

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Janja Petauer

**Avtomatizacija procesov v sistemski
administraciji**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Peter Peer

Ljubljana 2013

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.



Št. naloge: 00391/2013

Datum: 03.04.2013

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

Kandidat: **JANJA PETAUER**

Naslov: **AVTOMATIZACIJA PROCESOV V SISTEMSKI ADMINISTRACIJI**
PROCESS AUTOMATIZATION IN SYSTEM ADMINISTRATION

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Na primeru potreb realnega podjetja prikažite avtomatizacijo procesov ustvarjanja, spreminjanja ter odstranjevanja uporabniških računov znotraj aktivnega imenika. Primerjajte star in nov način dela.

Mentor:

doc. dr. Peter Peer

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Janja Petauer, z vpisno številko **63050318**, sem avtor diplomskega dela z naslovom:

Avtomatizacija procesov v sistemski administraciji

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Petra Peera
- 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 11. marec 2013

Podpis avtorja:

Zahvaljujem se vsem, zaradi katerih sem uspela spisati diplomu do konca; staršem za možnost študiranja, sodelavcema v sistemski administraciji za sodelovanje pri projektu ter vse podatke, ki sem jih lahko uporabila pri diplomu, mentorju in prijateljici za pregled diplome ter nasvete in fantu, ker mi je stal ob strani.

Večina ljudi odneha
tik preden dosežejo uspeh.
Ross Perot

Kazalo

Seznam uporabljenih kratic in simbolov

Povzetek

Abstract

1	Uvod	1
2	Osnovna orodja	3
2.1	RCG Assist	3
2.2	Aktivni imenik	6
2.3	Spletni servisi	8
2.4	Power Shell	11
3	Procesi v sistemski administraciji	13
3.1	Ustvarjanje novih uporabnikov	14
3.2	Spremembe na uporabnikih	15
4	Avtomatizacija procesov	17
4.1	Avtomatizacija ustvarjanja uporabnikov	17
4.2	Avtomatizacija odstranjevanja uporabnikov	28
4.3	Avtomatizacija sprememb na uporabnikih	31
4.4	Spletni servisi pri avtomatiziranih obrazcih	37
4.5	Varnost pri obrazcih za upravljanje uporabnikov	41

KAZALO

5	Primerjava	43
5.1	Prednosti	44
5.2	Slabosti	48
5.3	Ugotovitve	49
6	Zaključek	51

Seznam uporabljenih kratic in simbolov

ADO.NET ActiveX Data Objects, set računalniških programskih komponent, vključenih v .NET ogrodje

API Application Programming Interface

ASP Active Server Pages

C# programski jezik

HTML Hyper Text Markup Language

HTTP HyperText Transfer Protocol

IIS Internet Information Server/Services

MIME Multipurpose Internet Mail Extensions

MSSQL Microsoft Structured Query Language

OU Organization Unit

RPC Remote Procedure Call

SA System Administration

SOAP Simple Object Access Protocol

TCP/IP Transmission Control Protocol / Internet Protocol

UAG Unified Access Gateway

UDDI Universal Description, Discovery and Integration

UI User Interface

UM User Management

URI Uniform Resource Identifier

UTF-8 UCS Transformation Format 8-bit

VB.NET Visual Basic .NET

W3C WorldWideWeb Consortium

WML Website Meta Language

WSDL Web Service Definition/Description Language

XML Extensible Markup Language

Povzetek

Cilj diplomske naloge je predstaviti avtomatizacijo upravljanja uporabnikov na primeru podjetja Studio Moderna. Podjetje se je v zadnjih letih eksponentno večalo, zato je bilo potrebno najti hitrejši, lažji ter cenejši način upravljanja z uporabniškimi računi, kot je bil obstoječ. Avtomatizirali smo procese ustvarjanja, spreminjanja ter odstranjevanja uporabniških računov znotraj aktivnega imenika. V okviru avtomatizacije smo pripravili uporabniški vmesnik znotraj obstoječe aplikacije, katerega smo s pomočjo Java Scripta opremili s spustnimi meniji, spisali skripto v skriptnem programskem jeziku Power Shell ter postavili spletne servise, ki skrbijo za komunikacijo med aplikacijo ter aktivnim imenikom. Na koncu diplomske naloge je primerjava med starim ter novim načinom dela, ki pokaže na učinkovitost rešitve.

Ključne besede:

avtomatizacija, upravljanje z uporabniki, aktivni imenik, Power Shell

Abstract

The aim of the thesis is to present automatization of user management in company Studio Moderna. The company has grown exponentially in recent years, that is why we needed to find faster, easier and cheaper way of managing user accounts. We automatized processes of creating, changing and removing user accounts within Active Directory. We prepared user interface inside of existing application, used Java Script for drop down menus, wrote script in scripting programming language Power Shell and built web services for communication between application and Active Directory. At the end of the thesis there is comparison between old and new method, which shows the effectiveness of solutions.

Key words:

automatization, managing user accounts, Active Directory, Power Shell

Poglavje 1

Uvod

Točnost podatkov zaposlenih in ažurnost pri popravljanju le-teh so postali v podjetju Studio Moderna d.o.o. vedno bolj pomembni. Garažno podjetje, ki se je razvilo v mednarodno podjetje s podružnicami v 21 državah, je bliskovito povečevalo svoje zaposlene [1]. Iz nekaj deset zaposlenih je v kratkem času naraslo na sedanjih 6000 uslužbencev, kjer se dnevno dogajajo spremembe, kot so poroka, prezaposlitev, prihajanje ter odhajanje zaposlenih. Prav tako je potrebno vse te spremembe voditi ter jih redno osveževati. Sistemska administracija je sprva vse spremembe popravljala ročno, kmalu pa so ugotovili, da je dela vse več, časa pa vse manj. Pri taki količini sprememb se je povečala tudi možnost napake. Zato smo začeli razmišljati, katere ponavljajoče se procese bo najlažje avtomatizirati ter kako.

V diplomski nalogi bomo predstavili avtomatizacijo procesov, ki se izvajajo ob prihodu novega sodelavca, spremembe na njegovem uporabniškem računu med zaposlitvijo ter ob njegovem odhodu.

Avtomatizacija je procese poenostavila, pocenila ter pohitrila. Pripravili smo uporabniški vmesnik ter pravila delovanja le-tega, po katerih ga je podjetje RCG d.o.o. dodatno priredilo, sistemska administracija je spisala skripte v programskem jeziku Power Shell, podjetje RCG pa je postavilo spletne servise za komunikacijo med aplikacijo ter aktivnim imenikom.

V naslednjem poglavju so opisana orodja, s katerimi smo avtomatizirali

proces, uporabniški vmesnik, skriptni jezik ter aktivni imenik. V tretjem poglavju bomo opisali situacijo pred avtomatizacijo, izgled uporabniškega vmesnika ter proces. Nadaljevali bomo z opisom situacije po avtomatizaciji, delovanje uporabniškega vmesnika in kode. V zaključku bomo primerjali situaciji pred in po uvedbi sprememb, izpostavili prednosti ter slabosti in možne izboljšave.

Poglavje 2

Osnovna orodja

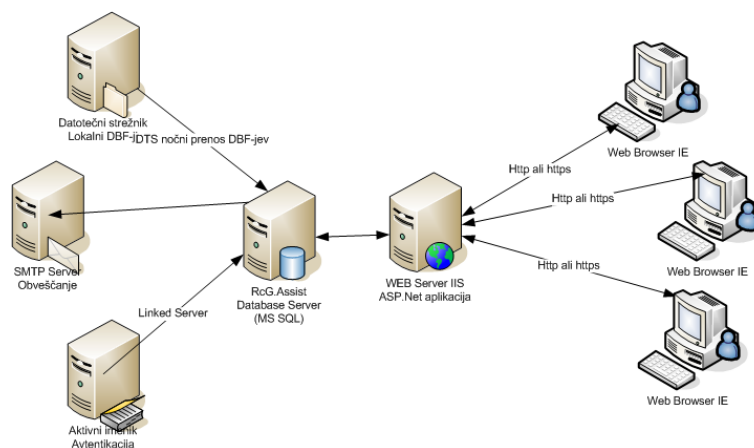
2.1 RCG Assist

RCG Assist je spletno zasnovano orodje za načrtovanje, izvajanje, upravljanje, nadziranje in analiziranje storitvenih poslovnih procesov znotraj podjetja. V našem podjetju se primarno uporablja kot aplikacija za pomoč uporabnikom (Helpdesk), kjer zaposleni v podjetju prijavljajo zahteve za celoten IT (okvarjene računalnike, potrebo po novi strojni ali programski opremi, nove uporabnike, vnos v šifrante ipd.). Vsaka zahteva je dodana v aplikacijo kot nov dogodek (angl. event) s svojo zaporedno številko ter privzetim lastnikom (angl. owner). Glede na tip zahteve se dogodki dodeljujejo (angl. redirect) agentom (zaposleni v IT). Komunikacija poteka prek odgovorov (angl. answer), prav tako ima vsak zahtevek svoj status, preko katerega končni uporabnik lahko vidi, kaj se dogaja z njegovo zahtevo. Na vsak zahtevek se lahko dodajo tudi udeleženci (angl. attendees), ki poleg kontakta (angl. contact – končni uporabnik, ki je oddal zahtevek) spremljajo reševanje zahteve. Ko je dogodek rešen, ima končni uporabnik možnost potrditi ali zavrniti rešitev. Aplikacija je zasnovana tako, da omogoča visoko prilagodljivost podjetju ter njihovem načinu dela. Obrazci (angl. forms) so enostavno sestavljivi ter omogočajo hitro popraviljanje, dodajanje ali odvzemanje polj ter tekstov. Veliko dodatnih funkcionalnosti si je moč narediti sam

brez posebnega znanja programiranja ali vključevanja razvijalcev. Prednosti aplikacije:

- Centralizirano prijavljanje zahtev
- Pregled nad reševanjem zahtevkov
- Mehanizem za merjenje ključnih mejnikov (čas prijave, odzivni čas, čas zaključka, čas zaprtja)
- Integracija s poštnim strežnikom (za zajem prijav in obveščanje vpletenih oseb)
- Revizijska sled za vsako podatkovno polje

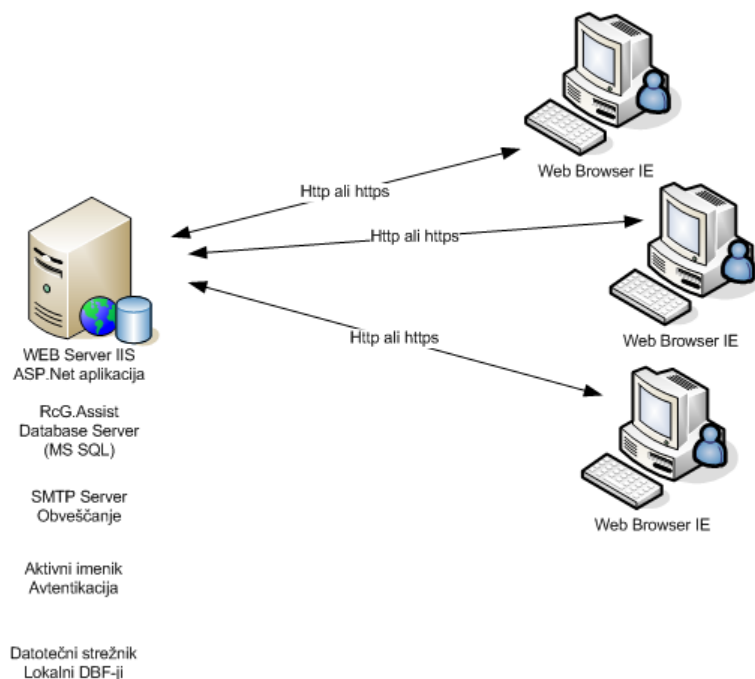
Aplikacija podpira tudi bazo znanja, kar pomeni, da je možno v njo dodati rešitve za ponavljajoče se probleme, ki so potem na voljo tudi ostalim agentom ter se tako olajšuje delo in pohitri reševanje problemov [2].



Slika 2.1: N-nivojska arhitektura postavitve

Programska rešitev RCG Assist je zgrajena na podlagi Microsoft spletnih tehnologij (.NET, ASP.NET, ADO.NET) za izdelavo dinamičnih spletnih aplikacij. ASP.NET aplikacije se izvajajo na spletnih strežnikih (IIS) ter generirajo HTML, WML, XML zapise, kateri se interpretirajo na namiznih oziroma mobilnih spletnih pregledovalnikih. Uporaba .NET tehnologij nam

omogoča potencialno uporabo različnih kompatibilnih programskih jezikov (VB.NET, C# ipd.) ter infrastrukturo za hitro izdelavo nadgradljivih spletnih aplikacij. Za izboljšano emuliranje uporabniških vmesnikov namiznih aplikacij, se programska logika RCG Assist izvaja na strani strežnika kot tudi na strani odjemalca (Java Script). Kot osnovo za delovanje uporablja programska rešitev RCG Assist podatkovno zbirko MSSQL, vendar za potrebe integracije z drugimi programskimi sistemi na njo ni omejena [3]. Na slikah 2.1 in 2.2 sta prikazani dve možni postavitev strežnikov za delovanje aplikacije. Slika 2.1 prikazuje N-nivojsko arhitekturo postavitve. Poleg podatkovnega ter aplikacijskega strežnika vsebuje tudi spletni strežnik.



Slika 2.2: Postavitev odjemalec-strežnik

Slika 2.2 prikazuje postavitev odjemalec-strežnik. Komunikacija med njima poteka s pomočjo niza standardnih aplikacijskih programskih vmesnikov (angl. API) in klicev oddaljenih procedur (angl. RPC).

Trinivojska arhitektura aplikacije je posledica dejstva, da se v večjih sistemih pri klasičnih aplikacijah tipa odjemalec-strežnik srečujemo s problemi vzdrževanja in razširjanja sistema z novimi funkcionalnostmi. Ločitev med

uporabniškim vmesnikom (angl. user interface, UI) in poslovno logiko je pomembna zaradi večje fleksibilnosti pri implementaciji, nadgradnji ter lažjega vzdrževanja aplikacije. Ločitev med podatkovnim nivojem in poslovno logiko je implementirano za zagotavljanje večje neodvisnosti aplikacije. To ne pomeni obvezno, da je logika shranjena drugje (npr. Stored Procedure na SQL strežniku), temveč da je ločitev narejena na konceptualnem nivoju. Pri več nivojskih aplikacijah lahko optimiziramo vsak nivo aplikacije posebej [3].

2.2 Aktivni imenik

Aktivni imenik je tip podatkovne baze, urejene v direktorij. Razlika med relacijsko bazo in direktorijem je, da je prva optimizirana za posodabljanje, medtem ko je druga optimizirana za branje. Zato je bil aktivni imenik razvit s predvidevanjem, da se objekti, vsebovani v direktoriju, ne bodo pogosto spreminjali, vendar bodo uporabljeni s strani uporabnikov, računalnikov in administratorjev za kontroliranje, managiranje in odkrivanje organizacijskih sredstev [4].

Aktivni imenik (angl. Active Directory, AD) je bil na ogled postavljen leta 1999, prvič je izšel skupaj z Windows 2000 Server in je skupaj z Windows Server 2003 doživel mnogo izboljšav ter dodatnih funkcionalnosti. Dodatne izboljšave so bile narejene skupaj z Windows Server 2003 R2, Windows Server 2008 in Windows Server 2008 R2, kjer je bil tudi preimenovan v Active Directory Domain Services [5].

Aktivni imenik je integriran, distribuiran imeniški servis, ki je vključen v večino Windows Server operacijskih sistemov. V aktivni imenik je integrirana množica aplikacij in storitev (slika 2.3), ki so prej zahtevale svojo imeniško strukturo in svoje uporabniško ime ter geslo za uporabo. Z uvedbo le-tega in prihodom aplikacij, ki so znale razširiti osnovno shemo aktivnega imenika in dodati svoje attribute, lahko administrator v organizaciji doda uporabnika ter ostale attribute, kreira poštni predal, elektronski naslov in podobno.

Sistem aktivnega imenika z uporabo sistema Kerberos skrbi za uporab-



Slika 2.3: Storitve aktivnega imenika [6]

nika, ki se prijavi v Windows okolje s svojim uporabniškim imenom in geslom, definiranim v aktivnem imeniku, da ga spremlja in odklene dostope do vseh aplikacij, kamor smo uporabniku dovolili. Na ta način se uporabnik samo enkrat prijavi v sistem, saj vzpostavljena povezava med aktivnim imenikom, poštno sporočilnim sistemom in integriranimi aplikacijami to omogoča. Na ta način se poenostavi tudi administracija uporabnikov, saj se v primeru sprememb te izvršijo na enem samem mestu znotraj administrativnega orodja aktivnega imenika.

V aktivnem imeniku se gradi hierarhična struktura uporabnikov in njihovih delovnih postaj. Mehanizem in način hierarhije se določi odvisno od potreb podjetja. Lahko se gradi glede na organizacijo v podjetju, fizične lokacije ali kaj drugega. Uporabnike in njihove delovne postaje med seboj v hierarhiji ločimo zaradi izvajanja skupinskih politik, saj se nekatere politike izvajajo nad delovnimi postajami, druge pa so lastne uporabnikom.

Omejeni mehanizem skupinske politike (angl. Group Policy), skupaj s prijavnimi skripti in možnostjo oddaljene administracije, skrbi za popolno obvladovanje delovnih postaj in uporabnikov v omrežju. Preko skupinske politike lahko uporabnikom omogočamo in onemogočamo uporabo posameznih funkcionalnosti delovnih postaj, predpisujemo izgled namizja, nameščamo tiskalnike, nastavljamo lastnosti internet brskalnika in podobno. S prijavnimi

skripti dosežemo, da se uporabniku na primer namestijo skupni direktoriji, izvedejo potrebni začetni (zagonski) programi in še veliko drugega [7].

2.3 Spletni servisi

Spletni servis predstavlja osnovo za nadgradnjo aplikacije ali uporabniškega programa v spletno aplikacijo, ki lahko ponudi uslugo ali servis drugim uporabnikom spleta ali medmrežja [8]. Spletni servisi imajo dvojni namen in uporabnost:

- *Ponovna uporaba aplikacij;*

Obstajajo mnoge funkcionalnosti, ki se v aplikacijah pogosto uporabljajo, kot na primer: menjava valut, napoved vremena, prevodi med jeziki itd. Zato se lahko takšne dejavnosti tipizirajo in nato večkrat uporabijo.

- *Povezovanje obstoječe programske opreme;*

Spletni servisi lahko zagotavljajo razne nivoje medopravnosti tako, da omogočajo povezavo in izmenjavo podatkov med različnimi spletnimi aplikacijami, bazami podatkov itd.

Spletni servisi uporabljajo na XML temelječe jezike za označevanje sestave podatkov ter servis SOAP za prenos sporočil (ob uporabi odprtih strojnih protokolov TCP/IP, MIME, HTTP itd.).

Spletni servisi temeljijo na večslojni tehnologiji (slika 2.4):

- SOAP
- WSDL
- UDDI



Slika 2.4: Večslojna tehnologija spletnega servisa

2.3.1 SOAP

SOAP je na XML temelječ jezik za izmenjavo podatkov med spletnimi aplikacijami preko protokola HTTP. Za dostop do spletnih servisov s uporablja RPC sporočil (slika 2.5). Njegove lastnosti so naslednje:



Slika 2.5: Osnovni elementi arhitekture oddaljenih klicev XML-RPC

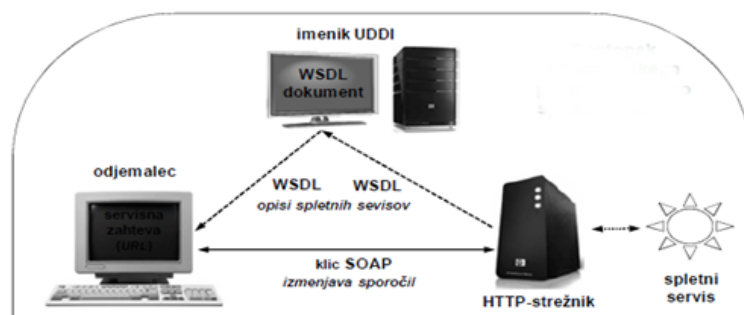
- pisni jezik za označevanje, ki temelji na jeziku XML;
- format za pošiljanje sporočil, ki deluje kot servis za komuniciranje;
- namenjen za komunikacijo po medmrežju;

- neodvisen od strojnega in programskega okolja (UTF-8);
- neodvisen od pogovornega ali programskega jezika;
- enostaven, razširljiv in nadgradljiv;
- SOAP sporočila ne ustavljajo požarni zidovi;
- SOAP je W3C standard.

2.3.2 WSDL

WSDL je na XML temelječ jezik. Zapis WSDL podaja metapodatke o določenem spletnem servisu (slika 2.6). WSDL ima vlogo vmesnika. Njegove lastnosti so naslednje:

- pisni jezik za označevanje, ki temelji na jeziku XML;
- uporablja se za predstavitev (metapodatki) in lociranje (URI) spletnih servisov;
- je W3C standard.



Slika 2.6: Postopek uporabniškega poizvedovanja po spletnem servisu

2.3.3 UDDI

UDDI je področni servis za organizacijo in objavo imenikov oziroma za XML-opis direktorijev, kjer se podjetja in uporabniki lahko registrirajo in hkrati iščejo ustrezne spletne servise.

Osnovne lastnosti UDDI so sledeče:

- imenik oziroma register za shranjevanje podatkov o spletnih servisih;
- je imenik vmesnikov do spletnih servisov, katerih opis in lokacija sta podana z jezikom WSDL;
- komunicira in posreduje sporočila s pomočjo protokola SOAP.

Med spletnimi storitvami različnih organizacij lahko pride do interakcije v primeru, ko:

- je možno zagotoviti ustrezno varnost;
- je znano, na kakšen način delujejo posamezni vmesniki WSDL;
- WSDL vmesniki dovoljujejo programskim vmesnikom drugih organizacij, da s pomočjo njih razširijo svojo uporabo (WSDL vmesniki so pridobljeni iz UDDI imenika) [9].

2.4 Power Shell

Lupina Windows Power Shell je ukazna lupina in skriptni jezik, ki se v celoti opira na ogrodje .NET. Sistemskim skrbnikom omogoča nadzirati množico avtomatiziranih procesov. Power Shell uporablja skrbniška opravila, ki se imenujejo cmdlets (command-lets). Vsi cmdlets-i (tabela 2.1) so poimenovani po vzorcu glagol-samostalnik, kot npr. dobi-proces (angl. Get-Process) ali dobi-vsebino (angl. Get-Content) [10]. Vsak vsebuje obvezne ali izbirne argumente, ki se imenujejo parametri, ki prepoznajo, s katerimi predmeti se ukvarja, ali nadzorujejo, kako cmdlet opravlja svoje delo. Ukaze cmdlet lahko

Power Shell ukazi	Opis
Get-ChildItem	Seznam vseh datotek / map v trenutnem direktoriju
Get-Content	Pridobi vsebino datoteke
Get-Command	Seznam vseh ukazov na voljo
Get-Help	Pomoč
Clear-Host	Izprazni zaslon
Copy-Item	Kopiraj eno ali več datotek / map
Move-Item	Premakni datoteko / mapo na drugo lokacijo
Remove-Item	Izbriši datoteko / mapo
Rename-Item	Preimenuj datoteko / mapo
Get-Location	Prikaži trenutno lokacijo (mapo)
Set-Location	Spremeni trenutno lokacijo (mapo)
Write-Output	Izpiši tekst / spremenljivke na standarden izhod
Get-Process	Seznam vseh trenutno delujočih procesov
Stop-Process	Končaj trenutni proces
Select-String	Prikaži vrstice, ujemaajoče z vzorcem
Set-Variable	Nastavi vrednost spremenljivki / ustvari spremenljivko

Tabela 2.1: Seznam najbolj pogosto uporabljenih cmdlet-ov

združujemo v skripte za opravljanje zahtevnih funkcij. Power Shell prav tako kot Unix ali Linux ukazne vrstice omogoča cevovod (angl. pipeline), s katerim urejamo, sortiramo in kopiramo podatkovni tok [11].

Poglavje 3

Procesi v sistemski administraciji

S hitro širitvijo podjetja je naraščalo tudi število zahtev po kreiranju in posodabljanju podatkov o uporabnikih. Aplikacija za pomoč uporabnikom se je v podjetje vpeljala leta 2009. Sistemski administratorji so najprej zahtevke, kjer je bila potreba po kreiranju, posodabljanju ali umiku uporabnikov, obdelovali ročno. Dokler je imelo podjetje še relativno malo zaposlenih, jim je to precej uspešno uspevalo obvladovati. Kmalu zatem se je podjetje bliskovito razširilo, obseg dela se je povečal, prav tako število projektov in za delo administrativnega tipa je začelo zmanjkovati časa, zato so začeli razmišljati o avtomatizaciji. Prvi korak je bil priprava izvoza (angl. export). RCG je glede na tip obrazca (System Administration – User – New) v aplikaciji pripravil pogled, ki je prikazoval vsa polja iz obrazca. Vsaka vrstica je podala enega uporabnika in je imela zapisane vse potrebne podatke za kreiranje. To se je izvozilo v .xls datoteko. Sistemski administrator je spisal skripto, ki je prebrala vrstico za vrstico v .xls datoteki ter le-te zapisovala v AD. Skripto so zagnali vsak drugi dan. Vendar je kmalu tudi to postalo prepočasi, saj so se stvari odvijale veliko hitreje. Zato so začeli razmišljati o popolni avtomatizaciji teh procesov. Naslednji korak sta bila nova obrazca User Management – New (UM – New) ter User Management – Remove (UM – Remove). Za

potrebe teh dveh obrazcev smo tudi povezali aplikacijo za pomoč uporabnikom ter AD in tako na določena polja dodali šifranke. Naslednji korak je bil uvedba obrazca User Management – Change (UM – Change), ki služi za spreminjanje podatkov uporabnikov. Vse tri obrazce se še zmeraj izpopolnjuje ter dopolnjuje, saj se skozi uporabo kažejo vedno nove ponavljajoče stvari, ki jih je možno avtomatizirati.

V spodnjih dveh podpoglavjih bomo opisali delovanje prvotnih obrazcev za kreiranje in spreminjanje uporabnikov, njihovo uporabo ter proces dela. Obrazec za kreiranje novih uporabnikov se je imenoval System Administration – User – New, obrazec za spreminjanje podatkov in ostale spremembe na uporabniku pa System Administration – User – Change. Obrazca za odstranitev uporabnika nismo imeli, saj tudi nismo imeli definiranega procesa. Če so že prijavi, da bo oseba prenehala delati, so to storili na obrazcu za spremembe na uporabniku.

3.1 Ustvarjanje novih uporabnikov

Zahteve za ustvarjanje novih uporabnikov so bile prvotno prijavljene na System Administration – User – New (slika 3.1).

Že ta obrazec je omogočal prijavo za ustvarjanje več uporabnikov naenkrat. Na glavnem zahtevku je bilo potrebno izpolniti naslov zaposlenih, na podzahtevkih pa ime, priimek, uporabniško ime, oddelek ter mesto zaposlitve, nadrejenega, določiti domeno za e-naslov ter izbrati začetne AD skupine. Uporabnike se je ustvarjalo vsak drugi dan. V aplikaciji smo sistemski administraciji dodali poseben pogled, ker so si glede na datum prijave razvrstili zahteve ter jih izvozili v .csv datoteko. To datoteko je potem skripta prebrala ter podatke zapisala v AD.

▼ Post event - **System administration - User - New**

Company
n/a

Contact (Petauer Janja)

Priority *
3 - Normal

Reported for *
IN900 - Slovenia, International

Short Description
Nov uporabnik

Long Description
Prosimo kreirajte novega uporabnika

Due date

Attendees

Approver 1

Header (the same for all users)


Street *
Litostrojska cesta 52

City *
Ljubljana

Zip Code *
1000

Please note that this form (User-New) is processed automatically therefore please create separate events for special requests.

User data (fill out when the event is saved)

#	first name	last name	username	department
 No data!				

Records found: 0 143 ms

Slika 3.1: Primer obrazca System Administration – User - New

3.2 Spremembe na uporabnikih

Prvotni obrazec (slika 3.2), ki je bil namenjen sporočanju sprememb pri podatkih uporabnikov, se je imenoval System Administration – User – Changes.

Na njem so bila navadna tekstovna polja, v katere si vnesel nove podatke. Preko obrazca se je sporočalo nov priimek, menjava oddelka dela, delovno mesto ali lokacijo, novo geslo in dodajanje e-naslova. Preko te forme se je sporočalo tudi, če je kdo prenehal z delom in ga je bilo potrebno deaktivirati v aktivnem imeniku. Načeloma pa tega procesa nismo imeli definiranega in smo ga uvedli šele pri avtomatizaciji. Zahtevki, ki so prišli, so bili ročno obdelani, kar pomeni, da je bilo potrebno v AD zbrisati staro vrednost ter

Post event - System administration - User - Changes

Company: n/a

Contact (Petauer Janja)

Priority: Choose

Reported for: Choose

Short Description: [Text Field]

Long Description: [Text Area]

Due date: [Date Picker]

Attendees: [Text Area]

Approver 1: [Text Field]

Attachment... [Button]

Last name: [Text Field]

Phone number: + () [Text Field]

Mobile (GSM): + () [Text Field]

Department: [Text Field]

Floor: [Text Field] Insert office number of similar identification?/ e.g. floor

Job function: [Text Field]

Street: [Text Field]

City: [Text Field]

Zip Code: [Text Field]

E-mail Address:

- @linea-directa.eu
- @sekt.si
- @studio-moderna.com
- Enable OCS license
- None

Admin Password: [Text Field]

OLAP group:

- Admin
- Finance
- Management
- MediaPlanner
- None
- SalesChannel
- SalesDRTV
- SalesInternet
- SalesOutbound
- SalesPrint
- SalesWhsRetail
- Supervisor
- SupervisorBasic
- DemandplanningBasic
- DemanplanningCalc
- DemanPlanningRetail

Obligatory: Please choose OLAP Group (if none apply choose None).

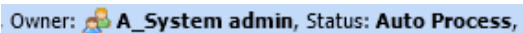
Slika 3.2: Primer obrazca System Administration – User – Changes

ročno vpisati novo. Povprečni odzivni čas pri tej vrsti zahtevka je bil 3 ure, povprečni čas reševanja enega takega zahtevka pa 6 ur. Odzivni čas pomeni čas od trenutka, ko je bil zahtevek prijavljen pa do trenutka, ko ga je eden izmed sistemskih administratorjev prevzel. Čas reševanja pomeni čas, ki se je porabil za reševanje, vendar v to ni všteti čas, ko se je čakalo na uporabnika v primeru dodatnih vprašanj. Kadar so imeli sistemski administratorji preveč dela z reševanjem drugih, kritičnih stvari, so bili taki zahtevki zelo pozno opaženi ter rešeni. To je znalo biti zelo nerodno v primeru, ko se je uporabnik zaklenil ali pa pozabil svoje geslo, saj ni mogel nadaljevati z delom. Zato smo začeli razmišljati tudi v smeri avtomatizacije, kar predstavljamo v naslednjem poglavju.

Poglavje 4

Avtomatizacija procesov

V tem poglavju bomo opisali vse tri obrazce, s katerimi v podjetju upravljamo z uporabniki. Vsi trije obrazci so avtomatizirani, skripta pa rešuje zahteve glede na status. Vsi zahtevki, ki imajo za lastnika skupino A_System admin ter so hkrati v statusu Auto Process, so v vrsti za obdelavo (slika 4.1).

A screenshot of a request condition in a software interface. The text is displayed in a light blue box with a white background. It reads: "Owner: A_System admin, Status: Auto Process,". There is a small icon of a person next to the text "A_System admin".

Slika 4.1: Pogoji za obdelavo zahtevka

V spodnji tabeli (tabela 4.1) so zapisana vsa polja na obrazcih, ki so povezana z AD, ter pripadajoča polja v AD.

4.1 Avtomatizacija ustvarjanja uporabnikov

Skozi obrazec User Management – New se kreirajo novi uporabniki (slika 4.7). Uporabnike ločujemo glede na to, ali so sodelavci notranji (torej zaposleni znotraj korporacije Studio Moderna) ali so zunanji (zunanji sodelavci, ki delajo na naših projektih). Zunanji sodelavci potrebujejo svoje uporabniško ime, da se preko UAG povezujejo v naše omrežje. Glede na izbiro uporabnika se določena polja različno obnašajo. Pri izpolnitvi zahtevka mora biti obvezno izpolnjeno polje Reported for, saj šele potem šifranti delujejo. Polje Reported for je namenjeno izbiri podjetja, kateremu bomo pripisali strošek

Ime polja na obrazcu	Ime polja v AD	Ime atributa v AD
Podjetje	Company	company
Ulica	Street	st
Poštna številka	Zip / postal Code	postalCode
Mesto	City	l
Država	Country / region	co
Uporabniško ime	User logon name	sAMAccountName
Ime	First name	givenName
Priimek	Last name	sn
Telefonska številka	Telephone numer	telephoneNumber
Mobilna številka	Mobile	mobile
FAX številka	Fax	
Delovno mesto	Job title	title
Oddelek	Department	department
Nadrejeni	Manager	manager
Nadstropje	Office	physicalDeliveryOfficeName
AD skupine	Member of	
Homefolder	Home folder	homeDirectory
Ustvari poštni nabiralnik		mail in proxyAddresses
Ostali poštni nabiralniki		mail in proxyAddresses
Primarni naslov	E-mail	Mail
Alias		mailNickname

Tabela 4.1: Povezava polj na obrazcu s polji v AD

tega zahtevka. Na en zahtevk se lahko zapiše kreiranje za več uporabnikov, vendar morajo biti vsi iz iste lokacije. V primeru, da vrednosti še ne obstajajo v AD (npr. kreiranje uporabnika, ki bo delal na novi lokaciji), je možen ročni vpis. V takem primeru te obrazec vsakič opozori na to. Te vrednosti se potem zapišejo v AD in so pri naslednji prijavi zahtevka na voljo za izbiro. Skripta za kreiranje novih uporabnikov [12] prek spletnih servisov pridobi vpisane podatke ter jih zapiše v aktivni imenik.

4.1.1 Notranji uporabnik

Slika 4.2 ter slika 4.3 prikazujeta glavni obrazec ter podobrazec za ustvarjanje novega uporabnika.

Zahtevana polja so:

- podjetje (angl. Company)
- naslov (angl. Street)
- mesto (angl. City)
- poštna številka (angl. ZIP code)
- država (angl. Country)
- ime (angl. First Name)
- priimek (angl. Last Name)
- oddelek (angl. Department)
- naziv delovnega mesta (angl. Job Function)
- nadrejeni (angl. Reports to)
- sekcija kreiranja poštnega nabiralnika in e-naslava (angl. Create mailbox and email address)

▼ Post event - User Management - New

Company: Studio Moderna d.o.o., Podvine 36, 1410 Zagorje ob Savi, http://www.studio-moderna.com, (+386)01/5899700

Contact (Petauer Janja): Petauer Janja, janja.petauer@studio-moderna.com

Priority *
3 - Normal

Reported for *
IN900 - Slovenia, International

Short Description
Novi uporabniki

User External Internal

Company
Studio Moderna d.o.o.

Street *
Litostrajska cesta 52


City *
Ljubljana

Zip Code *
1000

Country *
Slovenia

Attendees

User data

#	first name	last name	job function	department	floor	reports to	mail boxes	e-mail address	email	dont create e-mail	phone number	mobile (gsm)	fax number	home folder	ad group
 No data!															

Records found: 0 52 ms

Slika 4.2: Obrazec User Management – New (notranji uporabnik)

Opcijska polja so:

- telefonska številka (angl. Phone Number)
- številka mobilnega telefona (angl. Mobile (GSM))
- številka faksa (angl. FAX number)
- nadstropje (angl. Floor)
- AD skupine (angl. AD Group)

- geslo (angl. Admin Password)
- osebna mapa (angl. Home Folder)

V nadaljevanju so bolj podrobno opisala polja – katerega tipa so ter kako se obnašajo.

The screenshot shows a web-based form titled "User data new" with a "general" tab selected. The form contains the following fields and sections:

- First name ***: Janja
- Last name ***: Petauer
- Phone number**: (empty)
- Mobile (GSM)**: (empty)
- FAX number**: (empty)
- Department ***: IT
- Job function ***: Technical Support
- Reports To**: TestUA User
- Floor**: Second Floor
- E-mail Address ***: @studio-moderna.com, @linea-directa.eu, @selt.si
- Mail Boxes**: List of email addresses like FirstName.LastName@studio-moderna.com
- Alas:** (empty)
- AD Group**: (empty)
- Admin Password**: Geslo1
- Home Folder**: Tick if you want that home folder is created

Slika 4.3: Obrazec User Data (notranji uporabnik)

Polja Podjetje, Naslov, Mesto, Poštna številka: V primeru notranjega uporabnika se podjetje kopira iz polja Company (podjetje). Še vedno pa je možno vnos spremeniti. Če se izbere podjetje, se avtomatsko zapolnijo še polja Naslov, Mesto ter Poštna številka. Vsa 4 polja so tekstovna polja s spustnim seznamom.

Polji Ime in Priimek sta navadni tekstovni polji.

Polje Uporabiško ime: Na začetku so uporabniki sami vpisovali uporabniško ime, vendar smo polje, medtem ko pišem diplomo, ukinili. Ukinili smo ga zato, ker so se uporabniki velikokrat zatipkali v uporabniškem imenu ali ker je uporabniško ime že obstajalo, posledično so bili uporabniki z novim drugačnim uporabniškim imenom nezadovoljni. Na ta način smo se tem situacijam izognili. Uporabniško ime se ustvari avtomatsko s skripto po vzorcu ime.priimek (slika 4.4). V primeru če to uporabniško ime že obstaja, se novemu doda številko 1, torej ime.priimek1. Uporabniško ime ne more biti daljše od 20 znakov. Ko je uporabniško ime enkrat ustvarjeno, se ga ne da spreminjati.

```

Function GenerateUsername {
    Param ([String]$Name, [String]$Surname)
    $Name = $Name.Trim()
    $Surname = $Surname.Trim()
    if ($Name -match " ") {
        $Name = $Name.Substring(0,$Name.IndexOf(" "))
    }
    if ($Surname -match " ") {
        $Surname = $Surname.Substring(0,$Surname.IndexOf(" "))
    }
    $Username = $Name+'.'+$Surname
    if ($Username.length -gt 20) {
        $Username = ($Name.get_Chars(0))+'.'+$Surname
    }
    $Username = Convert-DiacriticCharacters $Username
    return $Username
}

```

Slika 4.4: Koda za ustvarjanje uporabniškega imena

Ker imamo zaposlene tudi v tujini, kjer uporabljajo drugačne znake, le-ta imena pretvarjamo s pomočjo funkcije Convert-DiacriticCharacters (slika 4.5).

Oddelek, Naziv delovnega mesta, Nadrejeni: Na vsa tri tekstovna polja je pripeljan šifrant vrednosti iz AD. Oddelek se bere iz polja Department, delovno mesto iz polja Job title, nadrejeni pa iz polja Manager.

```

Function Convert-DiacriticCharacters {
    param([string]$inputString)
        [string]$formD = $inputString.Normalize([System.text.NormalizationForm]::FormD)
        $stringBuilder = new-object System.Text.StringBuilder
        for ($i = 0; $i -lt $formD.Length; $i++) {
            $unicodeCategory =
                [System.Globalization.CharUnicodeInfo]::GetUnicodeCategory($formD[$i])
            $nonSPacingMark = [System.Globalization.UnicodeCategory]::NonSpacingMark
            if($unicodeCategory -ne $nonSPacingMark){
                $stringBuilder.Append($formD[$i]) | out-null
            }
        }
        $str = $stringBuilder.ToString().Normalize([System.text.NormalizationForm]::FormC)
        $str = $str -creplace 'đ', 'dj'
        $str = $str -creplace 'Đ', 'Dj'
        $str = $str -creplace 'Ě', 'L'
        $str = $str -creplace 'ě', 'l'
        return $str
    }
}

```

Slika 4.5: Del kode za pretvarjanje znakov

Sekcija kreiranja poštnega nabiralnika in e-naslava: Pri kreiranju poštnega predala in pripadajočega e-naslava so na voljo 3 možnosti, uporabnik pa lahko izbere samo eno. Ko je izbrana ena izmed možnosti se ostali dve zakleneta (oziroma posivita).

- Uporabnik izbere eno ali več domen, kjer želi imeti e-naslov (@studio-moderna.com, @selt.si, @linea-directa.si, @doremona.com). To se uporablja večinoma, ko gre za notranje zaposlene. Glede na izbiro se v polju Mailboxes pojavljajo variante. Vse, ki so notri zapisane, se bodo ustvarile, tista, ki pa je zapisana prva, pa bo postala primarni e-naslov. Vrstni red se lahko izbira s puščicami gor in dol. Dodano je tudi polje poštni vzdevek (angl. alias). V primeru, da je slednje izpolnjeno, se prav tako pojavi varianta poštni_vzdevek@studio-moderna.com. To polje se uporablja v primeru, ko ima zaposleni ime in priimek daljši od 19 znakov.
- Uporabnik vpiše že obstoječ e-naslov. To se uporablja večinoma za zunanje sodelavce, ki potrebujejo npr. dostop do UAG (zaradi avten-

tikacije je potrebno obvezno imeti v AD vpisan e-naslov).

- Uporabnik obkljuka Don't create e-mail. Ta opcija se uporablja za agente v klicnem centru, saj za svoje delo ne potrebujejo svojega poštnega predala.

Telefonska številka, Številka mobilnega telefona, Številka faksa: Polje dovoljuje vpis številke ter samo na 15., 16. ali 17. mestu dovoljuje vpis črke x. Telefonska številka se vpiše v formatu +xxxx_(xxx)_xxxxxxxx, kjer x predstavlja številke ter _ predstavlja presledek. V nekaterih državah (npr. Rusija) pa morajo posebej ločiti tudi interne telefonske številke, zato dodajo med telefonsko številko in interno številko znak X. Pri shranjevanju zahtevka se tudi preverja, če je vpisana črka X več kot enkrat ter javi napako. Prav tako pa tudi skripta preveri, če je telefonska številka pravilno zapisana in v primeru, da ni, javi napako (slika 4.6).

Nadstropje: Pripeljan je šifrant iz AD, ki črpa iz polja Office.

AD skupine: Polje črpa distribucijske (angl. distribution) in varnostne (angl. security), skupine, ki jih imamo v AD. Pri prijavi se lahko določene skupine izberejo, veliko pa se jih doda naknadno, glede na potrebe. V skripti imamo tudi vprogramiran (angl. hard-coded) seznam skupin, v katere se uporabnike ne da dodati avtomatsko (skupine z administratorskimi pravicami po državah).

Geslo: Privzeto je že vpisano neko geslo, ki se lahko tudi spremeni. Geslo, ki je vpisano, je geslo uporabnika, s katerim se prvič prijavi v operacijski sistem in ga mora tudi takoj obvezno spremeniti.

Osebna mapa (angl. Home Folder): V nekaterih državah že podpiramo možnost kreiranja osebne mape, za katero se redno delajo varnostne kopije. V primeru kljukice, se le-ta samodejno ustvari.

Vsa zgoraj našeta polja, so vključena v funkcijo za kreiranje novega uporabnika na sliki 4.7.

```

Function SyntaxOK {
    Param($user)
    [Boolean]$result=$true

    $result = $result -and (CheckSyntax-userPrincipalName ($user.userPrincipalName))
    $result = $result -and (CheckSyntax-UserName ($user.sAMAccountName))
    if (!(IsNullOrEmpty $user."telephoneNumber")) {
        $user.telephoneNumber = "+" + ($user.telephoneNumber.Trim())
        $result = $result -and (CheckSyntax-Phone $user."telephoneNumber")
    }
    if (!(IsNullOrEmpty $user."mobile")) {
        $user.mobile = "+" + ($user.mobile.Trim())
        $result = $result -and (CheckSyntax-Phone $user."mobile")
    }
    if (!(IsNullOrEmpty $user."faxN")) {
        $user.faxN = "+" + ($user.faxN.Trim())
        $result = $result -and (CheckSyntax-Phone $user."faxN")
    }
    }

    if (!$result) {
        Write-Host "Error in syntax for user: EventID - $($user.HD_ID) - check for details
        above" -ForegroundColor Blue
        $Global:HDanswer += "ERROR: Error in syntax. "
        return $false
    }
    else {
        return $true
    }
}

```

Slika 4.6: Koda za preverjanje zapisa telefonskih števil

4.1.2 Zunanji uporabnik

Pri kreiranju zunanjih uporabnikov se večina polj obnaša enako kot pri notranjem uporabniku, tako da bomo opisali samo spremembe. Če je izbrano polje zunanji uporabnik, se odklenejo polja za mesto, poštno številko in državo (slika 4.8).

```

Function CreateUser {
    Param($userinfo)
    $output += New-QADUser -name ($userinfo.givenName + " " + $userinfo.sn) `
        -parentContainer 'OU=Test,DC=sm-group,DC=local' `
        -SamAccountName $userinfo.sAMAccountName `
        -userPassword $userinfo.password `
        -Company $userinfo.company `
        -Department $userinfo.department `
        -Title $userinfo.Title `
        -StreetAddress $userinfo.streetaddress `
        -City $userinfo.l `
        -PostalCode $userinfo.postalcode `
        -StateOrProvince $userinfo.country `
        -FirstName $userinfo.givenName `
        -LastName $userinfo.sn `
        -UserPrincipalName $userinfo.userPrincipalName `
        -displayName ($userinfo.givenName + " " + $userinfo.sn) `
        -PhoneNumber $userinfo.telephonenumber `
        -MobilePhone $userinfo.mobile `
        -Fax $userinfo.facsimileTelephoneNumber `
        -Office $userinfo.physicalDeliveryOfficeName
        if (!(IsNullOrEmpty($userinfo.manager))) {
            $outL = Set-QADUser -identity $userinfo.UserPrincipalName -Manager
                $userinfo.manager -UserMustChangePassword $true
            $output += $outL
        }
}

```

Slika 4.7: Del kode za kreiranje uporabnika

Podjetje in pripadajoč naslov se lahko izbere iz šifrantu (če že obstaja), ali pa se vnese nove vrednosti.

Polji za vpis nadrejenega ter nadstropja sta v tem primeru zaklenjeni (sive barve), saj te podatke potrebujemo samo v primeru notranjih uporabnikov.

▼ Post event - **User Management - New**

Company Studio Moderna d.o.o., Podvine 36, 1410 Zagorje ob Savi, http://www.studio-moderna.com, (+386)01/5899700	Contact (Petauer Janja) Petauer Janja, janja.petauer@studio-moderna.com, (386)1/5899704
---	---

Priority *
3 - Normal

Reported for *
IN900 - Slovenia, International

Short Description
Zunanji uporabnik

User External Internal

Company
Zunanje podjetje

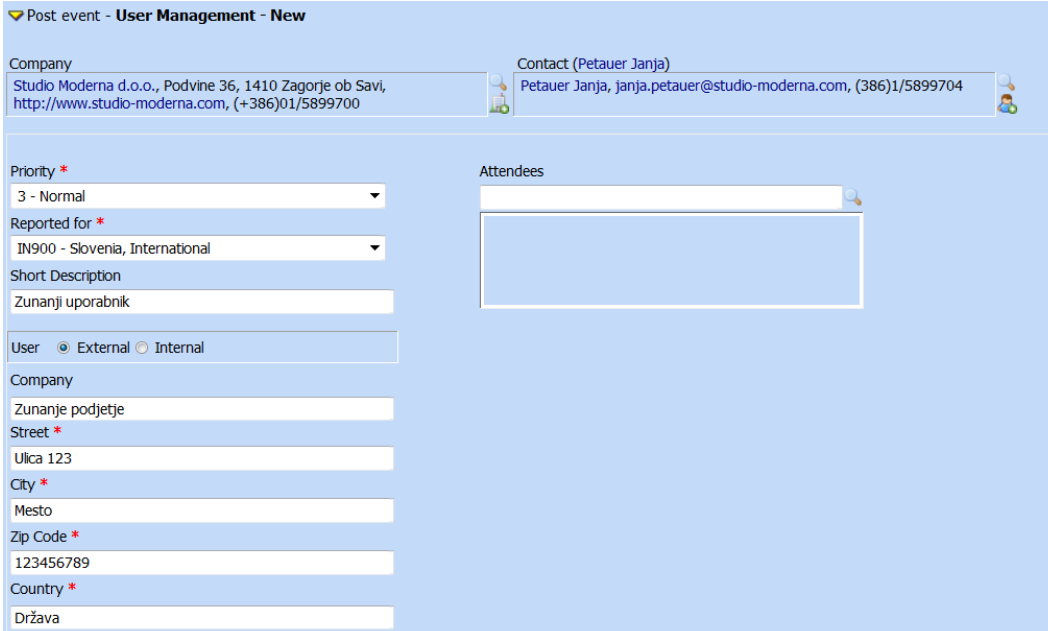
Street *
Ulica 123

City *
Mesto

Zip Code *
123456789

Country *
Država

Attendees



Slika 4.8: Obrazec User Management – New (zunanji uporabnik)

4.2 Avtomatizacija odstranjevanja uporabnikov

V primeru prenehanja zaposlitve je potrebno uporabniku odvzeti vse dostope do aplikacij (nekaj jih je še vedno dosegljivih tudi iz zunanje mreže), uporabniško ime pa onemogočiti. Za to uporabljamo obrazec User Management – Remove (slika 4.9). Na njej je potrebno označiti aplikacije, do katerih je imela oseba dostop. Tehnična podpora ročno preveri ter odvzame te dostope. V prihodnosti se bo ta korak avtomatiziral. Na en zahtevek je možno dodati več uporabnikov, vendar le tiste, ki so imeli iste dostope do aplikacij. Skripta za odstranjevanje uporabnikov [13] osebo v aktivnem imeniku deak-

#	ad group	department	email	e-mail address	enable ocs license	fax number	first name	floor	group membership	job function	last name	mobile (gsm)	owner	phone number	report to
141667		IT Technical support department				+__ (__)	Janja			Support Engineer	Petauer	+__ (__)		+__ (__)	

Slika 4.9: Zahtevek User Management – Remove

tivira, poštni nabiralnik začasno premakne v podatkovno skladišče (v pri-

meru, da bi bi potrebovali ponovno dostop, ga ne pobriše takoj), uporabnika umakne iz vseh distribucijskih in varnostnih skupin in ga premakne v drugo organizacijsko enoto za umaknjene uporabnike. Na podzahtevku (slika 4.10) je polje za uporabniško ime, ki je povezano z AD. V spustnem meniju se izbere uporabnika za odstranitev, nato se samodejno zapolnijo polja za ime, priimek, telefonsko številko, oddelek ter naziv delovnega mesta. Polji za ime ter priimek se obnašata isto; v vsako od teh treh polj lahko začneš vpisovati ime, priimek ali uporabniško ime in se bodo ostala polja zapolnila.

Slika 4.10: Podzahtevek User Management – Remove

Zadnja dopolnitev tega obrazca je pripis pomočnika oziroma kontakta ob odsotnosti (angl. out of office).

V primerih, ko je potrebno to nastaviti (slika 4.11), se v polje za e-naslov vpiše naslov, ki se ga doda v sporočilo samodejnega odgovora. Ob pritisku na gumb shrani se preveri, če ta naslov obstaja v AD. Če ne obstaja, javi

```

Function Set-UserAutoReply {
    Param ($User, $email)
    $country = (Get-QADUser -SamAccountName $User.sAMAccountName -IncludedProperties c).c
    $message = $null
    if (isnulloreempty $country) {
        $message = Get-AutoReplyMessage "EN" $email
    }
    else {
        $message = (Get-AutoReplyMessage $country $email) #+ "<P></P>" + (Get-AutoReplyMessage "EN"
                                                                $email)
        if (!(isnulloreempty $message)) {
            $message = $message + "<P></P>" + (Get-AutoReplyMessage "EN" $email)
        }
        else {
            $message = Get-AutoReplyMessage "EN" $email
        }
    }
    }
    Add-MailboxPermission -Identity $User.userPrincipalName -User
    ([System.Security.Principal.WindowsIdentity]::GetCurrent().Name) -AccessRights 'FullAccess'
    Start-Sleep -Seconds 10
    Set-EWSOofSettings -Identity $User.userPrincipalName -State Enabled -InternalReply $message -ExternalReply
    $message
    Write-Host "$message"
    Remove-MailboxPermission -Identity $User.userPrincipalName -User([System.Security.Principal.WindowsIdentity]::
    GetCurrent().Name) -InheritanceType 'All' -AccessRights 'FullAccess' -Confirm:$false
}

```

Slika 4.11: Koda za samodejni odgovor

napako. S tem smo želeli preprečiti vpisovanje zunanjih e-naslovov.

4.3 Avtomatizicija sprememb na uporabnikih

Zadnje ustvarjen obrazec je namenjen spremembam na uporabnikovi kartici v AD. Prvotna ideja je bila, da se naredi orodje, s katerim bi lahko lokalni IT po državah sam urejal podatke v AD, hkrati pa bi se spremembe tudi beležile. Preko tega obrazca se lahko popravi podatke (ime, priimek, lokacijo zaposlitve, delovno mesto, oddelek, telefonske številke), spremeni geslo, odklene geslo (v primeru, da se je uporabnik večkrat zmotil pri vnosu), doda ali odvzame AD skupino, doda ali odvzame poštni nabiralnik ter doda osebno mapo (samo za uporabnike iz držav, kjer to funkcionalnost že podpiramo). Skripta za spreminjanje uporabnika [14] prebere podatke ter jih prepíše čez obstoječe podatke v aktivnem imeniku.

Obvezno polje, ki se mora izpolniti, je polje Reported for. Ko se to polje izpolni, se ostala polja odklenejo, do takrat pa so sive barve. Naslednji korak je izbira zaposlenega. Na obrazcu lahko izbereš samo enega zaposlenega naenkrat. Na polje za uporabniško ime je pripeljan šifrant iz AD. Ko je uporabnik izbran, se samodejno zapolnijo polja s podatki, ki so zapisani na uporabnikovi kartici v AD. Polja, ki jih je potrebno popraviti, pobrišemo ter vanje vpišemo nove vrednosti. Na nekatera polja so vezani šifranti iz AD. Osvežujejo se na vsaki 2 uri.

Sprememba priimka: Po izbiri uporabnika, priimek pobrišemo ter zapišemo novega. Ko je zahtevek obdelan, se v AD spremenijo polja za priimek, prikazno ime (angl. display name) ter doda se nov e-naslov z novim priimkom. Na sliki 4.12 je del kode, ki prikazuje spremembo imena, priimka ter prikaznega imena.

Sprememba lokacije zaposlitve, oddelka dela, delovnega mesta: Po izbiri uporabnika, pobrišemo polje, ki ga je potrebno spremeniti. Na ta polja so vezani šifranti, zato v primeru da lokacija / delovno mesto / oddelek že obstaja, ga lahko izberemo iz padajočega menija. Koda na sliki 4.13 prikazuje spremembo podjetja.

Sprememba gesla: V primeru, da je potrebno spremeniti geslo, se v polje za geslo vpiše novo geslo. Ko bo spremenjeno, ga bo uporabnik moral pri

```

# Change Name
if ((!IsNullOrEmpty($user.givenName)) -and ($user.givenName -ne $thisUser.givenName)) {
    Set-QADUser -Identity $user.sAMAccountName-FirstName $user.givenName
    $changeDisplayName = $true
}
# Change Surname
if ((!IsNullOrEmpty($user.sn)) -and ($user.sn -ne $thisUser.sn)) {
    Set-QADUser -Identity $user.sAMAccountName-LastName $user.sn
    $changeDisplayName = $true
}
# Change Display name
if ($changeDisplayName) {
    Start-Sleep -Seconds 2
    $thisUser = get-qaduser -Identity $user.sAMAccountName -IncludedProperties 'c'
    $userDN = $thisUser.FirstName + " " + $thisUser.LastName
    Set-QADUser -Identity $user.sAMAccountName -DisplayName $userDN
    Rename-QADObject ('sm-group'+$user.sAMAccountName) -NewName $userDN
}
$message = $user.sAMAccountName+" - name and/or surname changed to:
"+$user.givenName+" "+$user.sn
LogMessage "$message" 1

```

Slika 4.12: Koda za spremembo imena, priimka in prikaznega imena

```

# Change Company
if($sprCompany) {
    if ((!IsNullOrEmpty($user.company)) -and ($user.company -ne $thisUser.company)) {
        Set-QADUser -Identity $user.sAMAccountName -Company $user.company
        $message = $user.sAMAccountName+" - company changed to: "+$user.company
        LogMessage "$message" 1
        $user.sortAD = $true
    }
    elseif($setBlanks) {
        Set-QADUser -Identity $user.sAMAccountName -Company ""
        $message = $user.sAMAccountName+" - company changed to: BLANK"
        LogMessage "$message" 1
    }
}

```

Slika 4.13: Koda za spremembo podjetja

prvi prijavi spremeniti. Gesla se v polju ne vidi, namesto tega se prikažejo samo črne pike.

Odklep uporabniškega računa: V primeru, da je uporabnik zaklenil svoj račun s tem, da je več kot petkrat napačno vnesel geslo, se aktivira polje za odklepanje računa (angl. Unlock account). Na sliki 4.14 je prikazana

funkcija za odklep uporabniškega računa.

```
Function UnlockAccount {  
    Param ($user)  
    Unlock-QADUser -Identity $user.sAMAccountName  
    $message = $user.sAMAccountName+" - account unlocked."  
    LogMessage "$message" 1  
}
```

Slika 4.14: Koda za odklep uporabniškega računa

Dodajanje ali odvzemanje uporabnika iz AD skupin: Če je potrebno uporabnika dodati ali odvzeti iz AD skupine, se v polju za AD skupino doda skupina s pritiskom na lupo. Odpre se šifrant z vsemi AD skupinami. Omogočeno je tudi iskanje glede na vpisano besedo. Na sliki 4.15 je del kode, ki prikazuje dodajanje uporabnika v skupino. Če je potrebno uporabnika odstraniti iz skupine, se jo s tipko Delete zbriše iz polja. Briše se lahko samo eno skupino naenkrat, čeprav je označenih lahko več.

Upravljanje z e-naslovi: Najnovejša funkcionalnost na obrazcu User Management – Change omogoča uporabniku dodajanje ali odvzemanje poštnega predala, dodajanje ali odvzemanje vzdevka ter spreminjanje primarno nastavljenega e-naslova (slika 4.16).

Področje smo razdelili na 3 dele:

- Ustvari poštni nabiralnik (angl. Create mailbox)
- Ostali e-naslovi (angl. Other e-mail addresses)
- Primarni e-naslov (angl. Primary e-mail)

Ustvari poštni nabiralnik: Vsak uporabnik, ki ima svoj poštni nabiralnik, ima tudi pripadajoča e-naslova ime.priimek@domena in priimek.ime@domena na eni izmed trenutno podprtih domen. Zaradi zgodovine podjetja in dejstva, da smo rastli hitreje, kot se je lahko urejalo podatke, ima veliko zaposlenih e-naslove na drugih domenah. Ko se obrazec odpre, se avtomatsko prebere iz AD domena (trenutno podpiramo 4) ter na obrazcu ustrezno prikaže. V primeru, da uporabnik poštnega nabiralnika še nima, se obkljuka, v kateri

```

if (!(IsNullOrEmpty($user.moNewGUID))) {
    $nGroups = $user.moNewGUID -replace "<br>", ","
    $nGroups = $nGroups.Split(",")

    Foreach ($group in $nGroups) {
        $group = guidToHex ($group.Trim())
        $GroupN = ((Get-QADGroup -Identity $group).samaccountname)
        if ((CheckExists-ADGroup $groupN) -and (DeniedGroups $group)) {
            Start-Sleep -Seconds 1
            Add-QADGroupMember -Identity $group -member ('sm-
            group\'+$user.sAMAccountName)
            $message = $user.sAMAccountName+" - added to group: "+$GroupN
            LogMessage "$message" 1
        }
        else {
            $message = $user.sAMAccountName+" - cannot be added to group:
            "+$groupN+" because group not exists in AD or contact not authorized for
            adding users to this group."
            LogMessage "$message" 3
        }
    }
}
}

```

Slika 4.15: Koda za dodajanje uporabnika v AD skupino

domeni ga potrebuje. Na sliki 4.17 je del kode za ustvarjanje poštnega nabiralnika.

V primeru, da uporabnik ne potrebuje več poštnega nabiralnika, se odkljuka domeno/e v sekciji ime.priimek in priimek.ime in skripta bo uporabniku poštni nabiralnik zbrisala. Na sliki 4.18 je koda za izbris poštnega nabiralnika.

V sklopu podprtih domen podpiramo tudi do en dodaten vzdevek, ki si ga lahko uporabnik sam izbere. V primeru, da ga še nima, se mu prikaže prazno polje za vpis. V primeru, da ga ima, se mu vzdevek izpiše ter ustrezno označi domena. Osebe, ki se poročijo ter s tem dobijo nova dva e-naslova, se jim oba naslova s starim priimkom zapišeta kot vzdevek. To je edina izjema, kjer dovoljujemo več kot en vzdevek na osebo.

Ostali e-naslovi: Pod ostale e-naslove se izpišejo naslovi na ostalih (še starih domenah) ter naslovi zunanjih partnerjev. Te naslove je mogoče samo

Slika 4.16: Uporavljanje s poštnimi nabiralniki

Create user mailbox with default email

```

Soutput += Set-Mailbox -Identity $userinfo.UserPrincipalName -EmailAddressPolicyEnabled $false
$contact = Get-Mailbox $userinfo.UserPrincipalName
$message = $userinfo.sAMAccountName+" - Primary email address created: "+$defMail
LogMessage "$message" 1

```

Slika 4.17: Koda za ustvarjanje poštnega nabiralnika

odstraniti, omogočamo pa možnost vpisa e-naslova zunanji osebi. Ker na tem obrazcu zunanje osebe nimamo posebej ločene, morajo za dovolitev vpisa biti izpolnjeni trije kriteriji:

- uporabnik nima poštnega nabiralnika ter pripadajočega e-naslova ime.priimek, priimek.ime;
- uporabnik nima poštnega vzdevka;
- uporabnik nima vpisanega še nobenega naslova v področju za ostale e-naslove.

V tem primeru se vnosna vrstica odklene in uporabnik lahko vpiše e-naslov.

```

# Delete additional email addresses
$NEmails = $contact.EmailAddresses | where {(!$_IsPrimaryAddress) -and ($_PrefixString -eq
    "SMTP") -and ($UserEmails -notcontains $_AddressString)}

if (!(isnulloreempty $NEmails)) {
    $Output += $contact | Update-List -Property EmailAddresses -Remove $NEmails
    Write-Host "Removed email addresses: $NEmails"
    $message = $userinfo.sAMAccountName+" - Removed email addresses: $NEmails"
    LogMessage "$message" 1
}

$Output += Set-Mailbox -Identity $userinfo.UserPrincipalName -EmailAddresses
    ($contact.EmailAddresses)
Write-Host "NNN: $UserEmails"
$Output += Set-CASMailbox -Identity $userinfo.UserPrincipalName -popEnabled:$false -
    ImapEnabled:$false -OWAEnabled:$true
}

```

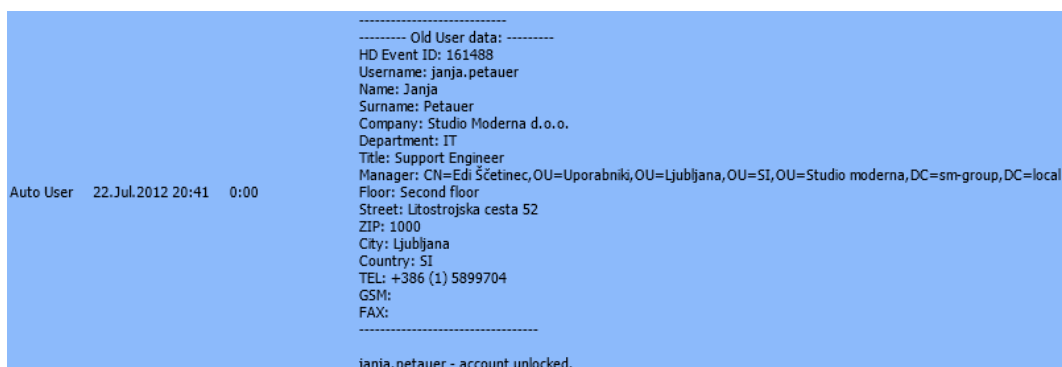
Slika 4.18: Koda za izbris poštnega nabiralnika

Preverja se, če je format vpisanega pravilen (v nasprotnem primeru javi napako), prav tako pa javi napako v primeru, če želi vpisati dva e-naslova.¹

Primarni e-naslov: V kolikor uporabnik ima poštni nabiralnik, se polje samodejno nastavi na trenutni primarni e-naslov. Polje je dinamični spustni meni (angl. drop-down menu), ki se samodejno polni, ko uporabnik dodaja nove naslove (npr. vpiše poštni vzdevek ter odkljuka domeno, vrednost se samodejno vpiše v spustni meni).

Ko je zahtevek oddan, gre avtomatsko v status Auto Process. Skripta se zažene avtomatsko vsakih 15 minut ter obdela oddane zahteve. Ko je zahtevek obdelan, vrne odgovor s starim stanjem podatkov ter spremembami, ki so bile narejene. Avtomatski uporabnik (angl. Auto User) je uporabnik, ki je dodan v aplikacijo samo za podajanje odgovorov (slika 4.19) iz skripte.

¹Slika nima sklica v tekstu! IMA:Upravljanje z enaslovi: Najnovejša funkcionalnost na obrazcu User Management – Change...(slika 4.19). Ali jo dodam višje ali se lahko v tekstu sklicuješ na neko sliko, čeprav sta vmes še 2 drugi?



Slika 4.19: Primer odgovora avtomatskega uporabnika

4.4 Spletni servisi pri avtomatiziranih obrazcih

Spletni servisi nam služijo za komunikacijo med različnimi aplikacijami, v našem primeru služijo za izmenjavo podatkov med aplikacijo za pomoč uporabnikom ter AD. Ukazna datoteka ali skripta (angl. script) iz spletnih servisov pridobi podatke, zapisane v zahtevku, ter jih zapiše v aktivni imenik. V spodnjih dveh podpoglavjih so predstavljeni spletni servisi, ki se uporabljajo pri izmenjavi podatkov med AD in aplikacijo za pomoč uporabnikov.

4.4.1 User Management – New / Remove

Skripta najprej pokliče spletni servis `dobiStatusZahtevka`, tako da dobi vse zahtevke, ki so v statusu `Auto Process`, torej čakajo na obdelavo. Potrebna parametra za klic tega spletnega servisa sta avtorizacijska vstopnica ter številka zahtevka. Odgovor spletnega servisa je v XMLformatu.

Na sliki 4.20 je primer klica spletnega servisa `dobiStatusZahtevka` iz skripte:

```
Sevents = $ws.get_event_status (avtorizacijska vstopnica, številka zahtevka)
```

Slika 4.20: Del kode, ki kliče spletni servis `dobiStatusZahtevka`

Na sliki 4.21 je primer odgovora spletnega servisa `dobiStatusZahtevka`.

Podatke, namenjene zapisu v AD, iz zahtevka na obrazcu `User Management – New / Remove` skripta dobi iz spletnega servisa `noviUporabnikiIzvoz`.

```

<?xml version="1.0" encoding="utf-8" ?>
<DataSet xmlns="http://rcg.com">
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
<xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocale="true">
<xs:complexType>
<xs:choice minOccurs="0" maxOccurs="unbounded">
<xs:element name="Table">
<xs:complexType>
<xs:sequence>
<xs:element name="idstat" type="xs:int" minOccurs="0" />
<xs:element name="status_desc" type="xs:string" minOccurs="0" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram xmlns:msdata="urn:schemas-microsoft-com:xml-msdata" xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
<NewDataSet xmlns="">
<Table diffgr:id="Table1" msdata:rowOrder="0">
<idstat>4</idstat>
<status_desc>Auto Process</status_desc>
</Table>
</NewDataSet>
</diffgr:diffgram>
</DataSet>

```

Slika 4.21: Odgovor spletnega servisa dobiStatusZahtevka

Potreben parameter je avtorizacijska vstopnica (angl. authentication ticket).

Na sliki 4.22 je del kode, ki kliče spletni servis noviUporabnikiIzvoz. Spletni servis vrne podatke za vse uporabnike, ki so dodani na zahteveke z obrazcem User Management – New in User Management – Remove in še niso rešeni.

```
Sevents = $ws.get_new_users_export (avtorizacijska vstopnica)
```

Slika 4.22: Del kode, ki kliče spletni servis noviUporabnikiIzvoz

Skripta potem glede na tip obrazca in ustrezen status pridobi tiste zahteveke, ki čakajo na ustvarjenje oz. odstranitev. Spodaj je del kode, ki poišče zahteveke za kreiranje novega uporabnika. Če je zahtevka v statusu Auto Process ter se obrazec imenuje User Management – New, bo skripta zahtevke obdelala.

Na sliki 4.23 je del odgovora, ki ga spletni servis vrne ob klicu. Odgovor je v XML formatu.

```

<Table diffgr:id="Table25" msdata:rowOrder="24">
  <event_no>Številka eventa</event_no>
  <sub_event_no>Številka podeventa</sub_event_no>
  <firstname>Janja</firstname>
  <lastname>Petauer</lastname>
  <username />
  <reported_for>International</reported_for>
  <department>Technical Support</department>
  <job_function>Support Engineer</job_function>
  <reports_to>janja.petauer</reports_to>
  <floor>Second floor</floor>
  <street>Litostrajska cesta 52</street>
  <zip_code>1000</zip_code>
  <city>Ljubljana</city>
  <reported_for_country>SI</reported_for_country>
  <reported_for_country_code>IN900</reported_for_country_code>
  <phone />
  <mobile />
  <fax />
  <password />
  <group_membership />
  <email>@studio-moderna.com</email>
  <ocs />
  <form_desc>User Management - New</form_desc>
  <homefolder>1</homefolder>
  <contact_username>janja.petauer</contact_username>
  <user>0</user>
  <company_1654>Studio Moderna d.o.o.</company_1654>
  <mail_boxes>FirstName.LastName@studio-moderna.com,LastName.FirstName@studio-moderna.com</mail_boxes>
  <alias />
</Table>

```

Slika 4.23: Del odgovora spletnega servisa novi_uporabniki_izvoz

4.4.2 User Management – Change

Podatke za obdelavo obrazca User Management – Change se pridobi preko spletnega servisa zahtevkiPoStatusu. Podani parametri so avtorizacijska vstopnica, ID obrazca (v tem primeru je to ID obrazca User Management – Change) ter ID statusa (v našem primeru je to status Auto Process, ki ima v bazi zapisan svoj ID). Na sliki 4.24 je del kode za klic spletnega servisa.

```
Sevents = $ws.get_forms_data_bystatus(avtorizacijska_vstopnica, ID_obrazca, ID_statusa)
```

Slika 4.24: Del kode za klic spletnega servisa zahtevekPoStatusu

Na sliki 4.25 je del odgovora spletnega servisa v XML formatu. Spletni servis za en zahtevek vrne 119 atributov.

```
<Table diffgr:id="Table1" msdata:rowOrder="0">
  <status_id>1021</status_id>
  <status_description>Auto Process</status_description>
  <company_short_name>International SM</company_short_name>
  <contact_fulname>Petauer Janja</contact_fulname>
  <obj1>Ljubljana</obj1>
  <obj2>Studio Moderna d.o.o.</obj2>
  <obj3>Second floor</obj3>
  <obj4>Support Engineer</obj4>
  <obj5>Petauer</obj5>
  <obj6>Janja</obj6>
  <obj7>Slovenia, International</obj7>
  <obj8>Litostrojska cesta 52</obj8>
  <obj9>1000</obj9>
  <obj10>AD skupine katerih član sem</obj10>
  <contact_username>janja.petauer</contact_username>
</Table>
```

Slika 4.25: Del odgovora spletnega servisa zahtevekPoStatusu

4.5 Varnost pri obrazcih za upravljanje uporabnikov

Ker ima dostop do aplikacije za pomoč uporabnikom približno 1800 ljudi, je bilo potrebno poskrbeti tudi za varnost podatkov v AD. Obrazci so avtomatsko obdelani, kar pomeni, da bi lahko zahtevo posredoval kdorkoli za kogarkoli in spreminjal podatke ali pa celo izbrisal račun. Države so nam posredovale imena največ dveh oseb odgovornih za upravljanje z uporabniki, primarno osebo ter rezervo v primeru bolezni ali dopusta. V AD smo ustvarili za vsako državo svojo varnostno skupino, ki se ločijo med seboj po kodi države. Skripta za vsak zahtevek najprej preveri (slika 4.26), če je oseba, ki je oddala zahtevek, v ustrezni skupini. V primeru da je, se zahtevek obdela, v primeru da ni, pa se zahtevek vrne tehnični podpori z obvestilom »Kontakt na zahtevku in uporabnik na zahtevku se ne ujemata. Kontakt mora biti avtoriziran za dodajanje novih uporabnikov.« (angl. »Contact on event and user on event do not match. Contact must be authorised for creating new users.«). V tem primeru tehnična podpora obvesti uporabnika, koga v državi kontaktirati ter prositi za pomoč.

```

Function checkLA {
    Param ($user, $la)
    $CCCode = CountryCode ((Get-QADUser -Identity $user).st)
    $OUUser = ((Get-QADUser -Identity $user).DN)
    if (IsNullOrEmpty $CCCode) {
        $CCCode = ((Get-QADUser -Identity $user -IncludedProperties 'c').c)
    }
    if ($OUUser -match ",OU=Fashion group,") {
        $CCCode = "SMFG"
    }
    if (IsNullOrEmpty $CCCode) {
        $CCCode = (((Get-QADUser -Identity $user).DN).split(", "))[((((Get-QADUser -Identity
        $user).DN).split(", ").count)-4)].substring(3)
    }

    if (!(IsNullOrEmpty $CCCode)) {
        $localAdminGroup = GetUMGroups $CCCode
        if (!(IsNullOrEmpty $localAdminGroup)) {
            $members = ((Get-QadgroupMember -Identity $localAdminGroup -Indirect | where {$_.Type -eq
            'user'}) | ForEach-Object {$_.samaccountname})
            if ($members -contains $la) {
                return $true
            }
            else {
                return $false
            }
        }
        else {
            return $false
        }
    }
    else {
        return $false
    }
}
}

```

Slika 4.26: Del kode, ki preverja članstvo v ustrezni skupini

Poglavje 5

Primerjava

Reševanje zahtevkov ter kakovost informacij se je z uvedbo novih obrazcev izboljšala. Zahtevki so hitreje rešeni, na račun šifrantov pa so informacije veliko bolj enotne. Kar nekaj časa je vzelo (in še ga bo) prečiščevanje že obstoječih uporabnik v AD ter urejanje informacij.

Prednosti:

- kakovost podatkov zaposlenih;
- hitrejše reševanje zahtevkov;
- manjša poraba časa za večjo količino zahtevkov;
- zapis sprememb.

Slabosti:

- izjeme niso podprte;
- povečan obseg izobraževanja končnih uporabnikov;
- povečano iskanje napak.

5.1 Prednosti

Kakovost podatkov zaposlenih

S pomočjo avtomatiziranih obrazcev, obveznih polj ter možnosti spreminjanja podatkov v AD tudi ostalim lokalnim administratorjem, se je kakovost in kvantiteta podatkov na zaposlenega precej dvignila. Prej je ustvarjanje uporabnika zahtevalo le ime in priimek. Šifranti z naslovi so se že precej poenotili, npr. samo za ulico v Ljubljani (Litostrojska cesta 52) je bilo kar nekaj variacij (Litostrojska 52, Litostrojska c. 52, Litostrojska ulica 52, Litostrojska ce. 52), ki so se sedaj poenotile. Zaradi obveznih polj, ki morajo biti izpolnjena, saj se drugače uporabnik ne skreira, imajo vsi novo-kreirani uporabniki vpisano lokacijo delovnega mesta, podjetje, v katerem so zaposleni, delovno mesto, oddelek ter ostale podatke, ki so pri staro-kreiranih uporabnikih bili velikokrat nepopolni ali celo prazni.

Hitrejše reševanje zahtevkov

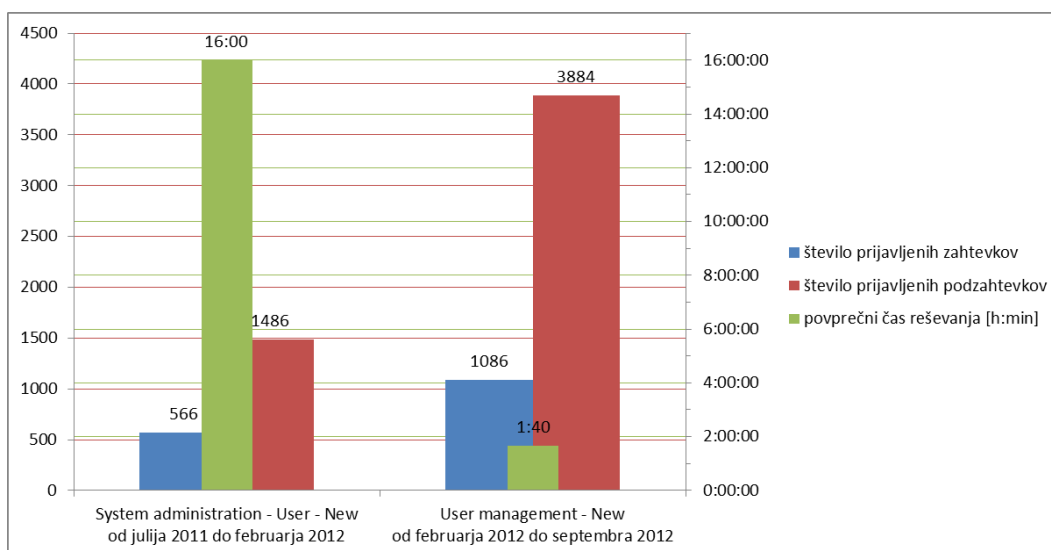
Čas reševanja se šteje od trenutka, ko je bil zahtevek prijavljen, do trenutka, ko se je njegov status spremenil na Solved. Če je bil zahtevek vmes dan na status CustWait, se ta čas ne šteje zraven. Status CustWait se uporablja v primeru, kadar uporabnik ni podal vseh informacij, ki jih potrebujemo za izvedbo. Tabeli 5.1 in 5.2 prikazujeta število zahtevkov ter čas reševanja le-teh pred uvedbo avtomatizacije ter po uvedbi, sliki 5.1 in 5.2 pa te vrednosti še grafično ponazorita za lažjo predstavo.

Manjša poraba časa za večjo količino zahtevkov

Poročani čas je čas, ki ga agent vpiše v aplikaciji za pomoč uporabnikom in podaja čas, ki ga je potreboval, da je prišel do rešitve. Razlika med časom reševanja in poročanim časom v urah je v tem, da se čas reševanja šteje avtomatsko v aplikaciji, poročani čas pa agent vpiše sam ter je zaokrožen na 5 minut. Tabela 5.3 nam prikazuje poročani čas iz zahtevkov pred uvedbo avtomatizacije in po uvedbi, slika 5.3 pa te podatke prikaže še v grafični obliki.

obrazec	SA-User-New	UM-New
Obdobje	julij 2011 do februar 2012	februar 2012 do september 2012
število prijavljenih zahtevkov	566	1086
število prijavljenih podzahtevkov	1486	3884
povprečni čas reševanja	2 delovna dneva	1 ura 40 min

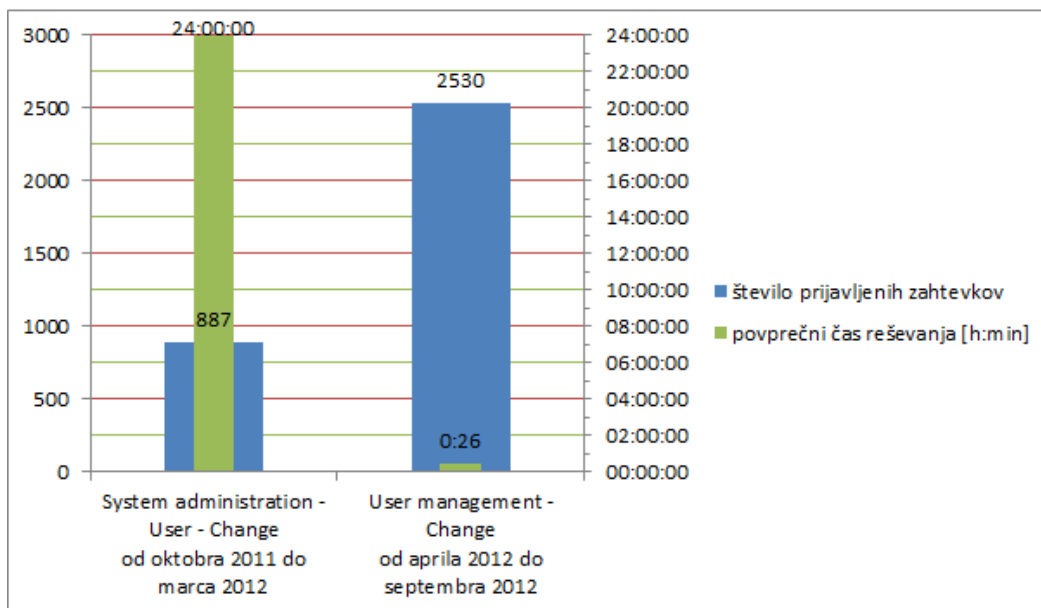
Tabela 5.1: Primerjava rešenih zahtevkov za novega uporabnika pred in po uvedbi avtomatizacije



Slika 5.1: Primerjava rešenih zahtevkov za novega uporabnika pred in po uvedbi avtomatizacije

obrazec	SA-User-Change	UM-Change
obdobje	oktober 2011 do marec 2012	april 2012 do september 2012
število prijavljenih zahtevkov	887	2530
povprečni čas reševanja	3 delovni dnevi	26 minut

Tabela 5.2: Primerjava rešenih zahtevkov za spremembe na uporabniku pred in po uvedbi avtomatizacije



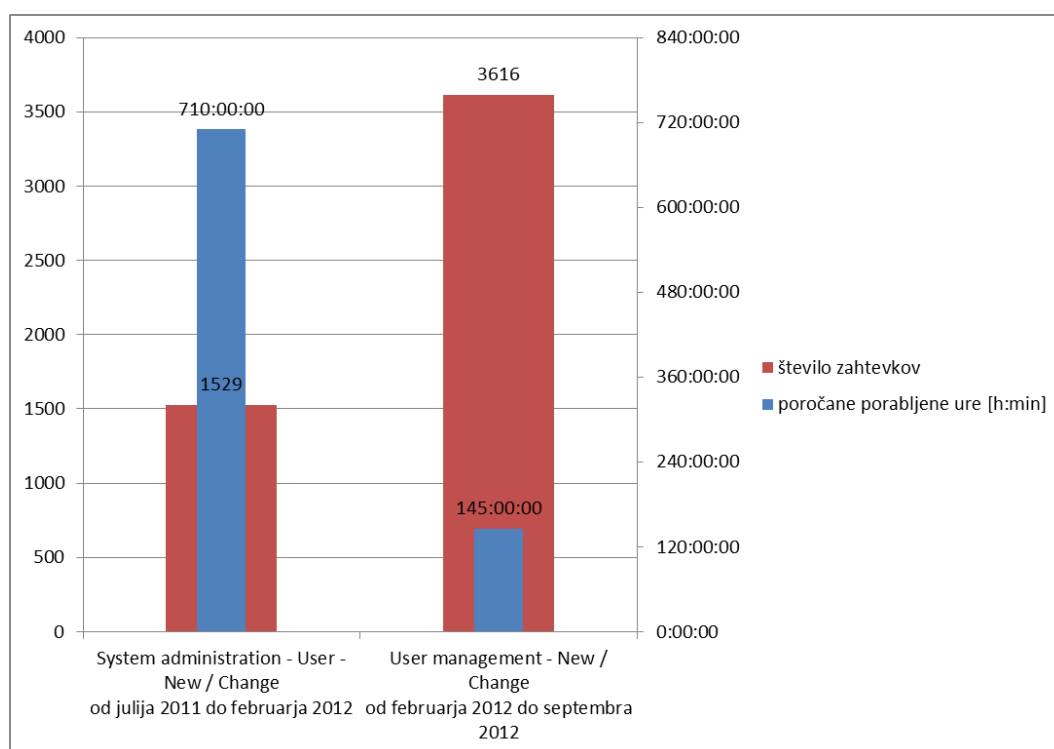
Slika 5.2: Primerjava rešenih zahtevkov za spremembe na uporabniku pred in po uvedbi avtomatizacije

obrazec	SA-User-New SA-User-Change	UM-New UM-Change
obdobje	julij 2011 do februar 2012	februar 2012 do september 2012
poročane porabljene ure	710	145
število zahtevkov	1529	3616

Tabela 5.3: Primerjava poročanega časa

Zapis sprememb

Ena izmed pomembnih prednosti je zapis sprememb. AD sam po sebi ne vodi evidence sprememb, tukaj pa imamo možnost preveriti, kdaj je bila sprememba narejena, kdo jo je zahteval in kaj se je spremenilo. Na tak način bi lahko tudi odkrili ter ustrezno sankcionirali zlorabo pravic, vendar do sedaj takega primera še ni bilo.



Slika 5.3: Primerjava poročanega časa

5.2 Slabosti

Izjeme niso podprte

V poplavi uporabnikov, ki jih premore Studio Moderna, so tudi t.i. izjeme, ki se jih ni dalo podpreti z avtomatiko. V to skupino spadajo uporabniki, kreirani v razne marketinške namene, ki imajo npr. poštni nabiralnik in pripadajoči e-naslov v eni izmed domen, ki niso podprte, razni stari uporabniki, ki še niso nadgrajeni na trenutno zahtevane podatke in jih je potrebno še ročno vpisati, itd.

Povečan obseg izobraževanja končnih uporabnikov

Z vsako uvedbo novosti pride zraven izobraževanje uporabnikov. Ker se je delo preneslo iz sistemskih administratorjev na lokalne sistemske administratorje po državah, se je tudi povečalo število vprašanj ter pomoči pri različnih operacijah. Za uporabnike smo spisali navodila, ki korak za korakom, skupaj s slikami, prikazujejo procese za vsako spremembo posebej. Navodila so bila razposlana in dodana na intranet, tako da lahko v vsakem trenutku do njih dostopajo. Sočasno s tem so imeli (in še imajo) vedno na voljo možnost kontakta preko e-pošte, komunikatorja ali telefona. Projekt, tak kot je, z vsemi funkcionalnostmi, je zaživel novembra 2012. Sedaj, 3 mesece kasneje (marec 2013), je stanje že veliko boljše. Uporabniki večinoma znajo sami brez problemov in dodatne pomoči izpolniti vse 3 obrazce. Potreba po pomoči se poveča (in ocenjujemo, da bo tako tudi v prihodnosti), ko se eden izmed lokalnih administratorjev zamenja, pa še to samo v tistih državah, kjer je edini (manjše države kot so npr. Kosovo, Češka). V večjih državah kot je npr. Rusija (ki ima trenutno 5 lokalnih administratorjev), se v primeru vprašanj vedno obrnejo najprej na svoje sodelavce.

Povečano iskanje napak

Sočasno z uvedbo nove rešitve je prišlo zraven tudi odpravljanje napak. Potrebno je zagotoviti nemoteno in brezhibno delovanje tako aplikacije za pomoč uporabnikom, sinhronizacije med njo in AD ter pravilno izvajanje

skripte v ozadju. Največkrat se zgodi, da skripta zavrne zahtevek zaradi napačnega kontakta (kontakt ni avtoriziran za oddajanje zahtevkov prek teh obrazcev, kontakt želi spremeniti podatke za osebo, za katero nima pravice (spada v drugo organizacijsko enoto)).

5.3 Ugotovitve

Projekt avtomatizacije procesov upravljanja z uporabniki smo pričeli januarja 2012. Novembra smo dodali na produkcijo še zadnje funkcionalnosti, ki se navezujejo na dodajanje ter odvzemanje poštnih nabiralnikov. Število zahtevkov se je vsaj podvojilo, čas reševanja pa se je zmanjšal za 8krat. Način dela nam sedaj omogoča shranjevanje vseh sprememb, narejenih v AD, hkrati pa se je delo iz sistemskih administratorjev preneslo na lokalni oddelek IT po državah. Obrazci so zastavljeni tako, da je vnos nekaterih podatkov obvezen in na ta način skrbimo tudi za pridobitev in vzdrževanje vseh potrebnih informacij o uporabniku. Vsi naslovi se bodo počasi poenotili, telefonske številke so v standardiziranem formatu. Slabost bi lahko bila odsotnost človeškega pregleda, ker skripta vsebinsko ne more pregledati vpisanih vrednosti, vendar istočasno učimo uporabnike pravilnega tako tehničnega kot vsebinskega izpolnjevanja obrazcev.

Poglavje 6

Zaključek

Pred začetkom projekta je bilo glavno vprašanje kako prenesti administrativni del urejanja uporabnikov iz systemske administracije na lokalni oddelek IT. Potrebovali smo rešitev, ki bo uporabniško prijazna in po možnosti znana, ki omogoča določeno vrsto zaščite pred zlorabo ter podpira zapisovanje sprememb. Najbolj logična izbira uporabniškega vmesnika je bila aplikacija za pomoč uporabnikom, saj jo poznajo ter uporabljajo vsi systemski administratorji ter lep delež končnih uporabnikov. Prav tako se v njej lahko vodi spremembe, saj je to ena izmed njenih prednosti. Ko smo uspeli povezati aplikacijo z našim AD tako, da je bila sposobna prebrati podatke ter jih prikazati, smo vedeli, da je to prava smer. Power Shell je logična izbira programskega jezika, saj je z njim najlažje pridobiti podatke iz aktivnega imenika. Veliko smo delali tudi na vnosnih poljih ter poskusili kar se da poenostaviti vnos podatkov. Kjer je bilo možno, smo dodali spustne menije ter dodali preverjanje vpisanih podatkov v poljih (npr. v polju za e-poštni naslov). Obrazce smo se trudili ohraniti čim krajše, s čim manj polji. Varnost smo zagotovili z varnostnimi skupinami znotraj AD in na ta način poskrbeli tudi za hitro ter enostavno zamenjavo odgovornih oseb. Funkcionalnosti smo dodajali počasi, eno za drugo, ter po korakih tako dobili tudi odzive končnih uporabnikov.

Na splošno je projekt dobro zaživel. Največjo pomanjkljivost vidim v

miselnosti uporabnikov. Ker so bili navajeni, da je dovolj navesti nekaj glavnih podatkov in počakati, da nekdo to uredi, je preskok na sedanji proces dolgotrajen. Potrebno je zbrati vse podatke, ki so zahtevani, preden se ali uporabnik ustvari ali spremeni, prav tako pa je pri odhodu potrebno naštetih aplikacije, ki jih je potrebno odstraniti. Na voljo so navodila, ki korak za korakom prikazujejo uporabo vseh 3 novih obrazcev, vendar jih večina le na hitro preleti. Zato to tudi večinoma prihaja do napak. Verjamem, da se bo to s časom izboljšalo.

V prihodnosti bi se lahko projekt širil. V načrtu smo imeli avtomatizacijo urejanja AD skupin (angl. group management) ter dodajanje računalnikov v skupine (angl. deploy groups) za namestitev programov (angl. computer management). Prav tako bi se lahko avtomatiziralo odstranjevanje uporabnika iz aplikacij. Trenutno tehnična podpora ročno preveri aplikacije ter uporabniku odstrani dostop.

Slike

2.1	N-nivojska arhitektura postavitve	4
2.2	Postavitev odjemalec-strežnik	5
2.3	Storitve aktivnega imenika [6]	7
2.4	Večslojna tehnologija spletnega servisa	9
2.5	Osnovni elementi arhitekture oddaljenih klicev XML-RPC	9
2.6	Postopek uporabniškega poizvedovanja po spletnem servisu	10
3.1	Primer obrazca System Administration – User - New	15
3.2	Primer obrazca System Administration – User – Changes	16
4.1	Pogoji za obdelavo zahtevka	17
4.2	Obrazec User Management – New (notranji uporabnik)	20
4.3	Obrazec User Data (notranji uporabnik)	21
4.4	Koda za ustvarjanje uporabniškega imena	22
4.5	Del kode za pretvarjanje znakov	23
4.6	Koda za preverjanje zapisa telefonskih števil	25
4.7	Del kode za kreiranje uporabnika	26
4.8	Obrazec User Management – New (zunanji uporabnik)	27
4.9	Zahtevek User Management – Remove	28
4.10	Podzahtevek User Management – Remove	29
4.11	Koda za samodejni odgovor	30
4.12	Koda za spremembo imena, priimka in prikaznega imena	32
4.13	Koda za spremembo podjetja	32
4.14	Koda za odklep uporabniškega računa	33

4.15	Koda za dodajanje uporabnika v AD skupino	34
4.16	Uporavljanje s poštnimi nabiralniki	35
4.17	Koda za ustvarjanje poštnega nabiralnika	35
4.18	Koda za izbris poštnega nabiralnika	36
4.19	Primer odgovora avtomatskega uporabnika	37
4.20	Del kode, ki kliče spletni servis dobiStatusZahtevka	37
4.21	Odgovor spletnega servisa dobiStatusZahtevka	38
4.22	Del kode, ki kliče spletni servis noviUporabnikiIzvoz	38
4.23	Del odgovora spletnega servisa novi_uporabniki_izvoz	39
4.24	Del kode za klic spletnega servisa zahtevkiPoStatusu	40
4.25	Del odgovora spletnega servisa zahtevkiPoStatusu	40
4.26	Del kode, ki preverja članstvo v ustrezni skupini	42
5.1	Primerjava rešenih zahtevkov za novega uporabnika pred in po uvedbi avtomatizacije	45
5.2	Primerjava rešenih zahtevkov za spremembe na uporabniku pred in po uvedbi avtomatizacije	46
5.3	Primerjava poročanega časa	47

Tabele

2.1	Seznam najbolj pogosto uporabljenih cmdlet-ov	12
4.1	Povezava polj na obrazcu s polji v AD	18
5.1	Primerjava rešenih zahtevkov za novega uporabnika pred in po uvedbi avtomatizacije	45
5.2	Primerjava rešenih zahtevkov za spremembe na uporabniku pred in po uvedbi avtomatizacije	45
5.3	Primerjava poročanega časa	46

Literatura

- [1] (2012) Studio Moderna. Dostopno na:
www.studio-moderna.com.
- [2] (2012) RCG Assist – informacijske tehnologije. Dostopno na:
www.rcg.si.
- [3] S. Kisovec, E. Ščetinec, *RCG Original Manual SLO HelpPages*, interno gradivo: Studio Moderna d.o.o., 2008.
- [4] J. A. Price, B. Price, S. Fenstermacher, *Mastering Active Directory for Windows Server 2008*, Indianapolis, Indiana, Wiley Publishing, Inc., 2008.
- [5] (2012) Active Directory. Dostopno na:
http://en.wikipedia.org/wiki/Active_Directory
- [6] (2012) Storitve aktivnega imenika. Dostopno na:
<http://www.activedirectorytutorial.org/wp-content/uploads/67383316.gif>.
- [7] M. Minasi, C. Anderson, M. Beveridge, L. Justice, *Mastering Windows Server 2003*, Alameda, CA: Sybex, 2003.
- [8] D. A. Chappell, T. Jewell, *Java Web Services*, Gravenstein Highway North Sebastopol, CA: O'Reilly & Associates, 2002.
- [9] (2012) Spletni servisi OGC. Dostopno na:
<ftp://ftp.fgg.uni-lj.si/Sendable/Geodezija>

- [10] L. Holmes, *Windows PowerShell CookBook*, Second Edition, Gravenstein Highway North Sebastopol, CA: O'Reilly, 2010.
- [11] (2012) Windows PowerShell. Dostopno na:
http://en.wikipedia.org/wiki/Windows_PowerShell.
- [12] M. Kovač, D. Ferbar, *Skripta za User Management – New*, interno gradivo: Studio Moderna d.o.o., 2012.
- [13] M. Kovač, D. Ferbar, *Skripta za User Management – Remove*, interno gradivo: Studio Moderna d.o.o., 2012.
- [14] M. Kovač, *Skripta za User Management – Change*, interno gradivo: Studio Moderna d.o.o., 2012.