

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleš Tepina

**Samodejna izdelava dokumentacije za sistem telefonije IP**

DIPLOMSKO DELO NA VISOKOŠOLSLEM STROKOVNEM ŠTUDIJU

Mentorica: Alenka Kavčič

Ljubljana, 2010



Št. naloge: 00505/2010

Datum: 15.03.2010

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

Kandidat: **ALEŠ TEPINA**

Naslov: **SAMODEJNA IZDELAVA DOKUMENTACIJE ZA SISTEM TELEFONIJE  
IP**

**AUTOMATIC PREPARATION OF DOCUMENTATION FOR THE IP  
TELEPHONY SYSTEM**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

V diplomski nalogi izdelajte ogrodje za samodejno pripravo dokumentacije za sistem telefonije IP. Ogrodje naj temelji na odprtih standardih, ki omogočajo prilagodljivo upravljanje s konfiguracijami omrežnih naprav ter pripravo omrežne dokumentacije v človeku berljivi obliki.

Proučite in podrobneje opišite sistem telefonije Cisco Unified Communications Manager ter raziščite možnosti upravljanja sistema preko Administrativnega XML, s pomočjo katerega rešite problem izvoza podatkov za potrebe samodejne izdelave dokumentacije v formatu HTML.

Realizirajte sistem za samodejno izdelavo dokumentacije ter ga preizkusite v praksi na simuliranem delovanju treh različnih konfiguracij sistemov za telefonijo IP. Ocenite prednosti uporabe sistema za samodejno izdelavo dokumentacije in podajte kritično presojo rezultatov realiziranega sistema. Navedite tudi predloge sprememb za njegovo nadgradnjo in izboljšave.

Mentor:

*Alenka Kavčič*  
viš.pred. dr. Alenka Kavčič



Dekan:

*Franc Solina*  
prof. dr. Franc Solina



**IZJAVA O AVTORSTVU**  
**diplomskega dela**

Spodaj podpisani Aleš Tepina,  
z vpisno številko 63020302,  
sem avtor diplomskega dela z naslovom:  
Samodejna izdelava dokumentacije za sistem telefonije IP

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom dr. Alenke Kavčič;
- 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:



## **ZAHVALA**

Za izkazano zaupanje, za mentorstvo, pomoč, vodenje in svetovanje pri pripravi diplomskega dela se najlepše zahvaljujem Alenki Kavčič.

Za pomoč pri naprednejših zagonetkah s programskim jezikom Perl se zahvaljujem Marku Zagožnu.

Za lektoriranje se iskreno zahvaljujem Katji Puhar.



## Kazalo vsebine

<b>1</b>	<b>Uvod</b> .....	<b>1</b>
1.1	Osnovni problem in cilj.....	1
<b>2</b>	<b>Teoretične osnove za rešitev problema</b> .....	<b>2</b>
2.1	<b>XML</b> .....	<b>2</b>
2.1.1	Kaj je XML?.....	2
2.1.2	Sintaksa.....	2
2.1.3	Pravilna oblika XML dokumenta.....	2
2.2	<b>Sistemi shem</b> .....	<b>3</b>
2.2.1	Web Services Description Language .....	3
2.2.2	XML Schema Definition.....	4
2.2.3	XSL Transformations .....	4
2.3	<b>Simple Object Access Protocol</b> .....	<b>6</b>
2.4	<b>Programski jezik Perl in XML</b> .....	<b>6</b>
<b>3</b>	<b>Cisco Unified Communications Manager</b> .....	<b>7</b>
3.1	Naloge strežnika Cisco Unified Communications Manager.....	7
3.2	Naprave v strežniku Cisco Unified Communications Manager.....	8
3.3	Končni uporabniki v strežniku Cisco Unified Communications Manager .....	9
3.4	<b>Usmerjanje klicev</b> .....	<b>11</b>
3.4.1	Usmerjevalni vzorci.....	13
3.4.2	Seznam poti.....	15
3.4.3	Usmerjevalna skupina.....	15
3.5	<b>Klicni privilegiji</b> .....	<b>16</b>
3.5.1	Razdelki in prostori klicnega iskanja .....	17
3.5.2	Prevajalni vzorci .....	17
3.6	<b>Vmesniki</b> .....	<b>18</b>
<b>4</b>	<b>AXL</b> .....	<b>19</b>
4.1	Definicija dokumenta AXL.....	20
4.2	Operacije vmesnika AXL.....	21
<b>5</b>	<b>Samodejna izdelava dokumentacije</b> .....	<b>22</b>
5.1	Shema za dokumentacijo.....	23
5.2	Opis in testiranje ukazne datoteke.....	26
5.2.1	Primer pridobivanja podatkov o napravah .....	27
5.3	Prikaz izdelane dokumentacije.....	30
<b>6</b>	<b>Sklepne ugotovitve</b> .....	<b>33</b>
6.1	Problemi z vmesnikom AXL.....	33
6.2	Smernice za nadaljnje delo .....	34
<b>A</b>	<b>Priloge</b> .....	<b>37</b>
	Seznam slik.....	38
	Seznam tabel .....	39
	Literatura .....	40

## Seznam uporabljenih kratic in simbolov

AAA	Administrative xml Application programming interface Access (dostop do vmesnika administrativnega XML)
API	Application Programming Interface (programski vmesnik, namenjen komunikaciji programa z operacijskim sistemom ali drugim računalniškim programom)
AXL	Administrative XML (administrativni XML)
CAR	Call detail records Analysis and Reporting database (podatkovna baza za hranjenje podrobnih podatkov o klicih, ki je uporabljena za analizo in poročila)
CAWS	Cisco Administrative XML Web Service (spletna storitev za administrativni vmesnik XML)
CCE	Contact Center Express (produkt podjetja Cisco Systems za kontaktne centre)
CDR	Call Detail Records (zapis, ki hrani podrobne informacije o posameznem telefonskem klicu)
Cisco JTAPI	Cisco Java Telephony Application Programming Interface (programski vmesnik za telefonijo za programski jezik Java)
Cisco TAPI	Cisco Telephony Application Programming Interface (programski vmesnik za telefonijo)
CMC	Client Matter Codes (storitev, ki omogoča vodenje in sledenje stroškov za določeno stranko)
COS	Class Of Service (skupina pravic za uporabnika, ki narekuje dodatne storitve, ki so uporabniku lahko omogočene)
CPAN	Comprehensive Perl Archive Network (zelo velik arhiv modulov in dokumentacije programskega jezika Perl)
CSS	Calling Search Space (urejen seznam usmerjevalnih razdelkov, ki so dodeljeni napravam)
CTI	Computer Telephone Integration (integracija telefona in storitev za telefon na računalniku)
CUCM	Cisco Unified Communications Manager
CUCMA	Cisco Unified Communications Manager Assistant (produkt podjetja Cisco Systems za pomoč tajnicam v podjetjih)
DRS	Disaster Recovery System (sistem za obnovitev in načrtovanje obnovitev po nesrečah)
EMS	Extension Mobility Service (storitev, ki uporabniku omogoča dostop do nastavitev in lastnosti svojega telefona preko drugega telefona)
FAC	Forced Authorization Codes (zahtevana koda za preverjanje dovoljenja pri uporabi omejevanja določenih klicev uporabnika)
HTML	Hyper Text Markup Language (označevalni jezik za oblikovanje večpredstavnostnih dokumentov)
HTTP	Hypertext Transfer Protocol (protokol za prenos hiperteksta)
HTTPS	Hypertext Transfer Protocol Secure (varni protokol za prenos hiperteksta)

## Seznam kratic in simbolov

IMSL	Identity Management System Library (knjižnica za upravljanje z imeniškimi storitvami)
IP	Internet Protocol (internetni protokol, namenjen prenosu podatkov)
ISDN	Integrated Services over Digital Network (digitalno omrežje z integriranimi storitvami)
IVR	Interactive Voice Response (interaktivni glasovni odziv – odziv naprave, ko razpozna zvočni signal ali glasovno sporočilo)
LAN	Local Area Network (lokalno omrežje)
LDAP	Lightweight Directory Access Protocol (lahki protokol za dostop do imenikov)
MIB	Management Information Base (skladišče upravljalških informacij, ki jih uporablja SNMP ali kak drug upravljalški protokol)
MPLS	Multi Protocol Label Switching (komutacija več protokolov z zamenjavo label oz. oznak)
PLAR	Private Line Automatic Ringdown (sistem za samodejno klicanje, ki ne uporablja telefonske centrale)
PSTN	Public Switched Telephone Network (javno komutirano telefonsko omrežje)
QOS	Quality Of Service (mehanizmi za ohranjanje kvalitete storitev)
SCCP	Skinny Call Control Protocol (protokol za nadzor in upravljanje govornih omrežnih naprav)
SIP	Session Initiation Protocol (protokol za zagon seje - protokol, ki specificira povezovanje omrežja IP in telefonskega omrežja)
SNMP	Simple Network Management Protocol (preprost protokol za upravljanje omrežja)
SOAP	Simple Object Access Protocol (protokol za izmenjavo podatkov pri spletnih storitvah, ki temelji na protokolu XML)
VoIP	Voice Over Internet Protocol (protokol, ki označuje skupino komunikacijskih tehnologij, ki za prenos govora uporabljajo omrežja IP)
W3C	konzorcij World Wide Web Consortium
WAN	Wide Area Network (prostrano omrežje)
WSDL	Web Services Description Language (jezik, ki narekuje opis spletnih storitev)
XML	Extensible Markup Language (razširljivi označevalni jezik - format podatkov za izmenjavo strukturiranih dokumentov v spletu)
XSD	Xml Schema Definition (definicijska shema protokola XML)
XSLT	Xml Transformation Schema (prevajalna shema za pretvorbo iz XML v kak drug format)

### POVZETEK

Diplomsko delo je namenjeno prikazu praktične uporabe tehnologij za upravljanje in sporočanje strežnika Cisco Unified Communications Manager. Le-ta nam kot nek centralni del služi za signalizacijo in prekodirni proces pri samih klicnih storitvah v našem omrežju. Z njegovo uporabo in uporabo telefonije IP lahko v podjetju drastično znižamo stroške in zvišamo produktivnost, hkrati pa zvišamo tudi sledljivost stroškov in obenem strankam ter zaposlenim ponudimo širši in bolj bogat nabor storitev.

Podjetje Cisco Systems je strankam ponudilo več možnosti upravljanja samega strežnika. Ena izmed njih je AXL oz. Administrativni XML, ki sem ga uporabil tudi pri tej diplomski nalogi. Z njim sem si pomagal pri reševanju problema izvoza podatkov oz. pri pripravi dokumentacije za večja podjetja, ki uporabljajo strežnik Cisco Unified Communications Manager. Izdelava dokumentacije, ki je vgrajena, je zelo skopa in nekonsistentna, kar pa nas lahko precej stane, če imamo podjetje z nekaj 100 uporabniki. V ta namen sem pripravil ukazno datoteko, napisano v jeziku Perl, ki nam izvoz podatkov avtomatizira. S pomočjo pretvornih datotek pa lahko izvožene podatke pretvorimo v obliko HTML ali kak drug željen format.

Cilj diplomske naloge je torej spremeniti večurno delo v nekajminutni avtomatiziran proces, ki je veliko bolj natančen, ne vsebuje človeškega faktorja in je zato precej bolj odporen na človeške napake.

### KLJUČNE BESEDE:

Perl, Cisco, telefonija IP, SOAP, XML, dokumentacija

### ABSTRACT

This Bachelor thesis provides the display of technology usage for managing and reporting of Cisco Unified Communications Manager. It serves as a central part of signalling and transcoding process with telephony services in our network. With its use and the use of IP telephony, we can drastically lower the costs and raise the productivity, improve the cost tracking and in the same time offer our customers and employees a wider and richer range of services.

Cisco Systems offered its customers several options for server management. One of them is AXL or the so called Administrative XML, which was used to achieve the goal, set in this thesis. It helped me to solve the problem of data export and documentation creation with larger companies that are using Cisco Unified Communications Manager. The built-in documentation creation is very miserly and inconsistent, and that alone can raise the cost of expenses, if we own a company with few hundred employees. I have created a Perl script for the purpose of automating the export process. The data can be converted to HTML or some other format with the help of transformation files.

The goal of this thesis is to change the usual hourly work to more accurate minute-long automated process that does not include the human factor, hence making it more error resistant.

### KEYWORDS:

Perl, Cisco, IP-telephony, SOAP, XML, documentation



## 1 Uvod

Na področju telefonije IP smo v zadnjih letih pričeli velikim spremembam. Številna podjetja so izvedla preklon z analogne telefonije na telefonijo IP zato, da so znižali stroške ter da so poenotili in poenostavili sam nadzor. Nekateri pa so se za prehod odločili zaradi številnega nabora dodatnih možnosti in aplikacij, ki jih sama telefonija IP ponuja. Telefonija IP pa je postala popularna tudi pri ponudnikih interneta za končne uporabnike, saj jo velika večina ponuja kar v paketu z internetno povezavo.

Telefonija IP se od klasične analogne (*PSTN*) tehnologije ter digitalne (*ISDN*) tehnologije razlikuje po tem, da za svoje delovanje uporablja kar omrežje IP. Tipi povezave so za telefonijo IP brezpredmetni. Lahko jo uporabimo znotraj lokalnega omrežja (*LAN*), v prostranem omrežju (*WAN*) ali v zadnjem času popularnih navideznih privatnih omrežjih (*MPLS*).

Eden izmed vodilnih ponudnikov strojne in programske opreme za telefonijo IP je ameriško podjetje Cisco Systems, ki na strežniškem tržišču telefonije IP prevladuje z izdelki »Cisco Unified Communications Manager« različnih velikosti. Sam izdelek je zastavljen tako, da ga pripravi in nastavi oseba, ki je prestala šolanje oziroma uradni tečaj za to vrsto izdelka.

### 1.1 Osnovni problem in cilj

Ko je strežnik Cisco Unified Communications Manager enkrat nastavljen in pripravljen za delo, je v večini primerov treba izdelati potrebno dokumentacijo za stranko, ki je izdelek kupila. V primeru velikega podjetja je izdelava takšne dokumentacije lahko precej zamudna, saj sam strežnik Cisco Unified Communications Manager ne omogoča izdelave dokumentacije.

Namen tega diplomskega dela je predstavitev uporabe administrativnega XML, ki ga izdelek Cisco Unified Communications Manager uporablja, ter izdelava programa oziroma ukazne datoteke, ki s pomočjo administrativnega XML jezika podjetjem olajša izdelavo same dokumentacije pri projektu.

Primer tipične dokumentacije vsebuje naslednje postavke:

- seznam vseh naprav,
- seznam končnih uporabnikov,
- seznam profilov mobilnih telefonskih priključkov (*Mobility extension*),
- seznam usmerjevalnih vzorcev (*Route pattern*),
- seznam usmerjevalnih seznamov (*Route list*),
- seznam usmerjevalnih skupin (*Route group*),
- seznam prostorov klicnega iskanja (*Calling Search Space*),
- seznam prevajalnih vzorcev (*Translation pattern*).

## 2 Teoretične osnove za rešitev problema

Za popolno razumevanje izdelane rešitve je potrebno razumeti osnove tehnologij, ki so uporabljene v samem strežniku Cisco Unified Communications Manager.

Namen tega poglavja je, da bralcu razjasnimo osnove teh tehnologij. S tem znanjem bo bralec lažje razumel dejansko rešitev in način delovanja le-te, hkrati pa bo razumel delovanje strežnika Cisco Unified Communications Manager.

### 2.1 XML

#### 2.1.1 Kaj je XML?

XML je odprt standard, ki narekuje skupek pravil, kako zapisati elektronske dokumente. Tekstovni podatkovni format lahko vsebuje zgolj besedilo, nikoli binarnih podatkov. Datoteko XML lahko odpremo z vsakim programom, ki zna brati tekstovne datoteke. Zapis v datoteki je zelo podoben zapisom HTML. Glavna razlika med zapisom XML in zapisom HTML je, da XML razlikuje velike in male črke.

Standard XML je definiran s specifikacijo XML 1.0, ki jo je izdal konzorcij W3C in je dostopna na [4].

Primer dokumenta XML:

```
<?xml version="1.0" encoding='UTF-8'?>
<Uporabnik>
  <UporabniskoIme>alest</UporabniskoIme>
  <Ime>Aleš</Ime>
  <Priimek>Tepina</Priimek>
  <TelefonskaStevilka>015556666</TelefonskaStevilka>
</Uporabnik>
```

#### 2.1.2 Sintaksa

Sintaksa XML je zapisana v specifikaciji XML. Je zelo preprosta in vključuje nekaj osnovnih pravil, ki so navedena v naslednjem razdelku. Opisani konstrukti so tisti nujni, ki še vedno omogočajo pravilno delovanje in dobro oblikovanost samega dokumenta.

Dokumenti XML naj bi se začeli (ni nujno) z deklaracijo XML. Če dokument XML vsebuje deklaracijo, potem mora biti deklaracija na prvem mestu v tem dokumentu. Pred njo se ne sme nahajati noben drug element (komentar, presledek itd.). Deklaracija XML bralcu pove nekaj osnovnih informacij o samem dokumentu.

#### 2.1.3 Pravilna oblika XML dokumenta

Vsak dokument XML (brez izjem) mora biti pravilno oblikovan. To pomeni, da mora upoštevati določena pravila. Nekaj najpomembnejših pravil je naštetih spodaj:

- vsaka začetna značka mora imeti svojo zaključno značko,
- elementi se lahko gnezdiijo, vendar se ne smejo prekrivati,
- datoteka mora imeti enega in ne več korenskih elementov,

- vrednosti atributov morajo biti citirane (zapisane v narekovajih),
- element ne more vsebovati dveh atributov z enakim imenom,
- komentarji in navodila za izvajanje se ne morejo pojaviti znotraj značk.

## 2.2 Sistemi shem

### 2.2.1 Web Services Description Language

Web Services Description Language ali WSDL je format dokumenta, ki temelji na jeziku XML in se uporablja kot slovnica za spletne storitve. Definira štiri vrste podatkov:

- javno dostopne metode – opise vmesnikov in formatov,
- tipe podatkov za zahteve in odgovore nanje,
- pripenjanje – kateri transportni protokol,
- informacija o naslovu – kje najti storitev.

Na sliki 1 so prikazani elementi WSDL, s katerimi opišemo spletno storitev. Mednje sodijo tipizacija podatkov, ki bodo izmenjani, definicija sporočil, lokacija spletnih storitev in drugi. S pomočjo teh opisov storitve lahko dostopamo do objektov in procedur z različnimi protokoli, kot so SOAP, MIME, HTTP.



Slika 1. Opisni elementi jezika WSDL.

Podjetje Cisco Systems nudi na strani za razvijalce [2] datoteko WSDL, ki lahko služi kot pripomoček pri komunikaciji z strežnikom Cisco Unified Communications Manager. S pomočjo te datoteke in ustreznega orodja (SoapUI, Eclipse z razširitvijo Axis) lahko brez znanja o protokolu SOAP enostavno zgradimo prave poizvedbe za pridobivanje ali nastavljanje podatkov. Za potrebe izdelave diplomske naloge sem uporabil orodje SoapUI, s katerim sem izdelal poizvedbe, ki sem jih nato uporabil v svoji ukazni datoteki. Več informacij o specifikaciji WSDL je na voljo na spletnih straneh konzorcija W3C [17],

brezplačno različico programske opreme SoapUI pa je mogoče dobiti na spletni strani programa [14].

### 2.2.2 XML Schema Definition

XML Schema Definition ali XSD je dokument XML, ki narekuje, kako naj bo sam dokument XML zgrajen. Narekuje elemente, ki se lahko pojavijo v dokumentu, in samo gnezdenje le-teh. Hkrati pa narekuje tudi, kolikokrat se nek element sploh lahko pojavi v dokumentu oz. znotraj vrhnjega elementa.

Dokument XSD vsebuje pravila, ki narekujejo, kako naj bo dokument XML zgrajen, da bo veljaven s to shemo XSD. Dokumenti sheme XML imajo ponavadi končnico .xsd.

Na sliki 2 lahko vidimo primer dokumenta sheme XML, ki na začetku vsebuje opis sheme, zatem sledi korenski element, ki vsebuje element `DeviceTable`. Ta element pa lahko vsebuje nič ali več elementov `Device` s podelementi `DeviceType`, `MacAddress` in `DeviceDescription`.

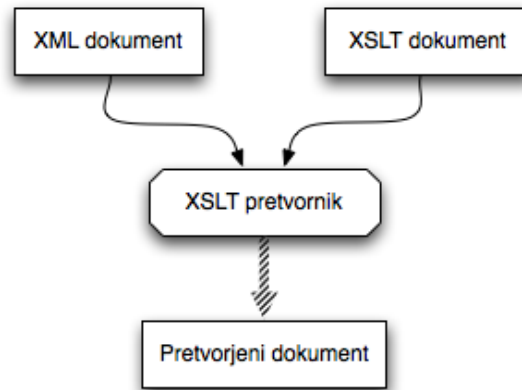
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- primer komentarja v xsd shemi -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fri="http://www.fri.uni-lj.si/"
targetNamespace="http://www.fri.uni-lj.si" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="CUCM">
    <xs:annotation>
      <xs:documentation>Root element</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="DeviceTable">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Device" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="DeviceType"/>
                    <xs:element name="MacAddress"/>
                    <xs:element name="DeviceDescription"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Slika 2. Primer dokumenta sheme XML.

### 2.2.3 XSL Transformations

XSL Transformations ali XSLT je dokument transformacijske sheme XML. To je dokument tipa XML, ki narekuje pravila za pretvorbo dokumenta XML v druge dokumente XML. Prvotni dokument se pri pretvorbi ohrani, zgradi pa se nov dokument, ki vsebuje podatke prvotnega dokumenta, pretvorjene po pravilih transformacijske sheme XML. Več informacij je na voljo na spletnih straneh konzorcija W3C [18].

Za izvedbo pretvorbe potrebujemo nek pretvornik. Pretvornik je lahko tudi internetni brskalnik ali samostojen program, ki zmore pretvarjanje datotek XSLT. Pretvornik običajno prejme dva parametra – transformacijsko shemo XSLT ter datoteko XML s podatki. Rezultat je pretvorjen dokument. Potek pretvorbe je prikazan na sliki 3.



Slika 3. Potek pretvorbe podatkov s pomočjo dokumenta XSLT.

Na sliki 4 je prikazan primer dokumenta XSLT, ki na začetku vsebuje opis dokumenta in format pretvorjenega dokumenta. Sledi predloga za izpis uporabnikov, ki bodo urejeni po polju `priimek`. Na koncu pa vsebuje še navedbo vseh polj, ki bodo prikazana pri izpisu uporabnikov.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/1999/xhtml">

  <xsl:output method="xml" indent="yes" encoding="UTF-8"/>

  <xsl:template match="/uporabniki">
    <html>
      <head> <title>XSLT primer</title> </head>
      <body>
        <h1>Uporabniki</h1>
        <ul>
          <xsl:apply-templates select="uporabnik">
            <xsl:sort select="priimek" />
          </xsl:apply-templates>
        </ul>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="uporabnik">
    <li>
      <xsl:value-of select="priimek"/><xsl:text>, </xsl:text>
      <xsl:value-of select="ime"/>
    </li>
  </xsl:template>
</xsl:stylesheet>
```

Slika 4. Primer dokumenta XSLT.

### 2.3 Simple Object Access Protocol

Simple Object Access Protocol ali SOAP [13] je prenosni protokol, ki skrbi za izmenjavo strukturiranih in tipiziranih podatkov, ki temeljijo na protokolu XML.

Specifikacija SOAP definira:

- format sporočila SOAP,
- kako pošiljati in sprejemati sporočila,
- kodiranje podatkov.

Na sliki 5 je prikazana struktura sporočila SOAP, ki vsebuje ovojnico z glavo in telesom sporočila. V telesu sporočila je lahko zapisana tudi napaka, s katero nam ponudnik storitve SOAP sporoči, da je prišlo do napake.



Slika 5. Struktura sporočila SOAP.

### 2.4 Programski jezik Perl in XML

Perl je programski jezik, katerega velika prednost je izvrstna manipulacija z besedilom. Njegova polna moč se pokaže v tem, da je popolnoma voden s strani razvojne skupnosti. Programerji jezika Perl napišejo module za reševanje različnih problemov oz. nalog, nato pa te module distribuirajo preko omrežja CPAN – Comprehensive Perl Archive Network. Manipulacija z moduli preko omrežja CPAN je zelo preprosta. Celoten sistem je zgrajen tako, da sam poskrbi za vse odvisnosti od programskega paketa – če nek modul za delovanje potrebuje še nek drug modul, uporabnik sam pa tega ne ve, ga CPAN pri nameščanju modula o tem obvesti in ponudi možnost namestitve tudi ostalih odvisnih modulov. Več o jeziku Perl in zmožnostih za delo z XML je na voljo na spletu [9] in v literaturi [10].

Nekaj modulov za delo z datotekami XML:

- XML::Parser,
- XML::Simple,
- XML::Dumper,
- XML::Grove,

- XML::Generator,
- XML::Writer,
- XML::SOAP.

### 3 Cisco Unified Communications Manager

Podjetje Cisco Systems je s pojmom »Cisco Unified Communications« predstavilo sistem, ki integrira komunikacije na način, da se podatki, zvok in video prenašajo preko ene mrežne infrastrukture s pomočjo protokolov IP. Družina produktov Cisco Unified Communications [16] je zasnovana tako, da omogoča popolno optimizacijo nabora funkcionalnosti, zmanjša čas postavitve produktov in njihovega vzdrževanja ter ponudi združljivost s širokim spektrom drugih naprav in aplikacij. Sistem ima visok nivo zagotavljanja kakovosti (*QoS*) in varnosti v omrežju.

Sistem Cisco Unified Communications združuje strojno in programsko opremo, kot so agenti za procesiranje klicev, telefoni IP (tako žični kot tudi brezžični), elektronske tajnice, video naprave in še mnogo drugih posebnih aplikacij.

Telefonija IP je tehnologija, ki za telefonske storitve uporablja kar mrežno infrastrukturo s pomočjo protokolov IP. Cisco Unified Communications Manager je sistem telefonije IP, ki je prvotno namenjen večjim podjetjem in nudi tako tradicionalne telefonske funkcije kot tudi napredne funkcije, na primer mobilnost uporabnikov, prisotnost uporabnikov, konference in druge.

#### 3.1 Naloge strežnika

Cisco Unified Communications Manager razširja telefonske storitve in funkcije podjetja do mrežnih naprav paketne telefonije. Te mrežne naprave paketne telefonije so: telefoni IP podjetja Cisco Systems, naprave za procesiranje multimedije, prehodi VoIP in multimedijske aplikacije.

Osnovne naloge strežnika Cisco Unified Communications Manager so:

- procesiranje klicev – se nanaša na celotni proces usmerjanja klica, ustvarjanja in končanja klica, vključno z zaračunavanjem in pridobivanjem statističnih podatkov,
- signalizacija in upravljanje naprav – priprava signalizacijskih povezav med končnimi točkami, kot so telefoni in prehodi, in konferenčnimi napravami za vzpostavitev in končanje tokovnih povezav,
- administracija klicnih poti – upravljanje s sezname, ki se uporabljajo za usmerjanje klicev,
- administracija dodatnih funkcionalnosti telefona – nudenje storitev, kot so »klic na čakanju«, »prenos klica«, »posredovanje klica«, »konferenca«, »hitro klicanje«, »ponovno klicanje zadnjega klica«, »parkiranje klica« in drugih storitev telefonov IP in prehodov,
- imeniške storitve – shranjevanje uporabniških informacij v lastni podatkovni bazi; uporabniki imajo možnost prestatati preverjanje pristnosti lokalno ali s pomočjo zunanjega imeniškega strežnika (Microsoft Active Directory 2000 in 2003, Sun ONE 5.2, iPlanet 4.1, LDAP),

- nudenje programskega vmesnika zunanjim napravam in aplikacijam – uporaba programskih telefonov npr. Cisco IP Softphone, Cisco IP Communicator, Cisco Unified IP Interactive Voice Response (*IVR*), Cisco Personal Assistant in Cisco Unified Communications Manager Attendant Console,
- upravljanje z varnostnimi kopijami sistema – uporaba sistema »Disaster Recovery Sistem« (*DRS*), ki nudi shranjevanje in upravljanje z varnostnimi kopijami konfiguracijske baze, baze podrobnih podatkov klicev (*Call Detail Records – CDR*) in baze analize in poročanja (*Cisco Unified Communications Manager CDR Analysis and Reporting database – CAR*).

### 3.2 Naprave

Cisco Unified Communications Manager omogoča priklop različnih končnih naprav podjetja Cisco Systems in tudi naprav različnih drugih proizvajalcev. Med končne naprave sodijo telefoni IP, video naprave in prehodi za analogno telefonijo, ki omogočajo analognim telefonom interakcijo s strežnikom Cisco Unified Communications Manager.

Protokoli, podprti v strežniku Cisco Unified Communications Manager, so naslednji:

- **SCCP - Skinny Call Control Protocol** [11] (protokol podjetja Cisco Systems, ki je večinoma uporabljen zgolj na končnih napravah Cisco, izjema so nekateri zunanji proizvajalci);
- **SIP - Session Initiation Protocol** [12] (protokol, ki sta ga prvotno razvila Henning Schulzrinne in Mark Handley leta 1996 in je pogosto uporabljen v multimedijskih komunikacijah);
- **H.323** [6] (protokol, ki ga je predlagala organizacija Telecommunication Standardization Sector v imenu organizacije International Telecommunication Union).

Uporaba protokola SCCP za komunikacijo s samim strežnikom Cisco Unified Communications Manager nudi bogat nabor telefonskih funkcij, medtem ko uporaba protokolov SIP in H.323 nudi manjši nabor funkcij – v večini primerov so to zgolj osnovne telefonske funkcije.

Razlika med modeli končnih naprav Cisco se odraža v:

- velikosti ekrana, barvi ekrana, zaslonu na dotik,
- podprtih kodirnikih,
- hitrosti lokalnega omrežja,
- gumbih,
- podpori za naglavno slušalko in vgrajenem zvočniku za prostoročno govorjenje,
- številu telefonskih linij in drugih posebnih lastnostih, kot so video, konferenčna postaja in brezžično omrežje.

Različne vrste končnih naprav, ki jih podpira strežnik Cisco Unified Communications Manager, so zbrane v tabeli 1, v kateri smo jih razvrstili tudi po tipu uporabljenega protokola.

Telefoni IP Cisco Unified SCCP in SIP	<b>Tip A:</b> 7940, 7960, 7905, 7912
	<b>Tip B:</b> 7906, 7911, 79[46][125],797[015]
Programski telefon Cisco	Cisco IP Communicator
Druge končne naprave Cisco (samo SCCP)	7902, 7910 in 7931 ( <b>telefoni IP</b> )
	7920 in 7921 ( <b>brezžični telefoni</b> )
	7935 in 7936 ( <b>konferenčne postaje</b> )
	7985 ( <b>namizni video telefon</b> )
Končne naprave drugih proizvajalcev	<b>SCCP:</b> Nokia dual mode cell phone SCCP client, Tandberg video končne naprave, IP Blue VTGO in drugi
	<b>SIP:</b> različni fizični in programski telefoni
	<b>H.323:</b> različni fizični in programski telefoni

Tabela 1. Prikaz telefonov IP podjetja Cisco Systems in drugih proizvajalcev, ki so podprti v samem strežniškem sistemu Cisco Unified Communications Manager.

### 3.3 Končni uporabniki

Cisco Unified Communications Manager vsebuje kar nekaj lastnosti, ki so povezane z uporabniškim računom končnih uporabnikov in administrativnimi pravicami. Uporabniški računi se upravljajo s konfiguracijskimi orodji ali preko integracije z imeniškim strežnikom.

Lastnosti strežnika Cisco Unified Communications Manager, ki koristijo uporabniške račune:

- Uporabniške in administrativne spletne strani:
  - Uporabniške spletne strani strežnika Cisco Unified Communications Manager,
  - Administrativne strani strežnika Cisco Unified Communications Manager,
  - Uporabnost (*servicability*) strežnika Cisco Unified Communications Manager,
  - Administracija operacijskega sistema v strežniku Cisco Unified Communications Manager,
  - Upravljanje z varnostnimi kopijami v strežniku Cisco Unified Communications Manager,
- Aplikacije:
  - Cisco Unified Communications Manager Attendant Console,
  - Cisco Unified Communications Manager Extension Mobility,
  - Cisco Unified Communications Manager Assistant,
- Imeniki,
- Storitve Cisco telefonije IP.

S telefoni IP podjetja Cisco Systems je možno brskati po imeniških storitvah in iskati uporabnika po imenu. Zato mora strežnik Cisco Unified Communications Manager poznati uporabnike in njihove telefonske številke. Storitve na samih telefonih so lahko nastavljene tako, da je potrebna prijava uporabnika z uporabniškim imenom in PIN-kodo. Takrat se zahteva za prijavo pošlje notranji knjižnici, imenovani »Identity Management System Library«, ki je odgovorna za preverjanje pristnosti uporabnikov.

V tabeli 2 sta predstavljena dva tipa uporabniških računov in glavne razlike med njima. Za izdelavo dokumentacije za podjetja je pomemben predvsem tip končnega uporabnika, ki je tudi edini prikazan v izpisu dokumentacije.

<b>Končni uporabniki</b>	<b>Aplikacijski uporabniki</b>
Asociacija s posamezno osebo	Asociacija z aplikacijo
Za osebno uporabo in interaktivno prijavo	Za ne-interaktivno prijavo
Za dostop do uporabniških lastnosti in posameznih administrativnih prijav	Za avtorizacijo aplikacije
Vključen v uporabniškem imeniku	Ni vključen v uporabniški imenik
Lahko se uporablja zunanji imeniški strežnik ( <i>LDAP</i> )	Zunanji imeniški strežnik ni možen

Tabela 2. Dva tipa uporabniških računov v strežniku Cisco Unified Communications Manager.

Vsak izmed končnih uporabnikov je asociiran s fizično osebo in interaktivno prijavo. V to skupino spadajo končni uporabniki in administratorji.

Aplikacijski uporabniki so povezani z lastnostmi strežnika Cisco Unified Communications Manager ali aplikacijami, kot so Cisco Unified Communications Manager Attendant Console, Cisco Unified Communications Manager Contact Center Express ali Cisco Unified Communications Manager Assistant. Takšne aplikacije morajo prestati preverjanje pristnosti s strežnikom Cisco Unified Communications Manager, da je njihovo delovanje omogočeno.

Tabela 3 prikazuje drugi tip uporabniških računov – aplikacijske uporabnike. Namen tabele je prikazati bralcu, da se aplikacijski uporabniki ločijo že po imenu, saj uporabljajo drugačne vrste uporabniških imen, kot bi jih administratorji navadno dodelili končnim uporabnikom. Na levi strani tabele pa so prikazane aplikacije, ki za delovanje potrebujejo dotične aplikacijske račune. Prva aplikacija v tabeli je ena izmed naših ciljnih, s pomočjo katere želimo pridobiti podatke za končni izpis dokumentacije pri tej diplomski nalogi.

Aplikacija	Aplikacijski račun
Cisco Extension Mobility Service	CCMSysUser
Cisco Unified Communications Manager Assistant Service	IPMA SecureSysUser, IPMASysUser
Cisco WebDialer Service	WDSEcureSysUser, WDSysUser

Tabela 3. Primer aplikacijskih uporabnikov v strežniku Cisco Unified Communications Manager.

### 3.4 Usmerjanje klicev

Klicni načrt (*Dial plan*) je eden izmed ključnih lastnosti telefonije IP. Sodi v samo jedro uporabniške izkušnje, saj narekuje pravila, po katerih bo uporabnik priklical končni cilj.

Naslavljanje končnih naprav in izbira poti sta najpomembnejši komponenti klicnega načrta.

Dosegljivost notranjih telefonskih priključkov je pogojena s pravilnim naslavljanjem in dodeljevanje imeniških števil (directory number) vsem končnim napravam (telefonom, faksom in analognim telefonom) in aplikacijam (sistemom za glasovno pošto, avtomatskim odzivnikom in konferenčnim sistemom).

Ločimo tri različne načine klicanja končnih naprav:

- **klicanje znotraj omrežja (*On-net*):** klici, ki ostajajo znotraj enega telefonskega sistema (primer: interni klic z enega telefona IP na drug telefon IP),
- **klicanje v drugo omrežje (*Off-net*):** klici, ki izvirajo v enem telefonskem sistemu in se končajo v drugem telefonskem sistemu (primer: klic s telefona IP v omrežje PSTN),
- **skrajšano klicanje (*Abbreviated*):** klic interne številke na neko številko zunaj telefonskega sistema preko kratke številke (primer: klic na krajšo številko s telefona IP v omrežje PSTN; v tem primeru mora strežnik Cisco Unified Communications Manager povezati skrajšano številko s polno številko PSTN).

Strežnik Cisco Unified Communications Manager omogoča različne vrste ciljev za usmerjanje telefonskega prometa. To pomeni, da klicana številka lahko ustreza kateri izmed usmerjevalnih komponent, ki so zapisane v tabeli 4, klic pa je nato usmerjen na eno izmed možnih entitet, ki so zapisane v tabeli 5.

Usmerjevalna komponenta	Opis
Imeniške številke	Številka, dodeljena vsaki končni napravi in aplikaciji. Uporablja se za notranje usmerjanje znotraj gruče.

Prevajalni vzorci	Za pretvorbo klicane številke in ponovni pregled cilja v usmerjevalni tabeli.
Usmerjevalni vzorci	Za usmerjanje klicev na cilje v drugem omrežju (s pomočjo prehodov) ali v druge gruče strežnikov Unified Communications Manager (s pomočjo povezave za povezovanje omrežij).
Številka usmerjevalnih storitev ( <i>hunt pilot</i> )	Usmerjanje klicev na člane skupine številke usmerjevalnih storitev, ki temelji na distribucijskem algoritmu (najdaljše čakanje v vrsti, krožno itd.).
Parkirne številke ( <i>Call park numbers</i> )	Postavitev klica na čakanje in sprejem istega klica nazaj z drugega telefona preko klicanja določene številke.
Konferenčne številke ( <i>Meet-me numbers</i> )	Ustvarjalcu konferenčne številke dopušča vzpostavitev konferenčnega klica, udeležencem pa dovoli pridružitve s klicanjem na konferenčno številko.

Tabela 4. Seznam usmerjevalnih komponent, ki so lahko uporabljene kot cilj nekega klica.

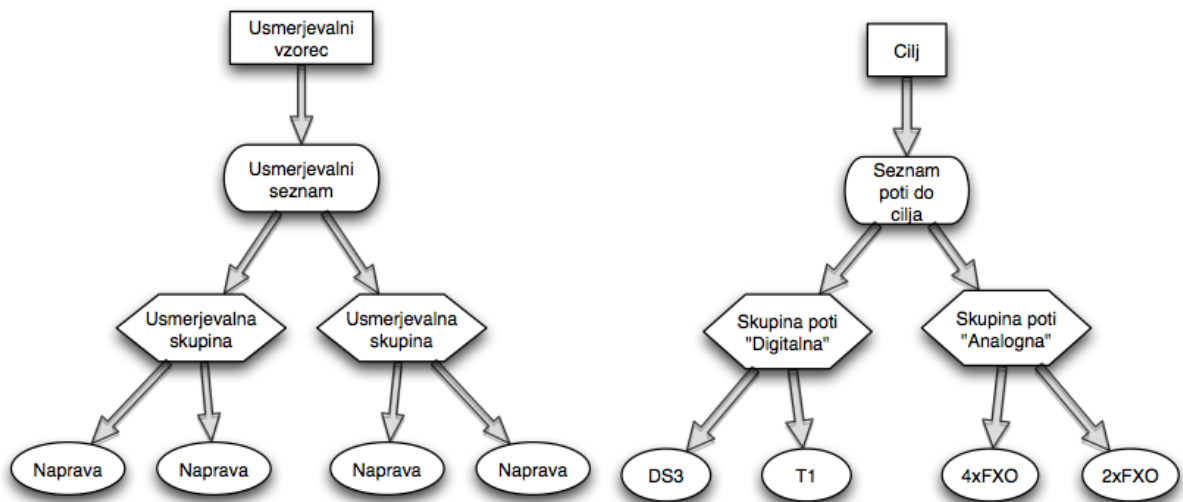
Entiteta	Opis entitete
Telefoni IP	Klic s telefona IP se poišče v usmerjevalni tabeli.
Povezave za povezovanje omrežij	Zahteva za klic, sprejeta preko povezave za povezovanje omrežij, se poišče v usmerjevalni tabeli.
Prehodi	Zahteva za klic, sprejeta preko prehoda, se poišče v usmerjevalni tabeli.
Prevajalni vzorci	Potem ko je bil najden prevajalni vzorec, ki se najbolj ujema, se pretvorjena številka ponovno poišče v usmerjevalni tabeli.
Glasovna pošta ( <i>voice mail</i> )	Sistem glasovne pošte se lahko nastavi tako, da omogoča klicanje na notranje številke ali na številke PSTN. V tem primeru je zahteva za klic sprejeta iz glasovne pošte.

Tabela 5. Seznam možnih entitet, na katere je moč preusmeriti klic.

Usmerjevalni načrt se vedno gradi od spodaj navzgor po naslednjem vrstnem redu:

1. Dodamo naprave (prehode in povezave za povezovanje omrežij).
2. Zgradimo usmerjevalne skupine iz naprav, ki smo jih dodali.
3. Zgradimo usmerjevalne sezname iz usmerjevalnih skupin.
4. Zgradimo usmerjevalne vzorce, ki kažejo na usmerjevalne sezname.

Na sliki 6 je prikazan primer arhitekture usmerjevalnega načrta, ki prikazuje levo drevo z entitetami in desno drevo z uporabnim primerom. Usmerjevalni vzorec predstavlja cilj, do katerega lahko pridemo z eno izmed skupin poti v usmerjevalnem seznamu. V tem primeru sta izdelani 2 skupini poti do cilja: digitalna in analogna. V vsaki izmed skupin je lahko ena ali več naprav, ki se uporabijo za doseg cilja.



Slika 6. Primer arhitekture usmerjevalnega načrta.

### 3.4.1 Usmerjevalni vzorci

Usmerjevalni vzorci lahko vsebujejo nadomestne znake in na ta način lahko samostojni usmerjevalni vzorec predstavlja več imeniških števil. To prispeva h krajši usmerjevalni tabeli, ki je tudi lažje berljiva.

Tabela 6 prikazuje posamezne nadomestne znake, ki so lahko uporabljeni v usmerjevalnih vzorcih.

Nadomestni znak	Opis
X	Samostojna številka (0-9)
@	Številčni plan Severne Amerike
!	Ena ali več števk (0-9)
[a-b] [cde]	Zapis generičnega razpona (od a do b oziroma c, d in e)
[^a-b]	Zapis izključenega razpona
.	Ločilo, ki loči dostopno kodo strežnika od klicne številke
#	Zaključek številčnega vnašanja
<nadomestni znak>?	Ujemanje z nič ali več ponovitvami katerekoli številke, ki se ujema s predhodnim nadomestnim znakom
<nadomestni znak>+	Ujemanje z eno ali več ponovitvami katerekoli številke, ki se ujema s prehodnim nadomestnim znakom

Tabela 6. Seznam najbolj pogostih nadomestnih znakov v usmerjevalnih vzorcih.

Primer uporabe teh znakov v usmerjevalnih vzorcih je prikazan v tabeli 7.

Vzorec	Rezultat
1234	Se ujema s številko 1234
1*1X	Se ujema s številkami od 1*10 do 1*19
12XX	Se ujema s številkami od 1200 do 1299
13[25-8]6	Se ujema s številkami 1326, 1356, 1366, 1376, 1386
13[^3-9]6	Se ujema s številkami 1306, 1316, 1326
13!#	Se ujema s številkami, ki se pričnejo s 13 in jim sledi ena ali več števk. Končati se mora z znakom #. Primer: 135# in 13579#

Tabela 7. Primeri usmerjevalnih vzorcev.

Pri uporabi usmerjevalnih vzorcev uporablja strežnik Cisco Unified Communications Manager logiko najboljšega ujemanja za izbiro najbolj primerne vzorca.

Ko je prisotnih več vzorcev, ki se ujemajo s številko, se izbere najboljši glede na naslednja pogoja:

- ujemati se mora s klicano številko,
- izmed vseh ujemajočih se vzorcev mora izbran vzorec zajeti najmanj celih števil (primer: uporabnik kliče številko 1200, vzorca, pa sta dva – 1XXX in 12XX; oba sta ustrezna, ampak 1XXX zajema 1000 različnih števil, v tem primeru od 1000 do 1999, 12XX pa zajame samo 100 števil, v tem primeru od 1200 do 1299, torej je kot cilj tega klica izbran vzorec 12XX).

Usmerjevalni vzorci so del klicne usmerjevalne tabele. Če se klic ujema z usmerjevalno logiko, lahko usmerjevalni vzorec klic usmeri direktno na napravo, kot sta prehod ali povezava za povezovanje omrežij, ali pa ga usmeri na seznam poti. Če obstaja več poti do končne številke, ki ustreza usmerjevalnemu vzorcu, nam seznam poti nudi prvi nivo izbire poti. Seznam poti vsebuje prioritetni seznam usmerjevalnih skupin in omogoča manipulacijo števk na posamezni usmerjevalni skupini. Usmerjevalna skupina je drugi nivo izbire poti in kaže na naprave, ki so izbrane glede na distribucijski algoritem (krožni ali od zgoraj navzdol).

Podjetje Cisco Systems priporoča uporabo celotnega konstrukta (usmerjevalni vzorec, seznam poti in usmerjevalna skupina), ker nudi največjo fleksibilnost pri usmerjanju klicev, manipulaciji števk, redundanci poti in rasti klicnega načrta v prihodnosti. Če usmerjevalni vzorci kažejo neposredno na naprave, se bo v primeru dodajanja novih naprav morala spremeniti tudi konfiguracija. Samostojna naprava ne more biti uporabljena v obeh primerih (biti član usmerjevalne skupine ali biti referenca, na katero neposredno kaže usmerjevalni vzorec).

### 3.4.2 Seznam poti

Seznam poti je prioritetni seznam usmerjevalnih skupin. Prva vnesena skupina ima višjo prioriteto, ker je bila prva dodana na prioritetni seznam. Seznam poti je možno začasno onemogočiti, kar pomeni, da bo sicer ostal v podatkovni bazi nastavitve sistema, a ne bo uporabljen.

Seznam poti je povezan z enim ali več usmerjevalnimi vzorci.

### 3.4.3 Usmerjevalna skupina

Usmerjevalna skupina je seznam naprav (prehodov in povezav za povezovanje omrežij), ki si delijo iste zahteve za manipulacijo števk, zato ker se manipulacija števk lahko nastavlja le enkrat na eni sami usmerjevalni skupini.

Usmerjevalna skupina je lahko nastavljena na krožno porazdelitev (*round robin*) ali na porazdelitev od zgoraj navzdol (*top down*), kjer ima prvi vnos največjo prioriteto. Krožna porazdelitev se uporablja za deljenje obremenitve (*load sharing*), porazdelitev od zgoraj navzdol pa se uporablja za implementacijo poti varnostnih kopij, ko predvidene poti ni na voljo.

### 3.5 Klicni privilegiji

Klicni privilegiji so pomemben del klicnega načrta. Uporabljajo se za implementacijo dovoljenj COS (*Class Of Service*), kar pomeni, da je glede na napravo, ki jo kličemo, ali telefonsko linijo določenim ciljem dovoljen dostop do klicne usmerjevalne tabele. Klicni privilegiji torej definirajo vnose v klicni usmerjevalni tabeli, ki so dostopni neki končni točki (telefonu, prehodu, povezavi za povezovanje omrežij), ki izvaja zahtevo po usmeritvi klica.

Klicni privilegiji se uporabljajo za kontrolo klicnih stroškov:

- blokiranje dragih storitvenih števil (primer: 090),
- prepoved mednarodnih klicev,
- varovanje zasebnosti določenih uporabnikov (primer: onemogočanje direktnega klica direktorju razen preko njegove tajnice).

Uporabljajo se tudi za posebne aplikacije:

- usmerjanje klicev z isto številko, ki imajo drugačnega uporabnika,
- drugačno usmerjanje klicev na isto številko glede na čas v dnevu.

Razred klicnih privilegijev	Dovoljeni cilji
Interni klici	Interno
	Klic v sili
Lokalni klici	Interno
	Klic v sili
	Klici v isti omrežni skupini
Medkrajevni klici	Interno
	Klic v sili
	Klici v isti omrežni skupini
	Medkrajevno
Mednarodni klici	Interno
	Klic v sili
	Klici v isti omrežni skupini
	Medkrajevno
	Mednarodno

Tabela 8. Primer zahtev klicnih privilegijev.

Tabela 8 opisuje tipično implementacijo razredov klicnih privilegijev in njihovih ciljev. Ti razredi so lahko dodeljeni končnim napravam ali uporabnikom.

Tabela 9 prikazuje možne elemente klicnih privilegijev, s katerimi lahko prirejamo privilegije, ki jih ima nek telefon, uporabnik ali telefonski klic.

Element	Karakteristike
Razdelki ( <i>Partitions</i> )	Skupina telefonskih števil s podobnimi karakteristikami dosegljivosti.
Prostor klicnega iskanja ( <i>Calling Search Space</i> )	Določa, kateri razdelki so dosegljivi določeni napravi.
Časovni razporedi in časovna obdobja	Dovoljuje dosegljivost določenih razdelkov zgolj v določenem času dneva.
Koda za sledenje števil ( <i>Client Matter Code</i> )	Sledenje števil; uporabnik mora vnesti kodo CMC, da lahko sledi določenim uporabnikom.
Zahtevana koda za preverjanje dovoljenja ( <i>Forced Authorization Codes</i> )	Prepoved izhodnih klicev na določene številke; uporabnik mora vnesti kodo FAC, da lahko pokliče na določeno številko.

Tabela 9. Elementi klicnih privilegijev.

### 3.5.1 Razdelki in prostori klicnega iskanja

Razdelek zajema skupino števil, ki je dosegljiva na enak način. V skupino lahko štejemo vse vzorce, ki so lahko del nekega razdelka. Primeri takih vzorcev so imeniška številka, usmerjevalni vzorec, prevajalni vzorec, številka glasovne pošte, konferenčna številka ipd.

Prostor klicnega iskanja je urejen seznam razdelkov in vključuje vse razdelke, ki so dosegljivi znotraj tega prostora klicnega iskanja. Naprava lahko pokliče le tiste številke, ki se nahajajo v razdelkih, ki so del tega prostora klicnega iskanja. Prostor klicnega iskanja je dodeljen vsaki entiteti, ki lahko ustvari zahtevo po usmeritvi klica (telefoni, telefonske linije, prehodi in aplikacije).

Privzeto so ob ustvarjanju vse entitete dodeljene razdelku <None> in prostoru klicnega iskanja <None>. To pa pomeni, da so vsi člani razdelka <None> ali tako imenovanega »ničelnega razdelka« vedno dosegljivi s strani virov zahtev klicnega usmerjanja ne glede na prostor klicnega iskanja, ki je dodeljen temu viru. Entitete, ki nimajo dodeljenega prostora klicnega iskanja in torej uporabljajo prostor klicnega iskanja <None>, lahko dostopajo le do usmerjevalnih ciljev, ki so v razdelku <None>.

### 3.5.2 Prevajalni vzorci

Prevajalni vzorci (*Translation patterns*) so zelo močno orodje za vse vrste klicev, ki jih strežnik Cisco Unified Communications Manager uporablja za manipuliranje števk pred

usmerjanjem klica ali pa za manipulacijo telefonskih števil, s katerih sprejme klic. V nekaterih primerih vhodna številka ni dejanska številka v sistemu ali pa ni številka, ki bi jo omrežje PSTN poznalo oz. dovolilo. V takih primerih, lahko številko spremenimo s pomočjo prevajalnega vzorca.

Prevajalni vzorci se lahko uporabijo za usmerjanje ali prepoved usmerjanja klicev določenih vzorcev.

Kadar skupek števk ustreza prevajalnemu vzorcu, strežnik Cisco Unified Communications Manager ne usmeri klica na zunanjo entiteto, ampak najprej izvede prevajanje številke in potem preusmeri klic na drug prevajalni vzorec ali zunanjo entiteto.

Manipulacija števk in prevajalni vzorci se pogosto uporabljajo v geografsko ločenih distribuiranih sistemih, kjer telefonske številke niso enake v vseh pisarnah. V tem primeru se pripravi enoten klicni načrt s prevajalnimi vzorci. Tako se pisarne lahko medsebojno kličejo brez dodatnih informacij o medkrajevnem predznaku same številke.

### 3.6 Vmesniki

Cisco Unified Communications Manager ima na voljo več različnih vmesnikov za komuniciranje s samim strežnikom [5]. Vmesniki se delijo po skupinah na:

- vmesnike, povezane z administracijo strežnika (namenjeni so oddaljeni konfiguraciji samega strežnika in primerni za večja okolja, ponudnike storitev ali sisteme z več geografskimi lokacijami),
- vmesnike za nadzor naprav oz. klicev in
- vmesnike za obvestila.

Med vmesnike, ki so povezani z administracijo strežnika, spadata:

- administracijski vmesnik XML (AXL) in
- vmesnik storitev prenosljivih telefonskih priključkov (*Extension Mobility Service API*).

Vmesniki za nadzor naprav oz. klicev so:

- Cisco CTI TAPI & Cisco Wave Driver,
- Cisco CTI JTAPI ter
- Cisco WebDialer.

Vmesnika za obvestila sta:

- Serviceability XML in
- SNMP/MIB.

Za razvijalce so najbolj pomembni vmesniki AXL, CTI-TAPI in CTI-JTAPI, saj nudijo dostop do velikega števila funkcij in dogodkov strežniškega sistema Cisco Unified Communications Manager. Tako ponujajo razvijalcu tudi več nadzora nad samo interakcijo s sistemom. Podjetje Cisco Systems jih tipsko poimenuje okvir interakcije (*Interaction Framework*).

Za razvijalce z nekoliko manj izkušnjami so pri podjetju Cisco Systems pripravili skupino tako imenovanih makro vmesnikov (*Macro API*). Ti nudijo razvijalcu predvsem hiter in enostaven dostop do funkcij strežniškega sistema Cisco Unified Communications Manager, hkrati pa mu zakrivajo granularne podrobnosti, ki zanj niso pomembne. V to skupino vmesnikov spadata tudi vmesnik storitev prenosljivih telefonskih priključkov in WebDialer.

V zadnjo skupino, poimenovano vmesniki uporabnosti (*Serviceability API*), spadajo vmesniki, ki nadzornim aplikacijam nudijo sistemske informacije v realnem času. V to zbirko servisov in orodij spadajo Serviceability XML, Performance Monitoring (*PerfMon*), Real-time Device/CTI feed (*RisPort*), Log Collection, Service Control, Call Detail Record on Demand in SNMP/MIBs.

Vsi zgoraj omenjeni vmesniki temeljijo na standardih (XML, SOAP, MS TAPI 2.1, Sun JTAPI 1.2, SNMP v1 & v2) in so združljivi s prejšnjimi različicami strežniških sistemov Cisco Unified Communications Manager. Na voljo so v okviru skupnosti »Cisco Developer Community« [2].

## 4 AXL

Administrativni vmesnik XML (*AXL*) omogoča aplikacijam, da oddaljeno dostopajo do konfiguracijskih podatkov strežnika Cisco Unified Communications Manager, ki so shranjeni v podatkovni bazi. Vmesniki za administracijo so edini način, preko katerega je podjetje Cisco Systems dovolilo dostop do podatkov strežnika Cisco Unified Communications Manager, saj lahko le na tak način zagotovijo, da so podatki v podatkovni bazi konsistentni.

AXL uporablja datoteke tipa XML in vmesnik SOAP za komunikacijo. Komunikacija z strežnikom poteka na naslednji način:

- uporabnik zapiše zahtevo v datoteko XML in jo preko protokola SOAP pošlje na strežnik,
- strežnik nato s pomočjo protokola SOAP vrne odgovor v obliki XML.

AXL omogoča zunanjim aplikacijam vstavljanje, branje, posodabljanje in brisanje podatkov v podatkovni bazi strežnika Cisco Unified Communications Manager. Skoraj vsak zapis, ki se nahaja v bazi strežnika, je v vmesniku AXL predstavljen kot objekt. Objekti v bazi so tako naprave, uporabniki, telefonske linije, govorni prehodi, povezave za povezovanje omrežij ipd. Vsak objekt ima več atributov, preko katerih lahko o samem objektu izvemo podrobnejše podatke ali pa te podatke nastavljamo.

Za uporabo vmesnika AXL je potrebno na strežniku Cisco Unified Communications Manager omogočiti dve storitvi. Prva storitev »Cisco AXL Web Service« se nahaja v »Communications Manager Serviceability« v razdelku »Database and Admin Services«. Druga storitev se imenuje »Cisco SOAP – CDRonDemand Service« in se nahaja v razdelku »CDR Services«.

Vse zahteve AXL morajo biti avtorizirane z osnovno avtorizacijo HTTPS, ki vsebuje glavo pooblastitve, sestavljeno v obliki <uporabniško ime>:<geslo> in zapisano v zapisu BASE64. Uporaba vmesnika AXL je varna zaradi protokola HTTPS ter uporabe namenskega uporabniškega računa, ki ga je potrebno ustvariti za ta namen. Uporabniškemu računu

dodelimo posebno vlogo v strežniku Cisco Unified Communications Manager, ki se imenuje »AXL API access«. Priporočljivo je, da je vloga tega uporabniškega računa omejena zgolj na dostop do vmesnika AXL.

#### 4.1 Definicija dokumenta AXL

Podjetje Cisco Systems je za potrebe vmesnika AXL razvilo svojo shemo XML, ki je precej velika in zajema vse elemente, ki jih je možno pridobiti s pomočjo vmesnika AXL. Problem te sheme je, da je zelo velika in ni namensko pripravljena za določeno nalogo. Tako bi v primeru izpisa določenih elementov morali datoteko XML programsko pregledati večkrat in ročno povezati nekatere elemente med seboj. Ker je shema namenjena zgolj shranjevanju in nastavljanju vrednosti podatkov, je za specifične naloge potrebna precejšnja programska logika, ki vso to goro podatkov pripravi za nadaljnjo uporabo.

Na spletni strani proizvajalca je možno dobiti celotno shemo za dokumente administrativnega vmesnika XML [1]. Shema je sestavljena iz sorodstvenih elementov, ki določajo razmerja med posameznimi elementi, in lahko vsebuje enostavne ali kompleksne elemente. Opis tipa elementov je prikazan v tabeli 10.






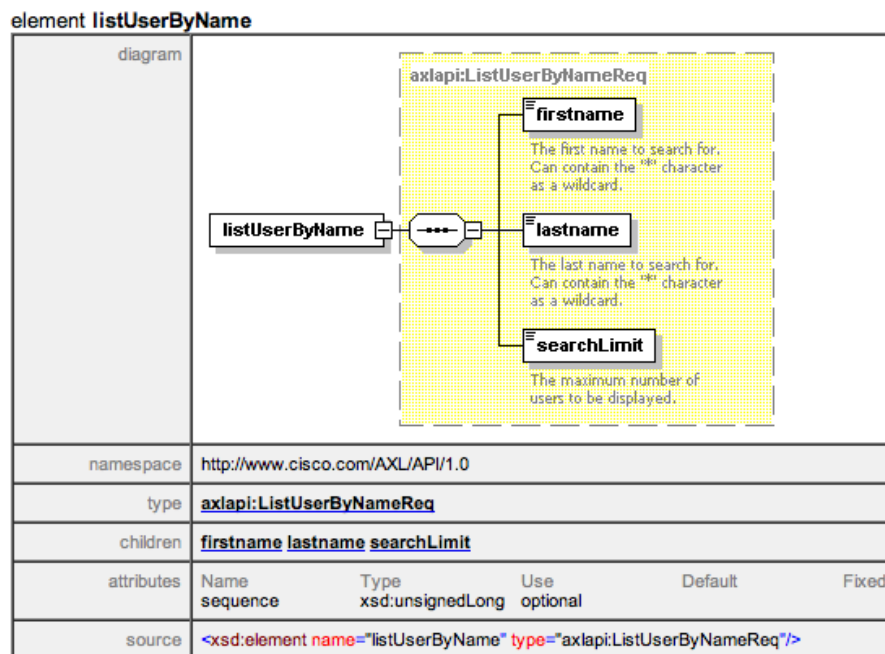
Ime elementa	Opis elementa	Simbol elementa v shemi
Kompleksen element	Vsebuje lahko attribute in druge elemente v razmerju starš – otrok. Element se mora pojaviti v instanci dokumenta XML.	
Enostaven element	Vsebuje lahko attribute, ne more pa vsebovati drugih elementov. Element se mora pojaviti v instanci dokumenta XML.	
Opcijski element	Vsak tip elementa je lahko opsijski, vključno z elementoma za zaporedje elementov in izbiro elementov. Ni potrebno, da se pojavi v instanci dokumenta XML.	
Zaporedje elementov	Element za zaporedje elementov narekuje, da se morajo vsi njegovi otroci pojaviti v dokumentu XML v vrstnem redu, kot so zapisani.	
Izbira elementov	Izbira elementov pomeni, da se lahko največ en izmed otrok tega elementa pojavi v datoteki XML.	

Tabela 10. Pomen posameznih grafičnih simbolov v shemi vmesnika AXL.

Na sliki 7 lahko vidimo primer iz celotne dokumentacije administrativnega XML podjetja Cisco Systems. Korenski element je v tem primeru `listUserByName` in vsebuje naslednje tri elemente v točno zapisanem zaporedju: `firstname`, `lastname` in `searchLimit`.



Slika 7. Primer uporabe sheme Cisco AXL.

## 4.2 Operacije vmesnika AXL

Operacije vmesnika AXL se delijo na več skupin glede na njihovo vlogo. Tako imamo operacije za:

- dodajanje (funkcije imajo predpono `add`),
- pridobivanje (funkcije imajo predpono `get`),
- izpisovanje (funkcije imajo predpono `list`),
- odstranjevanje (funkcije imajo predpono `remove`),
- posodabljanje (funkcije imajo predpono `update`).

Zgornje skupine operacij se nanašajo na povsem enake entitete, od katerih so najbolj pogosti uporabniki, naprave, strežniki v gručah Cisco Unified Communications Manager, linije za telefonske naprave, konferenčne številke, usmerjevalne skupine, usmerjevalne sezname, predloge za tipke na telefonskih napravah in prevajalni vzorci.

Poleg zgoraj omenjenih večjih skupin obstajata v vmesniku AXL še dve manjši skupini, ki vsebujeta vsaka le po nekaj operacij. To sta skupini za izvedbo in izvršitev posebnih operacij na strežniku Cisco Unified Communications Manager. V teh dveh skupinah sta najbolj uporabljeni operaciji za izvršitev poizvedbe SQL (`executeSQLQuery`) in za ponastavitev naprave (`doDeviceReset`).

## 5 Samodejna izdelava dokumentacije

Za potrebe diplomske naloge sem izdelal ukazno datoteko v programskem jeziku Perl, ki s pomočjo vmesnika AXL in uporabe protokola SOAP pridobi potrebne podatke od strežnika Cisco Unified Communications Manager. Ti podatki se zapišejo v datoteko XML in so tako shranjeni za kasnejšo uporabo. S pomočjo pretvornika datotek XSLT, to je transformacijske sheme za pretvorbo XML datotek v drug format, pa podatke iz dokumenta XML pretvorimo v zapis HTML in jih prikažemo v spletnem brskalniku.

Celotna rešitev predstavlja kar nekaj posameznih korakov, potrebnih za prikaz oz. izdelavo dokumentacije. Končni rezultat je torej pripravljena dokumentacija, ki jo je lažje in predvsem brez napak prenesti v dokumente dokumentacije sistema, namenjene tretjim osebam (največkrat sami stranki).

Koraki samodejne izdelave dokumentacije si sledijo v naslednjem zaporedju:

1. Uporabnik ali nek zunanji proces poskrbi za zagon ukazne datoteke Perl s pomočjo tolmača tega programskega jezika.
2. Ukazna datoteka pripravi zahteve SOAP in jih pošlje strežniku Cisco Unified Communications Manager, ki hrani podatke.
3. Strežnik odgovori na zahteve SOAP z odgovori SOAP in jih pošlje nazaj na računalnik, kjer se izvaja ukazna datoteka Perl.
4. Tolmač Perl po navodilih ukazne datoteke poskrbi za zapis prejetih podatkov v dokument XML.
5. Ustvarjen dokument XML mora biti na voljo preko spletnega strežnika oz. dostopen preko operacijskega sistema, v katerem zaženemo brskalnik.
6. Uporabnik usmeri svoj brskalnik na naslov ustvarjenega dokumenta XML in s tem ustvari zahtevo za prikaz podatkov.
7. Uporabniku se s pomočjo transformacijske sheme v spletnem brskalniku prikaže celotna zgrajena dokumentacija.

V prvem koraku se zagon ukazne datoteke lahko izvede ročno ali s pomočjo kakega zunanjega procesa. Kot primer zunanjega procesa lahko navedem proces cron [3], ki je nekakšen časovni razporejevalnik opravljen v operacijskem sistemu Unix in podobnih sistemih. Proces cron omogoča razporeditev in zagon računalniških opravil (posameznih ukazov ali ukaznih datotek lupine) ob točno določenem času.

V drugem koraku tolmač Perl prebere ukazno datoteko in sestavi več poizvedb, ki jih ovije v sporočilo SOAP in jih pošlje strežniku Cisco Unified Communications Manager. Strežnik sporočilo SOAP sprejme in iz njega izlušči poizvedbe.

Strežnik v tretjem koraku odgovori na poizvedbe. Odgovore prav tako ovije v sporočilo SOAP, katerega pošlje nazaj na računalnik, od koder je bila poizvedba narejena.

Odgovori na poizvedbe se v četrtem koraku s pomočjo tolmača Perl izluščijo iz sporočila SOAP in zapišejo v datoteko XML.

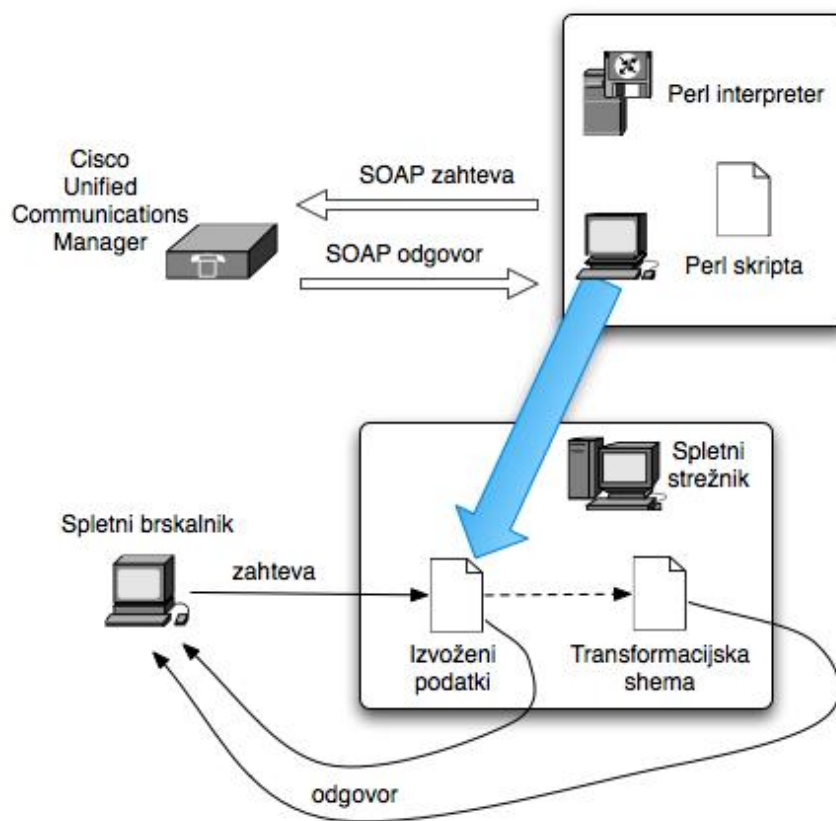
Ustvarjena datoteka XML se mora nahajati na mestu, katerega spletni strežnik servira odjemalcem. Datoteko lahko na ustrezno mesto prekopiramo ročno ali s pomočjo ukazne datoteke lupine, lahko pa v nastavitvah spletnega strežnika nastavimo, da je datoteka v datotečnem sistemu na voljo kar preko mape, v kateri se nahaja. Prav tako pa se mora na

dosegljivem mestu nahajati tudi transformacijska datoteka, ki narekuje prikaz pridobljenih podatkov v spletnem brskalniku.

Uporabnik v šestem koraku usmeri svoj spletni brskalnik na mesto datoteke XML s podatki (ta se lahko nahaja lokalno na računalniku ali na nekem spletnem strežniku). Tako ustvari zahtevo za prevod teh podatkov iz oblike XML v uporabniku prijaznejšo in lažje berljivo obliko.

V zadnjem koraku spletni brskalnik s pomočjo transformacijske datoteke XSLT pretvori datoteko XML s podatki v zapis HTML, ki omogoča uporabniku prijazen prikaz vseh podatkov v spletnem brskalniku.

Slika 8 prikazuje celoten krog dogodkov z vsemi koraki, ki so potrebni za izdelavo dokumentacije in njen prikaz uporabniku.

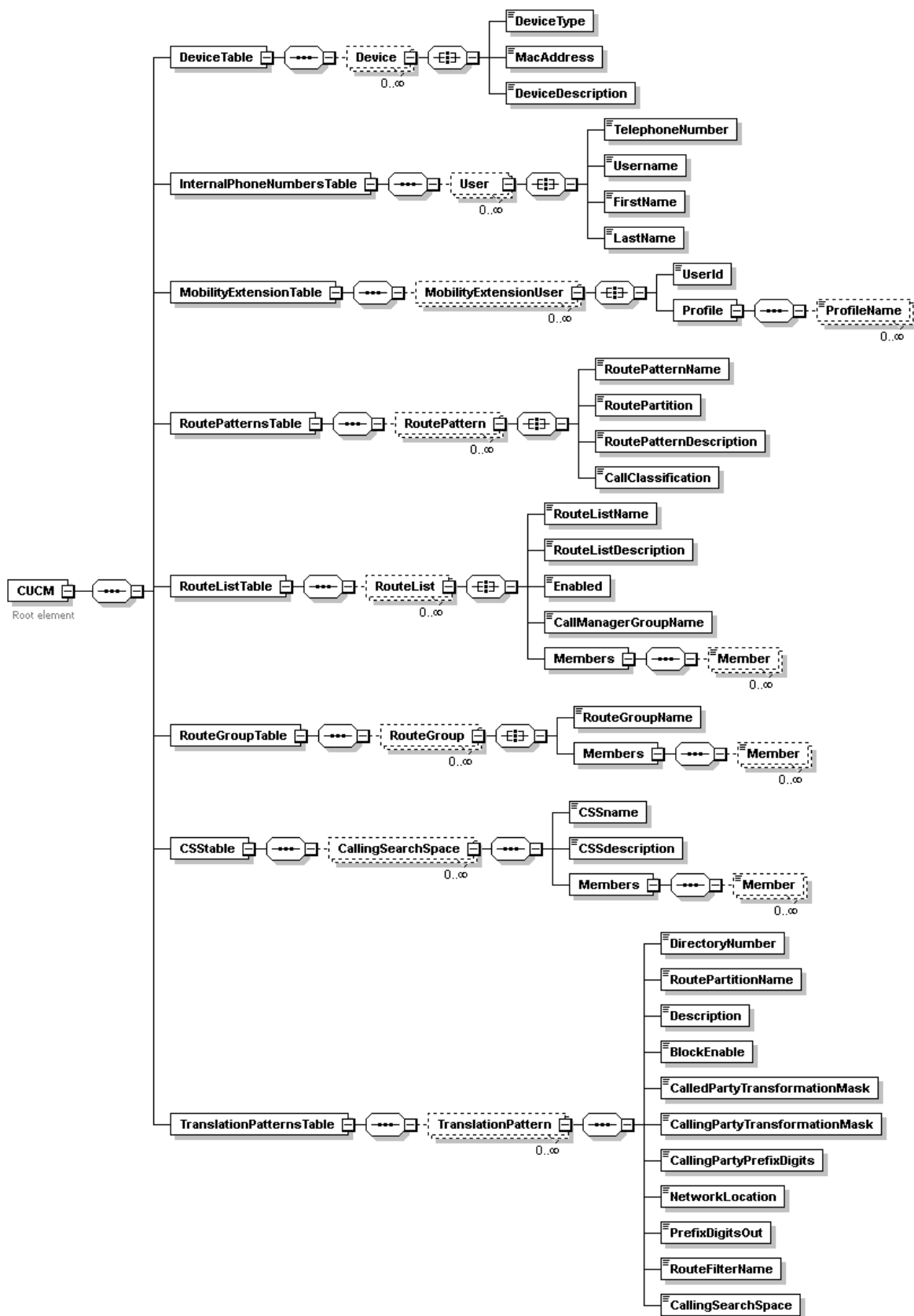


Slika 8. Celoten krog dogodkov pri samodejni izdelavi dokumentacije, od zahteve za pridobitev podatkov do prikaza v spletnem brskalniku uporabnika.

### 5.1 Shema za dokumentacijo

Za samodejno izdelavo dokumentacije je bila pripravljena posebna shema XML, ki je predstavljena na sliki 9.

Razlogi za izdelavo nove sheme XML, s pomočjo katere lahko shranimo dobljene podatke, so predvsem drevesna struktura shranjevanja podatkov, zmanjšanje nepotrebnih informacij in optimizacija dostopa do informacij za hitro iskanje posameznih elementov.



Slika 9. Shema XML, izdelana za diplomsko nalogo, ki služi za izvoz podatkov iz strežnika Cisco Unified Communications Manager.

Shemo sestavlja korenski element, ki se imenuje CUCM (kratica za Cisco Unified Communications Manager). Njegovi podelementi predstavljajo glavne lastnosti, ki jih želimo pridobiti preko vmesnika AXL: DeviceTable, InternalPhoneNumbersTable, MobilityExtensionTable, RouterPatternTable, RouteListTable, RouteGroupTable, CSSTable in TranslationPatternsTable.

DeviceTable predstavlja skupek telefonskih naprav, ki lahko vsebuje nič ali več telefonskih naprav. O vsaki izmed teh naprav pa beležimo tip naprave, njen naslov MAC in opis naprave (DeviceType, MacAddress, DeviceDescription). Pripadajoča koda sheme XML je prikazana na sliki 2.

InternalPhoneNumbersTable predstavlja skupino uporabnikov v podjetju, ki lahko vsebuje nič ali več uporabnikov. O vsakem uporabniku hranimo telefonsko številko, uporabniško ime, ime in priimek (TelephoneNumber, Username, Firstname, Lastname).

S poljem MobilityExtensionTable predstavljamo uporabnike storitve mobilnih telefonskih priključkov. V tabeli je lahko zapisanih nič ali več uporabnikov te storitve, o vsakem izmed njih pa hranimo njegovo identifikacijsko oznako in seznam profilov (UserId, Profile). Slednji lahko vsebuje nič ali več imen profila uporabnika.

RoutePatternsTable predstavlja skupek usmerjevalnih vzorcev. Vsebuje lahko nič ali več vzorcev. O vsakem izmed vzorcev hranimo naziv usmerjevalnega vzorca, usmerjevalni razdelek, opis usmerjevalnega vzorca in klasifikacijo klica (RoutePatternName, RoutePartition, RoutePatternDescription, CallClassification).

RouteListTable predstavlja skupek usmerjevalnih seznamov. Vsebuje lahko nič ali več usmerjevalnih seznamov. O vsakem usmerjevalnem seznamu hranimo njegov naziv, opis, informacijo o tem, ali je usmerjevalni seznam omogočen, skupino Call Manager, v kateri se seznam nahaja, in seznam članov tega usmerjevalnega seznama (RouteListName, RouteListDescription, Enabled, CallManagerGroupName, Members). Seznam članov lahko vsebuje nič ali več članov.

RouteGroupTable predstavlja skupek usmerjevalnih skupin, ki lahko vsebuje nič ali več usmerjevalnih skupin. Vsaka izmed usmerjevalnih skupin vsebuje naziv usmerjevalne skupine in skupino članov usmerjevalne skupine (RouteGroupName, Members). Skupina članov usmerjevalne skupine pa lahko vsebuje nič ali več članov.

CSSTable predstavlja skupek prostorov klicnega iskanja, ki lahko vsebuje nič ali več prostorov klicnega iskanja. Vsak prostor klicnega iskanja vsebuje naziv, opis in seznam razdelkov, ki so dodeljeni temu prostoru klicnega iskanja (CSSname, CSSdescription, Members). Seznam razdelkov lahko vsebuje nič ali več razdelkov.

TranslationPatternsTable predstavlja skupek prevajalnih vzorcev, ki lahko vsebuje nič ali več prevajalnih vzorcev. O vsakem vzorcu hranimo imeniško številko, naziv usmerjevalnega razdelka, opis, informacijo o tem, ali je vzorec omogočen, pretvorne maske klicnega, pretvorne maske kličočega, predpono številke kličočega, lokacijo omrežja, predpono številke izhodnega klica, naziv usmerjevalnega filtra in prostor klicnega iskanja (DirectoryNumber, RoutePartitionName, Description, BlockEnable, CalledPartyTransformationMask, CallingPartyTransformationMask,

CallingPartyPrefixDigits, NetworkLocation, PrefixDigitsOut, RouteFilterName, CallingSearchSpace).

## 5.2 Opis ukazne datoteke

Ukazna datoteka, ki poskrbi za pridobitev podatkov za izdelavo dokumentacije, je napisana v jeziku Perl. V njej so na začetku navedeni moduli, ki jih ukazna datoteka uporablja. Najpomembnejša dva sta »SOAP::Lite«, ki služi izdelavi sporočil SOAP, ter »XML::Writer«, ki služi za pisanje značk jezika XML. Nato sledi deklaracija spremenljivk oz. skalarjev, v katerih so shranjene vrednosti o lokaciji strežnika, poverilnicah in seznamih, ki se bodo tekom izvajanja ukazne datoteke napolnili s podatki, ki jih ukazna datoteka pridobi od strežnika. Za deklaracijo spremenljivk sledi glavni del ukazne datoteke, kjer se najprej izvede povezava na strežnik z namenom pridobitve njegove različice. Vmesnik Cisco AXL se med različicami strežnikov razlikuje, zato je pridobitev različice uporabljenega strežnika pomembna informacija, ki jo potrebujemo za pravilno izdelavo zahtev za pridobitev podatkov. Tu nastavimo tudi podatke o izhodni datoteki XML, med drugim so to ime datoteke, imenski prostor v shemi XML in ime korenskega elementa.

Glavne naloge ukazne datoteke so razbite na več podprogramov, ki se kličejo znotraj ukazne datoteke. Za vsako posamezno nalogo, to je za vsakega od osmih podelementov korenskega elementa CUCM v shemi XML (glej sliko 9), smo napisali dva podprograma – prvi je namenjen pridobivanju ustreznih podatkov o tem elementu od strežnika, drugi pa zapisu pridobljenih podatkov v datoteko XML. Podprograme potem kliče po vrsti, kot imamo zapisane podatke o elementih v shemi XML. V nadaljevanju so na kratko opisani posamezni pari podprogramov.

Podprogram `getDeviceTable` je namenjen pridobivanju podatkov o napravah, ki jih strežnik pozna. Podatke pridobimo s pomočjo poizvedbe SQL in jih zapišemo v skupek seznamov, ki se v drugem podprogramu, imenovanem `toXmlDeviceTable`, uporabijo za zapis pridobljenih podatkov v datoteko XML.

Podprogram `getEndUsers` je namenjen pridobivanju podatkov o končnih uporabnikih sistema. Tudi te podatke pridobimo s pomočjo poizvedbe SQL in jih zapišemo v skupek seznamov. Podprogram `toXmlEndUsers` uporabi te sezname za zapis pridobljenih podatkov v datoteko XML.

Podprogram `getMobilityExtension` je namenjen pridobivanju podatkov o prenosljivih telefonskih priključkih. Predpogoj za izvajanje tega podprograma je podprogram `getEndUsers`, v sklopu katerega pridobimo unikatno identifikacijsko številko vsakega uporabnika. Preko te številke in s pomočjo vgrajenega klica `getUser`, ki se nahaja v vmesniku Cisco AXL, pridobimo tudi podatke o posameznem uporabniku. Pri vsakem uporabniku se preveri, ali ima nastavljen profil za prenosljive telefonske priključke, in se nato shrani v seznam. Uporabnik ima lahko nastavljenih več profilov za prenosljive telefonske priključke. Podatki o profilih se nato v podprogramu `toXmlMobilityExtension` shranijo v datoteko XML.

Podprogram `getRoutePatterns` pridobi podatke o usmerjevalnih vzorcih. V ta namen podprogram najprej s samostojno poizvedbo SOAP pridobi podatke o imeniških številkah in razdelkih, v katerih se telefonski priključek nahaja. Identifikacijska številka posameznega

usmerjevalnega vzorca, ki je pridobljena s pomočjo klica vmesnika Cisco AXL, pa se nato uporabi za pridobitev podrobnejših podatkov o usmerjevalnih vzorcih. Ti podatki so shranjeni v sezname in se v podprogramu `toXmlRoutePatterns` zapišejo tudi v datoteko XML.

Podprogram `getRouteLists` pridobiva podatke o usmerjevalnih seznamih. S pomočjo dveh poizvedb SQL, ki sta oviti v sporočilo SOAP, pridobimo podatke o usmerjevalnih seznamih. Nato sestavimo še tretjo poizvedbo SQL, ki nam vrne identifikacijske podatke o posameznih usmerjevalnih seznamih. Sledi klic vmesnika Cisco AXL, s katerim pridobimo podrobnejše informacije o vsakem usmerjevalnem seznamu. Usmerjevalni seznam je kompleksnejša struktura in je zato shranjena v večdimenzionalnem seznamu. Ta seznam se nato v podprogramu `toXmlRouteLists` zapiše v datoteko XML.

Podprogram `getRouteGroups` skrbi za pridobitev podatkov o usmerjevalnih skupinah. Najprej s poizvedbo SQL pridobimo identifikacijske podatke o usmerjevalnih skupinah, nato pa za vsako skupino s pomočjo klica vmesnika Cisco AXL pridobimo še podrobne informacije o skupini. Sledi še poizvedba SQL, s katero pridobimo podatke o napravah, povezanih s to skupino. Skupine so zapisane v seznam, ki se v podprogramu `toXmlRouteGroups` zapiše v datoteko XML.

Podprogram `getCSS` pridobiva podatke o seznamih prostorov klicnega iskanja. Za vsakega izmed prostorov najprej pridobimo identifikacijske podatke s pomočjo poizvedbe SQL. Nato pa s pomočjo klica vmesnika Cisco AXL pridobimo podrobnejše informacije o posameznih prostorih klicnega iskanja ter pripadajočih razdelkih. Pridobljeni podatki so najprej zapisani v seznam in v podprogramu `toXmlCSS` še v datoteko XML.

Podprogram `getTranslationPatterns` skrbi za pridobivanje podatkov o prevajalnih vzorcih. Z dvema klicema vmesnika Cisco AXL pridobimo podatke o prevajalnih vzorcih, ki jih nato shranimo v skupek seznamov. Ti so nato v podprogramu `toXmlTranslationPatterns` zapisani tudi v datoteko XML.

### 5.2.1 Primer pridobivanja podatkov o napravah

Za primer si pogledajmo, kako pridobimo podatke o vseh napravah sistema. Kot smo že omenili, za to uporabimo podprograma `getDeviceTable` in `toXmlDeviceTable`, ki sta prikazana na sliki 10 in sliki 11, po vrsti.

Najprej s pomočjo podprograma `getDeviceTable` (slika 10) sestavimo poizvedbo SQL in jo ovijemo v sporočilo SOAP.

Primer tako sestavljenega sporočila SOAP prikazuje slika 12. Na začetku sporočila je vidna ovojnica sporočila SOAP, v telesu sporočila pa poizvedba SQL.

Sporočilo SOAP nato pošljemo strežniku Cisco Unified Communications Manager, ki na našo zahtevo odgovori z odgovorom SOAP, ki zajema vse zahtevane podatke.

Slika 13 prikazuje del sporočila SOAP, ki vsebuje odgovor strežnika. Tudi odgovor strežnika na začetku sporočila vsebuje ovojnico. V telesu sporočila pa so nanizane naprave in podrobni podatki o njih. Strežnik pošlje le tiste podatke, ki smo jih zahtevali. Tako so na sliki prikazani le tisti podatki, ki so potrebni za našo nalogo. Strežnik sicer hrani precej več informacij o posamezni napravi, kar je razvidno tudi iz dokumentacije API.

```
#####
# Subroutine for getting device table
# mac address, device type, description (sorted by device type)
#####
sub getDeviceTable{

    print "Getting device table from CUCM...";

    # Using SQL query as there is no other way to get list of
    # all devices
    my $res = $soap->executeSQLQuery(
        SOAP::Data->name(sql => 'select d.name as devicename,tm.name as
        devicemodel,d.description as description from device as d, typemodel as tm where
        (d.name like \'SEP%\'' or d.name like \'ATA%\') and d.tkmodel =tm.enum order by
        devicemodel desc')
    );

    # Filling appropriate arrays with values
    @DeviceTable_devicename = $res->valueof('//devicename');
    @DeviceTable_devicemodel = $res->valueof('//devicemodel');
    @DeviceTable_description = $res->valueof('//description');
    print "Done\n";

}

```

Slika 10. Izvorna koda za pridobitev informacij o vseh napravah.

```
#####
# Subroutine for writing device table to XML
#####
sub toXmlDeviceTable{

    $writer->startTag([$xmlNS,"DeviceTable"]);

    # Counter used for iterating through arrays
    my $counter = @DeviceTable_devicename;

    for(my $i=0; $i < $counter; $i++){
        $writer->startTag([$xmlNS,"Device"]);
        $writer->startTag([$xmlNS,"DeviceType"]);
        $writer->characters($DeviceTable_devicemodel[$i]);
        $writer->endTag([$xmlNS,"DeviceType"]);
        $writer->startTag([$xmlNS,"MacAddress"]);
        $writer->characters($DeviceTable_devicename[$i]);
        $writer->endTag([$xmlNS,"MacAddress"]);
        $writer->startTag([$xmlNS,"DeviceDescription"]);
        $writer->characters($DeviceTable_description[$i]);
        $writer->endTag([$xmlNS,"DeviceDescription"]);
        $writer->endTag([$xmlNS,"Device"]);
    }
    $writer->endTag([$xmlNS,"DeviceTable"]);
}

```

Slika 11. Izvorna koda za zapisovanje podatkov v datoteko XML.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://www.ci
  <soapenv:Header />
  <soapenv:Body>
    <ns:executeSQLQuery>
      <sql>select d.name as devicename,tm.name as devicemodel,d.description as description from
    </ns:executeSQLQuery>
  </soapenv:Body>
</soapenv:Envelope>
```

Slika 12. Sporočilo SOAP s poizvedbo SQL.

```
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
  <SOAP-ENV:Header />
  <SOAP-ENV:Body>
    <axl:executeSQLQueryResponse xmlns:axl="http://www.cisco.com/AXL/API/1
      <return>
        <row>
          <devicename>SEP000B821B1C69</devicename>
          <devicemodel>Third-party SIP Device (Basic)</devicemodel>
          <description>SIP client - Mac</description>
        </row>
        <row>
          <devicename>SEP001641E5E1F9</devicename>
          <devicemodel>Third-party SIP Device (Advanced)</devicemodel>
          <description>SIP client ADV1</description>
        </row>
        <row>
          <devicename>SEP00D050574959</devicename>
          <devicemodel>Third-party SIP Device (Advanced)</devicemodel>
          <description>SIP client ADV2</description>
        </row>
        <row>
          <devicename>SEP001C105C4D49</devicename>
          <devicemodel>Third-party SIP Device (Advanced)</devicemodel>
          <description>Linksys naprava</description>
        </row>
        <row>
          <devicename>SEP001DA238AA61</devicename>
          <devicemodel>Cisco TelePresence 500</devicemodel>
          <description>TelePresence 500 simulacija</description>
        </row>
      </return>
    </axl:executeSQLQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Slika 13. Del sporočila SOAP z odgovorom strežnika.

Podatke o napravah, ki smo jih prejeli v odgovoru strežnika (slika 13), nato s pomočjo podprograma `toXmlDeviceTable` (slika 11) zapišemo v datoteko XML.

Del vsebine tako pripravljene datoteke XML prikazuje slika 14. Na začetku datoteke XML je vidna deklaracija dokumenta. Sledi korenski element `CUCM`, ki vsebuje element `DeviceTable`. V tem elementu pa so drug za drugim nanizani elementi `Device`, ki ustrezajo posameznim napravam v sistemu. Vsak od teh elementov beleži podrobne informacije o napravi, ki v našem primeru zajemajo tip naprave (`DeviceType`), njen naslov MAC (`MacAddress`) in opis te naprave (`DeviceDescription`).

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="izvoz-CUCM.xslt"?>
<CUCM>
  <DeviceTable>
    <Device>
      <DeviceType>Third-party SIP Device (Basic)</DeviceType>
      <MacAddress>SEP000B821B1C69</MacAddress>
      <DeviceDescription>SIP client - Mac</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Third-party SIP Device (Advanced)</DeviceType>
      <MacAddress>SEP001641E5E1F9</MacAddress>
      <DeviceDescription>SIP client ADV1</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Third-party SIP Device (Advanced)</DeviceType>
      <MacAddress>SEP00D050574959</MacAddress>
      <DeviceDescription>SIP client ADV2</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Third-party SIP Device (Advanced)</DeviceType>
      <MacAddress>SEP001C105C4D49</MacAddress>
      <DeviceDescription>Linksys naprava</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Cisco TelePresence 500</DeviceType>
      <MacAddress>SEP001DA238AA61</MacAddress>
      <DeviceDescription>TelePresence 500 simulacija</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Cisco TelePresence</DeviceType>
      <MacAddress>SEP001D4526E1ED</MacAddress>
      <DeviceDescription>Telepresence 1000 simulacija</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Cisco TelePresence</DeviceType>
      <MacAddress>SEP001D4526E0B2</MacAddress>
      <DeviceDescription>Telepresence 3000 simulacija</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Cisco ATA 186</DeviceType>
      <MacAddress>ATA000BFD284713</MacAddress>
      <DeviceDescription>Ciscov faks vmesnik</DeviceDescription>
    </Device>
    <Device>
      <DeviceType>Cisco 7940</DeviceType>
      <MacAddress>SEP0006539E2EF1</MacAddress>
      <DeviceDescription>Cisco telefon 1</DeviceDescription>
    </Device>
  </DeviceTable>
</CUCM>

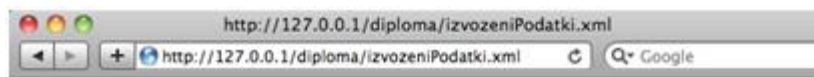
```

Slika 14. Prikaz dela podatkov, shranjenih v datoteki XML.

### 5.3 Prikaz izdelane dokumentacije

Primer izdelane dokumentacije, ki bi jo prikazal spletni brskalnik na uporabnikovem računalniku, lahko vidimo na sliki 15.

Gre za povsem osnoven prikaz pridobljenih podatkov, saj smo vsak element prikazali v enostavni tabeli, katere naslov ustreza nazivu elementa. Za ta namen smo ustvarili tri uporabnike. Vsakemu je bil dodeljen vsaj en profil za prenosljive telefonske številke. Tabela usmerjevalnih vzorcev prikazuje dva usmerjevalna vzorca, namenjena mobilnim in stacionarnim klicem. V zadnji tabeli pa sta prikazana še dva usmerjevalna seznama za dva različna ponudnika storitev, od katerih vsak vsebuje eno usmerjevalno skupino.



The screenshot shows a web browser window with the address bar containing 'http://127.0.0.1/diploma/izvozeniPodatki.xml'. The page content displays several tables of configuration data.

### Tabela naprav

Mac Naslov	Tip Naprave	Opis Naprave
SEP000B821B1C69	Third-party SIP Device (Basic)	SIP client - Mac
SEP001641E5E1F9	Third-party SIP Device (Advanced)	SIP client ADV1
SEP00D050574959	Third-party SIP Device (Advanced)	SIP client ADV2
SEP001C105C4D49	Third-party SIP Device (Advanced)	Linksys naprava
SEP001DA238AA61	Cisco TelePresence 500	TelePresence 500 simulacija
SEP001D4526E1ED	Cisco TelePresence	Telepresence 1000 simulacija
SEP001D4526E0B2	Cisco TelePresence	Telepresence 3000 simulacija
ATA000BFD284713	Cisco ATA 186	Ciscov faks vmesnik
SEP0006539E2EF1	Cisco 7940	Cisco telefon 1
SEP0006539E2E23	Cisco 7940	Cisco telefon 2

### Tabela uporabnikov

Uporabniško ime	Ime	Priimek	Telefonska številka
alest	Ales	Tepina	555
uporabnik1	Janez	Novak	556
uporabnik2	Franc	Novak	557

### Tabela MobilityExtension uporabnikov

Uporabniško ime	Profil
alest	AlesT_EM_Profile
uporabnik1	uporabnik1_EM_Profile
uporabnik2	Dialer Dialer2

### Tabela usmerjevalnih vzorcev

Vzorec	Usmerjevalna particija	Opis vzorca	Klasifikacija klica
0[3457][01]XXXXXX	Mobilna	Mobilni klici	OnNet
0[3457][^01]XXXXXX	Stacionarna	Stacionarni klici	OffNet

### Tabela usmerjevalnih seznamov

Seznam	Opis seznama	Omogocen	Ime skupine v CUCM	Člani
Tus_RouteList	TUS SIP Trunk	true	Default	Tus_RouteGroup
Telekom_RouteList	Telekom SIP Trunk	false	Default	Telekom_RouteGroup

Slika 15. Enostaven primer prikaza izdelane dokumentacije.

Obliko prikaza dokumentacije določa transformacijska shema, ki je zapisana v dokumentu XSLT. Za prikazan primer s slike 15 smo uporabili shemo, katere del je viden na sliki 16.

```

<?xml version="1.0" encoding="ISO-8859-2"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
  <h2>Tabela naprav</h2>
  <table border="1">
    <tr bgcolor="#808080">
      <th>Mac Naslov</th>
      <th>Tip Naprave</th>
      <th>Opis Naprave</th>
    </tr>
    <xsl:for-each select="CUCM/DeviceTable/Device">
      <tr>
        <td><xsl:value-of select="MacAddress"/></td>
        <td><xsl:value-of select="DeviceType"/></td>
        <td><xsl:value-of select="DeviceDescription"/></td>
      </tr>
    </xsl:for-each>
  </table>
  <br />

  <h2>Tabela uporabnikov</h2>
  <table border="1">
    <tr bgcolor="#808080">
      <th>Uporabniško ime</th>
      <th>Ime</th>
      <th>Priimek</th>
      <th>Telefonska številka</th>
    </tr>
    <xsl:for-each select="CUCM/InternalPhoneNumbersTable/User">
      <tr>
        <td><xsl:value-of select="Username"/></td>
        <td><xsl:value-of select="FirstName"/></td>
        <td><xsl:value-of select="LastName"/></td>
        <td><xsl:value-of select="TelephoneNumber"/></td>
      </tr>
    </xsl:for-each>
  </table>
  <br />

  <h2>Tabela MobilityExtension uporabnikov</h2>
  <table border="1">
    <tr bgcolor="#808080">
      <th>Uporabniško ime</th>
      <th>Profili</th>
    </tr>
  </table>

```

Slika 16. Izsek sheme, kjer je določen prikaz uporabnikov in njihovih mobilnih profilov.

Velika prednost uporabe datoteke XSLT je, da lahko sami določimo, kakšen bo prikaz pridobljenih podatkov v spletnem brskalniku ali kakšnem drugem programu. Obliko izdelane dokumentacije lahko tako popolnoma prilagodimo slogu podjetja in vanj vključimo zunanje elemente, kot je na primer logotip podjetja.

#### 5.4 Testiranje sistema za izdelavo dokumentacije

Za testiranje sistema za samodejno izdelavo dokumentacije in predvsem izdelane ukazne datoteke sem s pomočjo programskega paketa VirtualBox pripravil tri virtualne strežnike, na katerih so bili nameščeni strežniki Cisco Unified Communications Manager različice 6.0,

6.1.3 ter 7.0. Tri različice strežnika sem postavil zaradi preverjanja (in zagotavljanja) delovanja ukazne datoteke tudi na starejših različicah strežnika.

S pomočjo spletnega vmesnika in posebej pripravljene ukazne datoteke za vnos podatkov sem v strežnik vnesel podatke, na katerih sem kasneje preverjal delovanje sistema za izdelavo dokumentacije. Vneseni podatki so realno prikazovali nastavitve srednje velikega podjetja v Sloveniji. Imena uporabnikov in telefonske številke so bile izmišljene, vsi ostali podatki (usmerjevalni vzorci, usmerjevalne skupine ipd.) pa so ponazarjali scenarije v realnem svetu. Tako sem s pomočjo ukazne datoteke za vnos podatkov vpisal 1000 končnih uporabnikov.

Izdelana ukazna datoteka je delovala brezhibno na vseh treh testnih sistemih. Testiranje sistema za samodejno izdelavo dokumentacije je bilo uspešno, saj smo brez težav pripravili potrebno dokumentacijo v zelo kratkem času (za prikaz podatkov o vseh uporabnikih sistema potrebujemo le nekaj manj kot 17 sekund). Tako se čas priprave dokumentacije s pomočjo ukazne datoteke občutno skrajša, poleg tega pa se z avtomatizacijo procesa izognemo tudi možnostim napak.

## 6 Sklepne ugotovitve

Obstajajo podjetja z zelo strogo varnostno politiko, ki jim narekuje spremljanje sistema telefonije do te mere, da lahko dnevno preverjajo, ali so bile na strežniku narejene kakšne spremembe. Brez nekega avtomatiziranega sistema je to potrebno početi ročno. V tem primeru je vključen tudi človeški faktor, kar pomeni, da je možnost napake pri veliki količini podatkov precej velika. Ta proces lahko avtomatiziramo na tak način, da s pomočjo procesa cron v določenem časovnem obdobju zaženemo pripravljeno ukazno datoteko, ki preverja, ali je bila na strežniku narejena kakšna sprememba. V primeru spremembe nas ukazna datoteka o tem lahko obvesti. Tudi v tem primeru se z avtomatizacijo procesa izognemo človeškemu faktorju in povečani možnosti za napake.

V času pisanja diplomske naloge je na internetu možno dobiti kar nekaj primerov uporabe vmesnika AXL, vendar je velika večina primerov napisana za višjenivojske programske jezike, ki ne uporabljajo ukaznih datotek. Tudi v dokumentaciji vmesnika AXL so primeri napisani za programski jezik Java [7], kar je precej povečalo začetni čas raziskav uporabe vmesnika AXL, sporočil SOAP in programskega jezika Perl.

Razlog, da sem za potrebe izdelave diplomske naloge izbral programski jezik Perl, je predvsem v hitrosti in fleksibilnosti, ki jo ta programski jezik nudi. Pri svojem delovanju ne potrebuje prevajalnika in je s tega stališča bolj prijazen do sistemskih virov. Prav tako je integracija ukazne datoteke z operacijskim sistemom Linux enostavnejša, tolmač za programski jezik pa se v večini primerov kar privzeto namesti ob namestitvi operacijskega sistema Linux. Dodaten faktor pri izbiri programskega jezika je predstavljalo že obstoječe strežniško okolje, v katerem se za upravljanje v večini uporabljajo ukazne datoteke.

### 6.1 Problemi z vmesnikom AXL

Vmesnik AXL je precej dobro razvit produkt, vendar ima svoje pomanjkljivosti. Ena izmed teh, na katero sem naletel med izdelavo diplomske naloge, je brezdvomno primer spreminjanja podatkov na telefonskem aparatu oz. spreminjanja podatkov uporabnika

telefona. Za spreminjanje podatkov telefonskega aparata je potrebno vedeti njegov naslov MAC, ki služi kot enolični identifikator telefonskega aparata. Če ima administrator telefonski aparat pred sabo, to ne predstavlja nikakršnega problema, saj lahko naslov MAC prebere na zadnji strani telefonskega aparata. Problem pa nastane, ko hočemo telefonu, ki se fizično nahaja v drugem mestu ali drugi državi, spremeniti opis naprave in hkrati ne posedujemo celotnega seznama telefonskih aparatov v sistemu. V tem primeru zadane naloge ni možno rešiti na eleganten način. V vmesniku AXL namreč ne obstaja nikakršna poizvedba za pridobitev seznama vseh telefonskih naprav v sistemu. Obstaja pa poizvedba, ki nam omogoča neposredno branje iz podatkovne baze strežnika Cisco Unified Communications Manager, kar pomeni, da imamo neposreden dostop do podatkov na strežniku. Na tak način lahko s pravilno poizvedbo SQL pridobimo podatke o vseh telefonskih napravah v sistemu.

V skrajnih primerih, ko je podatkov zares veliko (primer nekaj tisoč naprav), je potrebno razmisliti o omejevanju prenosa količine podatkov. Omejevanja se lahko v prvi fazi lotimo kar s pomočjo samega vmesnika AXL, saj do neke mere omogoča omejevanje količine in hitrosti zaporednih zahtev. V nadaljevanju pa lahko sami omejimo dostop do strežnika AXL in do uporabniškega računa le na tisti čas, ko je sistem neuporabljen oz. je njegova uporaba zelo majhna (na primer ponoči oz. izven časa delovnih ur podjetja).

## 6.2 Smernice za nadaljnje delo

V diplomski nalogi izdelana ukazna datoteka pokriva zgolj določene tipe nalog, ki so se mi po ogledu dokumentacije podjetij zdele najbolj pomembne za opis stanja v sistemu. Z vmesnikom AXL je seveda možno storiti mnogo več, kot je to razvidno iz dokumentacije [1] in [8]. Vsako novo nalogo, ki bi jo želeli dodati v ukazno datoteko, bi morali realizirati z dvema podprogramoma: enim za pridobitev podatkov od strežnika, drugim za njihov zapis v datoteko XML. Za pridobitev podatkov je potrebno poiskati točno določene klice v dokumentaciji AXL API [1], ki pripravijo poizvedbo, s katero dobimo želene podatke. Za prikaz podatkov je potrebno dodati tudi ustrezno pretvorbo v transformacijsko datoteko XSLT.

Diplomsko delo bi lahko nadgradil z uporabo datotek XSD, tako da zapisovanja različnih elementov sheme XML ne bi vključevali neposredno v ukazno datoteko Perl, ampak bi datoteka XSD določala, kam se zapisujejo določeni podatki. V primeru spremembe na shemi bi bilo potrebno popraviti shemo XSD in dodati popravke za pridobivanje podatkov iz strežnika. Samo pisanje v dokument XML pa se ne bi spreminjalo.

Ukazno datoteko bi lahko preuredili, da bi podpirala dinamično omejevanje (*dynamic throttling*) količine podatkov, prenesenih preko klicev SOAP. Zahteve za poizvedbo podatkov in spremljanje sistema v realnem času pa naj bodo pripravljene na način, da količina podatkov in hitrost zaporednih poizvedb ne vplivata na zmogljivost procesiranja klicev. V tem primeru velja posebej opozoriti, da trenutna implementacija vmesnika AXL ne podpira omejevanja za naslednje zahteve:

- poizvedba `executeSQLQuery`,
- zahteva `doDeviceReset`,
- vse zahteve `get`,
- vse zahteve `list`.

Na koncu bi predstavil še nekaj primerov možnosti uporabe ukazne datoteke:

- **Aplikacija uporabniškega imenika**, s katero bi vsakdo oziroma samo tisti, ki so jim takšne pravice dodeljene, lahko v realnem času videl, kdo v podjetju je trenutno zaseden s telefonskim klicem in kdo ne. Seveda v tem primeru ne bi uporabljali celotne ukazne datoteke, ampak bi iz nje izluščili zgolj tiste klice vmesnika, ki so za to nalogo pomembni. Vredno je omeniti, da nekateri telefoni podjetja Cisco Systems že podpirajo to funkcionalnost in lahko na samem telefonu vidimo, ali je uporabnik trenutno zaseden.
- **Dokazovanje kakovosti storitev**, s čimer bi lahko na podlagi pridobivanja podatkov dokazali ponudniku telefonije IP, da njihova povezava za povezovanje omrežij v nekem časovnem obdobju ni delovala in smo za klice v oddaljeno pisarno morali uporabljati omrežje drugega operaterja, kar je povzročilo dodatne stroške v podjetju.
- **Oddaljeno upravljanje telefonskih aparatov**, ki ga omogoča ukazna datoteka, s pomočjo katere lahko v trenutku na vseh telefonih v podjetju zamenjamo strojno programsko opremo (*firmware*) in tako nadgradimo telefonske aparate na zadnjo različico, ki jo priporoča proizvajalec telefonov.
- **Izvoz nastavitv in podatkov** lahko izvedemo s pomočjo ukazne datoteke za potrebe varnostnih kopij ali kloniranja nastavitv strežnika na druge strežnike v gručah. Strežnik Cisco Unified Communications Manager nam sicer omogoča upravljanje z varnostnimi kopijami nastavitv strežnika, ne omogoča pa neposrednega kloniranja.

Prednost opisanega pristopa k izdelavi dokumentacije je tudi prožnost, ki nam omogoča, da podatke shranimo v nekem standardnem formatu, kasneje pa jih lahko s pomočjo datoteke XSLT pretvorimo v želen končen format. Ta je lahko Office Open XML ali Microsoft Office XML, ki ju je možno uporabiti za dokumentacijo v papirnati obliki. Lahko je CSV, ki ga je možno uporabiti za uvoz v druge programe, ali pa si po potrebi sami sestavimo tip formata, ki bi nam ustrezal za dosego ciljev. Format je lahko tudi XHTML ali HTML, kot to prikazuje diplomska naloga.



## A Priloge

## Seznam slik

Slika 1. Opisni elementi jezika WSDL.....	3
Slika 2. Primer dokumenta sheme XML. ....	4
Slika 3. Potek pretvorbe podatkov s pomočjo dokumenta XSLT. ....	5
Slika 4. Primer dokumenta XSLT. ....	5
Slika 5. Struktura sporočila SOAP. ....	6
Slika 6. Primer arhitekture usmerjevalnega načrta.....	13
Slika 7. Primer uporabe sheme Cisco AXL.....	21
Slika 8. Celoten krog dogodkov pri samodejni izdelavi dokumentacije, od zahteve za pridobitev podatkov do prikaza v spletnem brskalniku uporabnika.....	23
Slika 9. Shema XML, izdelana za diplomsko nalogo, ki služi za izvoz podatkov iz strežnika Cisco Unified Communications Manager. ....	24
Slika 10. Izvorna koda za pridobitev informacij o vseh napravah. ....	28
Slika 11. Izvorna koda za zapisovanje podatkov v datoteko XML.....	28
Slika 12. Sporočilo SOAP s poizvedbo SQL. ....	29
Slika 13. Del sporočila SOAP z odgovorom strežnika.....	29
Slika 14. Prikaz dela podatkov, shranjenih v datoteki XML.....	30
Slika 15. Enostaven primer prikaza izdelane dokumentacije.....	31
Slika 16. Izsek sheme, kjer je določen prikaz uporabnikov in njihovih mobilnih profilov. ....	32

## Seznam tabel

Tabela 1. Prikaz telefonov IP podjetja Cisco Systems in drugih proizvajalcev, ki so podprti v samem strežniškem sistemu Cisco Unified Communications Manager.....	9
Tabela 2. Dva tipa uporabniških računov v strežniku Cisco Unified Communications Manager.....	10
Tabela 3. Primer aplikacijskih uporabnikov v strežniku Cisco Unified Communications Manager.....	11
Tabela 4. Seznam usmerjevalnih komponent, ki so lahko uporabljene kot cilj nekega klica. .	12
Tabela 5. Seznam možnih entitet, na katere je moč preusmeriti klic.....	12
Tabela 6. Seznam najbolj pogostih nadomestnih znakov v usmerjevalnih vzorcih.....	14
Tabela 7. Primeri usmerjevalnih vzorcev.....	14
Tabela 8. Primer zahtev klicnih privilegijev.....	16
Tabela 9. Elementi klicnih privilegijev.....	17
Tabela 10. Pomen posameznih grafičnih simbolov v shemi vmesnika AXL.....	20

## Literatura

- [1] (2010) Cisco Unified Communications Manager XML Developers Guide Release 7.1(2). Dostopno na: [http://www.cisco.com/en/US/docs/voice\\_ip\\_comm/cucm/devguide/7\\_1\\_2/AXL\\_Dev\\_Guide.html](http://www.cisco.com/en/US/docs/voice_ip_comm/cucm/devguide/7_1_2/AXL_Dev_Guide.html)
- [2] (2010) Cisco Developer Community. Dostopno na: <http://developer.cisco.com/web/cdc/home>
- [3] (2010) Cron. Dostopno na: <http://en.wikipedia.org/wiki/Cron>
- [4] (2010) Extensible Markup Language (XML). Dostopno na: <http://www.w3.org/XML/>
- [5] G. Gary, Introduction to Cisco Unified Communications Manager Interfaces, 2008
- [6] (2010) H.323. Dostopno na: <http://en.wikipedia.org/wiki/H.323>
- [7] (2010) Java. Dostopno na: <http://java.com/>
- [8] J. Krohn, Extending Cisco Unified Communications Manager using the Administrative XML/SOAP (AXL) API, 2008
- [9] (2010) Perl. Dostopno na: <http://en.wikipedia.org/wiki/Perl>
- [10] E. T. Ray in J. McIntosh, Perl and Xml, Sebastopol: O'Reilly & Associates, Inc., 2002
- [11] (2010) SCCP. Dostopno na: [http://en.wikipedia.org/wiki/Skinny\\_Call\\_Control\\_Protocol](http://en.wikipedia.org/wiki/Skinny_Call_Control_Protocol)
- [12] (2010) SIP. Dostopno na: [http://en.wikipedia.org/wiki/Session\\_Initiation\\_Protocol](http://en.wikipedia.org/wiki/Session_Initiation_Protocol)
- [13] (2010) SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). Dostopno na: <http://www.w3.org/TR/soap12/>
- [14] (2010) SoapUI. Dostopno na: <http://www.soapui.org>
- [15] S. Srinivasan, Advanced Perl Programming, Sebastopol: O'Reilly & Associates, Inc., 1997
- [16] (2010) Unified Communications / Voice Solutions introduction. Dostopno na: [http://www.cisco.com/en/US/netsol/ns340/ns394/ns165/networking\\_solutions\\_packages\\_list.html](http://www.cisco.com/en/US/netsol/ns340/ns394/ns165/networking_solutions_packages_list.html)
- [17] (2010) Web services description language (WSDL) Version 2.0 Part 1: Core Language. Dostopno na: <http://www.w3.org/TR/wsdl20/>
- [18] (2010) XSL Transformations (XSLT). Dostopno na: <http://www.w3.org/TR/xslt>