivnom ili negativnom smjeru. Ostali atributi i njihovo definiranje je opisano u prethodnim odlomcima.

Nakon definiranja GUIDL koda, poziva se GUIDL kompajler uz pripadajući argument formata u koji se GUIDL kod prevodi.

*guidl c# < c#*gui*.guidl*

Ukoliko nema sintaktičkih pogrešaka ispisuje se poruka o uspješnom prevođenju GUIDL koda:

Succesfull generation of interface files. Press any key to end process...

Ukoliko dođe do sintaktičke pogreške ispisuje se poruka pogreška i upućuje se na mjesto nastanka sintaktičke pogreške.

Ukoliko ni jedna sintaktička pogreška nije nađena, generiraju se pripadajuće Visual C#.NET datoteke koje je potrebno kopirati preko postojećih u novom praznom Visual C#.NET projektu koji nosi isti naziv kao naziv projekta definiran u GUIDL kodu. GUIDL kompajler ima ugrađenu kontrolu pozicije forme, koja korigira poziciju forme, ukoliko zbog širine forme dođe do prelaska forme izvan granica ekrana. Ova korekcija se vrši automatski te za istu nije potrebna, niti se prikazuje posebna poruka. GUIDL kompajler također provjerava je li zbog količine kontrola na formi ili pozicije određene kontrole došlo do prelaska određenih kontrola van granica forme. Ukoliko je došlo do prelaska jedne ili više kontrola van granica forme, proces prevođenja će se uredno provesti uz generiranje pripadajućih datoteka, no doći će do poruke upozorenja o prelasku kontrola van granica forme:

Warning: Some of the form's controls are out of the form's boundaries

U prototipu GUIDL kompajlera su ugrađene neke restrikcije kako bi prototip bio što jednostavniji, no u daljnjem razvoju prototipa i sustava ugradit će se još veća razina detaljizacije, primjerice poruka o tome u kojem kvadrantu je došlo do prelaska granica forme, horizontalni i vertikalni pomaci (offset) forme, detekcija preklapanja kontrola i mogućnost određivanje broja kvadranata na ekranu tj. formi.

8. TESTIRANJE PROTOTIPA GUIDL SUSTAVA

Da bi se ocijenila funkcionalnost prototipa i mjera u kojoj ispunjava prethodno definirane zahtjeve za ovakvim sustavom, definiranim temeljem inicijalnog istraživanja, provedeno je dodatno istraživanje u kojem su programeri i dizajneri oštećena vida koristili razvijeni prototip GUIDL sustava i dali povratne informacije o iskustvu korištenja prototipa putem anketnog upitnika.

Kako bi se testiranje provelo što učinkovitije, bilo je potrebno pronaći odgovarajuću testnu grupu ispitanika. Prilikom inicijalnog istraživanja ispitanici su se mogli dobrovoljno prijaviti za kasnije testiranje prototipa. Ovim putem je dobiveno 37 prijava za testiranje prototipa, a nakon razvoja prototipa poslan je novi poziv kako bi se došlo do što više ispitanika. Poziv je upućen na sva prepoznata mjesta na kojima se okupljaju programeri oštećena vida, a koja su identificirana tijekom inicijalnog istraživanja (program-l: V.I. Programmers Discussion List, 2011; programmingblind: Programming Blind, 2011; Blind Geeks, 2011; JAWSScripts, 2011). Prije same distribucije prototipa, prigodni uzorak prijavljenih ispitanika je sadržavao 47 prijavljena sudionika testiranja od kojih je bilo 35 muških i 12 ženskih ispitanika.

Svim ispitanicima je putem e-maila poslana obavijest o linku na kojem mogu preuzeti prototip GUIDL sustava, zajedno s uputama za njegovo korištenje. Upute za korištenje sustava su osim opisa sintakse, sadržavale i tri testna primjera koja su ispitanici trebali riješiti prije početka samog istraživanja kako bi se upoznali sa sustavom na praktičan način i kako bi rad sa sustavom bio bez tehničkih poteškoća. Tri testna sučelja koja su bila uključena kao dio uputa i inicijalnog treninga su bila:

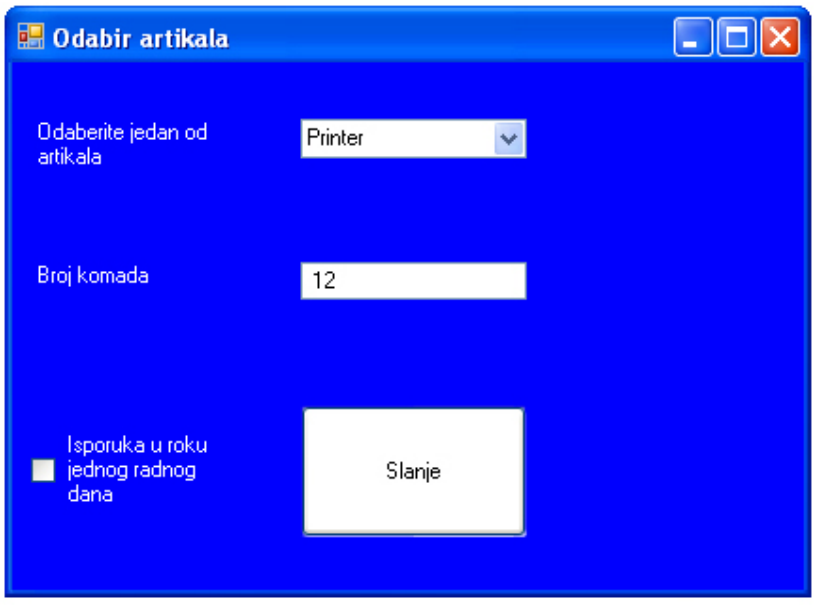
- Sučelje u kojem su redom na formi navedene sve moguće kontrole sa svim mogućim atributima kako bi se prikazale cjelokupne mogućnosti GUIDL jezika i kompajlera (ekvivalentno prikazanom primjeru 1)
- Sučelje za prijavu korisnika u neki zamišljeni sustav (ekvivalentno prikazanom primjeru 2)
- Sučelje u kojem korisnik iz popisa bira neki od ponuđenih artikala, u tekst kućicu upisuje broj odabranih artikala, označava putem kontrole za odabir (checkbox) da

artikli moraju ili ne moraju stići u roku od jednog radnog dana i klikom na gumb šalje narudžbu.

- Sučelje u kojem korisnik definira padajući izbornik sljedeće strukture:
  - Djelatnici
    - Unos
    - Pretraga
    - Ispis
  - Artikli
    - Stanje
    - Promet
  - Izlaz

Osim navedenog izbornika na sučelju je trebalo definirati polje za unos šifre djelatnika, polje za unos šifre artikla, kao i gumbe za pretragu i izlaz.

Prva dva primjera su već prikazana u ranijim odlomcima, a u nastavku slijedi prikaz sučelja pod rednim brojem 3 (slika 8.1) i 4 (slika 8.2). Sučelje 1 i sučelje 3 su definirani kao Visual Basic.NET sučelja, dok su sučelja 2 i 4 definirana kao Visual C#.NET sučelja.



Slika 8.1 Konačni izgled sučelja 3

GUIDL kod za sučelje 3 je definiran u nastavku:

```
project "Artikli"
frm "frmArtikli"
Text = 'Odabir artikala'
Size = frmsize1
Location = center middle
Windowstate = normal
color = white blue

Lbl 'lblArtikl'
Text = 'Odaberite jedan od artikala:'
Size = medium
location = left top
yoffsetplus = 20
Cend

Cmb 'cmbArtikli'
Text = 'Artikli'
Size = medium
location = center top
list = Printer Laptop Smartphone
yoffsetplus = 20
Cend

Lbl 'lblBroj'
Text = 'Broj komada:'
Size = medium
location = left middle
Cend

txt 'txtBroj'
Text = ' '
Size = medium
```

location = center middle

Cend

Chk 'chkRok'

Text = 'Isporuka u roku jednog radnog dana'

Size = medium

Location = left bottom

value = uncheck

yoffsetminus = 20

cend

Btn 'btnNarudzba'

Text = 'Slanje'

Size = medium

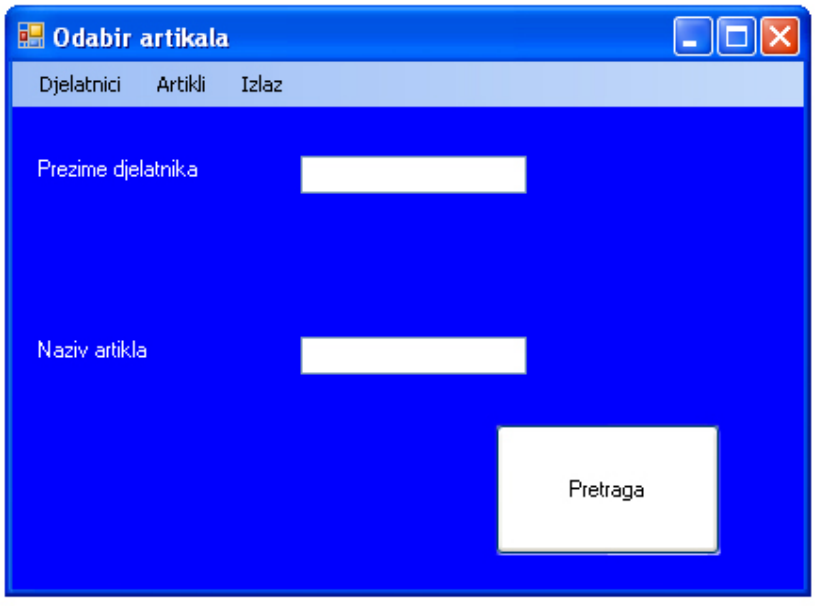
Location = center bottom

color = black white

yoffsetminus = 20

cend

End



Slika 8.2 Konačni izgled sučelja 4

GUIDL kod za sučelje 4 je definiran u nastavku:

project "Djelatnici"
frm "frmDjelatnici"
Text = 'Odabir artikala'
Size = frmsize1
Location = center middle
Windowstate = normal
color = white blue

DDMenu 'izbornik'
Menuitems = Djelatnici Unos~Pretraga~Ispis Artikli Stanje~Promet Izlaz
cend

Lbl 'lblPrezime'
Text = 'Prezime djelatnika:'
Size = medium
location = left top
yoffsetplus = 40
Cend

txt 'txtPrezime'
Text = ' '
Size = medium
location = center top
yoffsetplus = 40
Cend

Lbl 'lblArtikla'
Text = 'Naziv artikla:'
Size = medium
location = left middle
yoffsetplus = 40
Cend

txt 'txtArtikl'
Text = ' '
Size = medium
location = center middle
yoffsetplus = 40
Cend

Btn 'btnPretraga'
Text = 'Pretraga'
Size = medium
Location = right bottom
color = black white
yoffsetminus = 10
xoffsetminus = 40
cend
End

Svako testno sučelje je uključivalo GUIDL kod i upute kako iz GUIDL koda dobiti sučelje u željenoj tehnologiji tj. programskom jeziku. Koraci dolaska do konačnog koda ciljanog programskog jezika definirani u uputama su:

- Pisanje GUIDL koda
- Poziv GUIDL kompajlera s Visual Basic.NET ili Visual C#.NET parametrom
- Ispravljanje eventualno uočenih sintaktičkih ili logičkih pogreški
- Stvaranje novog praznog VB/C# projekta s nazivom definiranim kao naziv projekta u GUIDL opisu sučelja
- Kopiranje generiranih datoteka iz direktorija s GUIDL kompajlerom preko inicijalnih datoteka novostvorenog praznog VB/C# projekta ili uključivanje datoteka u projekt
- Pregled konačnog sučelja u Visual Basic.NET ili Visual C#.NET okolini

Nakon prolaska kroz upute i inicijalni trening, ispitanicima su dana dva nova zadatka. Prvi zadatak je bio izrada sučelja za evidenciju rada djelatnika gdje se iz izbornika bira djelatnik, u

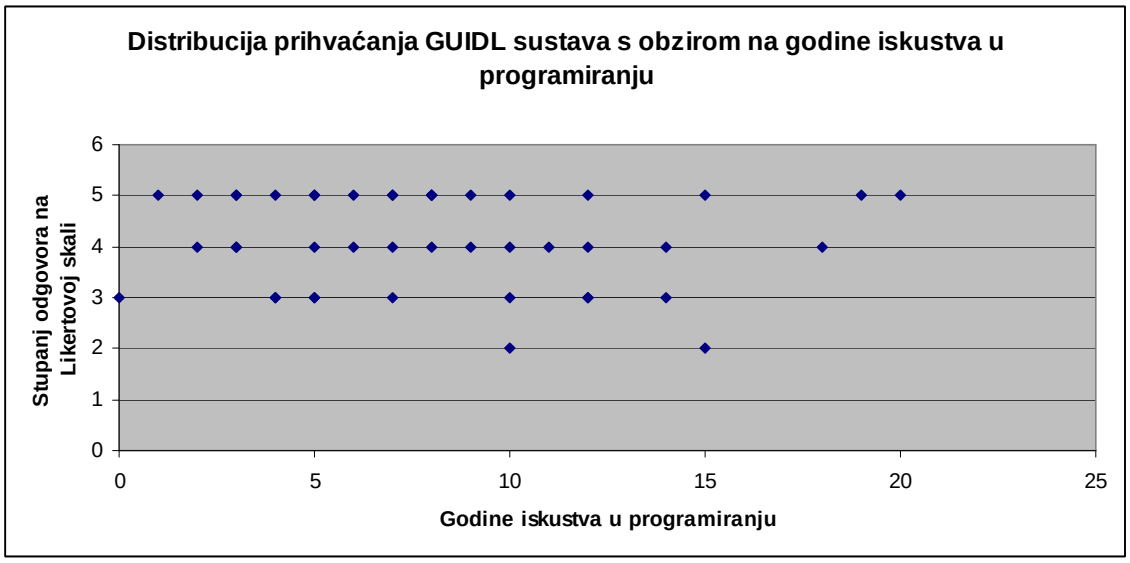
tekst polje se upisuje broj radnih sati, odabire se spol djelatnika te se označava je li djelatnik u radnom odnosu više od 10 godina. Sučelje je također trebalo sadržavati i gumb za upis podataka u bazu. Drugi zadatak je bio izrada sučelja u kojem se odabire marka i snaga vozila, upisuje se ime i prezime vlasnika vozila, odabire se je li automobil dizel ili benzinac, te se prikazuje umanjena slika automobila.

Ispitanici su zamoljeni da nakon izrade navedenih zadataka u GUIDL sustavu popune upitnik kojeg su dobili putem e-maila, zajedno s ostalim uputama vezanim za sustav. Većina pitanja je bila stupnjevana Likertovom skalom od 1 do 5 (1 – u potpunosti se ne slažem, 5 – u potpunosti se slažem) radi što realnije distribucije vrijednosti odgovora. Struktura upitnika i prosječne vrijednosti odgovara zajedno sa standardnom devijacijom su dani u tablici 8.1 a distribucija prihvaćanja GUIDL sustava s obzirom na godine iskustva u programiranju je prikazana na slici 8.3 na kojoj su u omjer stavljene godine iskustva programera i odgovori na treću izjavu iz tablice 8.1 kao indikator prihvaćanja GUIDL sustava.

Tablica 8.1 Rezultati testiranja prototipa

Izjava	**Prosječna vrijednost odgovora**	**Standardna devijacija**
Koliko imate godina?	33,17	8,37
Koliko godina programirate?	8,14	4,86
Korištenjem GUIDL sustava u mogućnosti sam na brži i jednostavniji način oblikovati grafička sučelja u odnosu na postojeća rješenja.	4,11	0,92
Korištenje GUIDL sustava mi je prekomplicirano.	1,57	0,47
GUIDL sustav mi omogućava razmjenu ideja sa svojim kolegama.	4,31	0,75
GUIDL sustav mi omogućava stvaranje sučelja u aktualnim programskim tehnologijama.	4,24	0,59
GUIDL sustav mi omogućava jednostavniji i brži način izrade grafičkih sučelja nego programske razvojne okoline poput Visual Basic-a, Visual C#-a, Visual C++-a, Delphi-a, Java razvojnih okolina i sličnih alata koji omogućavaju razvoj grafičkih sučelja.	4,08	0,64

GUIDL sustav sam u mogućnosti koristiti kao pomoćnu tehnologiju za izradu grafičkih sučelja u okviru postojećih programskih jezika i njihovih razvojnih okolina.	4,27	0,82
GUIDL sustav mi omogućava uključivanje u dijelove razvoja programskog proizvoda koji su mi do sada bili teško dostupni.	4,04	0,73
Tekstualni opis i sintaksa GUIDL sustava su primjereni i u mogućnosti sam koristiti jezik i pomoćni alat bez većih poteškoća.	3,98	0,86
Smatram da GUIDL sustav u svojoj prototipnoj inačici obećava razvoj adekvatnog rješenja za praktičnu primjenu u stvarnim projektima u domeni izrade grafičkih sučelja.	4,01	0,67
Želio/la bih vidjeti daljnji napredak na GUIDL sustavu i nove inačice sustava s dodanim funkcionalnostima i podržanim novim tehnologijama.	4,42	0,53



Slika 8.3 Distribucija prihvaćanja GUIDL sustava s obzirom na godine iskustva u programiranju

9. ZAKLJUČAK

U okviru ovog rada provedena su istraživanja koja su imala za cilj identificirati neke od aktualnih problema programera i dizajnera oštećena vida. Napravljen je pregled aktualnih tehnologija i nastojanja u području prikaza grafičkih elemenata osobama oštećena vida, kao i pregled problema koji se javljaju u području programiranja tj. razvoja grafičkih sučelja kod osoba oštećena vida. Dan je prikaz aktualnih pomoćnih tehnologija koje osobama oštećena vida omogućavaju korištenje računala i analizirana je njihova adekvatnost u pomoći kod korištenja novih programskih tehnologija i razvojnih okolina. Analizirana je razina problematičnosti razvoja grafičkih sučelja kod klasičnih stolnih i web aplikacija te je utvrđeno područje od posebnog interesa gdje je problem razvoja grafičkih sučelja za osobe oštećena vida najizraženiji. U radu su analizirani svi aspekti problema kreiranja grafičkih sučelja za osobe oštećena vida kao i mogući pristupi rješavanju navedenog problema. Identificirani mogući pristupi navedeni su u nastavku:

- Mogli bi biti izvedeni interpreteri koji bi interpretirali forme i njihove elemente zajedno s pripadajućim atributima.
- Programske razvojne okoline bi mogle podržati mogućnost stvaranja određenih grafičkih elemenata putem audio sučelja.
- Mogli bi biti razvijeni i posebni skriptni jezici za pojedine programske razvojne okoline i njihove inačice, koji bi omogućili tekstualni opis GUI elemenata unutar pojedinog programskog jezika.
- Mogao bi biti razvijen novi opisni jezik GUIDL (Graphical User Interface Description Language) koji bi bio razvijen na način da omogući jednostavan opis korisničkih sučelja i njihovih elemenata putem jezika koji je što bliži govornom jeziku tako da bude prikladan kako za programere tako i za dizajnere i koji bi bio centralni dio šireg sustava koji bi omogućio prevođenje opisanih sučelja u konkretan GUI pojedinog programskog jezika korištenjem pripadajućeg GUI generatora.

Odabran je adekvatan pristup rješavanja problema, koji bi doveo do generalnog i opće uporabljivog rješenja. Kao rješenje navedenog problema predložen je novi sustav čiji je centralni dio novi deskriptivni jezik za opis grafičkih sučelja pod nazivom GUIDL (Graphical User Interface Description Language). Temeljem analize i istraživanja definirane su temeljne

karakteristike novog GUIDL sustava koje bi morao zadovoljavati da bi bio primjenjiv i održiv. Spomenute karakteristike su navedene u nastavku:

- Lakoća uporabe: sustav bi trebao biti lak za uporabu kako programerima tako i ostalim zainteresiranim osobama iz informatičke struke, primjerice dizajnerima oštećena vida.
- Intuitivna, jednostavna i lako razumljiva sintaksa: deskriptivni jezik koji će se koristiti za opis grafičkih korisničkih sučelja mora biti intuitivan i lako primjenjiv svima koji se bave programiranjem u području klasičnih stolnih programa, ali i ostalim programerima oštećena vida.
- Neovisnost deskriptivnog jezika: deskriptivni jezik mora biti općenit, a ne specifičan ili usko vezan samo za jednu programsku razvojnu okolinu tj. jezik.
- Proširivost deskriptivnog jezika: deskriptivni jezik mora biti proširiv na način da omogućava uključivanje podrške za nove programske razvojne okoline i jezike, kao i za nove grafičke elemente (kontrole).

Temeljem analize i istraživanja problemske domene definirani su ciljevi rada i hipoteze. U skladu s ciljevima definiranim u poglavlju 1.1. postignuto je sljedeće:

1. Definirani su temeljni problemi i zahtjevi osoba oštećena vida u domeni klasičnog stolnog programiranja.
2. Istražena su i prikazana aktualna nastojanja u području prikaza grafičkog sadržaja osobama oštećena vida i rješavanja problema programera oštećena vida u domeni razvoja grafičkih sučelja.
3. Istraženi su i definirani mogući pristupi rješavanja problema razvoja grafičkih sučelja za osobe oštećena vida.
4. Izrađen je konceptualni model temeljem kojeg je moguće razviti rješenje koje bi omogućilo programerima oštećena vida razvoj GUI sučelja u domeni klasičnog stolnog programiranja na jednostavan i prihvatljiv način.
5. Razvijen je novi deskriptivni jezik GUIDL kao centralni dio osmišljenog modela.
6. Razvijen je cjelokupni GUIDL sustav temeljem osmišljenog modela koji omogućava dobivanje izvornog koda ciljanih programskih jezika.

7. GUIDL sustav je testiran i provjeren je stupanj njegove uporabljivosti na pojedinim zahtjevima iz prve točke te mjera u kojoj sustav rješava cjelokupni zadani problem razvoja GUI sučelja za programere oštećena vida.

S obzirom na hipoteze definirane u poglavlju 1.2. utvrđeno je sljedeće:

- Postoji realna potreba programera oštećena vida za rješenjem problema razvoja grafičkih sučelja što je utvrđeno analizom referentne literature i znanstvenih članaka, kao i vlastitim istraživanjem.
- Razvijeno je adekvatno rješenje koje osobama oštećena vida omogućava razvoj grafičkih sučelja. Navedena hipoteza je dokazana razvojem novog deskriptivnog jezika GUIDL i cjelokupnog GUIDL sustava temeljem utvrđenih zahtjeva za ovakvim sustavom te testiranjem razvijenog prototipa sustava na populaciji programera oštećena vida, čime je potvrđena njegova uporabljivost i adekvatnost.

Znanstveni doprinos rada se sastoji u:

1. Identificiranoj potrebi, problemima i zahtjevima programera oštećena vida u području razvoja grafičkih sučelja.
2. Definiranim mogućim pristupima rješavanju problema razvoja grafičkih sučelja za osobe oštećena vida i odabiru adekvatnog pristupa kao generalnog rješenja.
3. Izrađenom konceptualnom modelu temeljem kojeg je moguće razviti rješenje koje omogućava programerima oštećena vida razvoj GUI sučelja u klasičnom stolnom programiranju na jednostavan i prihvatljiv način.
4. Razvijenom deskriptivnom jeziku pod nazivom GUIDL kao centralnom dijelu osmišljenog modela i sustava te razvijenom cjelokupnom GUIDL sustavu temeljem osmišljenog modela koji omogućava dobivanje izvornog koda ciljanih programskih jezika.
5. Potvrdi uporabljivosti GUIDL sustava u populaciji programera oštećena vida za potrebe kreiranja grafičkih sučelja u domeni klasičnog stolnog programiranja.

Može se zaključiti da su u okviru ovog rada provedena istraživanja s ciljem definiranja potrebe i zahtjeva programera oštećena vida za novim konceptom rješenja problema razvoja

grafičkih sučelja. Istraženi su postojeći pristupi i napori na području prikaza grafičkih elemenata osobama oštećena vida i definirani mogući pristupi koji bi doveli do adekvatnog rješenja problema razvoja grafičkih sučelja za programere oštećena vida. Temeljem dobivenih spoznaja o mogućim pristupima, odabran je onaj koji bi mogao dovesti do adekvatnog rješenja te je izrađen konceptualni model takvog rješenja.

Temeljem konceptualnog modela razvijen je novi deskriptivni jezik pod nazivom GUIDL (Graphical User Interface Description Language) kao centralni dio sustava za dobivanje izvornog koda ciljanih programskih jezika kojim su opisana grafička sučelja u pojedinoj tehnologiji. Provedeno je testiranje razvijenog prototipa rješenja kako interno, tako i na odabranoj populaciji osoba oštećena vida. Rezultati testiranja su pokazali da je novorazvijeni koncept i prototip adekvatan za osobe oštećena vida u području razvoja grafičkih sučelja koja su uporabljiva u okviru aktualnih programskih tehnologija. Samim tim je postignut osnovni cilj ovog rada u obliku znanstvenog doprinosa koji se temelji na razvoju novog konceptualnog modela temeljem kojeg su razvijeni novi deskriptivni jezik i sustav koji omogućava osobama oštećena vida da na prihvatljiv i njima dovoljno jednostavan i zoran način razviju grafička sučelja, koja će moći uključiti u aktualne programske tehnologije i razvojne okoline te da se na taj način aktualiziraju sa svojim idejama kao ravnopravni članovi razvojnog tima i u području oblikovanja grafičkih sučelja što s postojećih pristupima i rješenjima nisu bili u stanju učiniti na njima dovoljno prihvatljiv način.

LITERATURA

Abrams, M., Phanouriou, C., Batongbacal, A. L., Williams S. M., Shuster, J. E.: UIML: An Appliance-Independent XML User Interface Language. Computer Networks, 31 (11-16): 1695-1708, 1999.

Alexander, S.: Blind Programmers Face An UncertainFuture. ComputerWorld, p. 1, 1998.

Alty, J., Rigas, D.: Communicating Graphical Information to Blind Users Using Music: The Role of Context. In CHI '98 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 574–81, ACM Press/Addison-Wesley Publishing Co., 1998.

Barnes, C.: Disabled People in Britain and Discrimination: a Case for Anti-discrimination Legislation. Hurst & Co., London, 1994.

Beckie, T. M., Hayduk, L. A.: Measuring Quality of Life, Social Indicators Research, vol. 42, pp. 21–39, 1997.

Bellik, Y., Burger, D.: Multimodal Interfaces: New Solutions to the Problem of Computer Accessibility for the Blind. In CHI '94 Conference Companion on Human Factors in Computing Systems, pp. 24–28, ACM Press, 1994.

Benjak, T: Izvješće o osobama s invaliditetom u Republici Hrvatskoj, Zagreb, 2013.

Bowling, A.: Measuring Disease: A Review of Disease Specific Quality of Life Measurement Scales. Open University Press, Buckingham, UK, pp. 1–16, 1995.

Bowling, A., Windsor, J.: Towards the Good Life: A Population Survey of Dimensions of Quality of Life. Journal of Happiness Studies, vol. 2, pp. 55–81, 1999.

Braz, L. M.: Visual Syntax Diagrams for Programming Language Statements. In ACM 8th International Conference on Systems Documentation, pp. 23–27, Visual Issues in Technical Communication, 1990.

Browne, J. P., O'Boyle, C. A., McGee, H. M., Joyce, C. R., McDonald, N. J., O'Malley, K., Hiltbrunner, B.: Individual Quality of Life in the Healthy Elderly. Qual. Life Res., vol. 3, pp. 235–44, 1994.

Bruce, I. W., McKennell, A. C., Walker, E. C.: Blind and Partially Sighted Adults in Britain: the RNIB Survey, vol. 1, HMSO, London, 1991.

Calman, K. C.: Quality of life in cancer patients - an hypothesis. J. Med. Ethics, vol. 10, no. 3, pp. 124–127, 1984.

Chomsky, N.: Some Methodological Remarks on Generative Grammar. Word, 17: 219-239, 1961.

Chomsky, N.: Aspects of the Theory of Syntax. Cambridge, Massachussetts: MIT Press, 1965.

Chomsky, N.: Topics in the Theory of Generative Grammar. The Hague: Mouton, 1966.

Chomsky, N.: Syntactic Structures, Second Edition. The Hague: Mouton, 2002.

Creswell, J. W.: Research Design: A Qualitative, Quantitative and Mixed Method Approaches (2nd ed.). Thousand Oaks, CA: Sage, 2003.

Damodaran, L.: User involvement in the systems design process - a practical guide for users. Behaviour and Information Technology, vol. 15, no. 6, pp. 363–377, 1996.

Day, H., Jutai, J.: Measuring the Psychosocial Impact of Assistive Devices; the PIADS. Canadian J. Rehabilitation, vol. 9, no. 2, pp. 159–168, 1996.

Day, H., Jutai, J., Campbell, K. A.: Development of a Scale to Measure the Psychosocial Impact of Assistive Devices: Lessons Learned and the Road Ahead. Disability and Rehabilitation, vol. 24, no. 1/2/3, pp. 31–37, 2002.

Demers, L., Weiss-Lambrou, R., Ska, B.: The Quebec User Evaluation of Satisfaction with Assistive Technology (QUEST 2.0): An overview and recent progress. Technology and Disability, vol. 14, pp. 101–105, 2002.

DeRuyter, F.: The importance of outcome measures for assistive technology service delivery systems. Technology and Disability, vol. 6, pp. 89–104, 1997.

Doward, L. C., McKenna, S. P.: Defining patient-reported outcomes. Value Health, vol. 7, Supplement 1, pp. S4–S8, 2004.

Fouygerollas, P., Noreau, L., Bergeron, H., Cloutier, R., Dion, S. A., St-Michel, G.: Social consequences of long-term impairments and disabilities: conceptual approach and assessment of handicap. Int. J. Rehabil. Res. vol. 21, no. 2, pp. 127–142, 1998.

Fowler, M., Scott, K.: UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd edition). Object Technology Series, Addison-Wesley, 2003.

Francioni, J. M., Smith, A. C.: Computer Science Accessibility for Students with Visual Disabilities. In Proceedings of 33rd SIGCSE Technical Symposium on Computer Science Education, Cincinnati, Kentucky, 2002.

Franqueiro, K. G., Siegfried, R. M.: Designing a Scripting Language to Help the Blind Program Visually. In ACM Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 241-242, ACM Press, 2006.

Fuhrer, M. J.: Subjectifying quality of life as a medical rehabilitation outcome. Disability and Rehabilitation, vol. 22, no. 11, pp. 481–489, 2000.

Fuhrer, M. J., Jutai, J. W., Scherer,M. J., DeRuyter, F.: A framework for the conceptual modelling of assistive technology device outcomes. Disability and Rehabilitation, vol. 25, no. 22, pp. 1243–1251, 2003.

Gabbrielli, M., Martini, S.: Programming Languages: Principles and Paradigms. Undergraduate Topics in Computer Science, Springer, 2010.

Galitz, W.O.: The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques (3rd Edition). Wiley, 2007.

Gelderblom, G. J., de Witte, L. P.: The Assessment of Assistive Technology, Outcomes, Effects and Costs. Technology and Disability, vol. 14, pp. 91–94, 2002.

Guyton, A. C., Hall J. E.: Medicinska fiziologija. Medicinska naklada, Zagreb, 2012.

Hersh, M. A., Johnson, M. A: Assistive Technology for Visually Impaired and Blind People. Springer, 2008.

Hickey, A. M., Bury, G., O'Boyle, C. A., Bradley, F., O'Kelly, F. D., Shannon, W.: A new short form individual quality of life measure (SEIQoL-DW). BMJ, vol. 313, pp. 29–33, 1996.

Hodson, B.: Sixties Ushers in Program To Train Blind Programmers. Computer World Canada, 2004.

Hughes, C., Heang, B., Kim, J., Eisenman, L. T., Killian, D. J.: Quality of life in applied research: A review and analysis of empirical measures. Am. J. on Mental Retardation. vol. 99, no. 6, pp. 623–641, 1995.

Jutai, J., Day, H.: Psychosocial Impact of Assistive Devices Scale (PIADS). Technology and Disability, vol. 14, pp. 107–111, 2002.

Kallmeyer, L.: Parsing Beyond Context-Free Grammars. Springer, 2010.

Kamel, H. M., Landay, J. A.: A Study of Blind Drawing Practice: Creating Graphical Information Without the Visual Channel. In Assets '00 Proceedings of the Fourth International ACM Conference on Assistive Technologies. ACM Press, 2000.

Kim, J. O., Curry, J.: The Treatment of Missing Data in Multivariable Analysis. Social. Meth. Res., vol. 6, pp. 215–240, 1977.

Kiš, M.: Englesko-hrvatski i hrvatsko-engleski informatički rječnik. Naklada Ljevak, Zagreb, p. 399, 2000.

Konecki, M., Kudelić, R., Radošević, D.: Challenges of the blind programmers. Proceedings of the 21st Central European Conference on Information and Intelligent Systems, pp. 473–476, 2010.

Konecki, M., Lovrenčić, A., Kudelić, R.: Making Programming Accessible to the Blinds. Proceedings of the 34th International Convention on Computers in Technical Systems, Croatian Society for Information and Communication Technology, Electronics and Microelectronics - MIPRO, pp. 820–824, 2011.

Konecki, M.: A New Approach Towards Visual Programming for the Blinds. Proceedings of the 35th International Convention on Information and Communication Technology, Electronics and Microelectronics - MIPRO, pp. 935–940, 2012.

Lazar, J., Dudley-Sponaugle, A., Greenidge, K.: Improving Web Accessibility: A Study of Webmaster Perceptions. Computers and Human Behavior, 20 (2): 269-288, 2004.

Lewis, H. R., Papadimitriou, C.: Elements of the Theory of Computation. Prentice Hall, Inc., 1981.

Manduchi, R., Kurniawan, S.: Assistive Technology for Blindness and Low Vision. CRC Press, New York, USA, 2012.

Maslow, A.: Toward a Psychology of Being, 2nd Edition. Van Nostrand, New York, 1968.

Mountain, L. A., Campbell, S. E., Seymour, D. G., Primrose, W. R., Whyte, M. I.: Assessment of individual quality of life using the SEIQoL-DWin older Patients. Q. J. Med. vol. 97, pp. 519–524, 2004.

Mršić, V.: Orijentacija i mobilitet u Hrvatskoj, Hrvatska udruga za školovanje pasa vodiča i mobilitet, Zagreb, 1995.

Noreau, L., Fougeyrollas, P., Vincent, C.: The LIFE-H: Assessment of the quality of social participation. Technology and Disability, vol. 14, pp. 113–118, 2002.

O'Boyle, C. A., McGee, H., Hickey, A., O'Malley, K., Joyce, C. R. B.: Individual quality of life in patients undergoing hip replacement. Lancet, vol. 339, pp. 1088–91, 1992.

O'Boyle C. A., McGee, H., Hickey, A., Joyce, C. R. B., Brown, J., O'Malley, K.: The Schedule for the Evaluation of Individual Quality of Life. User manual, Department of Psychology, Royal College of Surgeons in Ireland, Dublin, 1993.

Pascual, J., Massó, M., Vanderdonckt, J., Simarro, F. M., López, P. G.: Towards Virtualization of User Interfaces based on UsiXML. Proceedings of the 10th International Conference on 3D Web Technology, pp. 169–178, 2005.

Phillips, B., Zhao, H.: Predictors of Assistive Technology Abandonment. Assistive Technology, vol. 5, no. 1, pp. 36–45, 1993.

Pitt, I. J., Edwards, A. D. N.: Improving the usability of speech-based interfaces for blind users. In Proceedings of the Second Annual ACM Conference on Assistive Technologies (ASSETS), pp. 124-130 , Vancouver, Canada, 1996.

Pritchett, B. L.: Gramatical competence and parsing performance. The University of Chicago press, 1992.

Riccobono, M. A.: A Brighter Future for Blind Children. Braille Monitor, 2004.

Rosmaita, B. J.: Accessibility First!: A New Approach to Web Design. In ACM Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, pp. 270-274, ACM Press, 2006.

Rotard, M., Taras, C., Ertl, T.: Tactile web browsing for blind people. Multimedia Tools Appl., Springer, 37 (1): 53–69, 2008.

Rowley, J.: Towards a methodology for the design of multimedia public access interfaces. Journal of Information Science, vol. 24, no. 3, pp. 155–166, 1998.

Sanchez, J., Aguayo, F.: Blind Learners Programming Through Audio. In ACM CHI '05 Extended Abstracts on Human Factors in Computing Systems, pp. 1769-1772, ACM Press, 2005.

Sawicki, D. S.: Improving Community Indicator Systems: Injecting More Social Science into the Folk Movement. Planning Theory and Practice, vol. 3, no. 1, pp. 13–32, 2002.

Scherer, M. J.: Living in the State of Stuck: How Technology Impacts the Lives of People with Disabilities, 3rd. Edition. Broodline Books, Cambridge, MA, USA, 2000.

Scherer, M. J., Cushman, L.: Predicting satisfaction with assistive technology for a sample of adults with new spinal cord injuries. Psychological Reports, vol. 87, pp. 981–987, 2000.

Scherer, M. J., Craddock, G.: Matching Person and Technology (MPT) assessment process. Technology and Disability, vol. 14, no. 3, pp. 125–131, 2002.

Searle, J. R.: Chomsky's Revolution in Linguistics. The New York Review of Books, 1972.

Siegfried, R. M.: Visual Programming and the Blind: The Challenge and the Opportunity. In SIGCSE '06 Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, pp. 275-278, ACM Press, 2006.

Simon, S. E., Patrick, A.: Understanding and assessing consumer satisfaction in rehabilitation. Journal of Rehabilitation Outcomes Measurement, vol. 1, pp. 1–14, 1997.

Smith, R. O.: Measuring the outcomes of assistive technology: challenge and innovation. Assistive Technology, vol. 8, pp. 71–81, 1996.

Smith, A. C., Francioni, J. M., Matzek, S. D.: A Java Programming Tool for Students with Visual Disabilities. In Proceeding Assets '00 Proceedings of the 4th International ACM Conference on Assistive Technologies, ACM Press, 2000.

Smith, R. O.: OTFACT: Multilevel performance-oriented software with an assistive technology outcomes assessment protocol. Technology and Disability, vol. 14, pp. 133–139, 2002.

Spencer, K. L.: Assessing the Accessibility for the Blind and Visually Impaired of Texas State Agency Web Sites. Texas State University-San Marcos, Political Science Department, Public Administration, Texas State University, 2001.

Sullivan, T., Matson, R.: Barriers to Use: Usability and Content Accessibility on the Web’s Most Popular Sites. In Proceedings of the ACM Conference on Universal Usability, pp. 139-144, ACM Press, 2000.

Swain, J., French, S., Cameron, C.: Controversial Issues in a Disabling Society. Open University Press, Buckingham, UK, 2003.

Taylor, H. R., Keeffe, J. E.: World blindness: a 21st century perspective. Br. J. of Ophthalmol., vol. 85, no. 3, pp. 261–6, 2001.

Walker, E. C., Tobin, M. H. J., McKennell, A. C.: Blind and partially sighted children in Britain: the RNIB Survey, vol. 2, HMSO, London, 1992.

Wessels, R., Persson, J., Lorentsen, O., Andrich, R., Farrario, M., Oortwijn, W., VanBeekum, T., Brodin, H., de Witte, L.: IPPA; Individually Prioritised Problem Assessment. Technology and Disability, vol. 14, pp. 141–145, 2002.

Zhang, M., Weisser, V. D., Stilla, R., Prather, S. C., Sathian, K.: Multisensory cortical processing of object shape and its relation to mental imagery. Cognitive, Affective and Behavioral Neuroscience, vol. 4, no. 2, pp. 251–259, 2004.

Zovko, G.: Peripatologija 1. Školske novine, Zagreb, 1994.

Zovko, G.: Peripatologija 2. Školske novine, Zagreb, 1998.

Žugaj, M., Dumičić, K., Dušak, V.: Temelji znanstvenoistraživačkog rada, Metodologija i metodika. TIVA i FOI, Varaždin, 2006.

Information technology – Syntactic metalanguage – Extended BNF. ISO/IEC 14977. Geneva, Switzerland: ISO, 1996.

WHO, 1997, Global Initiative for the Prevention of Avoidable Blindness. WHO/PBL/97.61, WorldHealth Organization, Geneva

WHO, 1998, The World Health report – Life in the 21st century: a vision for all. World Health Organization, Geneva, WHO, 1998;47

WHO, 2001, International Classification of Functioning, Disability and Health. World Health Organisation, Geneva, Switzerland

UPIAS, 1976, Fundamental Principles of Disability. London: UPIAS (Union of the Physically Impaired Against Segregation)

Blind Programming Project in NetBeans IDE, http://netbeans.dzone.com/videos/netbeans-ide-for-the-blind/, 2010, preuzeto: 10.2.2011.

Screen Reader COBRA, http://www.baum.de/cms/en/cobra/, preuzeto: 10.3.2011.

GW Micro - Window-Eyes, http://www.gwmicro.com/Window-Eyes/, preuzeto: 25.3.2011.

Easy Web Browsing, http://www-03.ibm.com/able/accessibility_services/EasyWebBrowsing.html, preuzeto: 25.3.2011.

program-l: V.I. Programmers Discussion List, http://www.freelists.org/list/program-l, preuzeto: 30.3.2011.

programmingblind: Programming Blind, http://www.freelists.org/list/programmingblind, preuzeto: 30.3.2011.

http://BlindGeeks.org, preuzeto: 30.3.2011.

JAWSScripts, http://www.freelists.org/list/jawsscripts, preuzeto: 30.3.2011.

Cattani, R.: The employment of blind and partially-sighted persons in Italy: A challenging issue in a changing economy and society, http://www.euroblind.org/media/employment/employment_Italy.doc, 2009, preuzeto: 25.3.2011.

bfi Steiermark's European Labour Market Report, http://eurochance.brailcom.org/download/labour-market-report.pdf, 2004, preuzeto: 25.3.2011.

SuperNova Screen Reader, http://www.arante.pl/hal.html, preuzeto: 2.4.2011.

Configuring Your Machine For Testing With A Screen Reader, http://www.yuiblog.com/blog/2008/12/, 2008, preuzeto: 2.4.2011.

10 Facts About Blindness and Visual Impairement, http://www.who.int/features/factfiles/blindness/blindness_facts/en/index.html, preuzeto: 2.5.2013.

Pedijatrijski materijal za vizualnu dijagnostiku i istraživanje, http://www.zzjzpgz.hr/nzl/4/dodatak.htm, preuzeto: 5.5.2013.

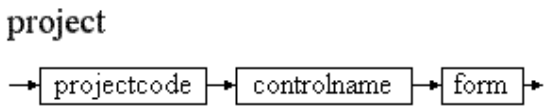
Blindness: Vision 2020 - The Global Initiative for the Elimination of Avoidable Blindness, http://www.who.int/mediacentre/factsheets/fs213/en/, 1999, preuzeto: 5.5.2011.

PRILOZI

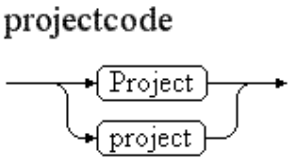
Prikaz gramatike GUIDL jezika putem sintaktičkih dijagrama i

PRILOG A: PRIKAZ GRAMATIKE GUIDL JEZIKA PUTEM SINTAKTIČKIH DIJAGRAMA

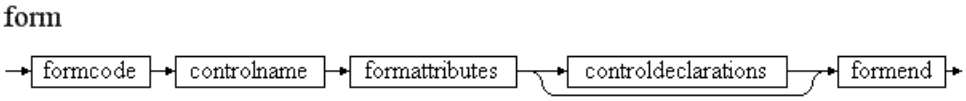
Tablica A.1 Sintaktički dijagrami gramatike GUIDL jezika



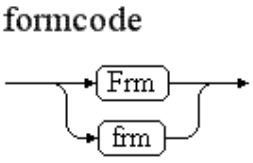
Slika A.1 Sintaktički dijagram produkcijskog pravila project



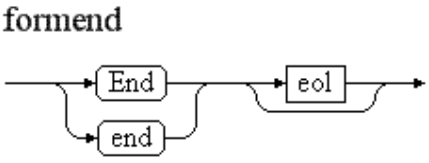
Slika A.2 Sintaktički dijagram produkcijskog pravila projectcode



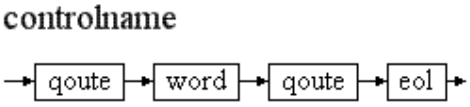
Slika A.3 Sintaktički dijagram produkcijskog pravila form



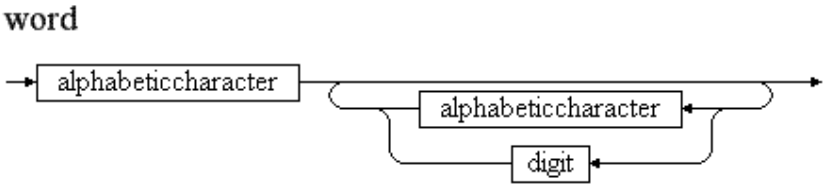
Slika A.4 Sintaktički dijagram produkcijskog pravila formcode



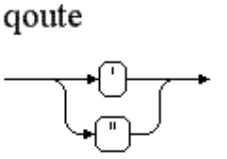
Slika A.5 Sintaktički dijagram produkcijskog pravila formend



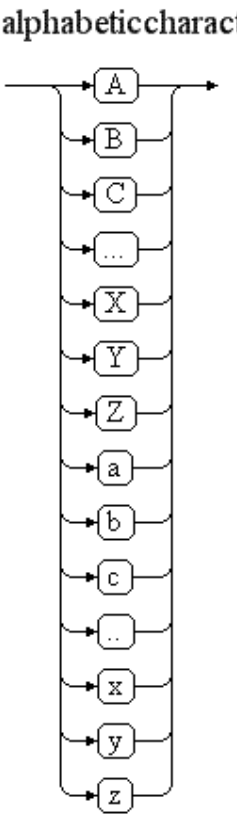
Slika A.6 Sintaktički dijagram produkcijskog pravila controlname



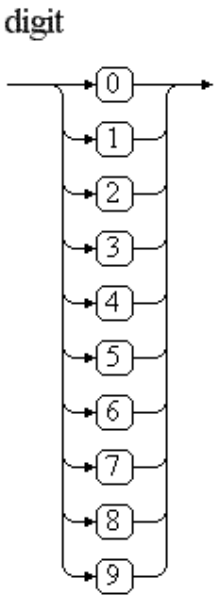
Slika A.7 Sintaktički dijagram produkcijskog pravila word



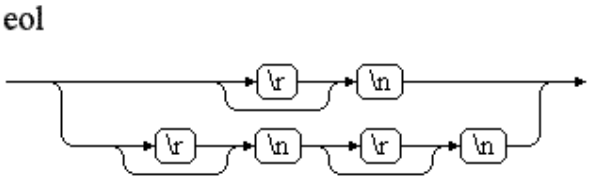
Slika A.8 Sintaktički dijagram produkcijskog pravila quote



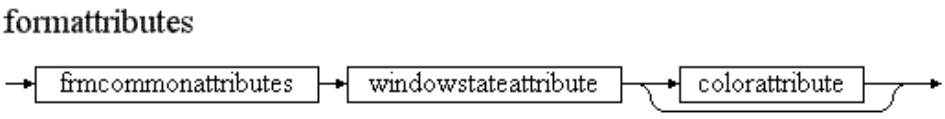
Slika A.9 Sintaktički dijagram produkcijskog pravila alphabeticcharacter



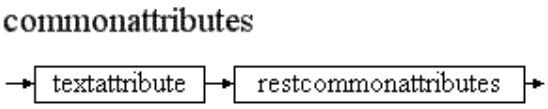
Slika A.10 Sintaktički dijagram produkcijskog pravila digit



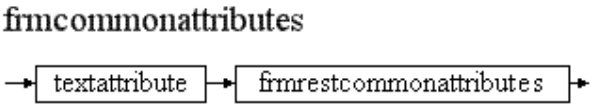
Slika A.11 Sintaktički dijagram produkcijskog pravila eol



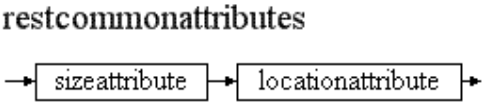
Slika A.12 Sintaktički dijagram produkcijskog pravila formattributes



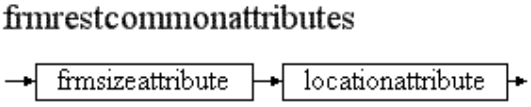
Slika A.13 Sintaktički dijagram produkcijskog pravila commonattributes



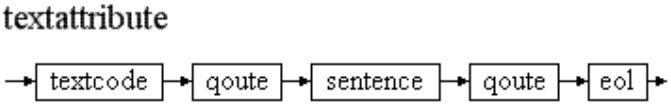
Slika A.14 Sintaktički dijagram produkcijskog pravila frmcommonattributes



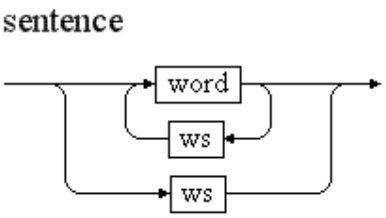
Slika A.15 Sintaktički dijagram produkcijskog pravila restcommonattributes



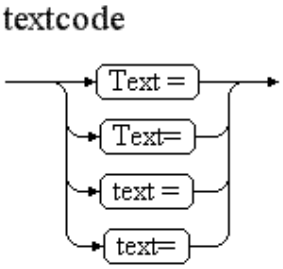
Slika A.16 Sintaktički dijagram produkcijskog pravila frmrestcommonattributes



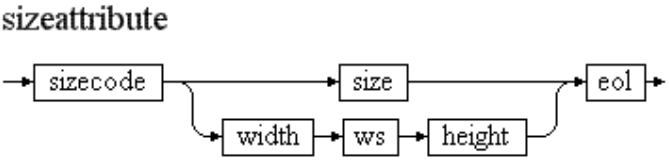
Slika A.17 Sintaktički dijagram produkcijskog pravila textattribute



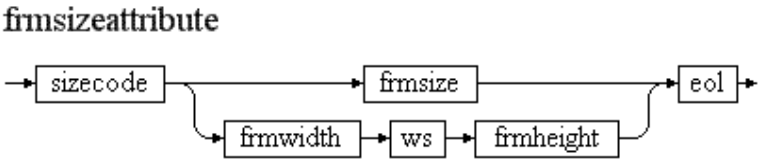
Slika A.18 Sintaktički dijagram produkcijskog pravila sentence



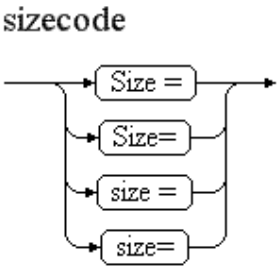
Slika A.19 Sintaktički dijagram produkcijskog pravila textcode



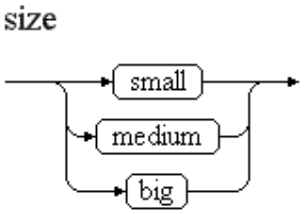
Slika A.20 Sintaktički dijagram produkcijskog pravila sizeattribute



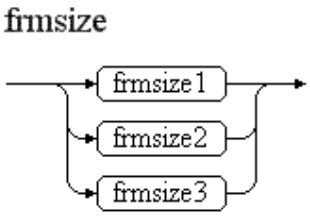
Slika A.21 Sintaktički dijagram produkcijskog pravila frmsizeattribute



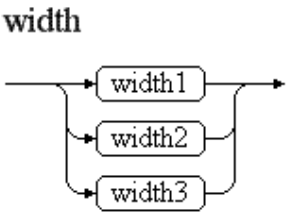
Slika A.22 Sintaktički dijagram produkcijskog pravila sizecode



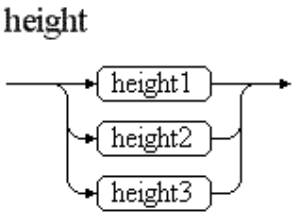
Slika A.23 Sintaktički dijagram produkcijskog pravila size



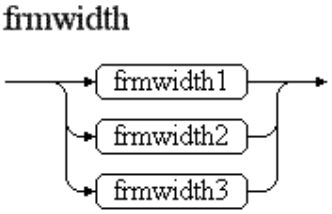
Slika A.24 Sintaktički dijagram produkcijskog pravila frmsize



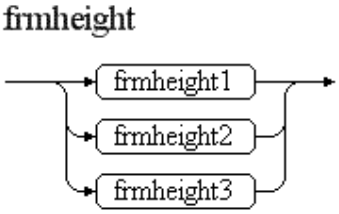
Slika A.25 Sintaktički dijagram produkcijskog pravila width



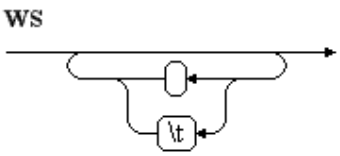
Slika A.26 Sintaktički dijagram produkcijskog pravila height



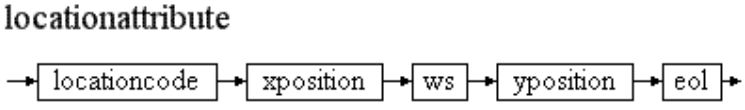
Slika A.27 Sintaktički dijagram produkcijskog pravila frmwidth



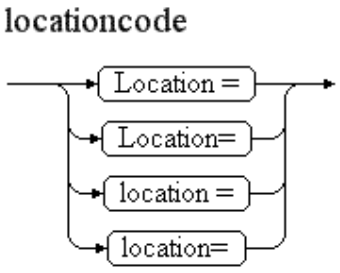
Slika A.28 Sintaktički dijagram produkcijskog pravila frmheight



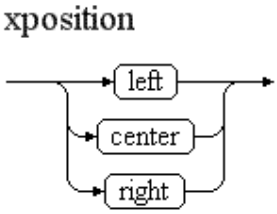
Slika A.29 Sintaktički dijagram produkcijskog pravila ws



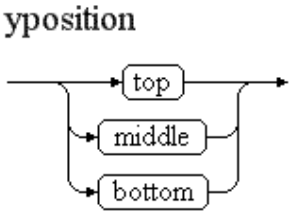
Slika A.30 Sintaktički dijagram produkcijskog pravila locationattribute



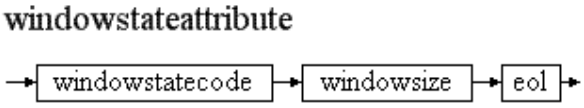
Slika A.31 Sintaktički dijagram produkcijskog pravila locationcode



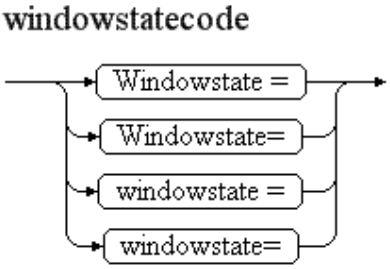
Slika A.32 Sintaktički dijagram produkcijskog pravila xposition



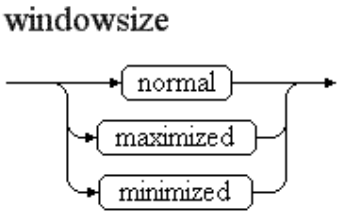
Slika A.33 Sintaktički dijagram produkcijskog pravila yposition



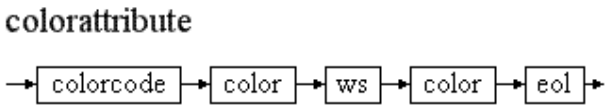
Slika A.34 Sintaktički dijagram produkcijskog pravila windowstateattribute



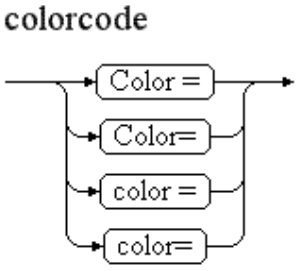
Slika A.35 Sintaktički dijagram produkcijskog pravila windowstatecode



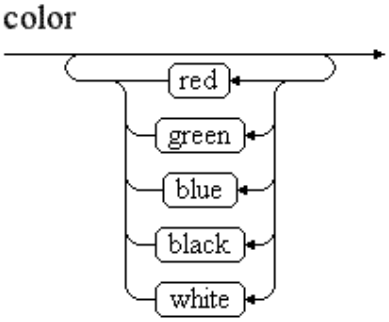
Slika A.36 Sintaktički dijagram produkcijskog pravila windowsize



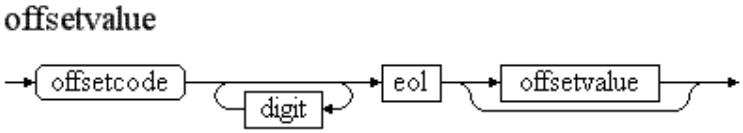
Slika A.37 Sintaktički dijagram produkcijskog pravila colorattribute



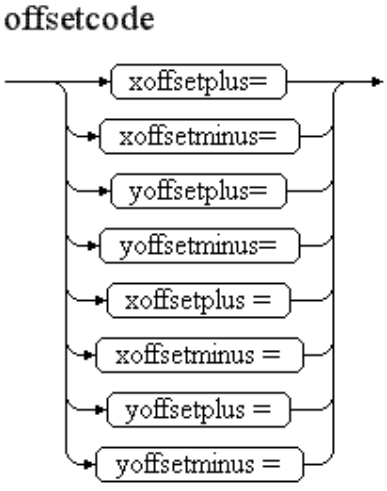
Slika A.38 Sintaktički dijagram produkcijskog pravila colorcode



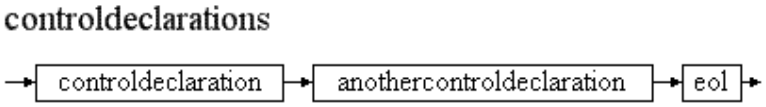
Slika A.39 Sintaktički dijagram produkcijskog pravila color



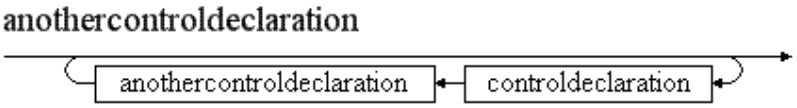
Slika A.40 Sintaktički dijagram produkcijskog pravila offsetvalue



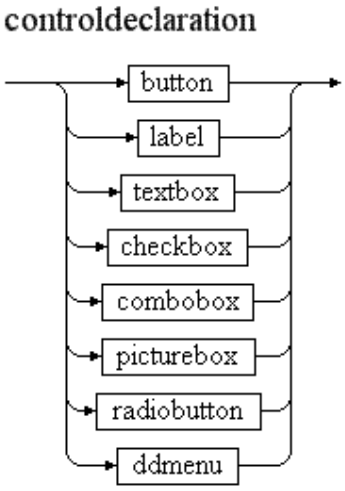
Slika A.41 Sintaktički dijagram produkcijskog pravila offsetcode



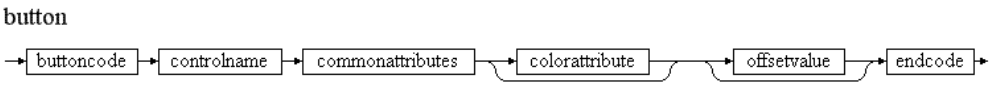
Slika A.42 Sintaktički dijagram produkcijskog pravila controldeclarations



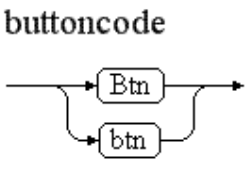
Slika A.43 Sintaktički dijagram produkcijskog pravila anothercontroldeclaration



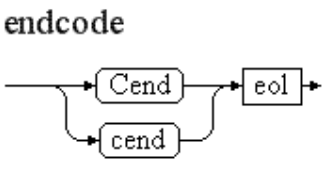
Slika A.44 Sintaktički dijagram produkcijskog pravila controldeclaration



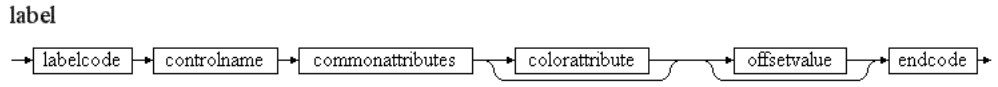
Slika A.45 Sintaktički dijagram produkcijskog pravila button



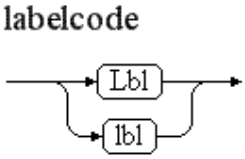
Slika A.46 Sintaktički dijagram produkcijskog pravila buttoncode



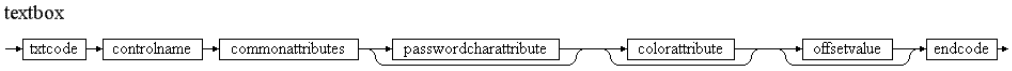
Slika A.47 Sintaktički dijagram produkcijskog pravila endcode



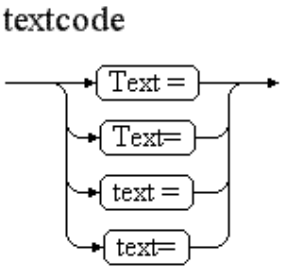
Slika A.48 Sintaktički dijagram produkcijskog pravila label



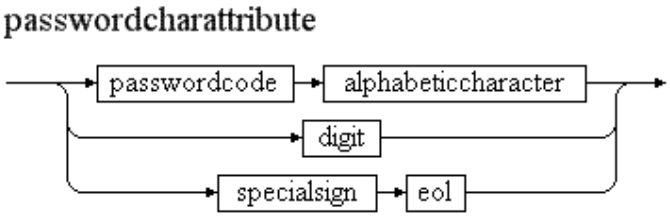
Slika A.49 Sintaktički dijagram produkcijskog pravila labelcode



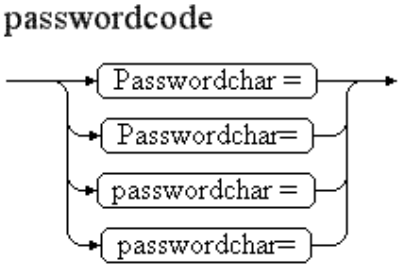
Slika A.50 Sintaktički dijagram produkcijskog pravila textbox



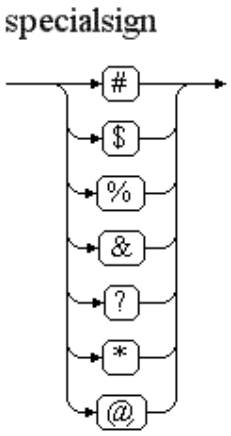
Slika A.51 Sintaktički dijagram produkcijskog pravila textcode



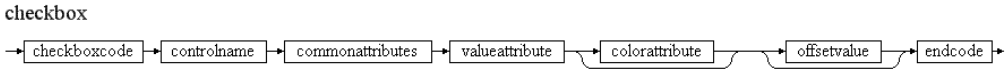
Slika A.52 Sintaktički dijagram produkcijskog pravila passwordcharattribute



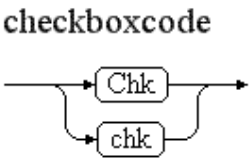
Slika A.53 Sintaktički dijagram produkcijskog pravila passwordcode



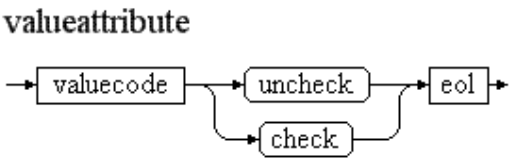
Slika A.54 Sintaktički dijagram produkcijskog pravila specialsign



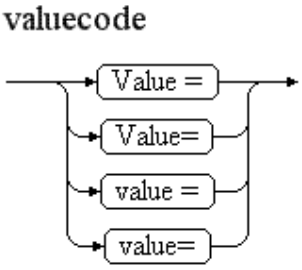
Slika A.55 Sintaktički dijagram produkcijskog pravila checkbox



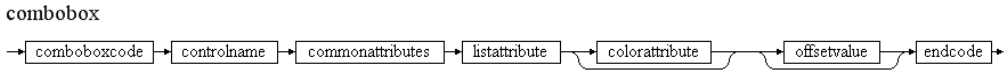
Slika A.56 Sintaktički dijagram produkcijskog pravila checkboxcode



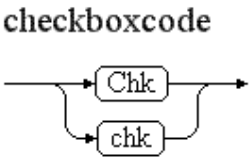
Slika A.57 Sintaktički dijagram produkcijskog pravila valueattribute



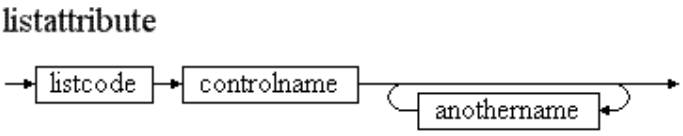
Slika A.58 Sintaktički dijagram produkcijskog pravila valuecode



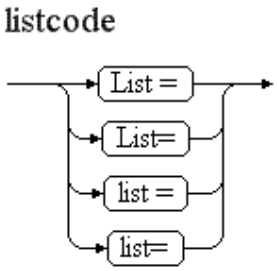
Slika A.59 Sintaktički dijagram produkcijskog pravila combobox



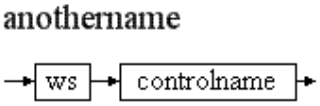
Slika A.60 Sintaktički dijagram produkcijskog pravila checkboxcode



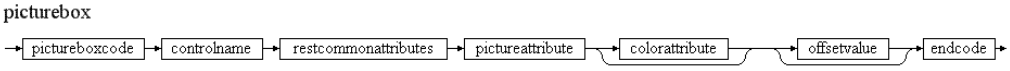
Slika A.61 Sintaktički dijagram produkcijskog pravila listattribute



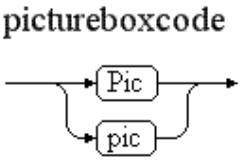
Slika A.62 Sintaktički dijagram produkcijskog pravila listcode



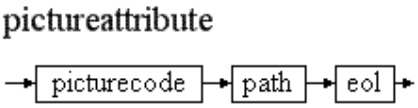
Slika A.63 Sintaktički dijagram produkcijskog pravila anothername



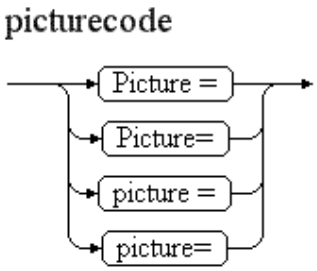
Slika A.64 Sintaktički dijagram produkcijskog pravila picturebox



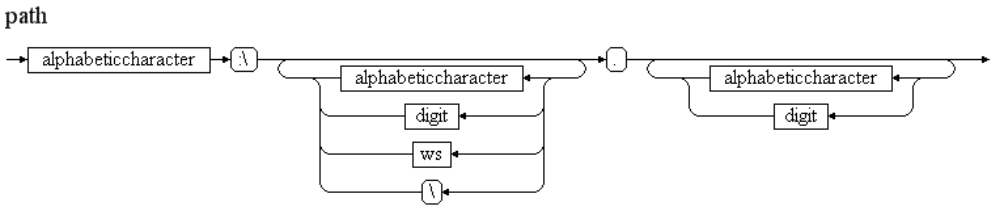
Slika A.65 Sintaktički dijagram produkcijskog pravila pictureboxcode



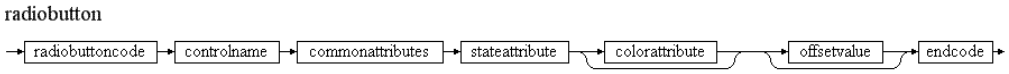
Slika A.66 Sintaktički dijagram produkcijskog pravila pictureattribute



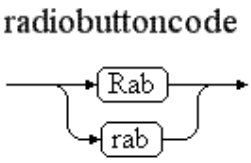
Slika A.67 Sintaktički dijagram produkcijskog pravila picturecode



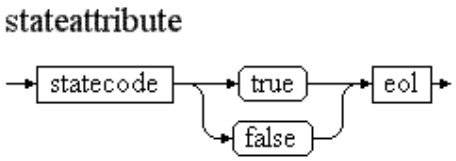
Slika A.68 Sintaktički dijagram produkcijskog pravila path



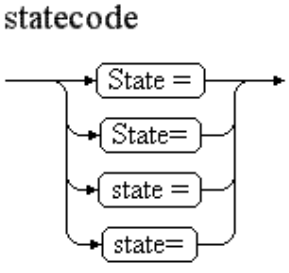
Slika A.69 Sintaktički dijagram produkcijskog pravila radiobutton



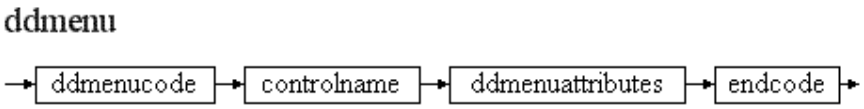
Slika A.70 Sintaktički dijagram produkcijskog pravila radiobuttoncode



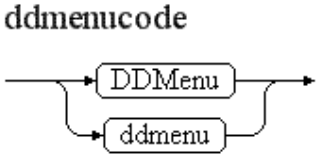
Slika A.71 Sintaktički dijagram produkcijskog pravila stateattribute



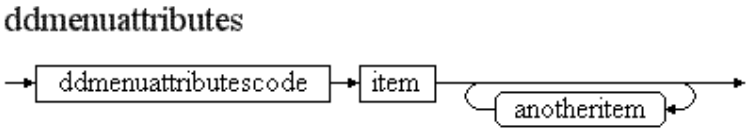
Slika A.72 Sintaktički dijagram produkcijskog pravila statecode



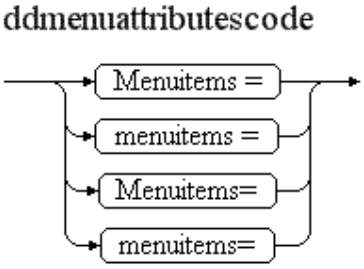
Slika A.73 Sintaktički dijagram produkcijskog pravila ddmenu



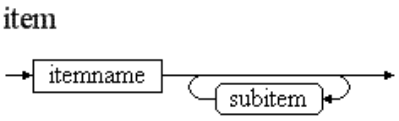
Slika A.74 Sintaktički dijagram produkcijskog pravila ddmenucode



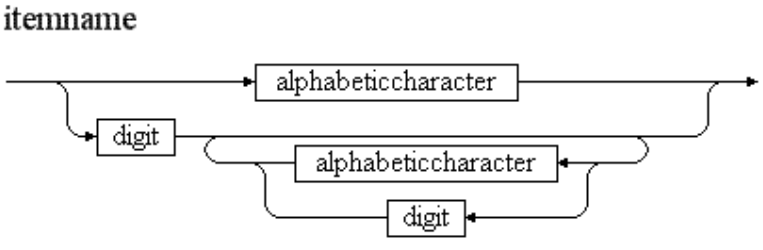
Slika A.75 Sintaktički dijagram produkcijskog pravila ddmenuattributes



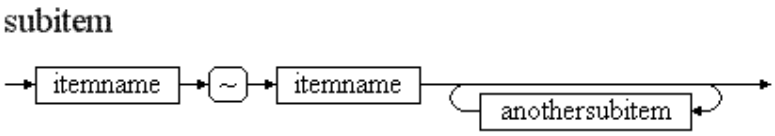
Slika A.76 Sintaktički dijagram produkcijskog pravila ddmenuattributescode



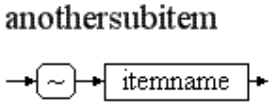
Slika A.77 Sintaktički dijagram produkcijskog pravila item



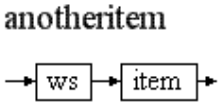
Slika A.78 Sintaktički dijagram produkcijskog pravila itemname



Slika A.79 Sintaktički dijagram produkcijskog pravila subitem



Slika A.80 Sintaktički dijagram produkcijskog pravila anothersubitem



Slika A.81 Sintaktički dijagram produkcijskog pravila anotheritem

ŽIVOTOPIS

Mario Konecki rođen je u Virovitici 3. kolovoza 1982. godine. Osnovnu školu je pohađao u mjestu prebivališta, Daruvaru (Češka osnovna škola Jana Amosa Komenskog) gdje je završio i srednju školu (Gimnazija Daruvar) s odličnim uspjehom. Diplomirao je 2005. godine na Fakultetu organizacije i informatike u Varaždinu, Sveučilišta u Zagrebu, smjer diplomirani informatičar, s temom "FOI LMS", pod mentorstvom prof. dr. sc. Alena Lovrenčića. Tijekom studija dva puta je dobio nagradu za najboljeg studenta, a u akademskoj godini 2002./2003. državnu stipendiju. Nakon završenog dodiplomskog studija 2006. godine zaposlio se kao asistent na Fakultetu organizacije i informatike u Varaždinu. U razdoblju od 2006. godine do danas sudjelovao je u izvođenju vježbi iz predmeta Programiranje I, Programiranje II, Programiranje i Dizajn programskih proizvoda. Autor je 18 znanstvenih i 7 stručnih radova. U razdoblju od 2008. do 2012. godine sudjelovao je u radu na projektu Automatizacija postupaka projektiranja informacijskih sustava (voditelj prof. dr. sc. Alen Lovrenčić) Ministarstva znanosti, obrazovanja i sporta Republike Hrvatske. Služi se engleskim, češkim i njemačkim jezikom.

POPIS RADOVA

Znanstveni radovi

1. **Konecki, M.** (2012): A New Approach Towards Visual Programming for the Blinds, Proceedings of the 35th MIPRO International Convention on Computers in Technical Systems, pp. 1076-1081. IEEE, Opatija
2. **Konecki, M.**, Lovrenčić, A., Kudelić, R. (2011): Making Programming Accessible to the Blinds, Proceedings of the 34th MIPRO International Convention on Computers in Technical Systems, pp. 180-184. IEEE, Opatija
3. **Konecki, M.**, Kudelić, R., Radošević, D. (2010): Challenges of the blind programmers, Proceedings of the 21st Central European Conference on Information and Intelligent Systems, pp. 473-476. Faculty of Organization and Informatics, Varaždin
4. **Konecki, M.**, Orehovački, T., Kermek, D. (2009): Design Patterns – education and classification challenge, Proceedings of the 20th Central European Conference on

Information and Intelligent Systems, pp. 375-381. Faculty of Organization and Informatics, Varaždin

5. Lovrenčić, A., **Konecki, M.**, Orehovački, T. (2009): 1957-2007: 50 Years of Higher Order Programming Languages, Journal of Information and Organizational Sciences, vol. 33, no. 1, pp. 79-150.
6. Orehovački, T., Bubaš, G., **Konecki, M.** (2009): Web 2.0 in Education and Potential Factors of Web 2.0 Use by Students of Information Systems, Proceedings of the 31st International Conference on Information Technology Interfaces, pp. 443-448. IEEE, Cavtat
7. **Konecki, M.**, Orehovački, T., Lovrenčić, A. (2009): Detecting Computer Code Plagiarism in Higher Education, Proceedings of the 31st International Conference on Information Technology Interfaces, pp. 409-414. IEEE, Cavtat
8. **Konecki, M.**, Orehovački, T., Stapić, Z. (2008): IT Users' Awareness about the Need of Strong Passwords Creation, DAAAM International Scientific Book 2008, pp. 387-394. DAAAM International, Vienna
9. Bubaš, G., Orehovački, T., **Konecki, M.** (2008): Factors and Predictors of Online Security and Privacy Behavior, Journal of Information and Organizational Sciences, vol. 32, no. 2, pp. 79-98.
10. Radošević, D., **Konecki, M.**, Orehovački, T. (2008): Java Applications Development Based on Component and Metacomponent Approach, Journal of Information and Organizational Sciences, vol. 32, no. 2, pp. 137-147.
11. **Konecki, M.**, Orehovački, T., Lovrenčić, A. (2008): Detecting code re-use potential, Proceedings of the 19th Central European Conference on Information and Intelligent Systems, pp. 627-631. Faculty of Organization and Informatics, Varaždin
12. Orehovački, T., **Konecki, M.**, Radošević, D. (2008): Web 2.0 technologies in university education, Proceedings of the 31st MIPRO International Convention on Computers in Education, pp. 269-273. Croatian Society for Information and Communication Technology, Electronics and Microelectronics, Opatija
13. Orehovački, T., **Konecki, M.**, Bubaš, G. (2008): Security and Privacy Related Online Behavior of Experienced ICT Users, Proceedings of the 27th International Conference on

Organizational Science Development, pp. 1932-1940. Faculty of Organizational Sciences, Portorož

14. Roško, Z., **Konecki, M.** (2008): Dynamic Data Access Object Design Pattern, Proceedings of the 19th Central European Conference on Information and Intelligent Systems, pp. 615-620. Faculty of Organization and Informatics, Varaždin
15. Radošević, D., Orehovački, T., **Konecki, M.** (2007): Web oriented applications generator development through reengineering process, DAAAM International Scientific Book 2007, pp. 443-458. DAAAM International, Vienna
16. **Konecki, M.**, Hutinski, Ž., Orehovački, T. (2007): Secure web applications?, Proceedings of 30th MIPRO Jubilee International Convention on Information System Security, pp. 162-166. Croatian Society for Information and Communication Technology, Electronics and Microelectronics, Opatija
17. Radošević, D., Orehovački, T., **Konecki, M.** (2007): PHP Scripts Generator for Remote Database Administration based on C++ Generative Objects, Proceedings of the 30th MIPRO Jubilee International Convention on Intelligent Systems, pp. 167-171. Croatian Society for Information and Communication Technology, Electronics and Microelectronics, Opatija
18. Kermek, D., **Konecki, M.**, Vrček, N. (2006): Filtering, aggregation and information system model, Proceedings of the 17th IIS International Conference, pp. 99-104. Faculty of Organization and Informatics, Varaždin

Stručni radovi

1. **Konecki, M.**, Stapić, Z., Orehovački, T. (2008): Razvoj aplikacija korištenjem uzoraka dizajna, CASE 20 - metode i alati za razvoj poslovnih i informatičkih sustava, pp. 203-209., Opatija
2. Orehovački, T., **Konecki, M.**, Stapić, Z. (2008): Primjena Web 2.0 tehnologija u poslovanju, CASE 20 - metode i alati za razvoj poslovnih i informatičkih sustava, pp. 197-202., Opatija

3. Stapić, Z., Orehovački, T., **Konecki, M.** (2008): OCL kao preduvjet automatiziranom generiranju programskog koda, CASE 20 - metode i alati za razvoj poslovnih i informatičkih sustava, pp. 87-94., Opatija
4. **Konecki, M.**, Bubaš, G., Kermek, D. (2007): Web aplikacija za izradu plana rada iz nastavnog predmeta i planiranje pedagoških aktivnosti u lekciji, CASE 19 - metode i alati za razvoj poslovnih i informatičkih sustava, pp. 125-130., Opatija
5. Orehovački, T., **Konecki, M.**, Radošević, D. (2007): Alati za e-obrazovanje 2.0, 9. CARNetova korisnička konferencija (CUC), Rijeka
6. Orehovački, T., **Konecki, M.**, Radošević, D. (2007): Web 2.0 i evolucija e-obrazovanja, Stručno-znanstveni skup "E-obrazovanje", Zbornik radova, pp. 145-155., Varaždin
7. **Konecki M.**, Kermek, D., Vrček, N. (2006): Model sustava za informiranje, filtriranje i agregaciju poruka, CASE 18 - metode i alati za razvoj poslovnih i informatičkih sustava, pp. 143-147., Opatija