

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Gerkman

Avtomatizacija testiranja prenosa faks
sporočila preko IP omrežja

DIPLOMSKO DELO NA
VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: izr. prof. dr. Miha Mraz
Somentor: doc. dr. Iztok Lebar Bajec

Ljubljana, 2011



Št. naloge: 00088/2011

Datum: 01.04.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ALEŠ GERKMAN**

Naslov: **AVTOMATIZACIJA TESTIRANJA PRENOSA FAKS SPOROČILA
PREKO IP OMREŽJA**
**AUTOMATISATION OF TESTING OF THE TRANSMISSION OF A FAX
MESSAGE OVER THE IP NETWORK**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj v svojem delu predstavi postopek avtomatizacije testiranja prenosa faks sporočila preko IP omrežja. Pri tem naj uporabi strojno in programsko opremo, ki jo za tovrstne namene uporablja znani slovenski proizvajalec komunikacijske opreme.

Mentor:

prof. dr. Miha Mraz

Dekan:

prof. dr. Nikolaj Zimic

Somentor:

doc. dr. Iztok Lebar Bajec



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Aleš Gerkman,

z vpisno številko 63060069,

sem avtor/-ica diplomskega dela z naslovom:

Avtomatizacija testiranja prenosa faks sporočila preko IP omrežja

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom
izr. prof. dr. Mihe Mraza
in somentorstvom
doc. dr. Iztoka Lebarja Bajca
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne _____ Podpis avtorja/-ice: _____

Zahvala

Zahvaljujem se vsem, ki so mi kakorkoli pomagali pri izdelavi diplomskega dela. Posebna zahvala gre mentorju izr. prof. dr. Mihi Mrazu ter somentorju doc. dr. Iztoku Lebarju Bajcu, ki sta mi pomagala z nasveti, ter me usmerjala pri izdelavi diplomskega dela.

Zahvalil bi se tudi mojim domačim, ki so mi v času študija in pisanja diplomskega dela vedno stali ob strani, me podpirali in mi pomagali.

Vsebinsko kazalo

Povzetek	1
Summary	3
1 Uvod	5
2 Opis problema	9
2.1 Problemi pri prenosu faks sporočila	9
2.2 Zahtevnost testiranja	10
3 Opis uporabljenih pojmov, testiranega sistema in merilnih okolij	13
3.1 Uporabljeni pojmi, specifični v testiranem sistemu	13
3.1.1 Kodek	13
3.1.2 Kodek G.711 A-law	13
3.1.3 Kodek T.38	14
3.1.4 Paketizacija	14
3.1.5 Zadušitev tišine	14
3.1.6 Vmesnik »jitter buffer«	14
3.1.7 Bližnjica za zagon aplikacij	15
3.1.8 Aplikacija WBM	16
3.1.9 Izkoristek	16
3.2 Testirani sistem	16
3.2.1 Opis in postavitve testiranega sistema	17
3.2.2 Ročni test prenosa faks sporočila in pregled parametrov	19
3.3 Merilna okolja	21
3.3.1 iTest	21
3.3.2 IxVoice	23
4 Avtomatizacija postopka testiranja	25
4.1 iTest scenarij	25
4.2 Avtomatizacija gradnje vhodnih podatkov	27
4.3 Avtomatizacija aplikacije WBM za zajem bližnjice	29
4.4 Avtomatizacija nastavitve parametrov v aplikaciji klicni strežnik	32
4.4.1 Zagon aplikacije klicni strežnik s pomočjo skripte	32
4.4.2 Snemanje nastavljanja parametrov na klicnem strežniku	34
4.4.3 Gradnja map z odzivi za vhodne podatke	36

4.4.4	Uporaba map z odzivi na primeru	38
4.5	Gradnja scenarija pošiljanja faks sporočila.....	39
4.6	Avtomatizacija izvajanja testov z IxVoice	41
4.7	Izračunana primerjava časov med ročnim in avtomatiziranim testiranjem	44
4.7.1	Test številka 1	45
4.7.2	Test številka 2.....	46
4.7.3	Test številka 3	47
4.8	Rezultati testiranja	49
4.8.1	Pričakovani rezultati testov	49
4.8.2	Dejanski rezultati testov	49
4.8.3	Razlike med pričakovanimi in dejanskimi rezultati	52
5	Zaključek	53
6	Viri	55

Seznam uporabljenih kratic

- SAK (angl. *Subscriber Analog version K*): analogna naročniška plošča z 64 priključki, ki omogoča priklop v IP omrežje; verzija K omogoča pretvorbo signalov v IP pakete;
- IP (angl. *Internet Protocol*): protokol, ki se uporablja za prenos paketov oziroma podatkov preko interneta;
- ES (angl. *Ethernet Switch*): plošča, ki povezuje naročniške plošče med seboj, zna pa tudi usmerjati promet glede na telefonske številke naročnikov;
- SSW (angl. *Soft Switch*): strojna naprava, na kateri teče aplikacija klicni strežnik;
- CS (angl. *Call Server*): aplikacija klicni strežnik, ki nadzoruje in upravlja omrežne elemente;
- LA (angl. *Lan Access*): vrsta Iskratelovega produkta, ki omogoča priklop na IP omrežje;
- JNLP (angl. *Java Network Launching Protocol*): protokol za spletno poganjanje javanskih aplikacij;
- XML (angl. *Extensible Markup Language*): način zapisa strukturiranih podatkov;
- JAR (angl. *Java Archive*): uporablja se za shranjevanje in kompresijo javanskih programov in z njimi povezanih datotek;
- ECF (angl. *Execution Configuration File*): datoteka z nastavitvami parametrov izvajanja testa v programu IxVoice;
- CSV (angl. *Comma-Separated Values*): datoteka, v kateri so vrednosti običajno ločene z vejico;
- URL (angl. *Uniform Resource Locators*): naslov spletnih strani v svetovnem spletu;
- V.17: protokol za prenos podatkov s hitrostjo 14.4kbps;
- V.34: protokol za prenos podatkov s hitrostjo 33.6kbps;
- SMS (angl. *Short Message Service*): tekstovna komunikacijska storitev, ki se uporablja v telefoniji in tudi na spletu;

Povzetek

IP omrežje je s svojim prihodom prineslo veliko prednosti, a hkrati tudi veliko novih problemov. Eden od njih so analogne komunikacijske naprave, med katere sodi tudi faks, ki same ne znajo komunicirati preko IP omrežja. Za to potrebujejo napravo, ki pretvori analogne signale v IP pakete in obratno. Na ta način lahko še vedno uporabljamo analogne naprave v IP omrežju. Podjetje Iskratel d.o.o. se ukvarja z razvojem takšnih naprav.

Toda nekdo mora preizkusiti, če naprave za pretvorbo signalov delujejo pravilno ali ne. Naloga proizvajalca je, da preveri, ob kakšnih pogojih oziroma nastavitvah so prenosi podatkov najbolj uspešni. Nastavitev je ogromno, podjetje pa nima dovolj materialnih in človeških virov za izvedbo vseh testov.

Za svojo diplomsko nalogo sem izbral temo avtomatizacije testiranja prenosa faks sporočila preko IP omrežja. Z avtomatizacijo sem želel skrajšati čas testiranja, hkrati pa omogočiti izvajanje večjega števila različnih testov, kot je le-to možno ročno. Z avtomatizacijo sem želel razbremeniti delo tistim, ki opravljajo omenjene teste.

Ključne besede: avtomatizacija, faks, IP omrežje

Summary

IP network brought a lot of improvements with its advent, but also many new problems. Analogue communication devices, which include fax, can not communicate over IP network without additional devices, which converts analogue signals into IP packets and vice versa. In this case we can still use analogue devices in IP network. Iskratel d.o.o. company deals with development of such devices.

Device for converting signals needs to be tested to find out if it works properly. Verification and validation department needs to test under what conditions and settings the data transfer is most successful. But combination of settings is huge and company does not have enough human and material resources.

Therefore I chose Automatisation of testing of the transmission of a fax message over the IP network for my thesis. I have shortened the time of testing with automatisation and I have also increased the number of different tests. Automatisation has also relieved those who perform these tests manually.

Keywords: automatisation, fax, IP network

1 Uvod

Podjetje Iskratel d.o.o. je eden vodilnih ponudnikov sodobnih komunikacijskih rešitev v regiji z več kot petdesetletnimi izkušnjami v svetu telekomunikacij. Razvija telekomunikacijske rešitve za lokalna in mestna omrežja, ter omrežja velikega dosega [1]. Snuje tudi celovite rešitve za komunikacijske potrebe informacijske družbe.

Osnovna dejavnost podjetja je razvoj celovitih rešitev za komunikacijske potrebe informacijske družbe prihodnosti na področjih fiksne in mobilne telefonije, konvergenčnih omrežij, omrežij naslednje generacije ter upravljanja omrežij. Poslovna filozofija podjetja poleg sodelovanja s partnerji vključuje še podporne storitve, načrtovanja omrežij, izobraževanja in poprodajne podpore [2].

Oddelek za verifikacijo in validacijo kjer delam, ima nalogo testiranja produktov in ugotavljanja, kaj na produktih deluje in kaj ne. Ko pride produkt v razvoj do mejnika, oziroma do faze razvoja, kjer storitve na produktu že delujejo, lahko oddelek za verifikacijo prične s testiranjem omenjenega produkta. V primeru najdene napake se pošlje zahteva skupaj z opisom problema oziroma napake k razvijalcem, da napako odpravijo. Na oddelku za verifikacijo in validacijo so ključnega pomena dokumenti z opisi, kako mora produkt delovati. Oddelek za verifikacijo in validacijo mora velikokrat najti inovativen način ali pristop k testiranju produkta, ki se ni še nikoli testiral. Zato letno prijavijo tudi več inovacij.

Zaradi vse hitrejšega tempa življenja je velikokrat premalo časa, da bi lahko delo dokončali v obljubljenem času. Tudi podjetja se srečujejo s podobnimi težavami. Tehnologija razvoja napreduje zelo hitro, kupci imajo vedno višje zahteve in želje in vsemu temu je treba slediti. Razvoj novih produktov se v podjetju Iskratel d.o.o. odvija skozi štiri faze. Te faze so načrtovanje, analiziranje, implementacija in testiranje. Po razvoju sledita še servis in vzdrževanje. Toda preden pride produkt v fazo vzdrževanja, ga je potrebno dobro stestirati. Testiranje je obsežno in ob pojavitvi problemov se produkt vrača nazaj v fazo razvoja ali celo analiziranja. Zato je običajno čas testiranja oziroma verifikacije in validacije nekega produkta

zelo dolg. Hitrejša izvedba testov je prihranek za podjetje, vendar le v primeru, ko je avtomatizacija smiselna. Smiselnost avtomatizacije se oceni na več načinov. Pregleda se:

- oceno vloženega dela v avtomatizacijo,
- čas ročnega testiranja,
- pogostost izvedbe testa,
- rabo materialne opreme.

Avtomatizacija je smiselna, ko je ocena vloženega dela v avtomatizacijo manjša od vloženega dela v ročno testiranje.

Zaradi pojavitve oziroma razširitve IP omrežij po svetu, so nastali problemi s protokoli za prenos podatkov. Pretvorba podatkov v IP pakete in obratno ter pošiljanje zahtevajo svoj čas, kar pa pomeni, da nastanejo zakasnitve na oddajni in sprejemni strani. Zaradi zakasnitev se na faks napravi lahko izteče časovna kontrola in zveza se poruši. Zato je potrebno testirati veliko različnih nastavitvev in ugotoviti, ob kakšnih nastavitvah so prenosi boljši in kdaj slabši. Tukaj pa nastane problem, ki ga želim rešiti. Za izvajanje velikega števila testov je potrebno ogromno človeških in materialnih virov ter časa.

Po pogovoru z vodji oddelkov verifikacije in validacije smo ugotovili, da avtomatizacija testiranja prenosa faks sporočila pomeni znaten prihranek v človeških in materialnih virih. Zato sem za svojo diplomsko nalogo izbral avtomatizacijo testiranja prenosa faks sporočila preko IP omrežja. Čeprav je faks naprava s tehnološkega vidika že zastarela, se še vedno zelo pogosto uporablja. Delavci na terenu jo uporabljajo za pošiljanje načrtov, podjetja pa za prenašanje računov in drugih pomembnih dokumentov. Testiranje prenosa faks sporočila je zaradi velikega števila različnih testov zelo zamudno in obsežno delo. Z avtomatizacijo sem znatno skrajšal čas testiranja in razbremenil večje število ljudi, ki sicer sodeluje pri takšnem testiranju.

Zaradi velike razvejanosti in kompleksnosti področja telekomunikacij prihaja do specializacij tudi na področju verifikacije in validacije. Naloge so zato porazdeljene in pri avtomatizaciji je potrebno ločiti med avtomatorji in izvajalci testov. Avtomatorji imajo nalogo, da v

sodelovanju z izvajalci testov naredijo avtomatizacijo testiranja. Izvajalci testov pa naredijo analizo rezultatov in prijavijo morebitne napake. Sodelovanje in izmenjava mnenj med avtomatorji in izvajalci testov sta velikega pomena, saj s tem zmanjšujeta napake pri pisanju programov za avtomatizirano testiranje.

Moja rešitev obsega celotno nastavljanje vrednosti parametrov na omrežnih elementih. Na strežniku z merilno opremo sem pripravil teste in jih zagnal. Ko se je izvajanje zaključilo, sem shranil vse pomembne rezultate v poročilo. Po končanem enem testu sem vse vrednosti parametrov nastavil na privzete vrednosti. S tem sem pripravil okolje za naslednje testiranje. Ko sem zaključil izvajanje vseh testov, sem pregledal ali je prišlo do uspešnega zaključka testov.

V drugem poglavju sem opisal probleme, ki so nastali zaradi pošiljanja faks sporočila preko IP omrežja (zakasnitve, izguba paketov ...). Nato sem opisal še probleme testiranja prenosa faks sporočila preko IP omrežja.

V tretjem poglavju sem opisal uporabljene pojme, testirani sistem ter merilna okolja, v katerih sem delal. Poglavje vsebuje postavitev okolja, kjer sem opisal povezavo merilnega okolja z ohišji, kjer so prisotni omrežni elementi. Na omrežnih elementih prihaja do pretvarjanja prometa, ki potuje med končnimi uporabniki.

V četrtem poglavju sem opisal potek celotne avtomatizacije. Poglavje sem razdelil na manjša podpoglavja in razdelke, v katerih sem ločeno opisal avtomatizacijo posameznih delov celotnega testnega postopka. Najprej sem napisal program v programskem jeziku Java, ki mi je kreiral vhodne podatke oziroma kombinacije testov. Nato sem opisal shranjevanje bližnjice za zagon aplikacije klicni strežnik, ki nadzoruje in upravlja omrežne elemente. Ko sem uspešno povezal aplikacijo klicni strežnik ter iTest, sem posnel in dopolnil procedure za nastavljanje parametrov omrežnih elementov. Nato sem napisal proceduro v iTest-u, ki preko telnet protokola oddaljeno dostopa in pripravi merilno opremo za testiranje. Oddaljeno zažene tudi test, rezultate testiranja pa shranjuje v datoteko. Po končanem testiranju sem nastavil privzete vrednosti na omrežne elemente. V podpoglavju 4.7 sem navedel razliko med časom testiranja in porabljenim časom izvajalca med ročnim in avtomatiziranim testiranjem. Ker

zaradi trenutne zasedenosti omrežnih elementov z drugimi prioritetskimi nalogami nisem uspel zagnati celotnega testnega scenarija, sem za boljšo ponazoritev testiranja prenosa faks sporočila v zadnjem razdelku dodal rezultate testiranja, ki jih je že pred tem ročno izvajal sodelavec Simon Grašič.

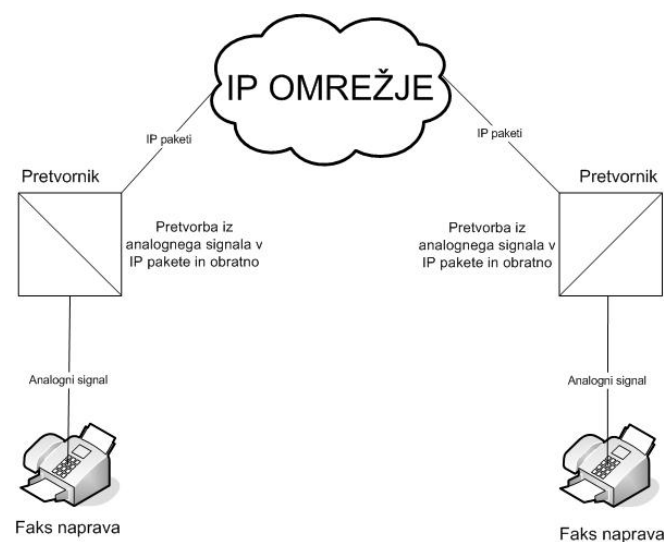
V zaključku sem na kratko opisal rešitve mojega diplomskega dela. Naštel sem ideje in izboljšave, ki bi lahko bile predmet novih raziskav.

2 Opis problema

V poglavju sem opisal probleme, ki se lahko pojavijo pri prenosu faks sporočila. Opisal sem tudi probleme testiranja, na podlagi katerih sem se odločil testiranje avtomatizirati.

2.1 Problemi pri prenosu faks sporočila

Prenos faks sporočila je prilagojen razmeram in težavam v analognem svetu. Želimo, da bi prenos faks sporočila deloval tudi v IP svetu. Za prenos analognega signala preko IP sveta ali omrežja potrebujemo pretvornik, ki zna pretvoriti analogne signale v IP pakete. IP paketi potujejo v IP omrežju. Slika 1 prikazuje preprosto shemo prenosa faks sporočila. Med faks napravo in IP omrežjem je pretvornik, ki omogoča priklop faks naprave v IP omrežje.



Slika 1. Prikaz preproste sheme prenosa faks sporočila.

Pri pošiljanju faks sporočila se lahko pojavijo težave:

- na sami faks napravi,
- na pretvorniku,
- v IP omrežju.

Faks naprava lahko pošilja »pokvarjeno« vsebino, na pretvorniku se lahko zgodi slaba pretvorba signalov. Težave, ki se lahko pojavijo v IP omrežju, so:

- zakasnitev paketov,
- nepravilen vrstni red paketov,
- izguba paketov.

2.2 *Zahtevnost testiranja*

Oddelek za verifikacijo in validacijo podjetja Iskratel d.o.o. ima merilno opremo IxVoice, s katero simulira promet analognih telefonov, modemov in faks naprav. Omenjeni promet se s pomočjo merilne opreme pošilja preko omrežnih elementov nazaj na merilno opremo, kjer se izmeri kvaliteta prenosa. Verifikacija faks funkcionalnosti zahteva veliko virov, tako človeških kot materialnih.

Zahteve za izvedbo testiranja so:

- postavitve testnega okolja,
- priprava testov,
- izvedba testov,
- pregled rezultatov,
- ponovitev testov s slabimi rezultati,
- izdelava poročila o testiranju.

Tabela 1 prikazuje zahteve in porabo virov za izvedbo zahtev. Postavitve testnega okolja izvaja ena skupina ljudi, ostale zahteve pa druga. S pričetkom uporabe orodja iTest se je oblikovala tudi skupina za pripravo avtomatiziranih testov. Zato v seštevku potrebnih ljudi upoštevamo vse omenjene skupine.

Osnovni in razširjeni nabor nastavitev sta najpogostejši ročni testiranja pri prenosu faks sporočila.

Osnovni nabor nastavitev prenosa faks sporočila vsebuje:

- kodek: G.711A-law in T.38,
- protokol za hitrost prenosa: V.17 in V.34,
- paketizacijo: običajno 20ms,
- eno kombinacijo »jitter buffer«-ja.

Razširjeni nabor nastavitev prenosa faks sporočila vsebuje:

- kodek in protokol za hitrost prenosa sta ista kot pri osnovnem naboru nastavitev,
- paketizacijo: 10ms do 40ms s korakom po 10ms,
- »jitter buffer«: vsaj deset kombinacij.

Zahteve	Osnovni nabor nastavitev prenosa faksa		Razširjeni nabor nastavitev prenosa faksa	
	Potreben čas	Potrebno število ljudi	Potreben čas	Potrebno število ljudi
Postavitev testnega okolja	2 uri	2	2 uri	2
Priprava testov	10 min	1	2 uri	2
Izvedba testov	1 ura	1	40 ur	1
Pregled rezultatov	5 min	1	8 ur	2
Ponovitev testov s slabimi rezultati	30 min	1	10 ur	2
Gradnja poročila o testiranju	30 min	1	15 ur	1
Σ	4 ure 15 min	3	77 ur	4

Tabela 1. Prikaz zahtev in poraba virov zanje.

Zaradi slabe izrabe materialnih virov in pomanjkanja človeških virov za testiranje faks funkcionalnosti, sem za temo moje diplomske naloge izbral avtomatizacijo testiranja prenosa faks sporočila preko IP omrežja. Čeprav je faks naprava s tehnološkega vidika že zastarela, se še vedno zelo pogosto uporablja. Delavci na terenu jo uporabljajo za pošiljanje načrtov, podjetja pa za prenašanje računov in drugih pomembnih dokumentov. Prenasjanje dokumentov preko faks naprave je varno in zanesljivo. Zato se je še vedno dobro držati načela, da delujoče opreme ne gre menjati. Ker se faks funkcionalnosti testirajo na vsakem novem produktu in v vsaki vzdrževalni izdaji programske opreme vidimo, da bi avtomatizacija te funkcionalnosti prinesla znaten prihranek materialnih in človeških virov.

Cilj mojega diplomskega dela je avtomatizacija naslednjih zahtev:

- priprave testov,
- izvedbe testov,
- pregleda rezultatov,
- izdelave poročila o testiranju.

S tem sem minimiziral čas testiranja in povečal nabor testov. Z avtomatizacijo sem tudi povečal izkoriščenost merilne opreme ter ljudi razbremenil opravljanja ponavljajočih se nalog.

3 Opis uporabljenih pojmov, testiranega sistema in merilnih okolij

V tem poglavju sem opisal uporabljene pojme, testirani sistem ter merilna okolja. Elementi testiranega sistema so Iskratelovi produkti. Merilno okolje IxVoice podjetje uporablja že dalj časa, z orodjem iTest pa želi podjetje avtomatizirati testiranja v čim večji meri.

3.1 Uporabljeni pojmi, specifični v testiranem sistemu

Pojmi, ki sem jih uporabljal, so specifični za moj testirani sistem. Opisal sem dva kodeka, paketizacijo, zadušitev tišine, vmesnik »jitter buffer«, bližnjico za zagon aplikacij, aplikacijo WBM in izkoristek.

3.1.1 Kodek

Kodek je programska oprema namenjena stiskanju in raztegovanju vsebine. Za shranjevanje vsebine se s pomočjo kodeka vsebina stisne, ob uporabi pa kodek vsebino razširi. Obstaja veliko vrst kodekov, ki so namenjeni različnim vsebinam.

3.1.2 Kodek G.711 A-law

Standard za kodeke vrste G.711 je nastal leta 1972. Ta kodek se pretežno uporablja v telefoniji in sicer v Evropi in drugod po svetu, razen v Severni Ameriki in na Japonskem. Kodek G.711 A-law se lahko zapiše tudi kot G.711 A. Kodek je algoritmičen in sloni na pulzno kodni modulaciji (angl. *pulse code modulation*). Prav tako se uporablja za komunikacijo med faks napravami v IP omrežjih.

3.1.3 Kodek T.38

Leta 1998 se je pojavil kodek z imenom T.38. Namen kodeka je bil zmanjšati izgube zaradi daljših prenosov preko satelitov, ki nastanejo s kodekom G.711 A. Kodek T.38 ima manjše pakete in dalj časa čaka na prihod paketa na cilj. Ima pa eno pomanjkljivost. Ker je bil razvit pred razvojem širokopasovnih povezav, ne more pošiljati s hitrostjo višjo od 14.4kbps. Torej se v primeru kodeka T.38 in protokola V.34 hitrost zmanjša na 14.4kbps.

3.1.4 Paketizacija

Paketizacija pomeni, koliko časa bo izvorna stran čakala, da napolni en paket, preden ga pošlje. Večja paketizacija pomeni večje pakete, ki se pošiljajo in s tem tudi večjo zakasnitev, toda manjšo režijo. Pri paketizaciji je pomembno najti optimalno razmerje med zakasnitvijo in režijo.

3.1.5 Zadušitev tišine

Opcijo zadušitev tišine (angl. *silence suppression*) se uporablja samo pri kodekih za zvok. V primeru, ko je opcija vključena, se na oddajni strani detektira govor. Če govor ni prisoten, oddajna stran ne bo pošiljala paketov. S tem se zmanjša velikost nepotrebne prometa v omrežju.

3.1.6 Vmesnik »jitter buffer«

»Jitter buffer« je prisoten na sprejemni strani. Na oddajni strani se paketi pošiljajo enakomerno. Ker v IP omrežjih prihaja do vpliva zakasnitev in izgub paketov, prihajajo do sprejemnika IP paketi zakasneni, v nepravilnem vrstnem redu, okvarjeni ali pa se celo izgubijo. Temu pojavu se reče tresenje (angl. *jitter*). Posledica tega pojava so lahko zavrženi paketi in s tem napake. Naloga »jitter buffer«-ja je, da na sprejemni strani čaka pakete in jih

razvršča v pravilni vrstni red. Šele po izteku maksimalne zakasnitve »jitter buffer«-ja, »jitter buffer« posreduje pakete naprej. Če vseeno prejme paket, ki bi ga že moral posredovati naprej, ga zavrže [3]. Zato se testira več kombinacij nastavitev »jitter buffer«-ja, da bi dobili optimalni čas sprejemanja paketov. Poznamo dva tipa »jitter buffer«-ja. Lahko je fiksni ali spremenljiv. V primeru fiksnega tipa »jitter buffer«-ja se upoštevata samo nominalna parametra, v primeru spremenljivega tipa pa se upoštevajo vsi parametri.

»Jitter buffer« ima štiri parametre, ki se merijo v milisekundah:

- minimalna zakasnitev »jitter buffer«-ja (angl. *min jitter buffer delay*); če je prenos IP paketov krajši in brez napak, se čas čakanja zmanjša na minimalno zakasnitev,
- maksimalna zakasnitev »jitter buffer«-ja (angl. *max jitter buffer delay*); če je prenos IP paketov daljši in se pojavijo napake, se čas čakanja poveča na maksimalno zakasnitev,
- nominalna zakasnitev »jitter buffer«-ja za zvok (angl. *nominal audio jitter buffer delay*); začetna zakasnitev »jitter buffer«-ja pri pošiljanju zvoka (telefonija),
- nominalna zakasnitev »jitter buffer«-ja za podatke (angl. *nominal data jitter buffer delay*); začetna zakasnitev »jitter buffer«-ja pri pošiljanju podatkov (modem ali faks naprava).

3.1.7 Bližnjica za zagon aplikacij

Bližnjica, o kateri govorim, se imenuje sso.jnlp. Uporablja se za zagon aplikacij, ki nadzorujejo različne Iskratelove produkte. V diplomski nalogi sem uporabljal bližnjico do aplikacije klicni strežnik. Aplikacije so narejene v java swing okolju, zato bližnjica vsebuje pravila, kako naj se javanska aplikacija zažene. Vsebuje tudi poti do JAR datotek, kjer so zbrani javanski prevedeni programi [4, 5].

3.1.8 Aplikacija WBM

Aplikacija WBM (angl. *Web Based Management*) je Iskratelova spletna aplikacija, kjer so po skupinah zbrane vse bližnjice nadzornih programov. Ob kliku na bližnjico se odpre nadzorni program za izbrani produkt.

3.1.9 Izkoristek

Pri testiranju prenosa faks sporočila se kot rezultat testiranja poda FOM (angl. *Figure of merit*), ki se izračuna na podlagi stanja testa, dosežene največje hitrosti prenosa in kvalitete slike. Tabela 2 prikazuje, kako se izračuna FOM za prenos faks sporočila. Vse tri kriterije vrne orodje IxVoice. Na podlagi FOM vrednosti se izvede nadaljnja analiza težav in prijava problemov.

Figure of merit			
FOM value	Test complete	Max speed	Image quality
1	Yes	Yes	Error-Free
2	Yes	Yes	Errored
3	Yes	Yes	Severely-Errored
4	Yes	No	Error-Free
5	Yes	No	Errored
6	Yes	No	Severely-Errored
7	No	Not Applicable	Not Applicable

Tabela 2. Prikaz računanja FOM vrednosti.

3.2 Testirani sistem

Testirani sistem predstavlja strojno in programsko opremo za prenos faks sporočil. Prav tako v ta sistem spadajo tudi povezave, s katerimi je strojna oprema povezana med seboj.

3.2.1 Opis in postavitve testiranega sistema

Testirani sistem vsebuje dva omrežna elementa:

- analogna naročniška SAK plošča: omogoča priključitev analognih naprav in pretvorbo analognih signalov v IP pakete in obratno,
- ES plošča: povezuje SAK plošče in skrbi za pravilno posredovanje paketov.

Testirani sistem vsebuje tudi Soft switch ploščo. Soft switch plošča je nadzorna plošča, ki krmili omrežne elemente. Na tej plošči teče klicni strežnik (angl. *Call server*). To je aplikacija, s pomočjo katere se izvaja krmiljenje klicev (angl. *Call Control*), krmiljenje povezav (angl. *Connection Control*) in storitvena logika (angl. *Service Control*). Na plošči se nastavlja parametre, ki so pomembni pred vzpostavitvijo klica in med samim klicem.

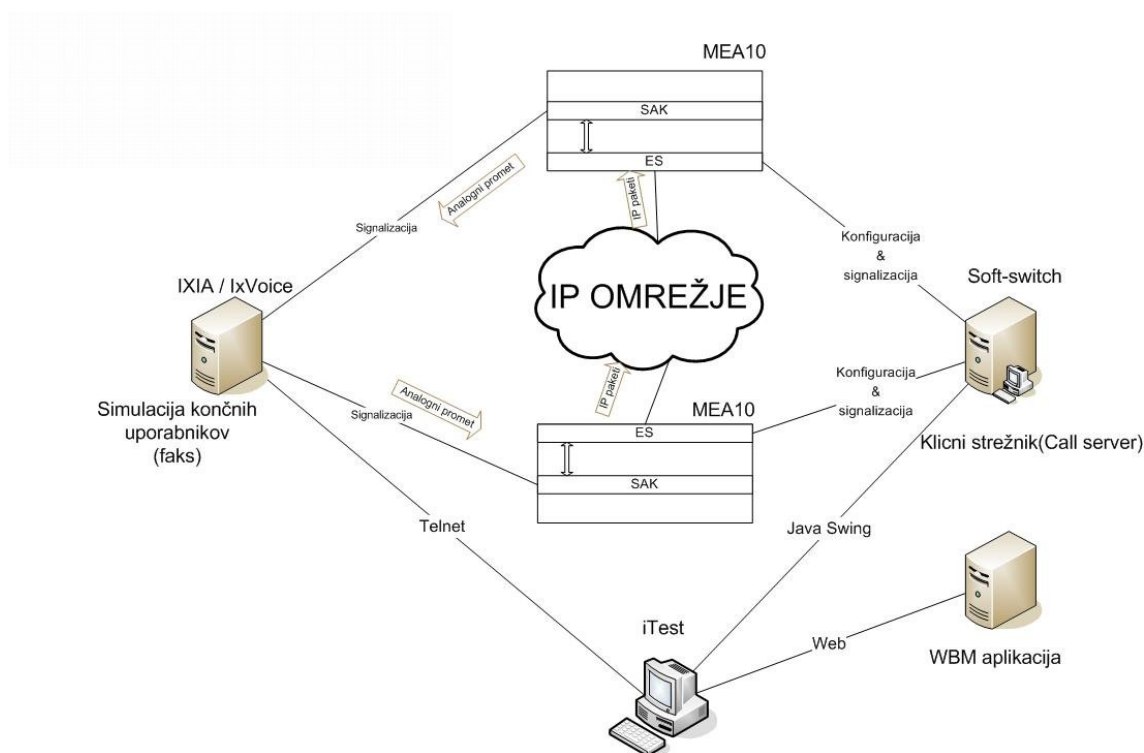
S pomočjo aplikacije se nastavlja:

- kodek: s tem kodekom se bo kodiral prenos,
- prioriteto vrsto kodekov: v primeru da ponorna stran ne podpira primarnega kodeka, se zveza poizkuša vzpostaviti z enim od naslednjih kodekov v prioritetni vrsti,
- parametre »jitter buffer«-ja,
- fiksni ali spremenljiv tip »jitter buffer«-ja,
- paketizacijo za prenos,
- opcijo zadušitev tišine: opcija je lahko vklopljena ali izklopljena,
- protokol za različne vrste prenosa.

Slika 2 prikazuje shemo testiranega sistema. Na levem delu slike 2 vidimo strežnik (IXIA/IxVoice), kjer teče simulacija dveh faks naprav. Faks napravi si med seboj pošiljata podatke. Primer bi v praksi izgledal tako, kot da imamo prvo faks napravo v podjetju Iskratel d.o.o., drugo faks napravo pa na Fakulteti za računalništvo in informatiko. Iz podjetja pošljemo faks sporočilo na Fakulteto za računalništvo in informatiko. Vsebina strani, ki jo pošiljamo, se najprej pošlje od prve faks naprave do analogne naročniške SAK plošče. Analogni promet se na SAK plošči pretvori v IP pakete in se posreduje naprej do ES plošče.

Paketi gredo nato preko IP omrežja do druge ES plošče, ta pa jih na podlagi telefonskih števil usmeri na ustrezno SAK ploščo. Na drugi SAK plošči se paketi pretvorijo nazaj v analogne signale. Signali gredo nato do končnega uporabnika, ki je v tem primeru faks naprava na Fakulteti za računalništvo in informatiko. Želimo si, da bo oseba na Fakulteti za računalništvo in informatiko lahko prebrala natisnjeno vsebino. Zato mora biti prenos uspešen in brez napak.

Z orodjem iTest sem se najprej preko spleta povezal z aplikacijo WBM, kjer sem shranil bližnjico za zagon aplikacije klicni strežnik. S pomočjo skripte za zagon sem se z iTest-om povezal na aplikacijo klicni strežnik, kjer sem nastavil vrednosti parametrov omrežnih elementov. Klicni strežnik nato z novimi nastavitvami konfigurira dve SAK plošči v ohišjih, skozi kateri bo tekel promet. Z orodjem iTest sem nato na merilni opremi IxVoice preko telnet protokola nastavil parametre za testiranje in zagnal test. Po končanem testu sem z orodjem iTest shranil rezultate testiranja in zgradil poročilo.



Slika 2. Shema testiranega sistema.

3.2.2 Ročni test prenosa faks sporočila in pregled parametrov

Pred postopkom avtomatizacije sem naredil ročni test prenosa faks sporočila, s katerim sem ugotovil katere profile in parametre moram uporabiti v aplikaciji klicni strežnik za uspešno konfiguriranje omrežnih elementov.

Najprej sem se prijavil v aplikacijo WBM. SAK plošče spadajo v skupino LA produktov, Soft switch-i pa spadajo v skupino CS produktov. Najprej sem izbral skupino CS produktov in našel identifikacijsko številko moje Soft switch plošče. Klik na ikono strežnika oziroma bližnjico poleg identifikacijske številke moje Soft Switch plošče požene program JavaWS (angl. *Java Web Start*), le-ta pa odpre aplikacijo klicni strežnik.

V aplikaciji klicni strežnik sem uporabljal pet različnih profilov:

- *Codec Profile*,
- *Codec Priority Profile*,
- *DSP IP Profile*,
- *RTP Profile*,
- *SIP Profile*.

Na profilu *Codec Profile* sem nastavil:

- kodek, s katerim se bo kodiral prenos,
- čas paketizacije,
- možnost vključitve ali izključitve opcije zadušitev tišine.

Na profilu *Codec Priority Profile* sem nastavil prioriteto vrsto kodekov. V njej sem dodal kodek, ki sem ga nastavil na profilu *Codec Profile*.

Na profilu *DSP IP Profile* sem nastavil:

- fiksni ali spremenljiv tip »jitter buffer«-ja,
- minimalno zakasnitev »jitter buffer«-ja,
- maksimalno zakasnitev »jitter buffer«-ja,
- nominalno zakasnitev »jitter buffer«-ja za zvok,
- nominalno zakasnitev »jitter buffer«-ja za podatke.

Na profil *RTP Profile* sem pripel:

- *Codec Profile*,
- *Codec Priority Profile*,
- *DSP IP Profile*.

Na profil *SIP Profile* sem pripel:

- *Codec Profile*,
- *Codec Priority Profile*,
- *DSP IP Profile*.

Profil *RTP Profile* se povezuje s protokolom MGCP, profil *SIP Profile* pa se povezuje s protokolom SIP. Protokola MGCP in SIP se uporabljata za signalizacijo med omrežnimi elementi v različnih omrežjih. Omenjena protokola v mojem primeru ne prideta do velikega izraza, saj imam samo eno omrežje. Vseeno pa sem moral protokola nastaviti in ju uporabljati.

3.3 Merilna okolja

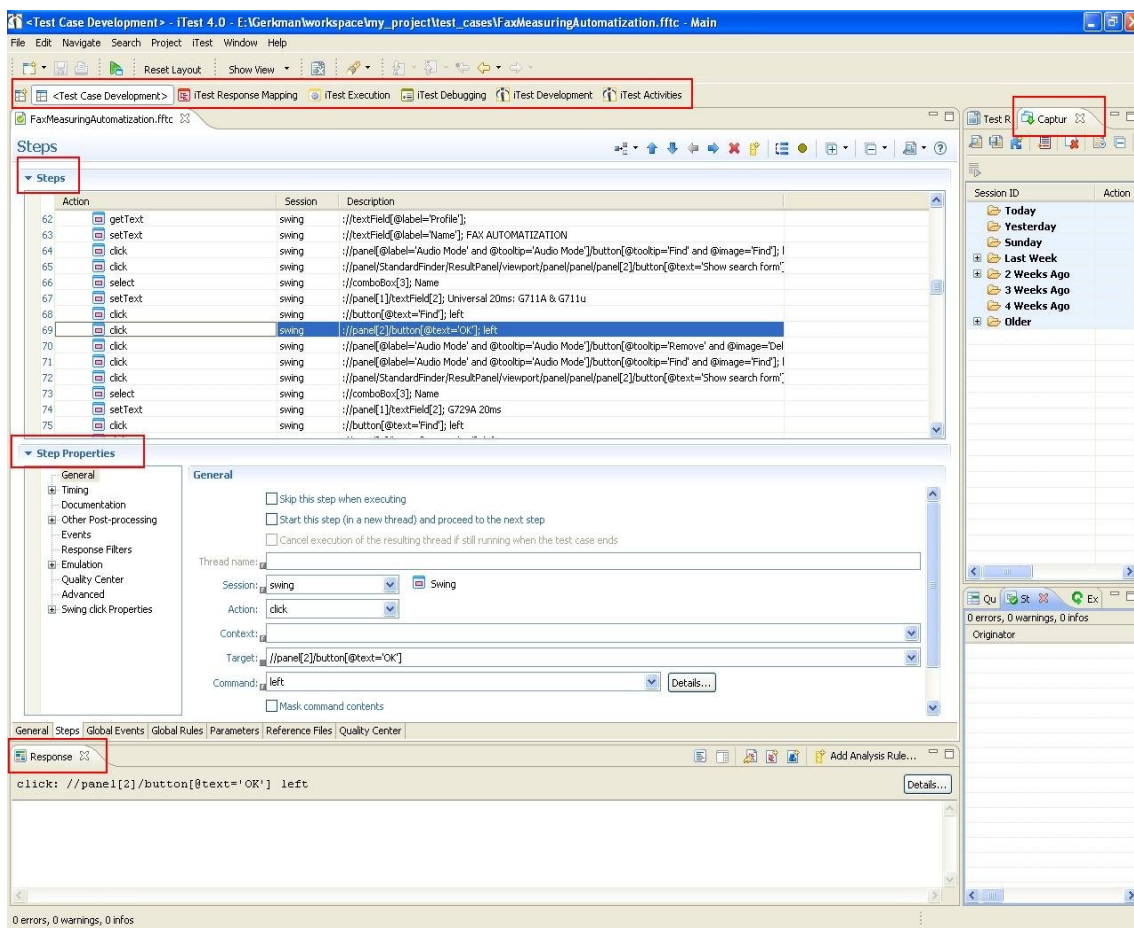
Merilna okolja predstavljajo programsko opremo, ki sem jo uporabljal. S pomočjo merilnega okolja IxVoice sem simuliral prenos faks sporočil, z orodjem iTest pa sem naredil avtomatizacijo testiranja.

3.3.1 iTest

Aplikacijo iTest so razvili v ameriškem podjetju Fanfare. Podjetje so leta 2004 ustanovili Kingston Duffie, Denise Savoie in Carl Hubbard, sedež pa ima v Kaliforniji. Podjetje Fanfare se ukvarja z iskanjem programskih rešitev za hitrejše in učinkovitejše testiranje telekomunikacijskih produktov.

Slika 3 prikazuje orodje iTest, s katerim lahko nadzorujemo in snemamo delo z več različnimi protokoli, aplikacijami in prometnimi generatorji. iTest za delo s protokoli, aplikacijami in prometnimi generatorji uporablja različne seje:

- *telnet session*: seja za snemanje telnet protokola,
- *web session*: seja za snemanje http in https protokolov,
- *ssh session*: seja za snemanje SSH protokola,
- *wireshark session*: seja za snemanje aplikacije wireshark,
- *comand prompt session*: seja za snemanje programa ukazna vrstica,
- *swing session*: seja za snemanje java swing programov,
- *IxNetwork session*: seja za snemanje prometnega generatorja IxNetwork,
- *IxTraffic session*: seja za snemanje prometnega generatorja IxTraffic.



Slika 3. Prikaz orodja iTest v načinu za urejanje programske kode.

V iTest-u s pomočjo orodja za izbiro seje (angl. *session profile*) izberemo sejo, s katero želimo komunicirati in jo snemati, ter jo zaženemo. S protokolom, aplikacijo ali prometnim generatorjem nato upravljamo kot običajno, iTest pa v ozadju snema vse korake in akcije. Po končanem snemanju zgradi program, ki se imenuje testni primer (angl. *test case*). Testni primer vsebuje ukaze oziroma korake, ki jih je iTest zajel med snemanjem.

iTest ima več pogledov, ki se uporabljajo v različnih fazah razvoja programa. Pogledi so:

- *Test Case Development* pogled: v njem pregledujemo in urejamo že ustvarjeno kodo ali pa jo napišemo sami,

- *iTest Response Mapping* pogled: omogoča nam gradnjo in urejanje odzivov (angl. *Response*), katere potem iTest avtomatsko shrani v spremenljivke ali strukture za nadaljnjo uporabo,
- *iTest Execution* pogled: uporablja se med izvajanjem programa,
- *iTest Debugging* pogled: v njem vidimo vse spremenljivke in strukture, ki jih iTest uporablja; lahko tudi ustavimo izvajanje in nadaljujemo korak za korakom,
- *iTest Development* pogled: omogoča nadzor nad vsemi profili in programi v iTest-u.

Pod menijem pogledov imamo okno, v katerem je koda oziroma koraki testnega primera. Urejamo lahko nastavitve za vsak korak (angl. *Step properties*). Na spodnjem delu slike 3 pa vidimo okno odziv (angl. *Response*). To okno je zelo pomembno, saj v njem vidimo, kako se je ukaz izvedel in kaj je seja vrnila kot rezultat tega ukaza. S pomočjo tega okna potem pišemo pravila (angl. *Analysis rule*) in pogoje za izvajanje. S pravili in pogoji določimo tok programa. Na desni strani slike 3 zgoraj, imamo okno z imenom zajem (angl. *Capture*). Tukaj vidimo korake, ki jih iTest posname. Pod oknom zajem imamo še eno okno, kjer vidimo status preteklega izvajanja in sporočila, ki nam jih vrnejo pravila, ki smo jih dodali v kodo programa [6, 7].

iTest ima zelo uporabno orodje, ki se imenuje mapa z odzivi (angl. *Response map*). S pomočjo tega orodja lahko strukturirano shranimo velike količine podatkov. Običajno se uporablja pri branju vhodnih podatkov. Orodje si vsebino shrani v drevesno strukturo, nato pa kreira poizvedbe (angl. *Queries*), s pomočjo katerih lahko dostopimo do podatkov v drevesni strukturi.

3.3.2 IxVoice

IxVoice je produkt podjetja IXIA, ki ima sedež v Calabasas-u v zvezni državi Kalifornija v Združenih državah Amerike. Podjetje se ukvarja z razvojem orodij in aplikacij za simulacijo naprav in prometa v IP omrežjih. Aplikacija skupaj s strojno opremo omogoča simulacijo končnih naprav kot so analogni telefoni, modemi in faks naprave. IxVoice lahko krmilimo preko grafičnega vmesnika ali preko telnet protokola. IxVoice aplikacija omogoča gradnjo

scenarijev za generiranje prometa, ki ga preko dveh analognih kartic pošilja in sprejema. Kartici sta:

- *T1/E1 analog card*: kartica omogoča analogni telefonski promet (priključitev analognih telefonskih naročnikov in vzpostavljanje zveze med njimi),
- *DTE analog card*: kartica omogoča prenose faks sporočil in modemskega prometa.

Vsi parametri, ki se nastavljajo v programu IxVoice, so zbrani v ECF datoteki. Parametri so:

- scenarij, ki se bo izvajal med testiranjem,
- število iteracij oziroma ponovitev enega testa,
- stopnja shranjevanja podatkov,
- številka kanala, ki je povezan na DTE kartico,
- oznaka, kateri kanal je v uporabi v trenutnem testu,
- klicne številke na vsakem kanalu,
- lastne telefonske številke za povratni klic na vsakem kanalu,
- priključki, ki se bodo uporabljali med testiranjem.

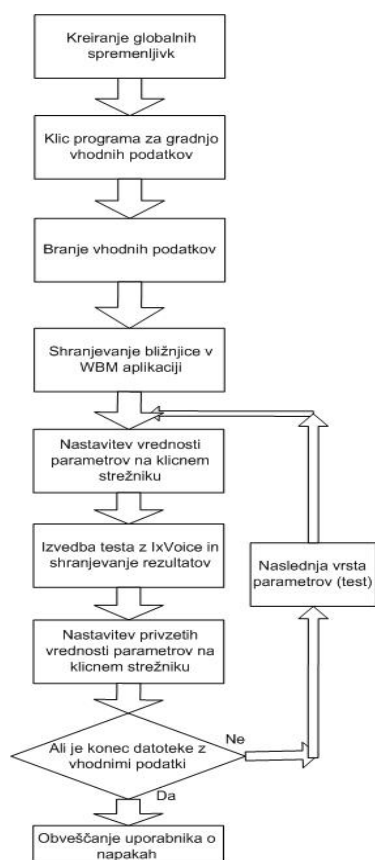
ECF datoteko je možno nastavljati tako preko grafičnega vmesnika, kot tudi preko telnet protokola.

4 Avtomatizacija postopka testiranja

Avtomatizacijo postopka testiranja sem razdelil na manjše dele in z orodjem iTest napisal procedure za vsakega od njih.

4.1 iTest scenarij

Slika 4 prikazuje korake avtomatizacije. Najprej sem ustvaril globalne spremenljivke, ki sem jih uporabljal tekom celotne avtomatizacije. Nato sem klical program, kjer sem vnesel vse potrebne podatke, program pa je sam ustvaril kombinacije testov oziroma vhodne podatke.



Slika 4. Prikaz avtomatizacije po korakih.

V orodju iTest sem prebral vhodne podatke in si kombinacije shranil s pomočjo map z odzivi. Slika 5 prikazuje branje vhodnih datotek z ukazom *readFile*. Ukaz *readFile* prebere vsebino datoteke in jo prikaže v svojem odzivu.

comment	First read the General input parameters file
readFile	file:/E:/Gerkman/iTestInputParametersFile/General_input_parameters.csv
comment	Then read the input data for Fax test settings
readFile	file:/E:/Gerkman/iTestInputParametersFile/Fax_test_settings.csv

Slika 5. Prikaz uporabe ukaza *readFile* za branje vhodnih datotek.

Nato sem se povezal z aplikacijo WBM, kjer sem shranil bližnjico do aplikacije klicni strežnik. Z zanko sem se sprehodil skozi vse kombinacije testov. Števec v zanki, ki se imenuje *testNumber*, predstavlja številko trenutnega testa. Spremenljivka je zelo pomembna, saj sem se kasneje z njeno pomočjo skliceval na vsako vrednost parametrov, ki sem jo nastavil na klicnem strežniku. Nato sem zagnal test prenosa faks sporočila v orodju IxVoice. Po zaključenem testu sem na klicnem strežniku nastavil privzete vrednosti in nadaljeval z naslednjim testom v datoteki. Ko se je zanka iztekla, sem uporabnika preko SMS-a in elektronske pošte obvestil o napakah, ki so se zgodile med izvajanjem testov. Slika 6 prikazuje zanko in klice procedur za nastavljanje parametrov na klicnem strežniku in proceduro za izvedbo testa z orodjem IxVoice.

comment	FOR loop goes through every test-configuration and performs tests
for	{set i 1} - {\$i <= [query /data/InputParameters_Fax_response_map rowCount()]} - {incr i}
comment	First configure settings on soft-switch
call	openMN_SetSettingsOnSoftSwitch_closeMN -testNumber \$i
comment	Perform test with IxVoice
call	makeTest -testNumber \$i -VoiceCodecName [query /data/InputParameters_Fax_response_map getVoiceCodec(\$i)] -VoicePacketization
comment	Set default settings on soft-switch
call	openMN_SetDefaultSettingsOnSoftSwitch_closeMN -testNumber \$i
comment	After testing is complete inform tester about status
call	InformTesterAboutStatus -messageSMS \$SMSErrorMessage -messageMail \$MailErrorMessage

Slika 6. Prikaz zanke, v kateri kličem procedure za nastavljanje parametrov in izvajanje testov.

Omenjene funkcionalnosti predstavljajo glavno proceduro v mojem programu za avtomatizacijo.

4.2 Avtomatizacija gradnje vhodnih podatkov

Eden glavnih vzrokov za avtomatizacijo je veliko število kombinacij testov. Na sliki 7 sem prikazal nekaj kombinacij testov. Ker je kombinacij testov še veliko več, sem v programskem jeziku Java napisal program, ki mi sam kreira vhodne podatke. Parametri so v vhodni datoteki ločeni z dvojnim narekovajem.

```
Test number"FaxProfile_name"Fax_codec"Fax_packetization"Fax_SS"Jitter buffer name"JB type"JBMin"JBMax"JBNomAudio"JBNomData
1"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
2"G711A_10ms_SS=1"G.711 at 64 kbit/s, A-law"10"true"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
3"G711A_20ms_SS=0"G.711 at 64 kbit/s, A-law"20"false"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
4"G711A_20ms_SS=1"G.711 at 64 kbit/s, A-law"20"true"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
5"G711A_30ms_SS=0"G.711 at 64 kbit/s, A-law"30"false"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
6"G711A_30ms_SS=1"G.711 at 64 kbit/s, A-law"30"true"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
7"T38_10ms_SS=0"T.38"10"false"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
8"T38_10ms_SS=0"T.38"10"false"JBAI_10_20_40_60"Adaptive immediately"10"80"40"40
9"T38_10ms_SS=0"T.38"10"false"JBAI_10_20_40_60"Fixed"10"40"20"20
10"T38_10ms_SS=0"T.38"10"false"JBAI_10_20_40_60"Fixed"10"80"40"40
11"T38_20ms_SS=0"T.38"20"false"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
12"T38_20ms_SS=0"T.38"20"false"JBAI_10_20_40_60"Adaptive immediately"10"80"40"40
13"T38_20ms_SS=0"T.38"20"false"JBAI_10_20_40_60"Fixed"10"40"20"20
14"T38_20ms_SS=0"T.38"20"false"JBAI_10_20_40_60"Fixed"10"80"40"40
15"T38_30ms_SS=0"T.38"30"false"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
16"T38_30ms_SS=0"T.38"30"false"JBAI_10_20_40_60"Adaptive immediately"10"80"40"40
17"T38_30ms_SS=0"T.38"30"false"JBAI_10_20_40_60"Fixed"10"40"20"20
18"T38_30ms_SS=0"T.38"30"false"JBAI_10_20_40_60"Fixed"10"80"40"40
19"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"10"40"20"20
20"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"10"40"30"30
21"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"20"50"30"30
22"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"20"50"40"40
23"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"30"60"40"40
24"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"30"60"50"50
25"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"40"70"50"50
26"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"40"70"60"60
27"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"50"80"60"60
28"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"50"80"70"70
29"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"60"90"70"70
30"G711A_10ms_SS=0"G.711 at 64 kbit/s, A-law"10"false"JBAI_10_20_40_60"Adaptive immediately"60"90"80"80
```

Slika 7. Prikaz nekaj kombinacij testov.

Prva vrstica vsebuje razlago parametrov – vrednosti parametrov so napisane s pisavo Courier New:

- Test number: številka posameznega testa,
- FaxProfile_name: ime profila,
- Fax_codec: kodek,
- Fax_packetization: paketizacija,
- Fax_SS: stanje opcije zadušitev tišine,
- Jitter buffer name: ime profila *DSP IP Profile*,
- JB type: vrsta »jitter buffer«-ja,
- JBMin: minimalna zakasnitev »jitter buffer«-ja,
- JBMax: maksimalna zakasnitev »jitter buffer«-ja,

- JBNomAudio: nominalna zakasnitev »jitter buffer«-ja za zvok,
- JBNomData: nominalna zakasnitev »jitter buffer«-ja za podatke.

Vsaka kombinacija testov vsebuje naslednje parametre – vrednosti parametrov za moj primer so napisane s pisavo Courier New:

- številko posameznega testa: 1,
- ime profila: G711A_10ms_SS=0,
- kodek: G.711 at 64 kbit/s, A-law,
- paketizacijo: 10,
- stanje opcije zadušitev tišine: false,
- ime profila *DSP IP Profile*: JBAI_10_20_40_60,
- vrsto »jitter buffer«-ja: Adaptive immediately,
- vrednost minimalne zakasnitve »jitter buffer«-ja: 10,
- vrednost maksimalne zakasnitve »jitter buffer«-ja: 40,
- vrednost nominalne zakasnitve »jitter buffer«-ja za zvok: 20,
- vrednost nominalne zakasnitve »jitter buffer«-ja za podatke: 20.

Ko se program za kreiranje vhodnih podatkov začne, lahko uporabnik izbere opcijo naj mu program kreira vse možne kombinacije testov. Uporabnik lahko vnese tudi nekaj podatkov in program na podlagi le-teh kreira kombinacije testov. V tem primeru mora uporabnik vnesti:

- klicne številke končnih uporabnikov,
- število iteracij za vsak test,
- kodek,
- stanje opcije zadušitev tišine,
- paketizacijo,
- vrednosti »jitter buffer« spremenljivk,
- protokol za hitrost prenosa faks sporočila,
- protokol za signalizacijo.

Slika 8 prikazuje primer programa za kreiranje vhodnih podatkov. Uporabnik vpiše podatke in program na podlagi teh ustvari kombinacije različnih testov.

V orodju iTest sem se preko seje *comand prompt session* povezal v ukazno vrstico in v novem oknu ukazne vrstice zagnal javanski program za gradnjo vhodnih podatkov. V ukazni vrstici sem uporabil ukaz *start*, skupaj s klicem programa za gradnjo vhodnih podatkov. Omenjena funkcionalnost predstavlja proceduro *Create_test_settings*.

```
Type the number to call to side A <prefix + DN>: 1569585
Type the number to call to side B <prefix + DN>: 1569586
Type the number of iterations for single test: 8
Select the codec:
1: G711 A-law
2: G711 u-law
3: G729 annex A
4: T.38
1
Enter the packetization: 20

Enter the minimum jitter buffer: 10
Enter the maximum jitter buffer: 40
Enter the nominal audio jitter buffer: 20
Enter the nominal data jitter buffer: 20

Which protocol for FAX will you use?
1: U17
2: U34
2

Which protocol for signalization will you use?
1: MGCP
2: SIP
1

Finished!!          Created files: General_input_parameters.csv
                               Fax_test_settings.csv
```

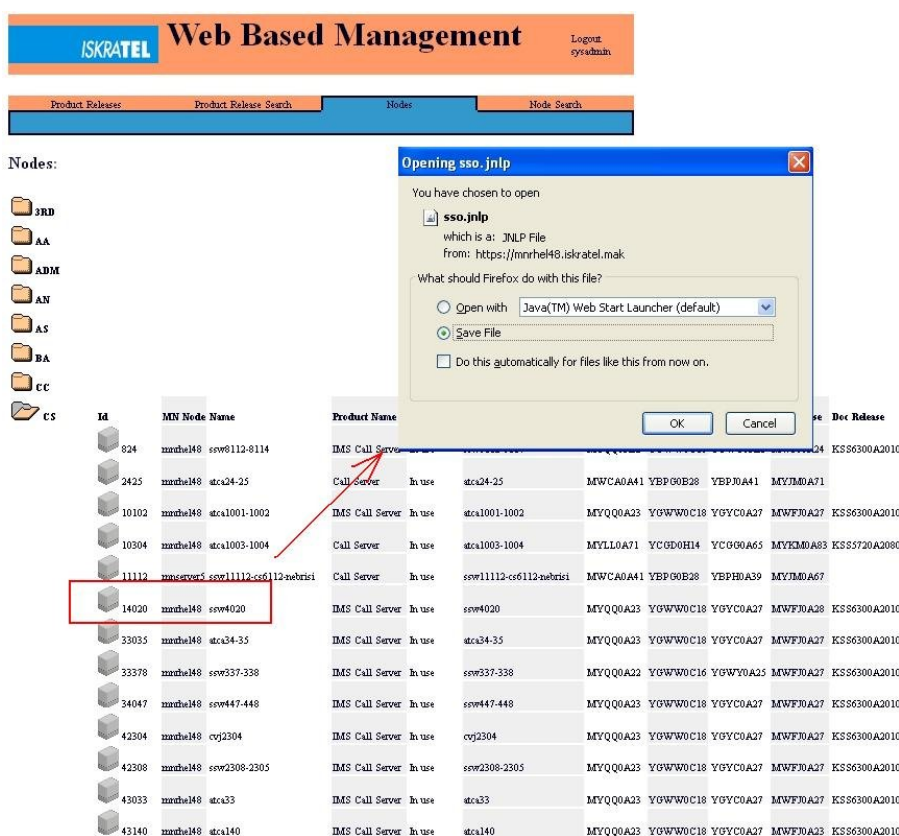
Slika 8. Prikaz programa za kreiranje vhodnih podatkov.

4.3 Avtomatizacija aplikacije WBM za zajem bližnjice

V orodju iTest sem izbral pogled *iTest Development*, v katerem sem ustvaril nov profil za snemanje spletne seje. Na spletni seji sem v polje *Base URL* vpisal URL naslov strežnika, kamor se je povezala seja. V polje *Session name* sem vpisal ime seje. Ime se prikaže ob vsakem ukazu, ki se navezuje na to sejo.

Shranil sem nastavitve in zagnal sejo. iTest ima vgrajen spletni brskalnik, ki je nastavljen kot privzeti brskalnik. Uporabimo lahko tudi katerega od spletnih brskalnikov, ki so naloženi na računalniku. Jaz sem izbral privzetega, ki ga ima iTest. Preverjen je s strani Fanfare podjetja in deluje. Ob kliku na gumb zaženi, iTest začne nadzorovati sejo, ki sem jo zagnal.

Po uspešni prijavi na strežnik se je odprla aplikacija WBM. V ozadju je iTest snemal vse, kar sem naredil, od klikov gumba na miški ali tipkovnici do premikov miške čez kak gradnik ipd. Tudi prijava se je posnela, tako da naslednjič uporabniku ne bo potrebno vpisovati uporabniškega imena in gesla. V skupini CS produktov sem izbral Soft switch z identifikacijsko številko 14020. Ob kliku na ikono strežnika se je odprlo okno, kjer sem izbral možnost »shrani bližnjico«. Slika 9 prikazuje aplikacijo WBM in okno, kjer sem izbral možnost »shrani bližnjico«.



Slika 9. Prikaz aplikacije WBM z bližnjicami ter oknom, kjer izberemo odpri ali shrani bližnjico.

Nato sem izbral še skupino s produkti LA, kjer so naročniške SAK plošče. Izbral sem ploščo z identifikacijsko številko 1014. Bližnjico sem shranil na isti način. Ob zaprtju aplikacije WBM je iTest zaključil s snemanjem. V pogledu *Test-Case Development* imamo okno z imenom zajeto. V njem sem označil zajeto kodo in izbral možnost dodaj med testne primere (angl. *Add to test case*). Slednja kreira novo proceduro in vanjo zapiše označeno kodo. Proceduro shrani v mapo s testnimi primeri. iTest je zelo dobro poskrbel za zaščito gesel, saj se le-ta ne prikažejo v kodi programa. Slika 10 prikazuje zaščito gesel v orodju iTest.

8		setText	mn	:name=username; sysadmin
9		setText	mn	*****

Slika 10. iTest skrije vsebino gesla z zvezdicami.

iTest ima težave pri določanju pravilnega zaporedja ukazov za shranjevanje bližnjice, saj zamenja ukaza *setFilePath* ter *click*. Z ukazom *setFilePath* iTest nastavi pot, kamor shrani datoteko. Ukaz *click* pa izvede klik na ikono za odpiranje ali shranjevanje. Če ukaza *setFilePath* ni pred ukazom *click*, iTest odpre datoteko, sicer pa jo shrani. Slika 11 prikazuje pravilno zaporedje ukazov za shranjevanje bližnjic.

	comment		Open CS products and download sso.jnlp file
	click	mn	:link=CS;
	setFilePath	mn	E:\Gerkman\iTestFiles\14020\sso.jnlp
	click	mn	:link=14020;
	comment		Open LA products and download sso.jnlp file
	click	mn	:link=LA;
	setFilePath	mn	E:\Gerkman\iTestFiles\1014\sso.jnlp
	click	mn	:link=1014;

Slika 11. Prikaz pravilnega zaporedja ukazov za shranjevanje bližnjic.

Omenjena funkcionalnost predstavlja proceduro *Save_sso_file*.

4.4 Avtomatizacija nastavitve parametrov v aplikaciji klicni strežnik

V tem podpoglavju sem opisal postopek avtomatizacije nastavljanja parametrov na klicnem strežniku ter gradnjo in uporabo map z odzivi.

4.4.1 Zagon aplikacije klicni strežnik s pomočjo skripte

V razdelku 3.2.2 sem opisal katere parametre je potrebno nastaviti na klicnem strežniku za uspešen prenos faks sporočila.

Aplikacija klicni strežnik je narejena v java swing okolju, zato sem v orodju iTest uporabil swing sejo. Neposreden vnos poti do bližnjice za zagon aplikacije klicni strežnik v swing sejo ni deloval, saj iTest ni prestregel aplikacije klicni strežnik za snemanje. Težava je bila v tem, da je v swing sejo potrebno neposredno vpisati pot do JAR datotek, ki jih zaganjamo. V mojem primeru pa imam JNLP datoteko, ki jo je potrebno zagnati s pomočjo programa javaWS. Ob zagonu programa javaWS sem moral nastaviti spremenljivko okolja (angl. *environment variable*) JAVA_TOOL_OPTIONS [8]. Ta spremenljivka okolja ima več opcij [9, 10]. Te so sledeče:

- *javaagent*: omogoča nadzorovanje java swing aplikacij,
- *Xbootclasspath/a*: z njo podamo pot do preprostih java programov, ki bodo ob zagonu zagnali kompleksnejše javanske programe,
- *D*: omogoča dodajanje XML parserja; z XML parserjem preberemo vsebino bližnjice.

Opisane opcije sem zapisal v skripto, ki je prikazana na sliki 12. Skripti sem dodal pot do programa javaWS in pot do iTest-ovih programov za delo z java swing aplikacijami. S pomočjo te skripte sem povezal iTest ter aplikacijo klicni strežnik, ki sem jo snemal [11].

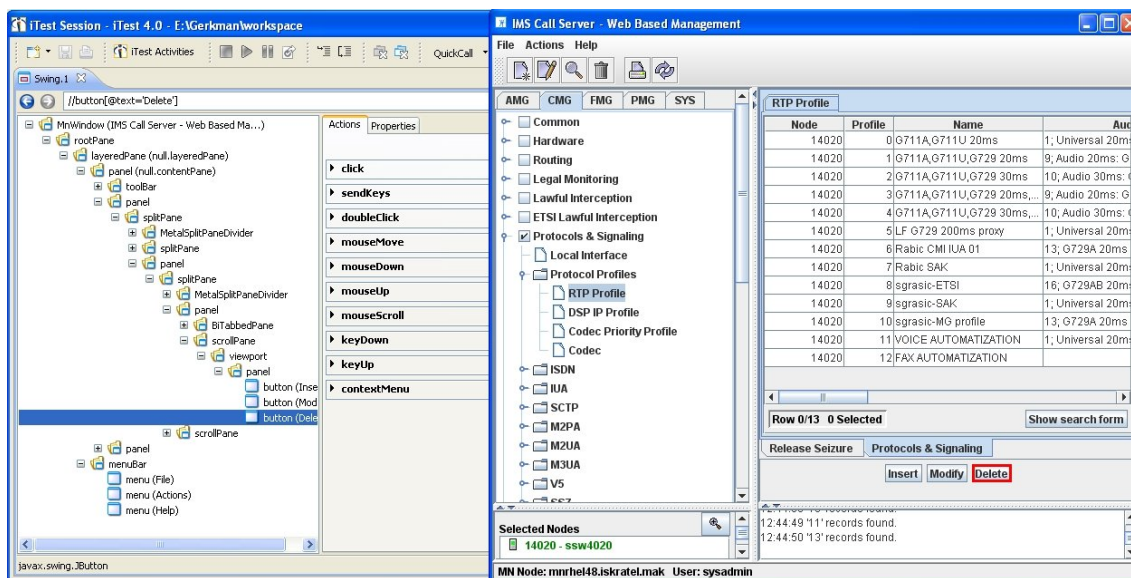
```

javaws1.launch.bat - Notepad
File Edit Format View Help
set ITEST_HOME=C:\Program Files\Fanfare\iTest 4.0\
set SWING_RESOURCES=%ITEST_HOME%\plugins\com.fnfr.svt.applications.java.swing.resources_4.0.0.49612\
set JAVAWS=C:\Program Files\Java\jre1.5.0_15\bin\javaws.exe
set JAVA_TOOL_OPTIONS=-javaagent:"%SWING_RESOURCES%\SwingTestHarness.jar"
                    -xbootclasspath/a:"%SWING_RESOURCES%\SwingTestHarnessInterfaces.jar";
                    "%SWING_RESOURCES%\SwingTestHarness.jar";
                    -Djavax.xml.parsers.DocumentBuilderFactory=com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl
"%JAVAWS%" -verbose -wait E:\Gerkman\iTestFiles\14020\ssw.jnlp

```

Slika 12. Prikaz vsebine skripte za povezavo orodja iTest in aplikacije klicni strežnik.

V swing seji sem izbral opcijo *Launch a custom application* in podal pot do skripte. Shranil sem nastavitve in zagnal sejo. Aplikacija klicni strežnik se je uspešno zagnala, iTest pa jo je prestregel. V orodju iTest se je izpisalo drevo z gradniki, ki so prisotni v aplikaciji, kot je to razvidno iz slike 13. Na desni strani slike 13 je vidna aplikacija klicni strežnik, ki sem jo snemal. Gumb, ki je izbran v drevesni strukturi na levi strani, je na desni strani obkrožen z rdečo obrobo. V mojem primeru je to gumb izbriši. V levem oknu vidimo tudi ostale akcije ter nastavitve, ki jih lahko posnamemo. Oboje lahko vidimo v zavihkih, desno od drevesa z gradniki na sliki 13.



Slika 13. Prikaz snemanja akcij v orodju iTest s swing sejo.

4.4.2 Snemanje nastavljanja parametrov na klicnem strežniku

Vhodne podatke, glavni program ter klice procedur v zanki sem že pripravil, zato sem pričel s snemanjem nastavljanja parametrov na klicnem strežniku. Z orodjem iTest sem posnel različne dogodke, ki bi se utegnili zgoditi med izvajanjem programa. Pri kreiranju novih profilov na klicnem strežniku ne sme biti podvojenih profilov. Zato sem uporabil pravila (angl. *analysis rule*), ki preverijo odziv ukaza in na podlagi odziva ukaza spremenijo tok programa.

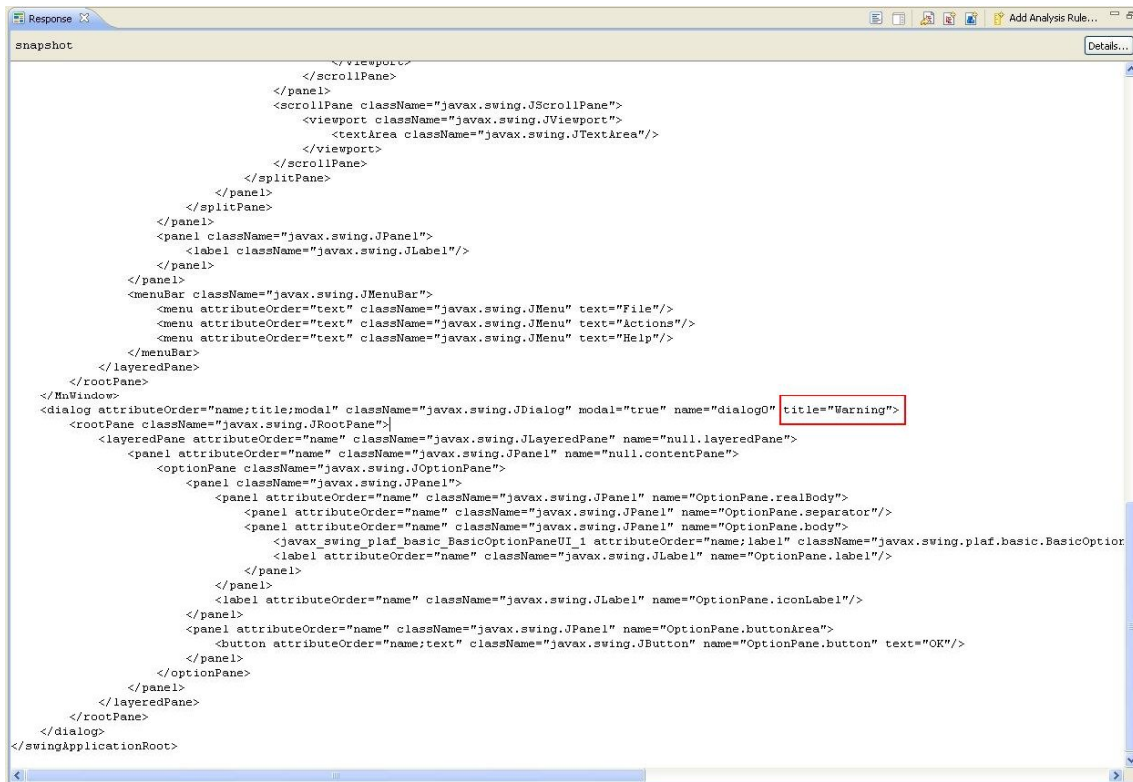
Slika 14 prikazuje:

- uporabo pravil na primeru, ko preverjam preseženo število ustvarjenih profilov; ustvarim lahko maksimalno 50 profilov,
- shranjevanje odziva v spremenljivko; na primeru sem shranil identifikacijsko številko v spremenljivko *SIPCodec_Number*.

click	swing	://button[@text='Insert']; left
getText	swing	://textField[@label='Codec'];
analyze		
regex		^\d+\$
store		SIPCodec_Number
if		[gget SIPCodec_Number] == 51
then		
click	swing	://button[@text='Cancel']; left
call		releaseCodecsAndProfiles
clickNode	swing	://FwTree; CMG/Protocols & Signaling/Protocol Profiles/Codec
click	swing	://button[@text='Insert']; left

Slika 14. Prikaz uporabe pravil in shranjevanja odziva.

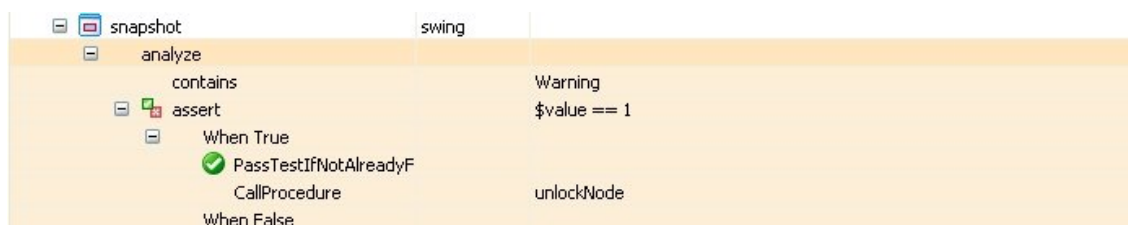
V primeru, ko sem presegel dovoljeno število profilov, sem klical proceduro *releaseProfiles*, ki preveri, ali so profili še aktivni in jih temu primerno izbriše. Program ob vsakem novem profilu shrani identifikacijske številke profilov s pomočjo pravil. Z identifikacijskimi številkami profilov program natančno ve, kateri profil nastavlja.



Slika 15. Prikaz odziva ukaza posnetek, ko se pojavi opozorilno okno.

Ko sem iskal dogodke, ki bi se lahko zgodili med avtomatizacijo in bi vodili k napaki, sem ugotovil, da so vozlišča lahko zaklenjena in se ne sprostijo, ko se aplikacija zapre. Težava se zgodi, ko je aplikacija klicni strežnik nepravilno ustavljena. Dogodek mi je uspelo posneti z ukazom posnetek (angl. *Snapshot*), ki zna prebrati drevesno strukturo gradnikov nadzorovanega programa. Ob dogodku se je pojavilo opozorilno okno z imenom *Warning*. Slika 15 prikazuje odziv ukaza posnetek, ko se je pojavilo opozorilno okno.

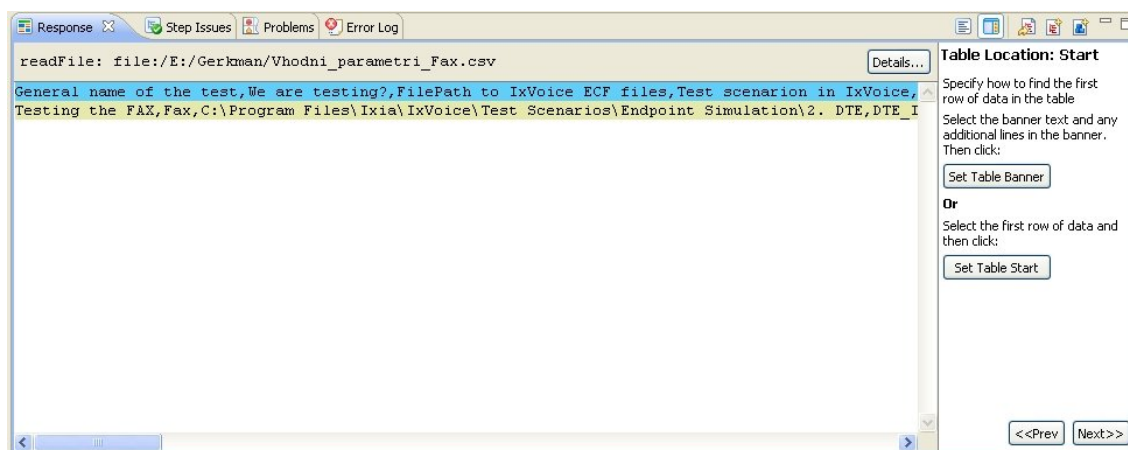
S pomočjo pravila sem preverjal ali odziv ukaza posnetek vsebuje ime opozorilnega okna. Če se je opozorilno okno pojavilo v odzivu, sem klical proceduro *unlockNode*. Procedura *unlockNode* se v aplikaciji klicni strežnik poveže na sistemske nastavitve, najde zaklenjeno vozlišče in ga odklene. Slika 16 prikazuje preverjanje zaklenjenega vozlišča in klic procedure *unlockNode*.



Slika 16. Prikaz preverjanja zaklenjenosti vozlišča.

4.4.3 Gradnja map z odzivi za vhodne podatke

Z ukazi *readFile*, ki so opisani v podpoglavju 4.1, sem v orodju iTest prebral vhodne datoteke, vsebino pa sem lahko takoj videl v polju odziv. Iz odzivov sem začel graditi mape z odzivi. Pogled *iTest Response Mapping* me je vodil skozi korake, kjer sem nastavil, kaj je glava oziroma začetek (angl. *Banner*) datoteke. Slika 17 prikazuje del kreiranja mape z odzivi, kjer nastavljam glavo vhodne datoteke. Glede na glavo datoteke potem iTest sam določi imena stolpcev. Imena stolpcev lahko določimo tudi sami.



Slika 17. Prikaz kreiranja mape z odzivi. Določitev glave v vhodni datoteki.

Med gradnjo mape z odzivi sem za ločilni znak (angl. *delimiter*) med stolpci izbral vejico. Pri vhodni datoteki, ki jo prikazuje slika 17, je vejica dober ločilni znak med parametri. Pri datoteki s kombinacijami testov se vejica ni izkazala kot dober ločilni znak. Vejica se pojavi v

imenu nekaterih kodekov, kot recimo pri *G.711 at 64 kbit/s, A-law* in mi tako razdeli ime kodeka na dva dela. Slika 18 prikazuje težavo z ločilnim znakom in rešitev težave ob izbiri boljšega ločilnega znaka – dvojni narekovaj.



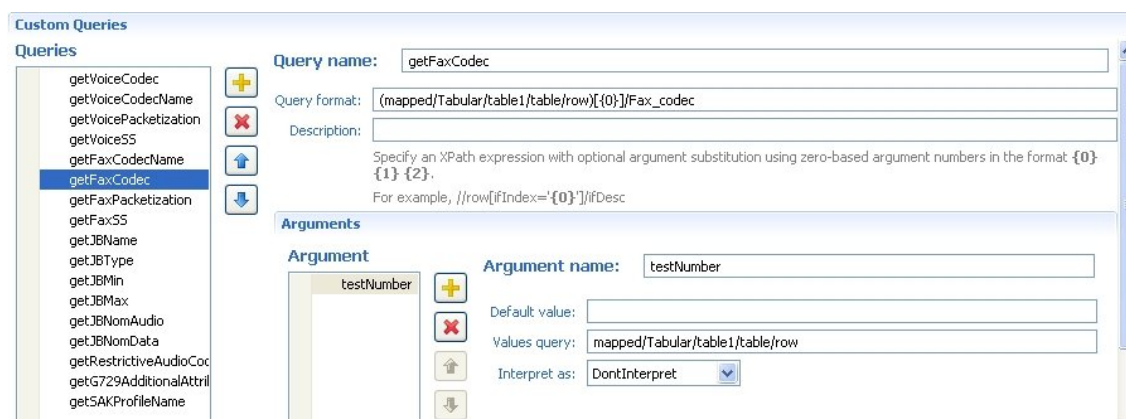
name	Codec & Voice Codec	Prj
G.711 at 64 kbit/s,	A-law,	10
G.711 at 64 kbit/s,	A-law,	10
G.711 at 64 kbit/s,	A-law,	20
G.711 at 64 kbit/s,	A-law,	20
G.711 at 64 kbit/s,	A-law,	30
G.711 at 64 kbit/s,	A-law,	30
G.711 at 64 kbit/s,	u-law,	10
G.711 at 64 kbit/s,	u-law,	10
G.711 at 64 kbit/s,	u-law,	20
G.711 at 64 kbit/s,	u-law,	20
G.711 at 64 kbit/s,	u-law,	30
G.711 at 64 kbit/s,	u-law,	30
G.711 at 64 kbit/s,	u-law,	40
G.711 at 64 kbit/s,	u-law,	40
G.711 at 64 kbit/s,	u-law,	10
G.711 at 64 kbit/s,	u-law,	10
G.711 at 64 kbit/s,	u-law,	20
G.711 at 64 kbit/s,	u-law,	20
G.711 at 64 kbit/s,	u-law,	30
G.711 at 64 kbit/s,	u-law,	30
G.711 at 64 kbit/s,	u-law,	40
G.711 at 64 kbit/s,	u-law,	40

name	Codec & voice Codec	Prj
"G.711 at 64 kbit/s, A-law"	"10"	
"G.711 at 64 kbit/s, A-law"	"10"	
"G.711 at 64 kbit/s, A-law"	"20"	
"G.711 at 64 kbit/s, A-law"	"20"	
"G.711 at 64 kbit/s, A-law"	"30"	
"G.711 at 64 kbit/s, A-law"	"30"	
"G.711 at 64 kbit/s, A-law"	"40"	
"G.711 at 64 kbit/s, A-law"	"40"	
"G.711 at 64 kbit/s, u-law"	"10"	
"G.711 at 64 kbit/s, u-law"	"10"	
"G.711 at 64 kbit/s, u-law"	"20"	
"G.711 at 64 kbit/s, u-law"	"20"	
"G.711 at 64 kbit/s, u-law"	"30"	
"G.711 at 64 kbit/s, u-law"	"30"	
"G.711 at 64 kbit/s, u-law"	"40"	
"G.711 at 64 kbit/s, u-law"	"40"	

Slika 18. Težava pri izbiri vejice, ločilnega znaka med podatki v vhodni datoteki (levo). Ob izbiri dvojnega narekovaja kot ločilnega znaka, je težava odpravljena (desno).

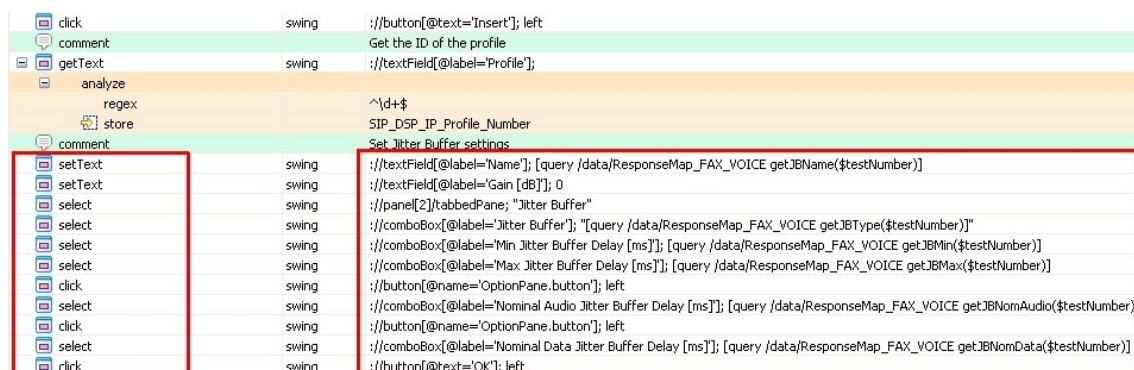
Nadaljeval sem z gradnjo map z odzivi. Dodal sem pravilo, da prva prazna vrstica v datoteki pomeni konec datoteke. Imena in število stolpcev je iTest znal razbrati sam, vedno pa ni tako. Če iTest ne razbere pravilno število stolpcev, lahko na koncu sami dodajamo stolpce in jih poimenujemo. Med gradnjo mape z odzivi si orodje *Response Map* sprti gradi strukturo podatkov, ter poizvedbe. S pomočjo poizvedb sem dostopal do želenih podatkov. Poizvedbam, ki jih zgradi orodje *Response Map*, sem dodal dodaten argument, s pomočjo katerega najdem želene podatke. Dodaten argument je številka vrstice oziroma številka testa (*testNumber*), kar prikazuje tudi slika 19. Poizvedba *getFaxCodec* vrne ime kodeka za faks v vrstici, določeni z argumentom *testNumber*.

V polju *Query format*, ki ga vidimo na sliki 19, sem dodal v zavutih oklepajih indeks argumenta, na katerega se bo poizvedba nanašala. Indeksi argumentov se začnejo z 0.

Slika 19. Prikaz nastavitve in dodatnega argumenta pri poizvedbi *getFaxCodec*.

4.4.4 Uporaba map z odzivi na primeru

Da sem v programu lahko uporabil več kot samo eno kombinacijo nastavitve omrežnih elementov, sem uporabil mape z odzivi. Na mestih, kjer program vpisuje podatke, sem dodal poizvedbe.



Slika 20. Prikaz uporabe poizvedb pri nastavitvah »jitter buffer«-ja.

Slika 20 prikazuje primer uporabe poizvedb za nastavljanje parametrov »jitter buffer«-ja. Na levem delu slike 20 je prikazan ukaz *setText*, ki vpiše vrednost v polje *textField*. Polje *textField* pa je prikazano na desnem delu slike 20. Polj *textField* je več in se razlikujejo po imenu. Preko imena polj se tudi sklicujem na tisto polje, ki mu želim spremeniti vrednost.

Ukaz *select* v gradniku kombinirano polje (angl. *Combo box*) izbere eno od možnosti. Vse te nastavitve lahko podam s pomočjo poizvedb, kot prikazuje slika 20. Opisal sem primer poizvedbe, ki vrne minimalno vrednost »jitter buffer«-ja:

- *[query /data/ResponseMap_FAX_VOICE getJBMin(\$testNumber)]*.

V iTest-u se na poizvedbe in globalne spremenljivke sklicuje v oglatih oklepajih. Nato beseda *query* določa, da imamo poizvedbo. Sledi pot od korena strukture, ki se imenuje *data*. Za tem je ime mape z odzivi, v mojem primeru je to *ResponseMap_FAX_VOICE*, nato pa sledi še ime poizvedbe. Imena poizvedb se začnejo z besedo *get*. V mojem primeru gre za poizvedbo *getJBMin()*, ki vrne minimalno zakasnitev »jitter buffer«-ja. V oklepaju ima poizvedba dodaten argument *testNumber*, s katerim sem poizvedbi določil v kateri vrstici se nahaja vrednost.

Funkcionalnosti, ki so predstavljene v razdelkih 4.4.2, 4.4.3 in 4.4.4, predstavljata proceduri:

- *OpenMN_SetSettingsOnSoftSwitch_closeMN*,
- *OpenMN_SetDefaultSettingsOnSoftSwitch_closeMN*.

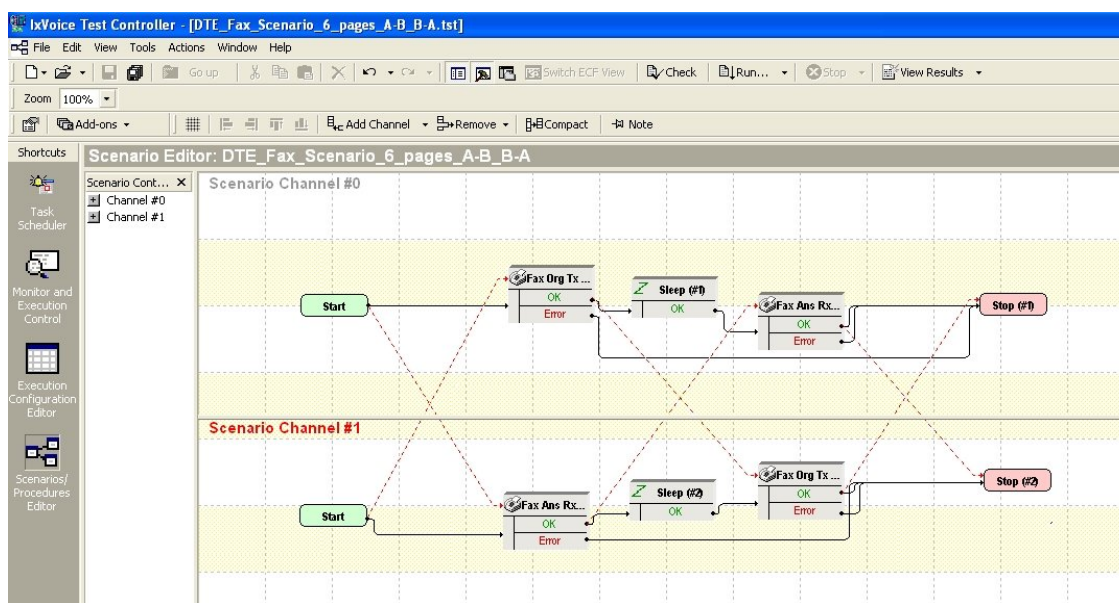
4.5 Gradnja scenarija pošiljanja faks sporočila

Lotil sem se gradnje scenarija pošiljanja faks sporočila med dvema končnima uporabnikoma z IxVoice aplikacijo. Ker imam dva končna uporabnika, sem naredil dva kanala. Scenarij vsebuje pet različnih gradnikov. To so gradniki:

- *Fax Org Tx*,
- *Fax Ans Rx*,
- *Sleep*,
- *Start*,
- *Stop*.

Kratica Tx pomeni prenos (angl. *transfer*) oziroma izvorno stran, Rx pa pomeni sprejem (angl. *receive*) oziroma ponorno stran. Pošiljanje se prične na prvem kanalu oziroma na *Channel #0*. Prvi gradnik je *Fax Org Tx* oziroma izvorna stran. Vzporedno z njim sem na drugi kanal oziroma *Channel #1* dodal gradnik *Fax Ans Rx* oziroma ponorno stran. Gradnik *Fax Org Tx* ima na levi strani vhod, na desni pa dva izhoda. Izhoda se imenujeta „OK“ in „Error“, vhod pa je neimenovan. Če je pošiljanje uspešno, se izvajanje nadaljuje skozi „OK“ izhod, sicer pa skozi „Error“ izhod.

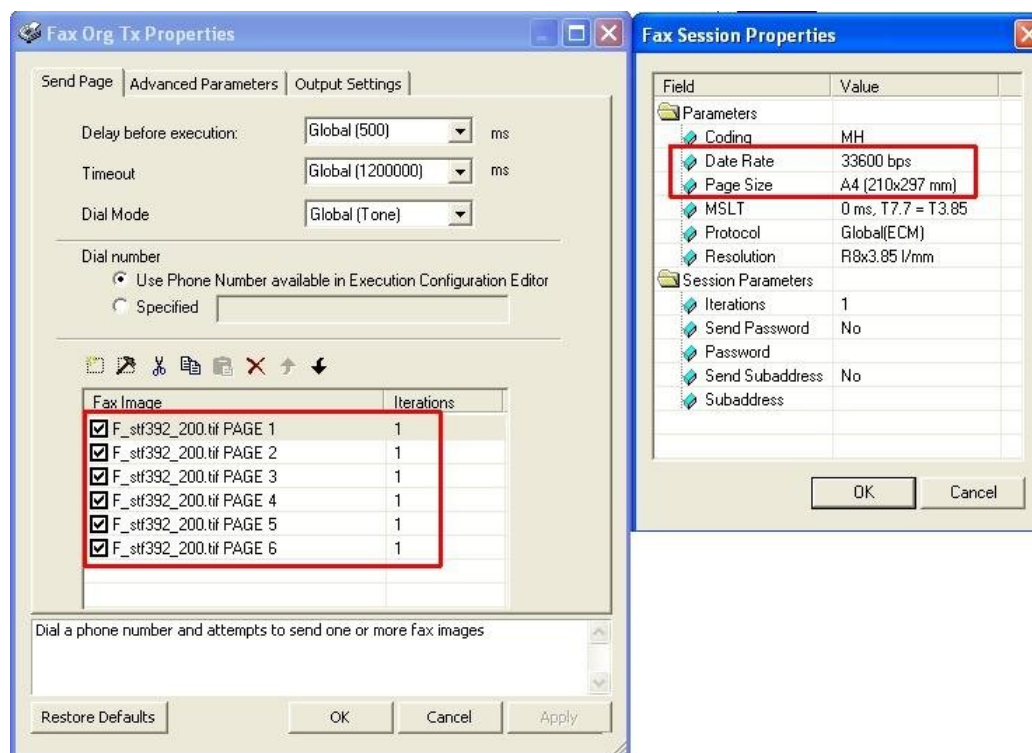
Vse izhode z napakami sem povezal z gradnikom *Stop*, ki konča simulacijo. Po uspešnem pošiljanju se izvajanje nadaljuje z gradnikom *Sleep*, ki za določen čas ustavi izvedbo scenarija. S tem počakam, da ponorna stran sprejme oddane podatke, preden se oddaja nadaljuje. Nato se izvede pošiljanje še v nasprotno smer. Slika 21 prikazuje scenarij pošiljanja faks sporočila med dvema končnima uporabnikoma.



Slika 21. Prikaz scenarija pošiljanja faks sporočila med dvema končnima uporabnikoma.

Na vsakem od gradnikov, ki predstavlja faks napravo, sem za začetek nastavil fiksno hitrost 33,6kbps ter katere testne strani bom prenašal. Izbral sem šest testnih A4 strani z različnimi vsebinami, od slik, podatkov, tabel do raznih likov in podobno. Različna vsebina strani

predstavlja vsestransko testiranje. Slika 22 prikazuje nastavljanje parametrov na gradniku *Fax Org Tx*.



Slika 22. Prikaz nastavljanja parametrov na gradniku *Fax Org Tx*.

4.6 Avtomatizacija izvajanja testov z *IxVoice*

Kot zadnja izmed procedur in hkrati tudi najpomembnejša je procedura *makeTest*. V iTest-u sem uporabil telnet sejo in se preko nje povezal na IxVoice. S pomočjo ukazov sem najprej odprl ECF datoteko ter nastavil vrednosti parametrov. Primer nastavljanja parametrov ECF datoteke preko telnet protokola prikazuje slika 23.

ECF datoteko sem nato shranil in zagnal test. Ob zagonu testa sem s pravili preverjal, ali se je test uspešno zagnal. Običajno se test ne zažene, če se kakšen od testov že izvaja. V tem primeru sem ustavil izvajanje vseh testov in zagnal trenutnega. Po uspešnem zagonu testa sem z ukazi *status-ecf*, *log-ecf* ter *summary-ecf* v zanki spremljal stanje testa.

```

ixv>
ixv>! CD C:\Program Files\Ixia\IxVoice\Test Scenarios\Endpoint Simulation\2. DTE
ixv>
ixv>set-lsf Iskratel_test.lsf
[OK] Current Library Settings File is 'C:\Program Files\Ixia\IxVoice\Pool\Iskratel_test.lsf'
ixv>
ixv>load-ecf DTE_IskratelFaxTest_Gerkman1.ecf
[OK] File C:\Program Files\Ixia\IxVoice\Test Scenarios\Endpoint Simulation\2. DTE\DTE_IskratelFaxTest_Gerkman1.ecf loaded.
ixv>
ixv>untag
[OK] All channels were deselected.
ixv>tag 47.49
[OK] New settings applied to 2 channels.
ixv>set-iteration 8 47.49
[OK] New settings applied to 2 channels.
ixv>set-log-level debug 47.49
[OK] New settings applied to 2 channels.
ixv>save-ecf DTE_IskratelFaxTest_Gerkman1.ecf
[OK] File C:\Program Files\Ixia\IxVoice\Test Scenarios\Endpoint Simulation\2. DTE\DTE_IskratelFaxTest_Gerkman1.ecf saved.
ixv>
ixv>run-ecf
[OK] Execution of 'C:\Program Files\Ixia\IxVoice\Test Scenarios\Endpoint Simulation\2. DTE\DTE_IskratelFaxTest_Gerkman1.ecf' ECF started.
ixv>
ixv>status-ecf
[OK] Connected with telnet client from 192.168.30.4.
Completed 93 ECF Execution Session(s) since started
Execution Session in progress: ECF 'C:\Program Files\Ixia\IxVoice\Test Scenarios\Endpoint Simulation\2. DTE\DTE_IskratelFaxTest_Gerkman1.ecf'
Execution Started at: 06/20/2011, 11:02:12.928
Current Time is: 06/20/2011, 11:04:22.537
Execution Elapsed Time: 129585 ms (<0 days, 00:02:09.609)
ixv>

```

Slika 23. Prikaz nastavljanja parametrov ECF datoteke preko telnet protokola.

Ko se je test zaključil, sem z ukazom *execution-flow* izpisal vse rezultate, ki jih vrne IxVoice za določen test. Rezultati, ki jih vrne ukaz *execution-flow*, so:

- stanje testa: ali je zaključeno ali prekinjeno,
- hitrost prenosa podatkov,
- paketizacija,
- zakasnitev paketov med pošiljanjem,
- število poslanih paketov,
- število sprejetih paketov,
- kategorija napak.

V orodju iTest sem dodal pravila, shranil rezultate v spremenljivke in jih nato dodal v poročilo. Poročilo je sestavljeno iz glave in telesa. Glava vsebuje:

- vrsto testiranja,
- čas začetka testiranja,
- ime ECF datoteke.

Telo poročila v prvem delu vsebuje nastavitve, s katerimi je bil test izveden:

- številka testa,
- kodek,
- paketizacija,
- stanje opcije zadušitev tišine,
- vrednosti parametrov »jitter buffer«-ja,
- protokol za hitrost prenosa faks sporočila.

V drugem delu vsebuje rezultate testiranja:

- stanje testa: ali je zaključeno ali prekinjeno,
- hitrost prenosa podatkov,
- čas prenosa podatkov,
- zakasnitev paketov med pošiljanjem,
- število poslanih paketov,
- število sprejetih paketov,
- kategorija napak,
- FOM.

Slika 24 prikazuje primer poročila, ki je v drugačni obliki kot originalno, zaradi lažje predstavitve podatkov in rezultatov. V originalu je poročilo zgrajeno tako, da ima glavo v svoji vrstici, imena parametrov in rezultatov v svoji vrstici, temu pa sledi serija parametrov in rezultatov, za vsak test v svoji vrstici.

Funkcionalnosti, ki so omenjene v tem poglavju, predstavlja procedura *makeTest*.

```

TEST RESULTS FOR MEASURING Fax
Test scenario name:,DTE_Iskratel\FaxTest_Gerkman1.ecf
Test started:,2011/06/17 12:37:01:409

Test number,                1
Codec,                      G.711 at 64 kbit/s A-law
Packetization,              20
Silence Suppression ON,    false
minJB,                      10
maxJB,                      40
nomAudioJB,                 20
nomDataJB,                  60
Protocol,                   v34
RESULTS
Channel status,             COMPLETED
Test duration,              0:03:29:030
Transmission speed,         33600bps
Expected Line Count,        1143
Received Line Count,        1143
Round Trip Delay,           226

Test number,                2
Codec,                      T.38
Packetization,              20
Silence Suppression ON,    false
minJB,                      10
maxJB,                      40
nomAudioJB,                 20
nomDataJB,                  60
Protocol,                   v17
RESULTS
Channel status,             COMPLETED
Test duration,              0:06:48:012
Transmission speed,         14400bps
Expected Line Count,        1143
Received Line Count,        1143
Round Trip Delay,           231

```

Slika 24. Prikaz primera poročila.

4.7 Izračunana primerjava časov med ročnim in avtomatiziranim testiranjem

Ker se faks funkcionalnosti zaradi deljene rabe materialnih virov ne morejo neprestano testirati, nisem mogel izvesti testnih scenarijev v celoti. Zato sem v tem poglavju opisal izračunane čase testiranja ter izračunan porabljen čas izvajalca.

Primerjavo med ročnim in avtomatskim testiranjem sem izračunal na podlagi krajših testiranj. Z orodjem IxVoice sem preizkusil trajanje prenosa faks sporočil v obe smeri. Prenos je trajal 18 minut. Gre za prenos dvanajstih slik, od katerih ima vsaka velikost 1 MB. Orodje IxVoice sicer zmanjša velikost slik za faktor osem (kompresija), potrebno pa je upoštevati še hitrost prenosa 14.4kbps. Upoštevamo še čas sinhronizacije med faks napravama ter čas čakanja (gradnik *Sleep*), da ponorna stran sprejme poslan promet.

Ročni prenos faks sporočil je, skupaj z nastavljanjem parametrov na klicnem strežniku in v orodju IxVoice, trajal 30 minut. Testiranje ene kombinacije je torej trajalo 30 minut. Enak postopek sem izvedel avtomatsko in testiranje ene kombinacije se je izvedlo v 20-ih minutah.

Čas, ki ga porabi izvajalec testov, je seštevek časa ročnega testiranja, časa potrebnega za pregled rezultatov in časa izdelave uradnega poročila o testiranju.

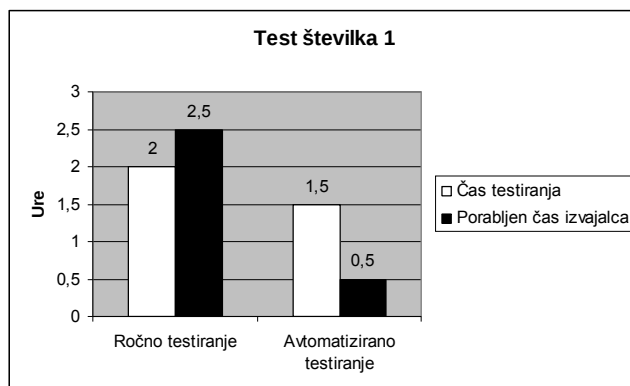
Na podlagi teh podatkov sem izračunal čase testiranja treh naborov testov. Čas izvajanja enega testa sem tako pri ročnem, kot pri avtomatiziranem testiranju pomnožil s številom testov. Pri vsakem od testov sem najprej prikazal nastavitve parametrov. Z grafom sem prikazal izračunano razliko časov testiranja in porabljenega časa izvajalca med ročnim in avtomatskim testiranjem. Pri spodnjih časih sem upošteval, da je okolje že postavljeno.

4.7.1 Test številka 1

S testom številka 1 testiramo osnovne nastavitve prenosa faks sporočila. Nastavitve so:

- kodek za prenos faks sporočila je G.711A ali T.38,
- hitrost prenosa faks sporočila je po protokolu V.17 (14.4kbps) ali V.34 (33.6kbps),
- paketizacija je 20ms,
- minimalna zakasnitev »jitter buffer«-ja je 10ms,
- maksimalna zakasnitev »jitter buffer«-ja je 40ms,
- nominalna zakasnitev »jitter buffer«-ja za zvok je 20ms,
- nominalna zakasnitev »jitter buffer«-ja za podatke je 20ms.

Slika 25 prikazuje izračunan čas testiranja in porabljen čas izvajalca med ročnim in avtomatiziranim testiranjem testa številka 1. Čas avtomatiziranega testiranja je manjši zaradi hitrejše izdelave poročila z rezultati.



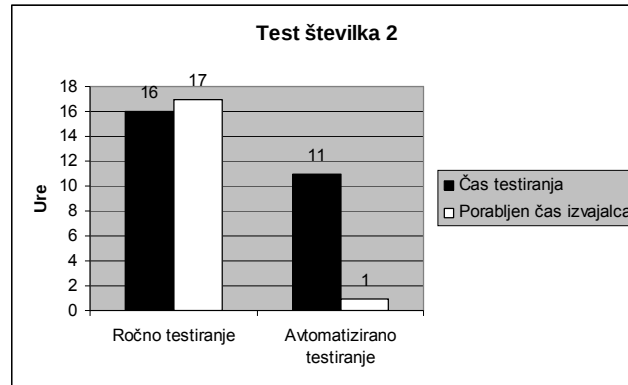
Slika 25. Grafični prikaz izračunanega časa testiranja in porabljenega časa izvajalca med ročnim in avtomatiziranim testiranjem.

4.7.2 Test številka 2

S testom številka 2 testiramo večje število kombinacij paketizacije in kodekov. Različnih kombinacij pri tem testu je 32. Nastavitve so:

- kodek za prenos faks sporočil, hitrosti prenosa faks sporočil ter vrednosti »jitter buffer«-ja so isti kot pri testu številka 1,
- kodek za prenos zvoka je G.711A ali G.729A,
- paketizacija se giblje od 10ms do 40ms.

Slika 26 prikazuje izračunan čas testiranja in porabljen čas izvajalca med ročnim in avtomatiziranim testiranjem testa številka 2. Čas avtomatiziranega testiranja je manjši zaradi hitrejšega nastavljanja vrednosti parametrov ter hitrejše izdelave poročila z rezultati.



Slika 26. Grafični prikaz izračunanega časa testiranja in porabljenega časa izvajalca med ročnim in avtomatiziranim testiranjem.

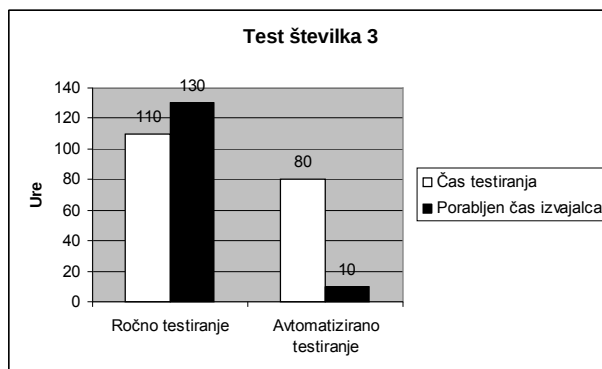
4.7.3 Test številka 3

S testom številka 3 testiramo manjše število kombinacij »jitter buffer«-ja, paketizacije in kodekov. Različnih kombinacij je približno 220.

Nastavitve so:

- kodek za prenos faks sporočil, hitrosti prenosa faks sporočil ter paketizacija so isti kot pri testu številka 2,
- minimalna zakasnitev »jitter buffer«-ja se giblje od 10ms do 30ms,
- maksimalna zakasnitev »jitter buffer«-ja se giblje od 20ms do 40ms,
- nominalna zakasnitev »jitter buffer«-ja za zvok se giblje od 20ms do 30ms,
- nominalna zakasnitev »jitter buffer«-ja za podatke se giblje od 20ms do 30ms.

Slika 27 prikazuje izračunan čas testiranja in porabljen čas izvajalca med ročnim in avtomatiziranim testiranjem testa številka 3. Čas avtomatiziranega testiranja je manjši zaradi hitrejšega kreiranja kombinacij testov, nastavljanja vrednosti parametrov ter hitrejšše izdelave poročila z rezultati.



Slika 27. Grafični prikaz izračunanega časa testiranja in porabljenega časa izvajalca med ročnim in avtomatiziranim testiranjem.

Segment, kjer avtomatizacija pridobi na času, je pri veliko hitrejšem nastavljanju parametrov na klicnem strežniku in na merilni opremi. Človek se je zmožen zelo velikokrat zmotiti pri vpisu kombinacij testov, medtem ko se računalnik ne. Avtomatizacija pridobi na času tudi pri pregledu rezultatov, saj so rezultati že v tabelarični obliki, primerni za Excel. Tako je tudi končno poročilo o testiranju veliko hitreje končano, saj se k uradnemu poročilu o testiranju enostavno doda tabela z rezultati. Človek običajno dela osem do deset ur na dan, medtem ko računalnik lahko dela 24 ur na dan in sedem dni v tednu. Pri izvajanju posameznega testa z merilno opremo pa razlik v času ni, saj je ta del isti tako pri ročnem kot pri avtomatiziranem testiranju.

Avtomatizacija ima največji prihranek pri porabljenem času izvajalca, ki je potreben za testiranje. Pri ročnem izvajanju testov mora biti ves čas prisotna oseba, ki nadzira izvajanje testov. Pri avtomatiziranem testiranju pa nadzor nad testiranjem ni potreben. Izvajalec testov mora s pomočjo programa kreirati vhodne podatke, zagnati postopek avtomatiziranega testiranja in na koncu rezultate testiranja dodati v uradno poročilo o testiranju. Med testiranjem lahko opravlja druga pomembna dela.

4.8 Rezultati testiranja

V tem razdelku sem za ponazoritev dodal rezultate merjenja prenosa faks sporočil, ki jih je pridobil sodelavec Simon Grašič [12] z ročnim testiranjem. Dokument, kjer sem dobil rezultate testiranja, je dostopen v Iskratelovi dokumentaciji. Simon Grašič je izvajalec testov, ki sem mu pomagal z avtomatizacijo testiranja.

4.8.1 Pričakovani rezultati testov

Štirje testi so odkrili, ali različne nastavitve kodekov na klicnem strežniku lahko privedejo do napak pri prenosu faks sporočila, zadnji test pa je odkril, če se tip spremenljivega »jitter buffer«-ja spremeni v fiksni tip »jitter buffer«-ja ob detekciji prenosa faks sporočila. Pri tem testu se mora opcija zadušitev tišine izklopiti, ko je zaznan prenos faks sporočila.

4.8.2 Dejanski rezultati testov

Omenjene teste v tem razdelku je ročno testiral sodelavec Simon Grašič. Rezultati so vzeti iz uradnega poročila o testiranju prenosa faks sporočil. Testiral se je prenos faks sporočila s kodekoma G.711 A-law in T.38, pri čemer se je zveza najprej vzpostavila s kodekom za zvok.

4.8.2.1 Prenos faks sporočila s kodekom G.711 A-law

Slika 28 prikazuje rezultate prenosa faks sporočila s kodekom G.711 A-law. Prenos je bil uspešen, saj je klicni strežnik uspešno detektiral prenos faks sporočila v vseh primerih.

Audio Mode: G.711 A-law, Fax Mode: G.711 A-law

With this measurement was tested if fax message is properly transmitted if Audio mode is set to G.711 A-law and fax mode is set to G.711 A-law. Tested were both fax classes 2.0 and 1.

Fax Modems used	Document - Page	Connection success	CLASS	Transmit time (s)	Connection speed (bps)	Errors/All lines
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	2.0	37	12751	4/2306
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	2.0	37	12752	0/2307

Slika 28. Prikaz rezultatov testiranja prenosa faks sporočila s kodekom G.711 A-law [12].

4.8.2.2 Prenos faks sporočila s kodekom G.729 A in G.711 A-law

Slika 29 prikazuje rezultate prenosa faks sporočila, kjer se je klic najprej začel s kodekom G.729 A, nato pa se je ob detekciji prenosa faks sporočila kodek uspešno zamenjal na kodek G.711 A-law v vseh primerih.

Audio Mode: G.729 A, Fax Mode: G.711 A-law

With this measurement was tested if fax message is properly transmitted if Audio mode is set to G.729 A and fax mode is set to G.711 A-law. Tested were both fax classes 2.0 and 1.

Fax Modems used	Document - Page	Connection success	CLASS	Transmit time (s)	Connection speed (bps)	Errors/All lines
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	2.0	37	12752	1/2307
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	2.0	37	12751	2/2307
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	2.0	37	12752	1/2307
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	1	32	14744	3/2301

Slika 29. Prikaz rezultatov testiranja prenosa faks sporočila s kodekoma G.729 A in G.711 A-law [12].

4.8.2.3 Prenos faks sporočila s kodekom G.711 A-law in T.38

Slika 30 prikazuje rezultate prenosa faks sporočila, kjer se je klic najprej začel s kodekom G.711 A-law, nato pa se je ob detekciji prenosa faks sporočila kodek zamenjal na T.38. V enem primeru se zveza s kodekom T.38 ni vzpostavila.

Audio Mode: G.711 A-law, Fax Mode: T.38

With this measurement was tested if fax message is properly transmitted if Audio mode is set to G.711 A-law and fax mode is set to T.38. Tested were both fax classes 2.0 and 1.

If CLASS2.0 was used, fax messages were not transmitted.

Fax Modems used	Document - Page	Connection success	CLASS	Transmit time (s)	Connection speed (bps)	Errors/All lines
U.S.Robotics-U.S.Robotics	FaxTest1.001	NOK	2.0	/	/	/
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	1	33	14298	0/2307
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	1	33	14296	2/2303

Slika 30. Prikaz rezultatov testiranja prenosa faks sporočila s kodekoma G.711 A-law in T.38 [12].

4.8.2.4 Prenos faks sporočila s kodekom G.729 A in T.38

Slika 31 prikazuje rezultate prenosa faks sporočila, kjer se je klic najprej začel s kodekom G.729 A, nato pa se je ob detekciji prenosa faks sporočila kodek zamenjal na T.38. V enem primeru se zveza s kodekom T.38 ni vzpostavila.

Audio Mode: G.729 A, Fax Mode: T.38

With this measurement was tested if fax message is properly transmitted if Audio mode is set to G.729 A and fax mode is set to T.38. Tested were both fax classes 2.0 and 1.

If CLASS2.0 was used, fax messages were not transmitted.

Fax Modems used	Document - Page	Connection success	CLASS	Transmit time (s)	Connection speed (bps)	Errors/All lines
U.S.Robotics-U.S.Robotics	FaxTest1.001	NOK	2.0	/	/	/
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	1	33	14297	1/2307
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	1	33	14296	2/2306

Slika 31. Prikaz rezultatov testiranja prenosa faks sporočila s kodekoma G.711 A-law in T.38 [12].

4.8.2.5 Testiranje »jitter buffer«-ja in opcije zadušitev tišine

Slika 32 prikazuje rezultate prenosa faks sporočila, kjer je bila pred vzpostavitvijo klica vklopljena opcija zadušitev tišine, »jitter buffer« pa je bil spremenljivega tipa. Ob detekciji prenosa faks sporočila se mora opcija zadušitev tišine izklopiti, »jitter buffer« pa mora spremeniti tip iz spremenljivega tipa v fiksni tip. Test se je uspešno končal v obeh primerih.

Adaptive Jitter buffer and Silence suppression ON

With this measurement was tested if application switches Jitter buffer to Fixed and sets Silence Suppression to OFF when recognizes fax connection.

Fax Modems used	Document - Page	Connection success	CLASS	Transmit time (s)	Connection speed (bps)	Errors/All lines
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	2.0	37	12750	0/2307
U.S.Robotics-U.S.Robotics	FaxTest1.001	OK	2.0	37	12751	0/2307

Slika 32. Prikaz rezultatov testiranja opcije zadušitev tišine in »jitter buffer«-ja [12].

4.8.3 Razlike med pričakovanimi in dejanskimi rezultati

Ročni testi so odkrili nekaj razlik med pričakovanimi in dejanskimi rezultati. Pri ročnem testiranju prenosa faks sporočila s kodekom T.38, je prišlo do težav v eni izmed ponovitev. Težava je najverjetneje bila na aplikaciji klicni strežnik, ki ni znala pravilno preklopiti iz kodeka za zvok na kodek za faks. Težava je potrebovala nadaljnjo analizo in odpravo problema v fazi razvoja.

5 Zaključek

V svojem diplomskem delu sem se lotil problema ročnega testiranja prenosa faks sporočila preko IP omrežja. Oddelku za verifikacijo in validacijo v podjetju Iskratel d.o.o. zelo veliko časa vzame testiranje faks funkcionalnosti na njihovih produktih. Ob vsaki uradni izdaji paketa ali ob končanju produkta je potrebno testirati prenose faks sporočil, njihovo kvaliteto in uspešnost prenosa. Kombinacij testov je zelo veliko in običajno ni časa in možnosti preveriti vseh. Izbere se nabor najpomembnejših in potem se testira samo te. Pred uradno izdajo produkta pa se testira večji nabor testov.

Zato sem avtomatiziral testiranje prenosa faks sporočila preko IP omrežja. Podjetje je kupilo orodje iTest, s katerim želi avtomatizirati čim večje število testov. Spoznal sem nova orodja in se srečal s težavami, ki sem jih tekom diplomskega dela, uspešno odpravil. Uspešno sem s spletno sejo shranil bližnjico do aplikacije klicni strežnik. S pomočjo skripte sem zagnal aplikacijo klicni strežnik in na njem avtomatsko nastavljal parametre z različnimi vrednostmi. Preko telnet protokola sem se povezal na orodje IxVoice, nastavil parametre in zagnal test prenosa faks sporočila. Na koncu sem zbral rezultate testiranja in kreiral poročilo z rezultati. V nadaljnje bi lahko h kreiranju poročila dodal še kriterije uspešnosti. Tudi v poročilu bi lahko bili nedoseženi kriteriji dobro označeni. Ob velikem naboru testov je poročilo lahko dokaj neberljivo, zato bi bilo dobro v prihodnje napisati program, ki bi iz vseh testov našel samo tiste, ki dosega izbrane kriterije.

V podpoglavju 4.7 sem napisal primerjavo izračunanih časov med ročnim in avtomatiziranim testiranjem. Glede na izračune, bi z avtomatizacijo prihranil veliko časa in povečal nabor različnih testov. Toda ta trenutek si ne morem privoščiti daljšega časa zasedanja omrežnih elementov z namenom testiranja večjega nabora testov. Omrežni elementi so nenehno v delovanju in se selijo iz enega maketnega prostora v drugega, kjer se izvajajo različni testi.

Ko sem izdeloval diplomsko nalogo, smo s sodelavci dobili več idej, kako bi lahko avtomatizacijo naredili uporabniku še bolj prijazno. Grafični vmesnik za vnos vhodnih podatkov bi naredil avtomatizacijo primerno za izvajanje ljudem, ki nimajo predhodnega

znanja o Iskratelovih produktih. Zamislili smo si grafični prikaz sheme omrežnih elementov, kjer bi s pomočjo miške povezal dva omrežna elementa, ter izbral vrsto testa. Program bi nato poskrbel za kombinacije testov in drugih vhodnih podatkov, ter sam zagnal testiranje preko orodja iTest.

Testiranje prenosa faks sporočil je zelo podobno testiranju analognih modemov in testiranjem v telefoniji. Moja diplomska naloga predstavlja osnovo za avtomatizacijo testiranja analognih modemov in testiranja v telefoniji preko IP omrežja.

V času izdelovanja diplomskega dela sem ogromno raziskoval ter testiral, da sem rešil probleme, s katerimi sem se srečal. Odkrival sem nove sposobnosti orodij, s katerimi sem delal, snemal in popravljaj procedure, ki jih je posnelo orodje iTest, pisal programe v javanskem okolju, da sem dosegel čim večjo stopnjo avtomatizacije. Želel sem doseči to, da bi uporabnik ali izvajalec testov moral vnesti čim manj podatkov, ki jih potrebuje program za izvajanje testov. S tem sem se izognil tudi številnim napakam, ki bi jih lahko prinesli napačno vpisani vhodni podatki.

6 Viri

[1] T. Vidmar, Informacijsko-Komunikacijski sistem, Založba Pasadena, Ljubljana, 2002

[2] (2011) Podatki o podjetju Iskratel d.o.o. Dostopno na:
http://www.iskratel.com/si/about_us/about_the_company/default.aspx

[3] (2011) »Opis »jitter buffer«-ja. Dostopno na:
<http://searchunifiedcommunications.techtarget.com/definition/jitter-buffer>

[4] (2011) Wikipedia, JavaWS in JNLP protokol. Dostopno na:
http://en.wikipedia.org/wiki/Java_Web_Start

[5] (2011) Oblika in vsebina JNLP datotek. Dostopno na:
<http://download.oracle.com/javase/1.4.2/docs/guide/jws/developersguide/syntax.html>

[6] (2011) Video vsebine o programu iTest. Dostopno na:
<http://wiki.fanfaresoftware.com/display/vid1/Learning+Center+-+Table+of+Contents>

[7] (2011) Forum na spletni strani podjetja Fanfare. Dostopno na:
<http://forums.fanfaresoftware.com/>

[8] (2011) Opis opcij okoljske spremenljivke JAVA_TOOL_OPTIONS. Dostopno na:
<http://download.oracle.com/javase/1.4.2/docs/tooldocs/solaris/java.html>

[9] (2011) Opis javaagent opcije. Dostopno na:
<http://javahowto.blogspot.com/2006/07/javaagent-option.html>

[10] (2011) Opis javaagent opcije. Dostopno na:
<http://download.oracle.com/javase/1.5.0/docs/api/java/lang/instrument/package-summary.html>

[11] (2011) Delna skripta za zagon swing aplikacij v iTest-u. Dostopno na:

<http://forums.fanfaresoftware.com/t5/GUI-Testing-Web-Java-Swing-etc/Swing-JNLP/mp/15209/highlight/true#M700>

[12] S. Grašič, Measurement test report fax and modem transmission quality on LA6111AX PSP070, koda dokumenta je WEV0376AD-MRE-010, 2005