

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Domen Mladovan

**Grafični uporabniški vmesnik za izbiro
časovnih pasov v orodju za upravljanje
elektroenergetskih naprav**

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2011



Št. naloge: 00105/2011

Datum: 04.04.2011

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

Kandidat: **DOMEN MLADOVAN**

Naslov: **GRAFIČNI UPORABNIŠKI VMESNIK ZA IZBIRO ČASOVNIH PASOV V
ORODJU ZA UPRAVLJANJE ELEKTROENERGETSKIH NAPRAV
GRAPHICAL USER INTERFACE FOR SELECTING TIME ZONES IN A
POWER SYSTEM MANAGEMENT TOOL**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V diplomski nalogi najprej predstavite načine hranjenja podatkov o časovnih pasovih s pomočjo ustreznih datotek. Nato predstavite nastavitve in uporabo strežniškega in odjemalčevega dela orodja PSM, ki služi za upravljanje in nadzor elektroenergetskih naprav. Orodje nagrađite z učinkovitim grafičnim vmesnikom za izbiro ustreznih časovnih datotek ter prenos le-teh na napravo. V nalogi se osredotočite na prikaz ustrezne tehnične rešitve nadgradnje.

Mentor:

viš. pred. dr. Igor Rožanc

Dekan:

prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Domen Mladovan,

z vpisno številko 63060204,

sem avtor/-ica diplomskega dela z naslovom:

Grafični uporabniški vmesnik za izbiro časovnih pasov v orodju za upravljanje

elektroenergetskih naprav

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

viš. pred. dr. Igor Rožanc

in somentorstvom (naziv, ime in priimek)

/

- 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 4.7.2011 Podpis avtorja/-ice: _____

ZAHVALA

Ob zaključku diplomskega študija bi se rad zahvalil svojemu mentorju viš. pred. dr. Igorju Rožancu za pomoč pri diplomskem delu.

Zahvala gre tudi celotni ekipi PRR-ZVE, še posebej Iztoku Kobalu za strokovno pomoč in spodbudo.

Posebna zahvala gre mojim staršem, ki so mi omogočili in me podpirali skozi celoten študij.

KAZALO

POVZETEK	1
ABSTRACT	2
1 UVOD	3
2 OPIS ČASOVNIH DATOTEK IN ZONEINFO DIREKTORIJA	5
2.1 Opis časovnih datotek.....	5
2.1.1 Definicija časovne datoteke	5
2.1.2 Na splošno o časovnih datotekah.....	5
2.1.3 Izvorna koda časovnih datotek	6
2.2 Sestava zoneinfo direktorija.....	9
2.2.1 Zoneinfo direktorij.....	9
2.2.2 Časovne datoteke razdeljene po celinah	9
2.2.3 Datoteka s pravilom za prehod iz poletnega v zimski čas	9
2.2.4 Direktorij z GMT datotekami.....	11
2.2.5 Direktorij Posix	11
2.2.6 Direktorij Right	12
2.2.7 Ostale časovne datoteke	12
2.2.8 Posodobitev in distribucija časovnih datotek	12
2.2.9 ZIC Prevajalnik	13
3 ORODJE PSM – Power System Manager.....	15
3.1 Uvod	15
3.2 Strežniška stran – iServer	15
3.2.1 Uvod.....	15
3.2.2 Funkcije.....	15
3.2.3 Nastavitve.....	15
3.2.4 Princip delovanja	16
3.2.5 Komunikacijski protokol.....	17
3.3 Stran odjemalca – PSM aplikacija.....	18
3.3.1 Uvod.....	18
3.3.2 Strategija izdaje PSM in IED programske opreme.....	18
3.4 Osnovne funkcionalnosti	19
3.4.1 Urejanje projekta in komunikacijska shema postaje	19
3.4.2 Struktura postaje	20
3.4.3 Komunikacijski model postaje	21
3.5 Nastavitve in diagnostika naprave.....	22
3.5.1 Parametracija funkcionalnosti naprave	22
3.5.2 Konfiguracija naprave.....	23
3.5.3 Komunikacijske nastavitve naprave	25
3.5.4 Lastnosti in diagnostika naprave	26
3.6 Online način dela z napravo.....	28
3.6.1 Online – Funkcionalne nastavitve	28
3.6.2 Online – Preslikava naprav	28
3.6.3 Online – Preslikava naprav	29
3.6.4 Online – Diagnostika naprave	29
3.7 Nivoji uporabnikov.....	29
3.8 Upload/download funkcionalnost	30
3.9 Predloga naprave	31
3.9.1 Dokument, ki definira predlogo (Device.xml)	31
3.10 Podatkovne strukture in dokumenti.....	31

3.10.1	Projektna podatkovna struktura.....	31
3.10.2	Objekti v podatkovni strukturi	32
3.10.3	Dokument z vsebino podatkovne strukture.....	34
4	GRAFIČNI VMESNIK ZA IZBIRO ČASOVNE DATOTEKE	36
4.1	Uvod v objekt Device_SystemSettings_TimeZone	36
4.2	Seznam spremenjenih oz dodanih datotek.....	37
4.3	Dodane lastnosti razredu Property_Field	37
4.4	Dodane lastnosti razredu PropertyInput	38
4.5	Dodane lastnosti razredu SubStaion.....	38
4.6	Dodan razred Device_SystemSettings_TimeZone.....	39
4.7	Dodan razred GuiNoTimeZoneFolder.....	40
4.8	Dodan razred GuiChooseTZ2	40
5	PRENOS IN POMNENJE ČASOVNE DATOTEKE.....	42
5.1	Pomnjenje časovne datoteke	42
5.2	Shranjevanje časovne datoteke na napravo.....	42
6	SKLEP	44
	KAZALO SLIK	45
	LITERATURA	46

SEZNAM UPORABLJENIH KRATIC IN SIMBOLOV

CEP	ang. Customized Embedded Platform; procesorska plošča za FPC 620 in CAU 360; tehnologija armv4l
FPC 620	ang. Feeder Protection Controller; modul vodenja in zaščite
CAU 360	ang. Control & Automation Unit; modul vodenja in nadzora distribucijskega omrežja
armv4l	ena od različic arm jedra
CID	dokument s komunikacijskimi nastavitvami naprave
DA	podatkovni atributi naprave
DAI	podatkovni atributi instance naprave
DO	podatkovni objekti naprave
DOI	podatkovni objekti instance naprave
DTBLK	podatkovni blok znotraj podatkovne baze RTDB
GMT	ang. Greenwich Mean Time : Greenwiški srednji časovni standard
GUI	grafični uporabniški vmesnik
IED	Intelligent Electronic Device (inteligentna elektronska naprava)
IEC61850	standard za komunikacijo v sistemu energetske podpostaje
ICD	XML dokument s predlogo naprave
LN	vozlišče naprave
NEO3000	platforma za izvedbo sistema zaščite, nadzora in vodenja v elektroenergetiki
OS	operacijski sistem
PSM	Power System Manager (upravljalca energetskega sistema)
RTDB	Real-Time DataBase (podatkovna baza v realnem času)
TAI	ang. International Atomic Time : Mednarodni atomski časovni standard
UTC	ang. Coordinated Universal Time : Univerzalni koordinirani časovni standard

POVZETEK

Večina sodobnih računalnikov, tako osebnih kot specializiranih, ima določeno uro v strojni ali programski opremi. Ura ali časovni žigi se uporabljajo skoraj pri vseh pojavih, dogodkih, napakah ali zapisih. V diplomskem delu predstavimo časovne datoteke, njihovo uporabo in implementacijo v sistemu PSM. Cilj naloge je implementirati dodatno funkcionalnost v že obstoječi program. Za lažje razumevanje tematike so v diplomskem delu predstavljene časovne datoteke, njihov nastanek, namen, sestava ter tudi opis sistema PSM. Za začetek smo opisali sestavo časovnih datotek, zoneinfo direktorija in PSM aplikacije ter ta znanja povezali v neko celoto, ki se je izrazila v obliki grafičnega vmesnika za izbiro časovne datoteke v aplikaciji PSM. Za izdelavo grafičnega vmesnika smo uporabili programski jezik Java in XML datoteke za opis nastavitev. Opisali smo tudi program zic, ki prevede izvorno kodo časovne datoteke v binarno ter jo uporabili v PSM sistemu. Rezultat dela je grafični vmesnik za izbiro časovne datoteke in prenos te na napravo. Zaradi tega bodo naprave družine NEO300 na različnih koncih sveta ustvarjale pravilne časovne žige.

Ključne besede:

Power System Manager, časovne datoteke, grafični vmesnik, zoneinfo direktorij, zic prevajalnik

ABSTRACT

Most modern computers personal or specialized, has a clock integrated in it's hardware or software. Time or time stamps are important for all events, errors or records that occur on computer. The thesis presents time zone files, their use and implementation in PSM system. The objective of a thesis is to implement the new functionality into the existing program. At the beginning we present time zone files, zoneinfo directory and PSM application. Finally, we connect all these knowledge together by producing graphical interface for time zone files in PSM application. Graphical user interface was developed in Java and all settings files have a XML structure. We also described program zic, which converts source code of time zone files to regular and correct time zone files and PSM system. The result is working graphical user interface for choosing time zone files and transfer of selected file to device. Therefore all the devices in NEO3000 family around the world will be able to produce correct time stamps.

Key words:

Power System Manager, timezone files, GUI, zoneinfo directory, zic compiler

1 UVOD

Večina sodobnih računalnikov, ne glede ali gre za osebne ali točno določene specializirane platforme, ima določeno uro s strani strojne ali programske opreme. Ura, oz. časovni žigi se uporabljajo skoraj za vse pojave, dogodke, napake ali zapise. Danes si ne znamo predstavljati računalnika, ki ne bi uporabljal ure. Za to je v sodobnih operacijskih sistemih poskrbljeno največkrat kar z NTP protokolom, tako da ima uporabnik čim manj dela z nastavitvami. Obstajajo pa tudi sistemi, ki nimajo možnosti dostopa do interneta ali javnih NTP strežnikov. Največkrat gre tu za specializirane sisteme ali pa za sisteme, ki imajo visoko prioriteto varnosti.

Prav ta problem se pojavlja tudi na napravah, ki se nahajajo v elektroenergetskih sistemih. Njihovi sistemi so večinoma zaprti in specifični. Takšen sistem imajo vsi večji proizvajalci elektroenergetskih naprav, med njimi tudi Iskra Sistemi d.d.. Ker se njihovi sistemi prodajajo po celem svetu, si ne morejo privoščiti, da bi naprave imele nastavljeno samo eno uro.

Za rešitev problema je bila predlagana rešitev, ki jo uporabljajo operacijski sistemi Unix. Tako smo razvili grafični uporabniški vmesnik za izbiro časovnih datotek in funkcionalnost za prenos le teh na določeno napravo.

Za začetek smo opisali strukturo časovne datoteke. Časovna datoteka je datoteka, ki se nahaja v `tz` podatkovni bazi in predstavlja čas v državi ali delu države. Obstajajo tudi datoteke, ki niso specifične za države, ampak jih večinoma uporabljajo pomorščaki. Takšne datoteke se večinoma nahajajo v `Etc` direktoriju.

Nadaljujemo z opisom izvornih datotek, predvsem njihovo sestavo in naštejemo kakšne tipe rezerviranih besed pozna prevajalnik `zic`. Opišemo tudi vrstice in njene attribute, ki jih prevajalnik zahteva za pravilno delovanje. Tu bi radi poudarili, da prevajalnik `zic` ni namenjen samo prevajanju časovnih datotek, ampak lahko z njim tudi zamenjamo časovni pas na Unix sistemih.

`Zoneinfo` direktorij je direktorij v katerem se privzeto nahajajo vse časovne datoteke sistema. Časovne datotek so v tem direktoriju shranjene po direktorijih, ki logično spadajo v neko zaključeno celoto. Izjema so časovne datoteke, ki se nahajajo na prvem nivoju in so tu predvsem zaradi kompatibilnosti s starejšimi verzijami podatkovne baze. `Zoneinfo` direktorij vsebuje tudi nekatere druge pomožne datoteke.

Omenimo tudi, kdo je človek, ki skrbi za posodobitev časovnih datotek in kdo skrbi za `zic` prevajalnik. Naveden je tudi naslov, kjer si lahko vsak uporabnik izbere najnovejše časovne datoteke. Na kratko opišemo program `zic` in njegovo uporabo.

V tretjem poglavju predstavimo orodje PSM in njegove značilnosti. Opis vsebuje vse od strategije izdaje do potrebnih datotek za pravilno delovanje sistema. V poglavju je tudi opis sistema `iServer`, ki služi kot strežnik PSM sistemu, ter mu nudi specifične naloge.

Naslednje poglavje je namenjeno opisu izdelave in funkcionalnosti grafičnega vmesnika za izbiro časovne datoteke. Za to poglavje pričakujemo, da bralec pozna vsaj osnove programiranja v Javi, saj smo dokaj podrobno opisali razrede in njihove metode, ki se uporabljajo za izris in funkcionalnost grafičnega dialoga za izbiro časovne datoteke.

Peto poglavje govori o zapisu izbrane časovne datoteke na želeno napravo in shranjevanju podatka o časovnem pasu, tako da ob naslednjem zagonu PSM prikaže pravilno vrednost. Vrednost se zapiše tudi v `neo3000.conf` datoteko, ki jo uporabi drugi program za svoje nastavitve.

2 OPIS ČASOVNIH DATOTEK IN ZONEINFO DIREKTORIJA

V tem poglavju smo zapisali definicijo časovnih datotek in njihov nastanek. Zapisana je tudi izvorna koda časovnih datotek in zic prevajalnik, ki pretvori izvorno kodo v binarni zapis razumljiv računalniku. Zic program shrani binarne datoteke v zoneinfo direktorij, ki je primarni direktorij Unix sistemov, za shranjevanje časovnih datotek. V zoneinfo direktoriju se nahajajo različne mape in datoteke (niso vse časovne datoteke), ki imajo vsaka svoj pomen in razlog, zakaj se nahajajo ravno na tem mestu.

2.1 Opis časovnih datotek

2.1.1 Definicija časovne datoteke

V t_z podatkovni bazi (tj. zbirka informacij o svetovnih časovnih conah) časovna datoteka vsebuje vsako državo ali regijo države, kjer so se prebivalci so strinjali glede lokalne ure od 1. januarja leta 1970 naprej. Ta definicija je precej drugačna od slednje, ki pravi, da je časovna datoteka konstanten odmik od meridijana (glavnega poldnevnika). Tu se pojavi predvsem problem poletnega časa, saj je mesto vedno na isti geografski lokaciji, odmik od meridijana pa se lahko poveča, oz. zmanjša [6].

Časovne datoteke imajo eksplicitna imena v obliki »območje/lokacija« z namenom, da bi jih ljudje lažje razumeli. Odločili so se tudi, da se uporabljajo večinoma angleška imena ter da datoteke nimajo pik in končnic. Območja so večinoma poimenovana po kontinentih ali oceanih, medtem ko lokacije predstavljajo predvsem večja mesta ali manjše otoke. Eden izmed razlogov, zakaj so se odločili za uporabo imen večjih mest in ne držav je, da so mesta veliko bolj robustna, država pa se lahko spreminja pogosto zaradi mej ali političnih ciljev. Med razlogi se najde tudi težnja, da bi bile časovne datoteke v isti mapi geografsko kompaktne.

2.1.2 Na splošno o časovnih datotekah

Skoraj vsi časovni pasovi na kopnem imajo definirane, oz. uzakonjene meje, ki pogosto sovpadajo z državnimi mejami ali so meje pripadajočega ozemlja države. Izmed 40 časovnih pasov na kopnem ima večina odmik od UTC enak celemu številu, seveda se najdejo tudi izjeme (Mjanmar, Iran, Indija). Tem 40-im časovnim pasovom pa je dodano še 25 morskih časovnih pasov [8]. Mejniki teh časovnih pasov so zemljepisne dolžine, ki se med seboj oddaljene za 15 stopinj. Teh 15 stopinj predstavlja ravno eno uro Zemljine rotacije relativno glede na Sonce. 25 morskih časovnih pasov nastane zato, ker je časovni pas, ki ima GMT odmik enak +/-12 ur razdeljen na 2 dela z mednarodno navtično črto, ki deli dan iz danes na jutri ali iz danes na včeraj. Navtična dnevna črta je legalna, oz. obstaja, ni pa nujno, da je narisana na vseh pomorskih kartah. Navtična dnevna črta načeloma sledi 180 stopinjskemu meridijanu. Odstopanja se lahko zgodijo, ko se ta linija nahaja v teritorialnih vodah, kjer je čas enak standardnemu času države.

Pred letom 1972 so bili vsi časovni pasovi predstavljeni kot odmik od GMT. Po tem letu pa se vse uradne, oz. državne ure sinhronizirajo preko radijskih valov. Po teh valovih se pošiljajo časovni signali, ki bazirajo na UTC časovnem standardu. Večina držav sedaj predstavi svoj

čas kot odmik od UTC standarda, čeprav nekaj držav še vedno predstavlja svoj čas kot odmik od GMT standarda. Med njimi je tudi Velika Britanija. UTC je standard, ki se uporablja povsod po svetu, kjer potrebujejo določiti časovni žig nedvoumno.

Originalno so vsa mesta merila čas s sončno uro [7]. Ob prihodu dobrih mehanskih ur v začetku 19. stoletja, je vsako mesto začelo uporabljati lokalni sončni čas. Prvi časovni pas je bil ustvarjen leta 1847 v Veliki Britaniji za železnice. Uporabljal je GMT standard. Sandford Fleming je leta 1879 predlagal standardizacijo časa. Do leta 1900 je bil čas že razdeljen po standardnih časovnih pasovih. Minilo je mnogo desetletij, preden je bil ves čas na Zemlji predstavljen kot odmik od GMT standarda, saj je večina držav uravnavala svoje ure glede na lokalne observatorije, ne meneč se za odklon od GMT standarda.

Čas merjen z lokalno sončno uro, je ob razširitvi železnice in vse večji uporabi telekomunikacij, postajal čedalje manj uporaben. Čas med 2 lokacijama se je razlikoval ponavadi za »čudno« število. Ta problem bi lahko rešili tako, da bi bila ura v vseh krajih enaka. Posledica takšne označbe časa so nepredvidljive, saj so ljudje vajeni, da čas predstavlja pozicijo sonca na nebu. Tako so časovni pasovi kompromis. Zrahljajo se geografske vezi med lokacijama in se hkrati pusti da lokacije ohranijo aproksimacijo lokalnega sončnega časa.

2.1.3 Izvorna koda časovnih datotek

Izvirne datoteke so vse pisane po pravilih. Vsi znaki, ki se pojavijo za lojtro (#), se ignorirajo, saj ta predstavlja vrstični komentar. Prav tako se ignorira vse neizpisljive znake (ang. white spaces), ki se pojavijo pred ali za vrstico, ki predstavlja uporabne podatke. Vrstice z uporabnimi podatki se delijo na 3 skupine:

- vrstice, ki predstavljajo pravilo,
- vrstice, ki generirajo časovno datoteko in
- vrstice, ki predstavljajo povezavo (ang. link).

Vrstica s pravilom ima naslednjo obliko [5]:

Rule	NAME	FROM	TO	TYPE	IN	ON	AT	SAVE	LETTER/S
npr:									
Rule	US	1967	1973	-	Apr	lastSun	2:00	1:00	D

- Besedica `Rule` definira, da bo v tej vrstici podano pravilo.
- Stolpec `NAME` predstavlja poljubno ime pravila
- Stolpec `FROM`: tu je podano prvo leto na katerega se nanaša pravilo. Tu se lahko vpiše katero koli celo število. Rezervirani besedi `minimum` oz. `maximum` ali njeni kratici predstavljata minimalno oz. maksimalno število, ki se še lahko predstavi kot integer. Pravila lahko opisujejo čase, ki niso predstavljeni kot vrednosti časa. V takšnih primerih se ta pravila ignorira. To omogoča, da so pravila prenosljiva med različnimi napravami, ki imajo različne časovne vrednosti.
- Stolpec `TO`: predstavlja zadnje leto, v katerem še velja to pravilo. Kot zgoraj se lahko tudi tu uporabijo rezervirani besedi `minimum` in `maximum`, dodano pa ji je še besedica `only`, ki ponovi vrednost iz polja `FROM`.
- Stolpec `TYPE`: predstavlja obdobje v letu, na katerega se nanaša pravilo. Če je tip enak pomišljaju, potem to pravilo velja skozi vsa leta (od leta podanega pod `FROM` do

tistega podanega pod TO stolpcem vključno z njim). Če tip ni enak -, potem program zic izvede komando *yearistype year type*, ki vrne 0, če je leto podanega tipa, 1 sicer.

- Stolpec IN: predstavlja ime meseca v katerem se bo pravilo izvedlo. Ime meseca je lahko predstavljeno kot kratica.
- Stolpec ON: predstavlja dan, v katerem bo pravilo začelo veljati. Tukaj se lahko nahajajo različne vrednosti: *število* – predstavlja dan v mesecu, *lastSun* – predstavlja zadnjo nedeljo v mesecu, *lastMon* – predstavlja zadnji ponedeljek v mesecu, *Sun>=8* – predstavlja prvo nedeljo, ki je osmega ali pozneje v mesecu, *Sun<=25* – predstavlja zadnjo nedeljo, ki je 25. ali pa prvo, ki je pred 25. Imena dni v tednu so lahko zapisana v celoti. Posebna pozornost velja stolpcu ON, saj med znaki ne sme biti neizpisljivih znakov,
- Stolpec AT: predstavlja čas, kdaj se pravilo zgodi. Prepoznane forme so: *število* – čas podan v urah, *2:00* – čas podan kot ure in minute, *15:00* – čas predstavljen v 24 urni notaciji in *1:28:14* – čas predstavljen kot ura, minute in sekunde. Število 0 predstavlja začetek dneva, medtem ko 24 predstavlja konec dneva. Katerikoli od teh form lahko sledijo črke *w* – če je čas podan kot lokalni čas »stenske ure«, *s* – če je čas podan kot lokalni standardni čas ali [*u*, *g*, *z*] – če je podani čas univerzalni. Če ni nobenega indikatorja se privzame lokalni čas »stenske ure«.
- Stolpec SAVE: poda čas, ki bo dodan lokalnemu standardnemu času. Forme so enakega formata kot v stolpcu AT, z izjemo, da se ne doda *w* in *s* pripono.
- Stolpec LETTER/S: predstavlja »variabilni del«, kjer S pomeni EST, D pa EDT, katerih časovni standard se uporabi, ko je pravilo veljavno. Če je polje enako -, potem je variabilni del enak null.

Vrstica, ki generira časovno datoteko:

Zone	NAME	GMTOFF	RULES/SAVE	FORMAT	[UNTIL]
npr:					
Zone	Australia/Adelaide	9:30	Aus CST	1971 Oct 31	2:00

- Besedica *Zone* pove prevajalniku, da bodo v tej vrstici in tudi nadaljnjih, vse dokler ne pride do naslednje rezervirane besede, podani podatki za izdelavo časovne datoteke.
- Stolpec *NAME*: predstavlja ime, ki ga bo vsebovala binarna časovna datoteka.
- Stolpec *GMTOFF*: predstavlja čas, ki se mora dodati UTC standardu, da prevajalnik pravilno ustvari standardno časovno datoteko. Oblika zapisa je enaka kot pri AT stolpcu pravila. Polje se lahko začne z minusom, če se časovni pas nahaja na desni strani meridijana.
- Stolpec *RULES/SAVE*: ima podano ime pravila, ki se uporabi za to časovno datoteko, ali čas, ki se mora dodati lokalnemu standardnemu času. Če podatka ni, se pravi da polje vsebuje vrednost, potem se standardni čas vedno uporabi v tem časovnem pasu.
- Stolpec *FORMAT*: predstavlja format okrajšave tega časovnega pasu. Par %s predstavlja, kje se lahko nahaja variabilni del okrajšave časovnega pasu. Poševnica (/) izmenično ločuje okrajšave za standardni oz poletni čas.

- Stolpec UNTIL: poda čas, kdaj se spremeni s poletnega na zimski čas s pomočjo pravila ali odmika od UTC standarda (stolpec RULES/SAVE). Zapis je podan kot *leto, mesec, dan in čas v dnevu*. Če je čas podan, se časovni pas ustvari iz podanega UTC odmika in pravila, vse dokler tako določa čas v tem stolpcu. Stolpec ima enak format kot AT stolpec iz pravil. Podatki v tem stolpcu niso nujno potrebni. V primeru da podatkov ni, se vzame najmanjši možen čas. Ker UNTIL stolpec določa čas samo za določeno časovno obdobje, je potrebno, ob podanem tem stolpcu, v naslednji vrstici podati tudi pravila za nadaljnja leta. V tako imenovanih »naslednjih vrsticah«, se namesto rezervirane besede *Zone* in imena časovnega pasu vstavi 2 neizpisljiva znaka, ostala pravila stolpcev ostanejo nespremenjena.

Vrstica, ki predstavlja povezavo:

Link	LINK-FROM	LINK-TO
npr:		
Link	Europe/Istanbul	Asia/Istanbul

Vrednost stolpca LINK-FROM se mora pojaviti nekje v NAME stolpcu Zone vrstice. LINK-TO pa se uporabi kot alternativno ime za to časovno datoteko.

Vsa našeta pravila, oz. vrstice se lahko nahajajo v datoteki v naključnem vrstnem redu. Samo »naslednje vrstice« (to so vrstice, zapisane pod pravilom Zone, ki ga nadaljujejo) se morajo držati skupaj.

Obstajajo pa tudi še vrstice, ki opisujejo prestopne sekunde in se morajo nahajati v svojih datotekah. Slednje vrstice imajo naslednjo strukturo:

Leap	YEAR	MONTH	DAY	HH:MM:SS	CORR	R/S
npr:						
Leap	1974	Dec	31	23:59:60	+	S

Polja YEAR MONTH DAY HH:MM:SS povedo, kdaj se je zgodila prestopna sekunda. Polje CORR doda sekundo, če je + ali odšteje sekundo, če je -. Polje R/S vsebuje 2 črki. R, če se prestopna sekunda podana v drugih poljih interpretira kot lokalna ura in S, če se prestopna sekunda interpretira kot UTC.

Primer takšne datoteke je tudi v podatkovnem paketu izvirne kode časovnih datotek in se imenuje *leapseconds*. Če se ob prevajanju programu zic doda stikalo *-L* in argument pot do datoteke *leapseconds*, so ustvarjene binarne datoteke enake tistim, ki se nahajajo v right direktoriju.

2.2 Sestava zoneinfo direktorija

2.2.1 Zoneinfo direktorij

Zoneinfo mapa se na Unix sistemih nahaja na poti `/usr/share/zoneinfo`. V tej mapi so zapisane binarne časovne datoteke za nastavitve časovnega pasu računalniku. Mapa vsebuje večje število map, ki predstavljajo zaključeno celoto časovnih datotek, ki po geografski legi ali drugih značilnostih sodijo v neko celoto. Vsebuje pa tudi nekaj drugih časovnih datotek in link `localtime`, ki dejansko kažejo na izbrano časovno datoteko, ki jo uporablja sistem. Mapa vsebuje tudi nekatere druge datoteke (`posixrules`, `iso1366.tab`, `zone.tab`). V splošnem lahko vsebuje tudi uporabniško generirane časovne datoteke ali katere koli druge datoteke.

Značilnost časovnih datotek, ki so grupirane pod celine je, da ne nosijo imen po državah, ampak po njenih glavnih mestih ali večjih mestih.

2.2.2 Časovne datoteke razdeljene po celinah

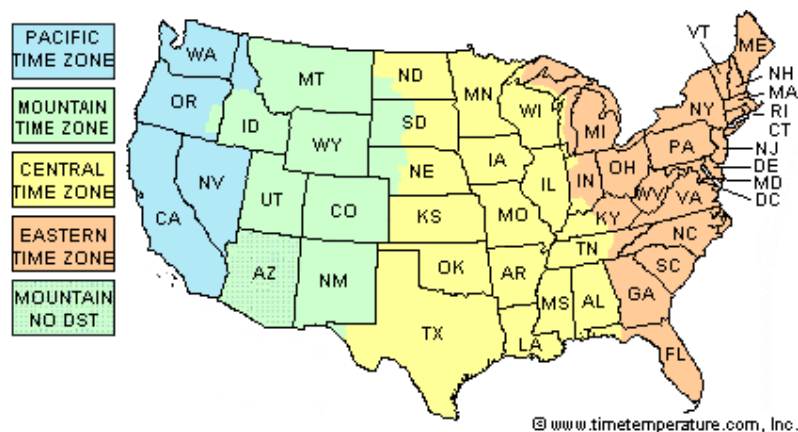
Mape, ki so poimenovane po celinah, ne potrebujejo posebne razlage, saj vsakdo, ki prebere ime mape ve, da v mapi *America* ne bo iskal časovne datoteke za Slovenijo. Čeprav tudi tu obstajajo izjeme. Ena izmed njih je *Ciper*, ki ga najdemo v mapi *Asia* in tudi pod *Evropa*. Razlog za to je zgodovinski, saj naj bi po Herodotovem pričanju *Ciper* pripadal Aziji (Herodotus, *Histories*, I.72.) [3]. Ker pa je država *Ciper* članica Evropske Unije, nas večina pričakuje, da bomo našli *Ciper* v mapi *Europe*.

V mapi *Indian* pa se nahajajo vsi majhni in veliki otoki v indijskem oceanu. Prav tako mapa *Pacific* združuje vse otoke v Tihem oceanu.

Mapa *US* vsebuje vse večje časovne pasove, ki se nahajajo na ozemlju ZDA. Te so lahko tako na Severno Ameriški celini (kar prikazuje Slika 1), ali pa na otoku Havaji. V to skupino spadajo časovne datoteke, ki razdelijo kontinentalno ZDA. Zanimivost te skupine je, da vsebujejo najbolj skrajne dele ameriškega ozemlja [9]. Tako v tej mapi najdemo Aleutsko otočje, ki je najbolj zahodna točka ZDA. Najbolj severna točka ZDA je Aljaska, najbolj južna točka je Ameriška Samoa (otok sredi južnega Pacifika). Michigan je edina Ameriška država, ki ima 2 polotoka. Arizona se nahaja v mapi zato, ker je največja ameriška država po številu prebivalcev, ki ne leži ob obali. Havaji pa so najmlajša država v ZDA in hkrati edina otoška država.

2.2.3 Datoteka s pravilom za prehod iz poletnega v zimski čas

`Posixrules` datoteka je kopija časovne datoteke *America/New_York*. `Posixrules` naj bi se uporabljala za časovne datoteke, ki so zapisane v POSIX stilu. Pomembno pa je, da se s pomočjo te datoteke lahko izračuna točen čas v določenem časovnem pasu, se pravi da so vštete vse preskočne sekunde do tistega časa.



Slika 1: Časovni pasovi v celinski ZDA

Zakaj se uporablja, bolje rečeno, se je uporabljala datoteka `posixrules`? Za to se je potrebno vrniti v preteklost in pogledati opis pomoči programa `tzset` [10]. Program `tzset` inicializira informacije za pretvorbo časa, ki ga uporablja program `localtime`. Okoljska spremenljivka `TZ` specificira, kako se ta sprememba izvrši. Če se za konverzijo uporabi okoljska spremenljivka `TZ`, mora imeti naslednji format zapisa:

`std offset [dst [offset] [,rule]]`, kjer nas najbolj zanima spremenljivka `rule`, ki pove kdaj se zgodi prehod iz standardnega časa v poletni in obratno. V primeru, da pravila `rule` ni, se za prehod iz zimskega na poletni čas uporabi pravilo, ki je zapisano v datoteki `posixrules`. Tako je bilo v starejših različicah Unix sistemov. V novejših različicah je potreba po `posixrules` datoteki odpadla, saj ta datoteka večinoma kaže, ali pa je kopija `localtime` datoteke.

V mapi `Mideast` se nahajajo tri časovne datoteke in vse se nanašajo na Savdsko Arabijo med leti 87 do 89. Zanimivost teh datotek je, da nimajo zapisanega dne, kdaj se preide v poletni čas, ampak imajo za vsak dan v letu zapisano, koliko je ura in kdaj se je spremenila. Čas je izračunan s pomočjo formule, priskrbljene s strani U. S. Naval Observatory-ja. Formula

$$eqt = -105.8 * \sin(l) + 596.2 * \sin(2 * l) + 4.4 * \sin(3 * l) \\ -12.7 * \sin(4 * l) - 429.0 * \cos(l) - 2.1 * \\ \cos(2 * l) + 19.3 * \cos(3 * l);$$

kjer l predstavlja »povprečno dolžino sonca« izračunano po

$$l = 279.642 \text{ stopinj} + 0.985647 * d$$

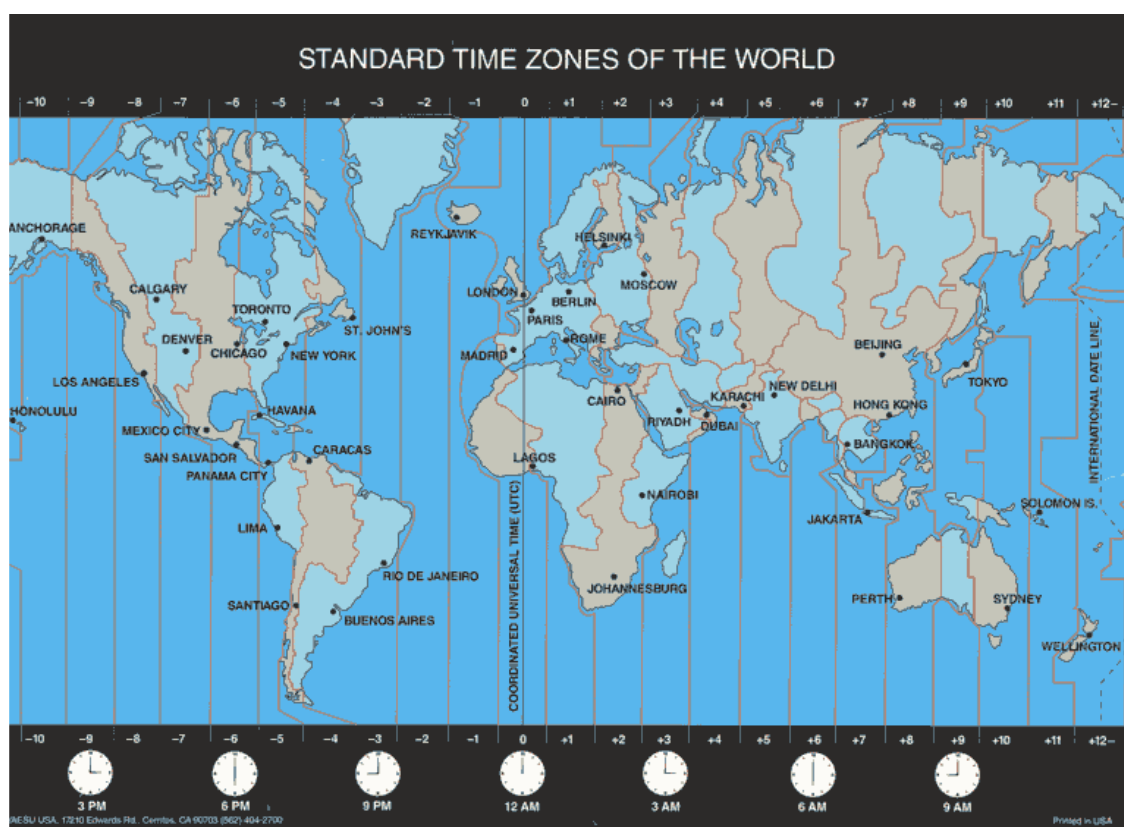
in d predstavlja dan v letu od 1. januarja naprej. Formula ima napako samo +/- 3 sekunde. Ker pa računalnik ne more shraniti točnega zapisa vseh decimal, je zapisanih samo 256 različnih časov. To je limit, ki je dosežen, ker je čas v računalniku zapisan kot nepredznačen znak(ang. unsigned char). Zato so te tri datotek tudi skoraj 2x večje kot ostale časovne datoteke.

Mapa `SystemV` vsebuje stare časovne datoteke, če se slučajno kdaj v prihodnosti pojavi potreba po njihovi ponovni uporabi.

2.2.4 Direktorij z GMT datotekami

V direktoriju `etc` se nahajajo časovne datoteke z imeni GMT+/-neko število, Universal, UTC, UCT, Zulu, Greenwich in še GMT0 in GMT, ki nosita enako informacijo kot GMT+0 in GMT-0. Te datoteke so večinoma prisotne samo zaradi zgodovinskih razlogov. Tako da lahko ljudje, ki ne živijo v območjih, ki jih pokrivajo ostale časovne datoteke, nastavijo ustrezen časovni pas. V današnjem času pa imajo te datoteke uporabno vrednost samo za pomorščake.

Zanimivost GMT datotek pa je tudi, da imajo imena datotek ravno nasprotni predznak odmika od GMT, kar prikazuje Slika 2. Za to je kriva POSIX notacija, ki ima pozitivne znake zahodno od Greenwicha. To pa je za mnogo ljudi sporno, saj pričakujejo, da bodo imele časovne datoteke zahodno od Greenwicha negativen predznak, saj zaostajajo za njim. Na primer časovna datoteka GMT+4 predstavlja čas, ki je 4 ure za UTC, čeprav jih veliko pričakuje, da bo ta datoteka 4 ure pred UTC.



Slika 2: Prikaz Posix notacije

2.2.5 Direktorij Posix

Posix direktorij je po svoji zgradbi kopija zoneinfo direktorija. Vsebuje vse časovne datoteke razen `map posix` in `right`, ter specifičnih datotek `iso3166.tab`, `link localtime`, `posixrules`, `zone.tab`. Časovne datoteke v Posix direktoriju bazirajo na Coordinated Universal Time (UTC). Primerjava istoležnih datotek v `posix mapi` in datotek v `zoneinfo mapi` razkrije, da so vse datoteke dejansko enake.

UTC [11] ne pozna poletnega časa in ima skozi celo leto enak odmik od GMT. UTC izhaja iz TAI (International Atomic Time), njuna odvisnost je povezana z linearno funkcijo. V UTC lahko občasno pride do nezveznosti (diskontinuitete). Vzrok za to je sprememba iz ene v drugo linearno funkcijo TAI-a. Nezveznosti se poznajo samo ob koncu meseca po Gregorijanskem koledarju. UTC je tudi močno vezan na UT1 standard (UT1 je glavna oblika Universal Time in je enaka na vseh krajih na Zemlji. UT1 je proporcionalen glede na rotacijo zemlje in njene oddaljenosti od ostalih kvazarjev, tj. ostalih zvezdi podobnih izvirov sevanja), saj sme od njega odstopati za manj kot 0.9 sekunde. UTC pa ima še slabo lastnost, saj mu zaradi svoje nezveznosti ni mogoče izračunati točnega časa med 2 UTC časovnima žigoma brez pomožne tabele, kjer so zapisane vse prestopne sekunde, ki so se pojavile med časovnima žigoma.

2.2.6 Direktorij Right

Right direktorij ima enako zgradbo kot posix direktorij. Razlika je v tem, da right direktorij temelji na International Atomic Time (TAI), časovne datotek pa vsebujejo vse sekunde v letu. Tako čas izračunan iz časovnih datotek v right mapi ne vsebuje prestopnega dneva.

TAI standard [12] se uporablja za sisteme, ki ne znajo uporabljati prestopnih sekund. Je zelo natančen atomsko koordinatni časovni standard, ki temelji na fiktivnem prehodu časa prek Zemlje prikazane kot geoid (geoid je ekvipotencialna ploskev, ki približno sovпада s srednjo gladino oceanov. Največkrat jo poimenujejo »zadosti dober« prikaz fizičnega modela Zemlje). TAI je izračunan kot tehtano, oz. uteženo povprečje časov 200 atomskih ur, ki se nahajajo v 70 nacionalnih laboratorijih po svetu. Zaradi povprečja je čas veliko bolj natančen. Kot že omenjeno, je TAI osnova za UTC ter tudi Terrestrial Time, ki se uporablja za izračune v astronomiji. V začetku leta 1958 je bil sinhroniziran z UT, nato pa sta se počasi začela razhajati zaradi spreminjajočega gibanja Zemlje. Kot zanimivost naj povemo, da je bil 1. januarja 2011 ob 00:00:00 po UTC, TAI natančno 34 sekund pred UTC.

2.2.7 Ostale časovne datoteke

Nekatere časovne datoteke pa se ne nahajajo v mapi, ampak se jih najde takoj pod glavnim direktorijem. GB, GB-Eire, GMT, GMT+0, GMT-0, GMT0, NZ, NZ-CHAT, PRC, ROC, ROK, UCT, UTC časovne datoteke se nahajajo tukaj, ker so se skozi čas njihova imena spremenila. Tako so te časovne datoteke kot nekakšen most med starimi in novimi. Veliko imen se je spremenilo leta 1993.

Datoteke CET, CST6CDT, EET, EST, EST5EDT, HST, MET, MST, MST7MDT, PST8PDT, WET pa se nahajajo zaradi kompatibilnosti s starejšimi sistemi. V prejšnjih različicah časovnega paketa, so te datoteke bile legalne, sedaj pa so jih nadomestile pravilnejše, oz. popravljene različice. Te datoteke so dejansko časovne datoteke in ne povezave, ki bi kazale na novejšje datoteke, zato ker se datoteke spreminjajo ali države začnejo uporabljati poletni čas.

2.2.8 Posodobitev in distribucija časovnih datotek

Vse informacije o časovnih datotekah so spravljene v tako imenovani t z podatkovni bazi (tj. zbirka informacij o svetovnih časovnih conah) ali zoneinfo podatkovni bazi. To je zbirka

izvornih datotek, ki se s pomočjo zic prevajalnika prevedejo v binarne datoteke, katere se nahajajo v zoneinfo direktoriju. Prvotno je bila ta zbirka namenjena za uporabo v programih in operacijskih sistemih, nato pa je prerasla te okvirje, verjetno tudi zaradi vse širše uporabe računalniških sistemov. Zbirko se včasih poimenuje tudi Olsonova zbirka po Arthur David Olsonu, ki je bil njen ustanovitelj in pobudnik. Kot je omenjeno že zgoraj, se imena datotek skladajo s celino na kateri se nahajajo in z glavnim ali večjim mestom. Za to je zaslužen Paul Eggert, ki popravlja in posodablja zbirko izvornih datotek. Prizadevajo si, da bi zbirka vsebovala trenutne čase po svetu, kot tudi vse prejšnje standarde, ki so se pojavili od leta 1970 naprej. To leto je tudi leto začetka štetja časa na računalniških sistemih.

Da bi lahko 2 človeka urejala celotno zbirko časovnih datotek in hkrati še brskala za novostmi, ki se dogajajo po svetu, je skoraj preveč za pričakovati. Tako jim lahko vsakdo pomaga po svojih najboljših močeh in pošlje svoje verodostojne podatke, oz. predloge na tz poštni seznam (ang. mailing list).

Izvirne datoteke so navadne tekstovne datoteke in so prosto berljive, medtem ko so časovne datoteke binarne.

Navadno se časovne datoteke spremenijo nekajkrat na leto. Za posodobitev velja, da na Unix sistemih za to poskrbi Upravljalnik posodobitev. Za vse, ki imajo to možnost izklopljeno, pa lahko najdejo izvirne datoteke časovnih datotek na naslovu <http://elsie.nci.nih.gov/pub/>, kjer si prenesejo tzdata2011g.tar.gz stisnjeno datoteko, nato pa jih s pomočjo zic prevajalnika prevedejo v binarne datoteke. Če nimamo prevajalnika, si ga prav tako lahko potegnemo iz zgoraj omenjene povezave.

2.2.9 ZIC Prevajalnik

Zic [5] program pretvori tekst, ki je zapisan v izvorni datoteki, ki je podana kot vhodni argument programa v binarno datoteko. Če ima namesto datoteke podan vezaj (-), program pričakuje podatke iz standardnega vhoda.

Zic program ima lahko podanih več argumentov:

- `--version` izpiše verzijo programa in zaključi z delovanjem,
- `-d <direktorij>` ustvari časovne datoteke v podani mapi namesto v standardni (/usr/share/zoneinfo),
- `-l <časovna datoteka>` program bo uporabil podano časovno datoteko in naredil link na /etc/localtime. S tem se nastavi časovni pas na računalniku,
- `-p <časovni standard>` program bo uporabil podano časovno datoteko in naredil kopijo na /usr/share/zoneinfo/posixrules,
- `-L <datoteka s prestopnimi sekundami>` prebere podatke o prestopnih sekundah iz podane datoteke in jih zapiše v izhodne binarne datoteke. Na takšen način je ustvarjena mapa right,
- `-v` program se pritoži, če je leto prebrano iz datoteke izven obsega let, ki so podane z time(2) vrednostmi. V praksi to pomeni, da se izpišejo opozorila pri generiranju časovnih datotek,
- `-s` omeji časovne vrednosti shranjene v časovnih datotekah na tiste, ki so enake ne glede ali je to predznačeno ali nepredznačeno število,

- `-y <command>` uporabi podan ukaz za pregled let, namesto, da bi uporabil `yearistype` ukaz.

Vhodna datoteka mora biti napisana po pravilih, da jo lahko program prebere in prevede.

3 ORODJE PSM – Power System Manager

3.1 Uvod

Aplikacija PSM je na strani odjemalca, kjer na uporabniku prijazen način prikažemo vse funkcionalnosti za parametriranje postaje in njihovih naprav, kot tudi komunikacije na nivoju posameznih naprav in na nivoju celotne postaje.

V tem poglavju predstavimo orodje PSM in njegove značilnosti. Opis vsebuje vse od strategije izdaje do potrebnih datotek za pravilno delovanje sistema. V poglavju je tudi opis sistema iServer, ki služi kot strežnik PSM sistemu, ter mu nudi specifične naloge.

iServer je v tem sistemu strežnik, ki omogoča PSM aplikaciji nadzor in upravljanje NEO3000 naprav. NEO3000 je platforma za izvedbo sistema zaščite, nadzora in vodenja v elektroenergetiki.

3.2 Strežniška stran – iServer

3.2.1 Uvod

iServer [2] je strežniška aplikacija, ki se izvaja na napravah NEO3000 in omogoča uporabniškim aplikacijam (Power System Manager in drugim) upravljanje in nadzor naprave. Na začetku se je iServer razvijal v Javi kot java servlet, kasneje pa se je razvoj nadaljeval v programskem jeziku C++, ohranil pa se je princip komuniciranja po TCP/IP protokolu, ter princip prenosa vsebine zahtevkov (request) in odgovorov (response) po HTTP protokolu. Zaradi tega se lahko iServer uporablja tudi kot enostaven HTTP strežnik.

3.2.2 Funkcije

- odzivanje uporabniškim aplikacijam, ki iščejo naprave v omrežju,
- dostop do diagnostičnih podatkov na napravi,
- dostop do datotečnega sistema naprave, spreminjanje strukture le-tega, prenos datotek z in na napravo,
- inštalacija programskih modulov na napravo,
- delo z XML dokumenti,
- zajem živih podatkov iz posameznih programskih modulov naprave, dostop do njihovega konzolnega izpisa ter nastavljanje delovanja le-teh,
- izvajanje sistemskih ukazov operacijskega sistema naprave.

3.2.3 Nastavitev

Nastavitev delovanja aplikacije se nahaja v datoteki `iServer.xml` na datotečnem sistemu naprave na lokacijah `/usr/local/NEO3000/XML/iServer.functionality` (Linux) ali `C:/NEO3000/XML/iServer.functionality` (Windows) ali pa na poljubni lokaciji, ki jo določimo z vhodnim parametrom `-c`. Nastavitvena datoteka mora obstajati, drugače se aplikacija ne zažene. Če vsebino nastavitvene datoteke spremenimo ali

pa prepišemo obstoječo datoteko, iServer to pravočasno zazna in upošteva nove nastavitve (ni potreben ponovni zagon). Nastavitveno datoteko je priporočljivo urejati z iClient-om, ker se pri spreminjanju in dodajanju parametrov upoštevajo pravila, ki so navedena v pripadajoči XML shemi.

3.2.4 Princip delovanja

Izvajalni cikel aplikacije se deli na tri faze:

- inicializacija: če ta ni uspešna se izvajanje aplikacije prekine. Izvaja se v naslednjem vrstnem redu:
 - branje nastavitvene datoteke,
 - inicializacija SSL za kriptiran prenos vsebine med aplikacijo in odjemalcem,
 - inicializacija TCP vtičnika (ang. socket), na katerega odjemalci pošiljajo svoje zahteve (ang. requests),
 - inicializacija niti, ki se izvajajo vzporedno z glavno zanko:
 - `BroadcastListener`: prisluškuje klicem v broadcast omrežju, s katerimi klienti iščejo naprave v omrežju ter jim pošlje odgovor s katerimi se na kratko opiše (TCP port, ime naprave, verzija...).
 - `MulticastListener`: prisluškuje klicem v multicast omrežju, s katerimi klienti iščejo naprave v omrežju ter jim pošlje odgovor s katerimi se na kratko opiše (TCP port, ime naprave, verzija...).
 - `SystemStatus`: namenjen periodičnemu (20s) zajemanju diagnostični podatkov o delovanju iServer-ja in naprave.
 - `UserStatus`: spremlja aktivnost uporabnikov. V primeru, da klient ne pošilja periodičnega potrjevanja seje, se smatra, da je prišlo do prekinitve seje. V primeru, da v dalšem časovnem obdobju ni prišlo do nobene druge zahteve kot le periodičnega potrjevanja seje, se smatra, da je uporabnik neaktiven. V primeru da ni aktiven noben uporabnik, se datotečni sistem nastavi v bralni način, če je to potrebno.
 - inicializacija `SignalHandler`-ja: ki prestreza signale za prekinitev izvajanja aplikacije.
- sprejemanje in obdelava zahtev odjemalcev: ta del se izvaja, dokler aplikacije ne prekinemo. Vzporedno se izvajajo vse zgoraj naštetje niti:
 - ko se na TCP vtičniku pojavi odjemalčeva zahteva po vzpostavitvi povezave, se kreira nov vtič, preko katerega se bo komunikacija z odjemalcem nadaljevala. Potem se kreira nova nit, v kateri se bo obdelala zahteva odjemalca. Glavna zanka aplikacije nadaljuje s čakanjem na nove zahteve klientov. Na ta način se lahko vzporedno obdeluje več zahtev enega ali večih klientov;
 - nit, ki obdeluje zahtevo klienta, najprej prebere vsebino zahteve. Vsebina mora ustrezati standardu HTTP 1.1;
 - iServer razbere, če je zahteva namenjen njemu, kakšnega tipa je in katere parametre vsebuje. Sledi del, ki vsebuje podatke o samem klientu, na koncu pa je pripeta vsebina. Ta ni obvezna in jo vsebujejo le nekateri tipi zahtev, kot na primer zahteva po nalaganju datoteke na napravo;
 - glede na tip zahteve iServer ustrezno odreagira. Če je tip napačen to odjemalcu sporoči s sporočilom o napaki. Če klient zahteva vsebino datoteke z datotečnega sistema, se prebere vsebina datoteke in pošlje njeno vsebino. V primeru, da zahteva vsebuje parameter "kaj", ki se navezuje na ustrezno akcijo

ali poizvedbo, iServer preveri ali so ostali parametri ustrezni. V primeru da niso, to odjemalcu sporoči z obvestilom o napaki, drugače pa akcijo ali poizvedbo izvede, odjemalcu pa vrne pridobljeno vsebino ali pa sporočilo o napaki/uspešni izvedbi. Sporočilo, ki ga iServer pošlje odjemalcu kot odgovor, ravno tako ustreza standardu HTTP 1.1. Odjemalčeva aplikacija mora prepoznati sporočila o napakah ali uspešno izvedeni akciji, ki jih vrača iServer, ker te ne nosijo vsebine napake, temveč so predefinirani nizi znakov.

- Če odjemalec pošlje zahtevo po vsebini datoteke z datotečnega sistema, mu ni potrebno predhodno vzpostaviti seje z iServer-jem, za izvajanje akcij in poizvedb pa je to potrebno. Seja se vzpostavi s parametrom `kaj=connect`, ki mora vsebovati še uporabniško ime, geslo in verzijo iServerCommunication protokola. Odjemalcu se vrne uporabniški `id`, katerega mora vključiti kot parameter pri vsaki naslednji zahtevi.
- Uporabniške aplikacije morajo periodično pošiljati zahtevo s parametrom `kaj=connected` ali `kaj=connectedXML` s katerim vzdržujejo trenutno sejo z iServer-jem, kot rezultat pa jim ta vrača nekatere osnovne podatke o delovanju (število uporabnikov, stanje datotečnega sistema...).
- seja se konča z zahtevo, ki vsebuje parameter `kaj=disconnect`.
- prekinitev: se izvede, ko aplikacija prejme ustrezen signal po prekinitvi (SIGKILL / SIGTERM). Faze potekajo v obratnem vrstnem redu kot pri inicializaciji:
 - ustavitev SignalHandlerja,
 - ustavitev vseh niti, ki se izvajajo vzporedno z glavno zanko,
 - deinicializacija TCP socketa,
 - deinicializacija SSL,
 - nastavitev datotečnega sistema v bralni način, če je to potrebno.

3.2.5 Komunikacijski protokol

iServer Communication je komunikacijski protokol v sistemu SPM (iServer <-> odjemalec) za prenos podatkov in izvajanje komand. Podatki se prenašajo preko TCP/IP protokola in morajo sintaktično ustrezati HTTP komunikacijskem standardu.

Komunikacija med iServer-jem in odjemalcem izgleda v grobem takole: iServer sprejme zahtevo odjemalca, iz parametra "kaj" razbere, za kakšen tip komande gre. Če zahteva vsebuje vse obvezne parametre in so izpolnjeni tudi drugi pogoji (vzpostavljena seja, pisalni način datotečnega sistema...), se komanda izvede, drugače pa se odjemalcu vrne sporočilo o napaki (npr. `iServerErrorParameterMissing`, `iServerErrorNotWritable`...). Po uspešno izvedeni komandi klient dobi sporočilo z zahtevano vsebino ali le potrditev, da se je komanda uspešno izvedla (`iServerOK`). V primeru napake klient-u vrne sporočilo o tipu napake (`iServerErrorSCUUpdateFailed`...), če pa tip napake ni znan, pa le `iServerErrorUnknown`.

3.3 Stran odjemalca – PSM aplikacija

3.3.1 Uvod

Na NEO3000 platformi se z uvedbo podpore standarda IEC 61850, pojavlja potreba po parametraciji komunikacije med napravami v sistemu [1]. S tem orodjem želimo uporabniku omogočiti enostaven način parametracije posameznih funkcionalnosti različnih tipov naprav, kot tudi komunikacije na nivoju posameznih naprav in na nivoju celotne postaje.

Orodje bo združevalo več funkcionalnosti, ki bodo uporabniku omogočale:

- izdelavo hierarhičnega modela postaje v več nivojih: postaja, napetostni nivo, celica in naprava na najnižjem nivoju,
- parametracijo posameznih sklopov funkcionalnosti naprave,
- konfiguracijo naprave s povezovanjem posameznih modulov naprave,
- definicijo komunikacijskih objektov in navezava le-teh na ustrezne module naprave,
- pregled lastnosti naprave in prikaz diagnostičnih podatkov,
- izdelavo in urejanje komunikacijskega modela na nivoju naprav v sistemu,
- opazovanje procesnih spremenljivk v realnem času,
- arhiviranje in restavriranje naprave.

Aplikacija je razvita v programskem jeziku Java, na platformi Java 1.6. Razlogi za uporabo Java platforme so: podprtost na različnih platformah, zanesljivost in fleksibilnost, velik nabor (brezplačnih) razvojnih orodij, znanje in izkušnje iz preteklosti.

Za hranjenje podatkovne strukture projekta, nastavitvev naprave, parametracijo po IEC61850... se bodo uporabljali XML dokumenti. Podatkovna struktura nekaterih (ne vseh) dokumentov bo definirana z XML shemo.

Pri razvoju aplikacije je bilo, poleg osnovne funkcionalnosti, potrebno upoštevati naslednje zahteve naročnika:

- poleg dela z napravami v sistemu, mora biti omogočeno tudi delo s posamezno napravo
- zagotoviti podporo različnih tipov naprav, preko predlog, in možnost dodajanja novih tipov naprav
- opcija, s katero uporabnik v projekt doda obstoječo napravo, ki je prisotna v omrežju, kot tudi možnost izdelave lastnega profila naprave,
- delo z napravami v offline in online načinu,
- podpora različnim tipom uporabnikov (inženir, zaščitni inženir),
- podpora večjezičnosti,
- enostavna distribucija PSM aplikacije preko namestitvenega paketa.

3.3.2 Strategija izdaje PSM in IED programske opreme

PSM aplikacija se distribuira preko namestitvenega paketa, ki vsebuje vse potrebne datoteke za delovanje na Windows in/ali Linux platformi – lahko dva tipa instalacij.

V namestitvi PSM so že vključene tudi določene verzije Device\Templates.

Po namestitvi PSM ali kadarkoli naknadno mora biti omogočen uvoz enega ali več paketov Device\Templates, s katerimi bo PSM aplikacija urejala projekte.

Posamezna verzija Device\Template bo imela tudi preverjanje ali sta Template in PSM kompatibilna (npr.: zapis v `device.xml` element `PSM_version`, ki mora biti nižja ali enaka različici aplikacije PSM).

Strategija izdaje IED različic programske opreme je razdeljena na vsaj dva sklopa:

- namestitev OS, ostalih aplikacij (in strojna oprema(ang.firmware)) (`OS_release`),
- namestitev NEO 3000 aplikacije in strojne opreme (`App_release`),
- namestitev celotne naprave (`Device_release=OS_release + App_release`).

Poleg IED programske opreme je potrebno posodobiti/izdelati tudi pripadajoče Device\Templates.

`App_release` vsebuje verzijo v `software_info.xml`. Sintaksa je `version.subversion.revision` (npr.: 1.4.8).

Razlaga:

- `version` – glavna posodobitev programske opreme,
- `subversion` – različne manjše posodobitve – spremenjena shema, ...,
- `revision` – odprava napak, ki ne vplivajo na spremembo parametrov.

Pri posodobitvi IED-ja s programom PSM je potrebno preveriti naslednje elemente:

- družino, v katero spada naprava (ang. Family), ki se nahaja v `device_info.xml` datoteki,
- tip naprave (ang. Type), ki je zapisan v `device_info.xml`,
- izdana različica (ang. Release_version), ki je zapisana v `software_info.xml`.

V primeru neujemanja Family in Type elementov med PSM projektom in IED se na PSM pojavi pogovorno okno z opozorilom.

V primeru, da se ne ujema `Release_version` v `version` in `subversion` delu, se na PSM pojavi pogovorno okno z opozorilom o neujemanju verzij konfiguracije, v primeru neujemanja `revision` dela pa ne opozarjamo ničesar.

3.4 Osnovne funkcionalnosti

3.4.1 Urejanje projekta in komunikacijska shema postaje

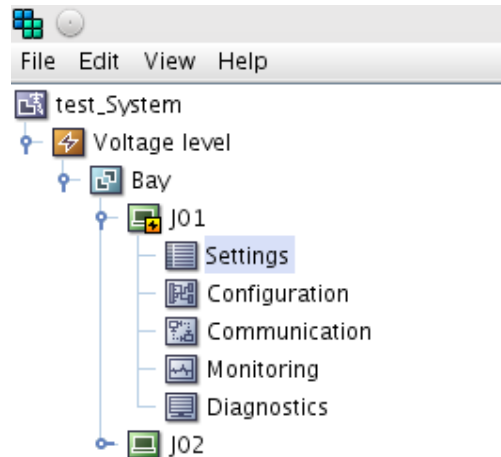
Aplikacija omogoča izdelavo novega projekta in urejanje obstoječega projekta. Projekt določa strukturo energetske postaje, ki je sestavljena iz naslednjih tipov objektov: napetostni nivo, celica in naprava.

Poleg tega aplikacija omogoča tudi izdelavo komunikacijske sheme postaje, kjer uporabnik definira način komunikacije med posameznimi napravami, ki so prisotne v sistemu.

Projekt se nahaja v lastnem direktoriju na lokalnem datotečnem sistemu, vsebina in struktura projekta se shranjuje v XML dokument z imenom `Project.xml`.

3.4.2 Struktura postaje

Postaja ima lahko definirano poljubno število napetostnih nivojev, ki se med seboj razlikujejo po oznaki (npr. 20kV, 110kV). Napetostni nivo lahko vsebuje poljubno število celic, ki morajo imeti določeno unikatno ime znotraj napetostnega nivoja (npr. E01, E02). Posamezna celica lahko vsebuje eno ali več naprav enakega ali različnega tipa.



Slika 3: Struktura postaje

Struktura postaje je predstavljena z drevesno komponento (glej Sliko 3), vsak objekt v drevesu je predstavljen z lastnim elementom. Uporabnik ima možnost dodajanja, premikanja in odstranjevanja posameznih objektov v projektu. Pravila za poimenovanje posameznih objektov niso določena. Vsak objekt, ki se doda v projektno strukturo, ima na datotečnem sistemu, v projektnem direktoriju, tudi lasten direktorij. Struktura v projektnem direktoriju ustreza strukturi objektov v projektu, zaenkrat je predvidena dodatna vsebina v obliki datotek/dokumentov, le na najnižjem nivoju – napravi, kjer so bodo nahajali dokumenti, ki so lastni napravi (`Settings.xml...`).

Napravo, ki predstavlja najnižji nivo strukture postaje, lahko uporabnik doda v celico, na enega od naslednjih načinov:

- dodajanje neobstoječe naprave NEO3000 iz liste vnaprej definiranih naprav. Naprava je vnaprej definirana, če se v poddirektoriju `Templates/Devices`, znotraj direktorija v katerem je nameščena aplikacija, nahaja predloga z ustreznim naborom datotek, ki definirajo napravo.

Sem spada `Device.xml` dokument z naborom in podatki o posameznih funkcionalnostih naprave. Poleg tega se v predlogah nahajajo še dokumenti, ki vsebujejo privzete nastavitve naprave (`Settings.xml...`) in XML sheme, ki definirajo podatkovno strukturo posameznih funkcionalnosti naprave. Ko uporabnik napravo doda, se določeni dokumenti prekopirajo v poddirektorij znotraj projektnega direktorija in vse spremembe, ki jih napravi uporabnik, se shranjujejo v dokumente

znotraj projektnega direktorija. Dokumentov iz predloge, ki niso lastni napravi, temveč tipu naprave (npr. XML sheme), ni potrebno kopirati v direktorij naprave;

- dodajanje neobstoječe naprave drugih proizvajalcev (IED): v primeru, da naprava ni tipa NEO3000, mora uporabnik profil naprave definirati preko vnosnih mask;
- dodajanje obstoječe naprave, ki se nahaja v omrežju. Pogoji, da aplikacija lahko najde napravo v omrežju je, da pozna broadcast/multicast naslov omrežja, v katerem se naprave iščejo. Poleg tega se mora naprava znati odzvati na telegrame, ki iščejo naprave v omrežju. Zaenkrat znajo to le naprave tipa NEO3000, standard IEC61850 ne predvideva, da bi se naprave v omrežju iskalo na ta način, zato mora uporabnik v tem primeru vnesti IP naslov naprave;
- izdelava kopije ene ali večih naprav, ki so prisotne v projektu: z opcijama Kopiraj/Prilepi, uporabnik na enostaven način ustvari kopijo ene ali večih naprav, ki se lahko nahajajo v isti ali drugi celici.
- Uvoz naprave iz drugega projekta.
 - Tu gre za uporabo projektnega poddirektorija, ki se nanaša na posamezen IED, v drugem projektu.
 - Funkcionalnost bo omogočala:
 - kopiraj/prilepi IED projektnega vozlišča med dvema istočasno odprtima projektoma na istem inženirskem računalniku,
 - izvoz (ang. export) v obliki, ki omogoča tudi prenos preko e-pošte (ZIP, TGZ) ter uvoz (ang. import) v drug projekt.

3.4.3 Komunikacijski model postaje

Naprave, ki se nahajajo v sistemu, so priključene v omrežje in med seboj komunicirajo tako, da pošiljajo telegrame v omrežje in sprejemajo telegrame iz omrežja. Posamezna naprava lahko sočasno telegrame le pošilja ali sprejema.

Podatki, ki jih naprava pošilja v omrežje, se definirajo na nivoju naprave (poglavje 3.5.3), kateri od telegramov, ki jih pošiljajo v omrežje druge naprave, se bodo zajemali, pa na nivoju ene naprave ni mogoče definirati. Komunikacija med napravami v omrežju se ureja v posebnem načinu, v katerem se definirajo prejemniki GOOSE sporočil, ki jih pošiljajo naprave znotraj celotne postaje.

Uporabniški vmesnik je sestavljen iz dveh delov:

- na levi strani se nahaja dvonivojska drevesna struktura, na prvem nivoju se nahajajo naprave, na drugem pa seznam podatkov, ki jih posamezna naprava pošilja v omrežje (GOOSE DataBlock),
- na desni strani se nahaja večnivojska drevesna struktura, v kateri so prikazane naprave v sistemu, njihova vozlišča (LN), podatkovni objekti (DO) in atributi (DA).

V zgornjem delu uporabniškega vmesnika poteka povezovanje; uporabnik lahko za posamezen podatek iz GOOSE DataBlock-a (na levi) določi enega ali več podatkovnih atributov (DA) (na desni). S tem, ko se določajo povezava, se naprave naročajo na telegrame, ki jih pošiljajo v omrežje druge naprave (naročanja na telegrame z lastne naprave aplikacija ne dovoljuje).

Posamezna naprava filtrira telegrame, ki jih prejme iz omrežja. Izločijo se tisti, na katere naprava ni naročena, ostali se obdelajo in posredno shranijo v ustrezen register v podatkovni

bazi naprave. Tak register je v konfiguraciji naprave prikazan kot virtualni izhodni register, ki ga uporabnik lahko poveže na vhodni konektor drugega modula.

Več o samem uporabniškem vmesniku v poglavju 3.5.3.

3.5 Nastavitve in diagnostika naprave

Aplikacija vsebuje različna orodja, s katerimi nastavljam delovanje posamezne naprave, poleg tega pa tudi orodja, ki omogočajo nadzor nad delovanjem naprave. Uporabnik ima pri delu z napravo tipa NEO3000 na voljo celoten nabor orodij; če naprava ni tipa NEO3000, podpira pa protokol IEC61850 je uporabniku na voljo le orodje za komunikacijske nastavitve (ang. Communication mapping).

Vsaka naprava NEO3000, ki je prisotna v projektu, vsebuje v pripadajočem elementu na projektnem drevesu pet različnih opcij, ki se v delovnem panelu prikažejo kot ustrezen gradnik za:

- parametracijo posameznih funkcionalnosti oz. modulov naprave,
- konfiguracijo naprave s povezovanjem vhodnih in izhodnih registrov posameznih modulov naprave,
- komunikacijske nastavitve naprave,
- pregled lastnosti naprave in prikaz diagnostičnih podatkov,
- pregled oscilografskih posnetkov, dogodkov in števcov.

Funkcionalnost se bo izvedla z zagonom 3rd party aplikacije, ki se je bo PSM zavedal, na oscilografski (Comtrade) datoteki, ki smo jo predhodno prenesli iz IED.

Nastavitve naprave se shranjujejo samo v dokumente, ki se nahajajo na lokalnem datotečnem sistemu. Nastavitve se na napravo prenesejo le na zahtevo uporabnika, aplikacija pa na podlagi definicije iz predloge naprave, razbere ali nove nastavitve zahtevajo ponovni zagon določenega modula ali celotne naprave.

Sistem mora biti tudi zasnovan tako, da omogoča zaznavo nekonsistence med lokalnimi nastavitvami naprave in podatki, ki se nahajajo na napravi.

V prvi fazi se pri preverjanju nekonsistence med lokalnimi nastavitvami v PSM projektu in nastavitvami na napravi, preverja kot je opisano v poglavju 4.2.2. strategije izdaje psm in ied programske opreme.

3.5.1 Parametracija funkcionalnosti naprave

V predlogi za posamezen tip naprave se nahaja dokument s spiskom vseh funkcionalnosti naprave. V predlogi se nahajajo tudi XML sheme, ki definirajo podatkovno strukturo in vsebino podatkov za posamezno funkcionalnost naprave. Podatki so lahko hierarhično umeščeni v eno ali več skupin, pravila za spreminjanje in prikaz vrednosti in imena parametrov so ravno tako določeni v XML shemi. Začetno stanje parametra je lahko določeno v XML shemi, lahko pa je vrednosti določena tudi v dokumentu s privzetimi nastavitvami naprave, ki se ravno tako nahaja v predlogi (`Settings.xml`).

Aplikacija na podlagi definicije iz XML sheme, ustvari večnivojski uporabniški vmesnik, preko katerega uporabnik lahko pregleduje in ureja nastavitve posameznih funkcionalnosti naprave.

Na orodni vrstici v pogledu za konfiguracijo naprave (glej poglavje 3.5.2) se nahajajo vsi funkcionalni moduli naprave. Nastavitve določene funkcionalnosti lahko uporabnik pregleduje/ureja le, če je prisotna v konfiguraciji naprave.

Uporabniški vmesnik za parametracijo je trnivojski:

- aktivne funkcionalnosti naprave, ki so na voljo za parametracijo, npr. zaščite, oscilografija... so navedene na najvišjem nivoju,
- podatkovna struktura, ki je sestavljena iz podatkovnih elementov, vsebuje pa lahko tudi podatkovne podstrukture. Število nivojev podatkovnih struktur ni omejeno,
- podatkovni element: je končni element v podatkovni strukturi. Predstavljen je z labelo, ki prikazuje ime elementa, vnosnim poljem, ki prikazuje njegovo vrednost, ter mersko enoto (V, A...), ki jo je potrebno določiti z vpisom ustreznega atributa na ustrezna mesta v XML shemi. Uporabnik lahko vrednost elementa spremeni le, če XML shema to dovoljuje. V XML shemi je določeno tudi, kakšnega tipa je vrednost elementa (celo ali decimalno število, string...), uporabniški vmesnik pa za vnos nove vrednosti, uporabniku ponudi ustrezno vnosno polje za ta tip podatka (textBox, checkBox, comboBox...).

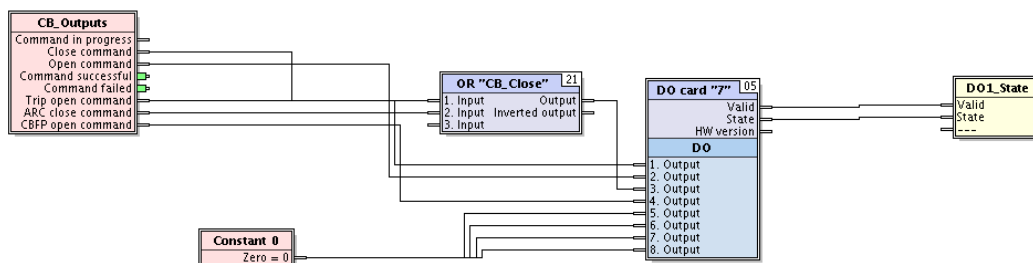
Formularni prikaz :

- omogoča razpiranje ali zapiranje prikaza vsebine posameznih funkcionalnosti in vsebine podatkovnih struktur znotraj nje,
- omogoča osnovne funkcije za urejanje teksta: iskanje, kopiranje, lepljenje...,
- validacija vnešenih vrednosti glede na pravila iz XML sheme (minimalna, maksimalna vrednost, zahtevane in opsijske vrednosti, unikatne vrednosti...),
- možnost izračuna vnešene vrednosti v vnosnem polju,
- v online načinu so (tam kjer je to možno) v vnosnem polju podatkovnega elementa, namesto referenc, prikazane „žive“ vrednosti, ki se zajamejo z naprave,
- v offline načinu niso prikazane reference na registre, v online načinu so prikazane „žive“ vrednosti registrov na napravi (online). Za potrebe izračunavanja vrednosti iz delcev v analogne vrednosti, se lahko formula za preračunavanje, priredi elementu v XML shemi (atribut formula),
- podprt je prikaz merskih enot. Merske enote bodo z ustreznim atributom (ang. unit) določene v definiciji podatkovnega elementa v XML shemi.

V primeru, da ne gre za napravo tipa NEO3000, parametracija funkcionalnosti naprave ni možna.

3.5.2 Konfiguracija naprave

Ta način uporabniku omogoča konfiguracijo naprave z dodajanjem, odstranjevanjem in grafičnim povezovanjem različnih modulov naprave. Slika 4 prikazuje primer povezovanja različnih modulov naprave.



Slika 4: Primer povezovanja različnih modulov naprave

Moduli, ki so na voljo pri posameznem tipu naprave, so navedeni v predlogi naprave, poleg tega je v predlogi določeno tudi maksimalno število instanc modula določenega tipa, ki so lahko prisotne v vezalni shemi naprave. Obstajati mora tudi možnost grupiranja različnih modulov, v skupine, npr. zaščitni moduli. Moduli ali skupine modulov so prikazani v orodni vrstici, uporabnik lahko modul iz orodne vrstice, doda na površino za prikaz vezalne sheme na poljubno mesto. S tem se ustvari instanca modula in hkrati funkcionalnost postane na voljo tudi za parametracijo (glej poglavje 3.5.1). Poleg tega aplikacija, če ima ta modul v XML shemi definirane izhodne registre, v nastavitvenem dokumentu naprave (*Settings.xml*) ustvari znotraj registerske baze – RTDB, lasten podatkovni blok (DTBLK) z vsemi potrebnimi registri. Registri so tudi ustrezno poimenovani, ime se ustvari na podlagi imena modula, imena instance modula (če je možnih več instanc), imena izhodnega registra in instance registra (če je možnih več instanc). Pri tem je potrebno upoštevati omejitve, ki so določene v XML shemi (dolžina max. 31 znakov).

Moduli so prikazani s kvadratom, v zgornjem delu kvadrata je naveden tip modula. Na levi strani kvadrata se lahko nahajajo vhodi (sponke), vsak vhod predstavlja povezavo na register v registerski bazi naprave. Na desni strani se nahajajo izhodi (sponke), ki ravno tako predstavljajo povezavo na register iz registerske baze naprave. Število in tip vhodnih/izhodnih sponk je določeno v XML shemi z definicijami elementov, ki predstavljajo reference na registre (elemente v podatkovnih blokih – DTBLK iz registerske baze – RTDB). Po potrebi se lahko v XML shemi določi grupe registrov, s čimer se zmanjša število sponk na nadrejenem modulu in poenostavi delo uporabniku pri povezovanju modulov. Če je število vhodnih ali izhodnih registrov pri posameznem modulu spremenljivo, obstaja možnost dodajanja in odstranjevanja sponk preko uporabniškega vmesnika.

Povezovanje modulov poteka tako, da uporabnik poveže izhodno sponko enega modula, z vhodno sponko drugega modula. Povezava se lahko ustvari tudi v obratnem vrstnem redu, ni pa možno povezovanje vhodnih sponk z vhodnimi in izhodnih sponk z izhodnimi. Ostale restrikcije pri povezovanju morajo biti določene v XML shemi.

Na podlagi atributa (*Ref_type* = Digital (boolean), Analog (integer) in Positive Analog (unsigned integer)) v XML shemi (pri vseh in izhodi – reference na registre) se pri povezovanju izvedejo ustrezne restrikcije.

Ko uporabnik ustvari povezavo, se v nastavitvah naprave (*Settings.xml*) znotraj podatkovne strukture začetnega in končnega modula, ustvarita referenci, ki kažeta na register, v katerega prvi modul vpisuje vrednost, drugi pa bere.

Povezave med moduli so prikazane z ravnimi ali lomljenimi črtami med dvema sponkama. Aplikacija sama poskrbi za čimbolj pregleden prikaz povezav.

V načinu, ko aplikacija ni povezana z napravo (offline), so povezave med moduli obarvane z enotno barvo (črna), v online načinu pa je z ustrezno barvo prikazana vrednost digitalne povezave (`Ref_type=Digital`)(`modra=0`, `rdeča=1`), če gre za analogno vrednost (`Ref_type = Analog (integer)` in `Positive Analog (unsigned integer)`) (povezava je zelena), je ta vrednost prikazana znotraj prekinitve na vseh koncih vezice, da uporabnik ve kateri povezavi vrednost pripada.

Vežalna shema je lahko razbita na več strani, število strani je omejeno na 12, uporabnik doda lahko novo stran po potrebi. Omogočeno je tudi poimenovanje, oz. preimenovanje strani.

Pozicije modulov, ki so prisotni na vežalni shemi naprave, se shranjujejo v lasten XML dokument, ki se nahaja znotraj lokalnega projektnega direktorija v direktoriju naprave. Če je dokument prisoten na datotečnem sistemu naprave, aplikacija omogoča, da se dokument prekopira z naprave ali sinhronizira z dokumentom na lokalnem datotečnem sistemu, s čimer je možen vpogled v vežalno shemo naprave tudi uporabniku, ki nima na voljo celotne vsebine projekta.

Znotraj delovnega področja za konfiguracijo naprave se nahaja gradnik za prikaz nastavitvev označenega modula, ki se v celoti pokrivajo z vsebino, ki je določena v nastavitvenem dokumentu naprave in XML shemi, ki definira podatkovno strukturo modula. Način prikaza je podoben prikazu, ki se uporablja za parametrizacijo funkcionalnosti naprave (opisan v tem poglavju), le da v tem primeru ni prikazana vsebina vseh funkcionalnosti naprave, temveč le nastavitve označenega modula.

V primeru, da naprava ni tipa NEO3000, konfiguracija naprave na ta način ni možna.

3.5.3 Komunikacijske nastavitve naprave

V tem načinu poteka povezovanje izhodnih registrov modulov, ki so prisotni v konfiguraciji, na ustrezne komunikacijske objekte, ki so podprti s strani IEC61850.

Predloga naprave vsebuje XML dokument (ICD), v katerem se določene predloge (ang. `DataTypeTemplates`) vseh komunikacijskih objektov (`LnodeType`), ki se lahko uporabijo za komunikacijo pri določenem tipu naprave. V predlogi so definirani še vsi podatkovni objekti (DO) ter njihovi atributi (DA), ki so lahko prisotni v instanci posameznega komunikacijskega objekta.

Preko uporabniškega vmesnika je možno dodajati, odstranjevati in povezovati instance komunikacijskih objektov z izhodnimi registri modulov v konfiguraciji. Podatkovna struktura novo ustvarjene instance objekta se generira na podlagi vsebine predloge za objekt tega tipa. Posamezne instance komunikacijskih objektov (LN), njihovi podatkovni objekti (DOI) ter atributi objektov (DAI), so uporabniku prikazani v obliki drevesne strukture.

Posameznih objektov ni potrebno poimenovati, imena objektov se prenesejo iz objektov v predlogi, kar pa za objekt tipa LN ne velja popolnoma. Objektu tega tipa je možno določiti

poljuben prefiks, ostanek imena pa vsebuje ime, ki je določeno v predlogi ter indeks instance objekta (pravila za poimenovanje so določena v XML shemi `SCL.xsd`).

Povezovanje podatkovnih atributov v komunikacijskem objektu z izhodnimi registri modulov v konfiguraciji, poteka na enega od dveh načinov:

- v uporabniškem vmesniku za konfiguracijo naprave se lahko na izhodni sponki modula, preko kontekstnega menija izbere obstoječ podatkovni atribut v komunikacijskem objektu,
- v uporabniškem vmesniku za komunikacijske nastavitve naprave, se določi podatkovni atribut komunikacijskega objekta. Preko kontekstnega menija se določi modul v konfiguraciji in njegov izhodni register.

Povezave se hranijo samo v dokumentu s komunikacijskimi nastavitvami naprave (CID), dokument s konfiguracijo naprave podatkov o povezavi ne hrani. Uporabniški vmesnik za konfiguracijo naprave, kljub temu zna prikazati na kateri komunikacijski objekt (oziroma ustrezen atribut v podatkovnem objektu), je povezan posamezen izhodni register. Povezave so v CID dokumentu zapisane kot vrednosti atributa `sAddr`, ki pripada elementu, ki predstavlja atribut podatkovnega objekta. Vrednosti atributa `sAddr` vsebuje indeks podatkovnega bloka (DTBLK), v katerem se nahaja izhodni register ter indeks registra znotraj podatkovnega bloka.

Tudi v primeru, ko naprava ni tipa NEO3000, mora biti možno pregledovati komunikacijske nastavitve naprave.

3.5.4 Lastnosti in diagnostika naprave

Ta način je namenjen prikazu lastnosti naprave, kot tudi prikazu podatkov o samem delovanju naprave (Slika 5).

V primeru da je PSM v On-line načinu, se lastnosti in diagnostika naprave prikazuje neposredno iz naprave.

V primeru, da je PSM v offline načinu, prikazuje le-to, za kar ima podatke. V aplikacijo je vgrajena funkcionalnost, ki zna diagnostične podatke in dnevnike zajeti (na zahtevo) in jih shraniti v projektni direktorij, tako da so ti uporabniku na voljo tudi v offline načinu.

Prikazana vsebina se deli na naslednja področja:

- osnovni atributi naprave: kot so: družina in tip naprave, serijska in naročniška številka, leto izdelave... so navedeni v datoteki `device_info.xml`, ki se nahaja na datotečnem sistemu naprave. Datoteka z privzetimi atributi se ravno tako nahaja v predlogi naprave; dokler se aplikacija prvič ne poveže z napravo, so uporabniku prikazani privzeti atributi. Ob prvi povezavi aplikacija sama poskrbi, da se datoteka `device_info.xml` z naprave prekopira v projektni direktorij, ter od tega trenutka dalje, v offline načinu, prikazovati podatke iz te datoteke. Prikaz `software_info.xml` podatkov o verziji programske opreme.

NEO3000 Power System Manager 0.47 - test_System	
J01	
Settings Configuration Communication	
Name	J01
Description	
Type	FPC 680 - F04
Device family	NEO3000
Version	1.0
Ordering code	FPC680-F04/L1/D11V/FG/FG/FN/FN/3/B3/B3
Serial number	
Production date	Junij 2011
Hardware	
Device up time	1 day 17 hours 49 minutes
CPU load	99%
RAM usage	63%
File system usage	59%
File system status	Read / Write
Network address	10.9.72.103
Software	
Version	2.0.0
Release date	May 2012
Platform	ppc-Linux
Distribution	NEO3000-powerpc-ppc440fp_debug-linux
Installed packages	
Calibration-N3K 1.0 (0.20110527)	
Name	Calibration-N3K
Version	1.0
Release	0.20110527
powerpc-linux	
Summary	powerpc-linux
Dependencies	NRTsystem DOM IServerCommunication
Change log	* Mon Mar 28 2011 Uros Malensek <uros.malensek@iskrasistemi.si> - Derivate of Calibration-N3Kcompact adopted to ACQ_Main.
Calibration-N3Kcompact 1.8 (0.20110527)	
DOM 1.0 (0.20110527)	
IEC61850 0.99 (7.20110527)	
IServer 1.0 (14.20110527)	
IServerCommunication 0.99 (2.20110527)	
LDU 3.1 (0.20110527)	
Neort 4.5 (0.20110527)	
NRT-Supervisor 1.3 (2.20110527)	
NRTsystem 2.6 (3.20110527)	
Oscillography 5.3 (3.20110527)	
PackageManager 0.5 (0.20110527)	
RegScan 1.2 (0.20110527)	
RT-Acquisition-N3Kcompact 0.1 (36.20110527)	
RT-Acquisition-N3KHw02 0.99 (5.20110527)	
RT-Acquisition-N3KMain 0.99 (10.20110527)	

Slika 5: Prikaz diagnostike naprave

- Diagnostika delovanja sistema: sem spada prikaz podatkov o času neprekinjenega delovanja naprave, operacijskem sistemu, ki teče na napravi, stanju določenih procesov, ki se izvajajo na napravi, obremenjenosti CPU, porabi systemskega pomnilnika, zasedenosti datotečnega sistema. Poleg tega so tu na voljo še opcije za zajem sistemskih logov (ang. dmesg). Vsi ti podatki se lahko zajamejo preko iServer-ja le, ko obstaja aktivna povezava z napravo. V aplikacijo je vgrajena funkcionalnost, ki zna diagnostične podatke zajeti (na zahtevo) in jih shraniti v projektni direktorij, tako da so ti uporabniku na voljo tudi v offline načinu.

- Diagnostika delovanja programskih modulov: vključuje prikaz delovanja real-time modulov in ostale programske opreme NEO3000 ter opcije za zajem dnevnikov delovanja posameznih procesov.
Vsi ti podatki se lahko zajamejo preko iServer-ja le, ko obstaja aktivna povezava z napravo. V aplikacijo je vgrajena funkcionalnost, ki zna vsebino dnevnikov zajeti (na zahtevo) in jo shraniti v projektni direktorij, tako da so ti uporabniku na voljo tudi v offline načinu.
- Zagon namenskih skript: tu mora biti omogočeno poganjanje skript
Definicija nekaterih skript:
 - `reboot_IED` (ponovni zagon cele naprave)
 - `restart_APP` (restart NEO3000 aplikacije, brez ponovnega zagona OS)

Pri prikazu diagnostičnih podatkov in log-ov je zaželeno (vendar ne nujno) izfiltrirati vse izpise, oz. informacije, ki bi nakazovale na operacijski sistem, oz. uporabljeno arhitekturo naprave.

V primeru, da naprava ni tipa NEO3000, ta način ni omogočen.

3.6 Online način dela z napravo

Poleg Offline dela s projektom in napravo, poznamo tudi Online delo z napravo.

Pri Offline delu s projektom (in napravo), s posamezno napravo ne komuniciramo direktno, ampak na osebem računalniku spremenimo parametre, konfiguracijo, ... in vse spremembe naložimo na napravo preko Upload procedure, skupaj z uveljavitvijo novih nastavitev.

Pri Online delu z napravo pa poteka živi prenos podatkov.

V primeru, da ne gre za napravo tipa NEO3000, Online delo z napravo ni možno.

3.6.1 Online – Funkcionalne nastavitve

V Online načinu so v oknu funkcionalnih nastavitev (ang. Functional Settings) prikazani poleg parametrov posameznih funkcij tudi vrednosti vhodov in izhodov funkcijskih blokov.

V Online načinu je nad posamezno funkcijo možna tudi akcija `restart_INSTANCE`. Perioda osveževanja je nastavljiva v globalnih parametrih PSM.

3.6.2 Online – Preslikava naprav

V Online načinu so v oknu preslikave naprav (ang. Device mapping) prikazani poleg sheme povezav funkcijskih blokov tudi vrednosti vhodov in izhodov funkcijskih blokov. Opis je v poglavju 3.5.2.

V Online načinu je nad posameznimi funkcijskimi bloki možna tudi akcija `restart_INSTANCE`.

Perioda osveževanja je nastavljiva v globalnih parametrih PSM.

3.6.3 Online – Preslikava naprav

V Online načinu so v oknu preslikava naprav (ang. Communication mapping) v prvi fazi ne prikazuje ničesar (enako kot offline).

V drugi fazi lahko implementiramo v PSM funkcionalnost IEC61850 Client funkcionalnost (kot npr. IEDScout), ki bo znala na podlagi ICD datoteke, ki jo imamo za posamezno napravo, komunicirati preko IEC61850 protokola.

3.6.4 Online – Diagnostika naprave

V Online načinu so v oknu diagnostike naprave (ang. Device diagnostics) prikazane dejanske vrednosti posameznih parametrov delovanja naprave (npr.: ang. system resources, logs, ...). Perioda osveževanja je nastavljiva v globalnih parametrih PSM.

3.7 Nivoji uporabnikov

Aplikacija bo omogočala tudi prijavljanje v aplikacijo, oz. na napravo z različnimi uporabniki.

V prvi fazi je izvedeno fiksno prijavljanje v sistem, oz. napravo z uporabniškim imenom nivoja 2 (npr.: root / .root). Uporabnik vidi vse.

V drugi fazi se izvede upravljanje z uporabniki preko že znanega mehanizma iServer avtentikacije tako v online in offline načinu. To pomeni da se v Template vključi iServer.xml, kjer so predefinirani uporabniki in gesla:

- NEO Administrator (nivo 1) - omogoča dodajanje/urejanje uporabnikov z nivoji 1 ali višjimi.
- NEO Inženir (nivo 2) - omogoča dodajanje/urejanje uporabnikov z nivoji 2 ali višjimi.
- Uporabnik Administrator (nivo 3) – omogoča dodajanje/urejanje uporabnikov z nivoji 3 ali višjimi.
- Uporabnik1 (nivo 4) – ne omogoča upravljanja z uporabniki.
- Uporabnik2 (nivo 6) - ne omogoča upravljanja z uporabniki.

Pri uploadu na napravo se iSever.xml na napravo prenese samo, če je prijavljen z NEO uporabniki ali Uporabnik Administrator, sicer se prenaša le v smeri Downloada.

Nivoji uporabnikov bodo omogočali skrivanje parametrov glede na nivo uporabnika (Settings in Configuration), izvajanje določenih akcij (npr.: restart_instance, zaganjanje določenih skript preko okna diagnostike), gledanje sistemskih orodij (logs, system resources, ...), ...

Uvesti je potrebno tudi „device_change.log“, v katerega se beležijo dostopi in nekatere akcije (Download, Upload, Restart, ...) uporabnikov. Kasneje ta log dopolnjuje tudi LDU aplikacija za spreminjanje parametrov in se na ta način uvede tudi preverjanje konsistence podatkov.

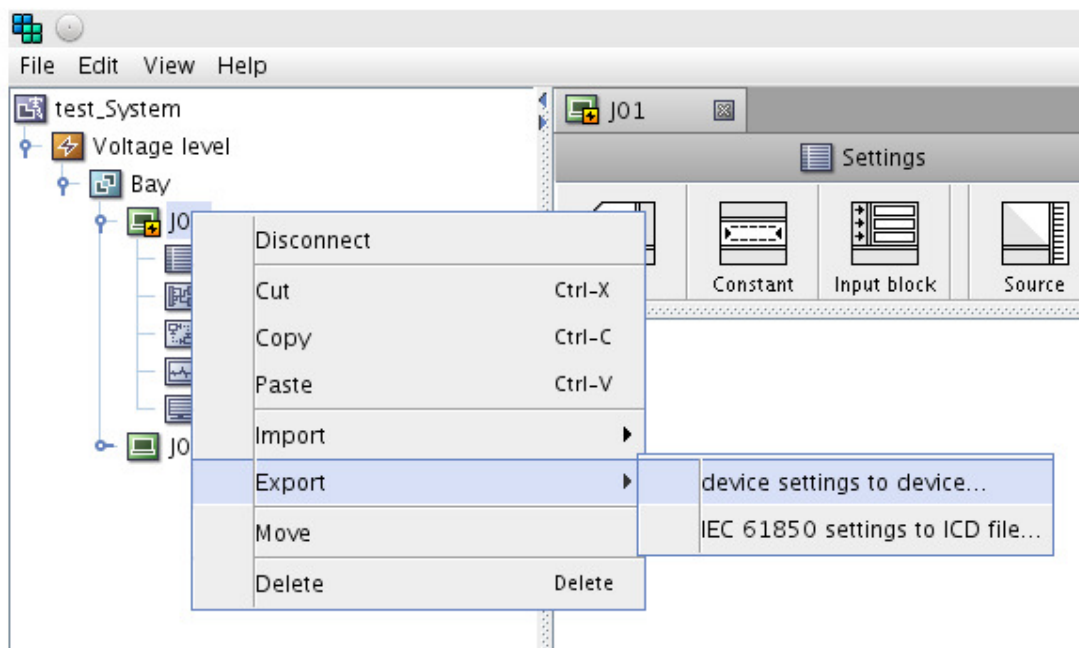
Če se na kopiji naprave v PSM in na napravi zadnji zapis ne ujema, je potrebno opozorilo da je nekdo spreminjal parametre.

Podatki v log: (iServer_)username, PSM_hostname, datum+ura, tip akcije (Download, Upload, Restart, ...)

Ta log se prikazuje v oknu diagnostika.

3.8 Upload/download funkcionalnost

Funkcionalnost Upload omogoča prenos aplikativnih in sistemskih konfiguracijskih datotek iz projekta na ciljno napravo. Selektivni prenos posameznih konfiguracijskih datotek ni podprt. Slika 6 prikazuje primer prenosa nastavitve na želeno napravo.



Slika 6: Prenos datotek na napravo

Prenos se inicializira s pritiskom na gumb na GUI in izvede ob potrditvi prenosa spremenjenih nastavitvev. Pri tem se opravi preverjanje, kot je opisano v poglavju 3.3.2.

Ob potrditvi prenosa projekta na IED je potrebno preko GUI dialoga še določiti akcijo (activate, restart, reboot, ...), ki sledi upload-u.

V prvi fazi PSM se po uploadu ponudi možne akcije:

- reboot_IED (default) – za tem stoji neka skripta, ki opravi restart celega IED
- restart_APP - za tem stoji neka skripta, ki opravi restart NEO3000 aplikacije (brez ponovnega zagona OS)

Funkcionalnost Download omogoča prenos aplikativnih in sistemskih nastavitvenih datotek iz projekta na ciljno napravo. Selektivni prenos posameznih nastavitvenih datotek ni podprt.

Prenos se inicializira s pritiskom na gumb na GUI in izvede ob potrditvi prenosa spremenjenih nastavitvev. Pri tem se opravi preverjanje, kot je opisano v poglavju 3.3.2 - STRATEGIJE IZDAJE PSM in IED PROGRAMSKE OPREME.

3.9 Predloga naprave

Predloge naprav se nahajajo v direktoriju *Templates/Devices* – ta se nahaja v direktoriju, v katerem je nameščena aplikacija. Vsak poddirektorij v direktoriju *Templates/Devices* predstavlja predlogo za določeno napravo iz družine naprav NEO3000.

V korenskem direktoriju predloge se mora obvezno nahajati datoteka *Device.xml*, poleg tega pa še vsi dokumenti, ki jih aplikacija potrebuje za to, da naprava lahko postane del projekta in da uporabnik lahko parametrira ter konfigurira napravo.

Format datotečne strukture predloge je spremenljiv, vsi potrebni dokumenti se lahko nahajajo znotraj korenskega direktorija ali pa so organizirani drevesno. Vsebina, ki nima neposredne zveze z aplikacijo, se lahko nahaja v predlogi. Z vidika nadaljnjega razvoja aplikacije v prihodnosti, je smotrno, da je predloga organizirana na enak način, kot datotečni sistem naprave.

3.9.1 Dokument, ki definira predlogo (*Device.xml*)

Device.xml je XML dokument, ki je za GUI obvezno prisoten znotraj korenskega direktorija, ki predstavlja predlogo naprave. V dokumentu se nahajajo informacije o sami napravi, kot tudi lokacije dokumentov, ki vsebujejo privzete podatke o nastavitvah naprave.

Ko uporabnik doda v projekt novo napravo, ki je bila izbrana iz liste naprav, ki so definirane s predlogo, se v projektne direktoriju ustvari direktorij, ki je lasten napravi. V ta direktorij se prekopirajo dokumenti iz predloge, ki so definirani v elementih *Functional_settings*, *Device_mapping* in *Communication_mapping*. Če kateri od teh dokumentov ne obstaja, aplikacija ustvari nov dokument s privzeto vsebino.

3.10 Podatkovne strukture in dokumenti

3.10.1 Projektna podatkovna struktura

Projektna podatkovna struktura je drevesna struktura, sestavljena iz štirih različnih objektov: postaja, napetostni nivo, celica in naprava. Vsak objekt se nahaja na svojem nivoju v podatkovni strukturi. Pravila kateremu objektu lahko nek drug objekt pripada in katere podobjekte lahko vsebuje, so enolično določena (Slika 7).

Podatkovna struktura projekta se hrani v XML dokumentu z imenom *Project.xml*, ki se nahaja znotraj projektne direktorija. Vsak objekt iz projektne podatkovne strukture, je v tem dokumentu predstavljen z ustreznim tipom elementa. Poleg tega vsakemu objektu pripada še lasten direktorij, ki se nahaja znotraj projektne direktorija.

3.10.2 Objekti v podatkovni strukturi

Project (projekt)

Objekt vsebuje podatkovno strukturo postaje, poleg tega pa tudi vrednosti atributov, ki so jih definirani v fazi kreiranja novega projekta, ter spremembe, ki jih uporabnik izvede v odprtem projektu.

Objekt ima definirane naslednje attribute:

- Title: ime projekta; atribut mora imeti vrednosti. Vrednost atributa je enaka imenu projektnega direktorija,
- Path: pot do projektnega direktorija na lokalnem datotečnem sistemu,
- SubStation: referenca na objekt, ki predstavlja postajo v energetskega omrežju,
- ChangeLog: dnevnik sprememb, ki jih uporabnik izvede nad enim ali večimi objekti, ki pripadajo podatkovni strukturi postaje.

SubStation (postaja)

Objekt predstavlja postajo v energetskega omrežju in se nahaja na prvem nivoju podatkovne strukture. Objekt ima definirane naslednje attribute:

- Name: ime postaje; atribut mora imeti vrednosti,
- Description: opis postaje; vrednost atributa ni obvezna,
- Project: referenca na nadrejeni objekt,
- VoltageLevels: lista objektov tipa VoltageLevel. Število objektov v listi ni omejeno, objekti v listi se med seboj ločijo po imenu.

VoltageLevel (napetostni nivo)

Objekt predstavlja napetostni nivo postaje; nahaja se na drugem nivoju projektne podatkovne strukture, kot podobjekt objekta tipa SubStation. Objekt ima definirane naslednje attribute:

- Name: ime napetostnega nivoja; atribut mora imeti vrednosti, ki je unikatna glede na vrednost istega atributa v ostalih objektih enakega tipa,
- Description: opis napetostnega nivoja, vrednost atributa ni obvezna,
- SubStation: referenca na nadrejeni objekt,
- Bays: lista objektov tipa Bay. Število objektov v listi ni omejeno, objekti v listi se med seboj ločijo po imenu.

Bay (celica)

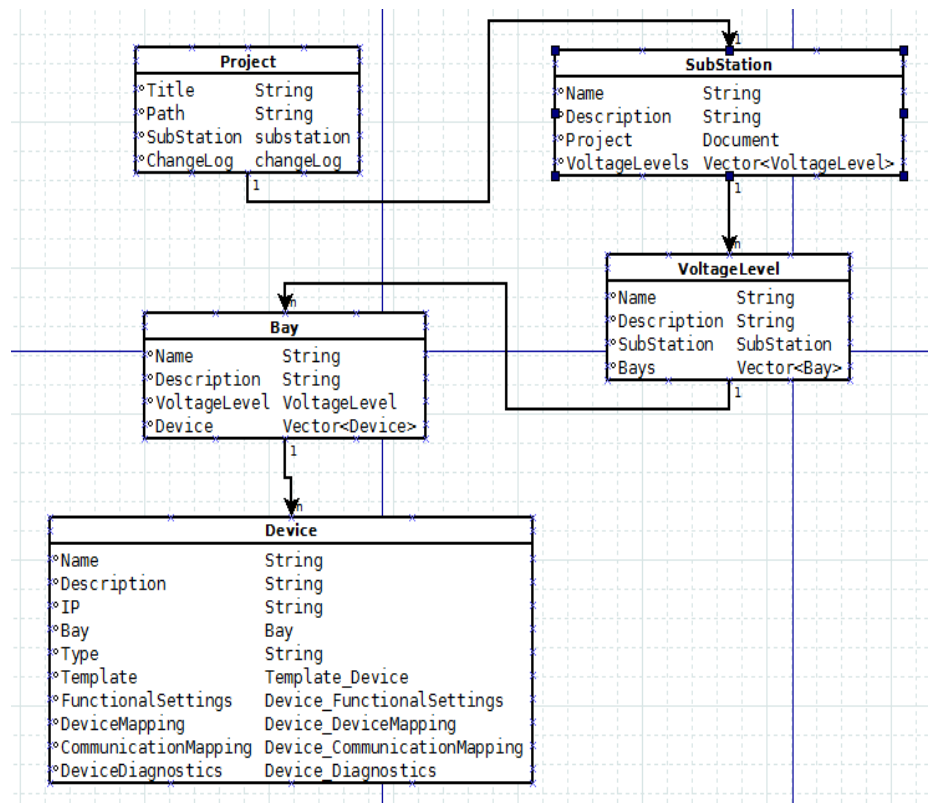
Objekt predstavlja posamezno celico znotraj enega napetostnega nivoja; nahaja se na tretjem nivoju projektne podatkovne strukture, kot podobjekt objekta tipa VoltageLevel. Objekt ima definirane naslednje attribute:

- Name: ime celice; atribut mora imeti vrednosti, ki je unikatna glede na vrednost istega atributa pri ostalih celicah znotraj istega napetostnega nivoja,
- Description: opis celice, vrednost atributa ni obvezna,
- VoltageLevel: referenca na nadrejeni objekt,
- Device: lista objektov tipa Device. Število objektov v listi ni omejeno, objekti v listi se med seboj ločijo po imenu.

Device (naprava)

Objekt predstavlja napravo v celici; nahaja se na četrtem nivoju projektne podatkovne strukture, kot podobjekt objekta tipa Bay. Objekt ima definirane naslednje attribute:

- Name: ime naprave; atribut mora imeti vrednosti, ki je unikatna glede na vrednost istega atributa pri ostalih napravah, ki pripadajo isti celici,
- Description: opis naprave, vrednost atributa ni obvezna,
- IP: naslov naprave v omrežju, vrednost atributa ni obvezna,
- Bay: referenca na nadrejeni objekt,
- Type: tip naprave, vrednost atributa je prazna, če naprava ni tipa NEO3000,
- Template: referenca na objekt, ki predstavlja predlogo za ustrezen tip naprave. Podatkovna struktura objekta je opisana v poglavju 3.10,
- FunctionalSettings: objekt s podatkovno strukturo, ki se navezuje na nastavitve posameznih funkcionalnosti naprave, ki so prisotne v konfiguraciji,
- DeviceMapping: objekt s podatkovno strukturo, ki se navezuje na konfiguracijo naprave,
- CommunicationMapping: objekt s podatkovno strukturo, ki se navezuje na komunikacijske nastavitve naprave,
- DeviceDiagnostics: objekt s podatkovno strukturo, ki se navezuje na diagnostiko naprave.



Slika 7: Prikaz relacij objektov

3.10.3 Dokument z vsebino podatkovne strukture

Podatkovna struktura projekta se ob shranjevanju pretvori v XML dokument, vsebina dokumenta se zapiše v datoteko `Project.xml`. Datoteka je locirana na prvem nivoju v projektnem direktoriju.

Hierarhija elementov v XML dokumentu je enaka, kot hierarhija objektov opisana v poglavju 3.9.2, vendar pa se v dokumentu ne hrani podatkovna struktura vseh objektov. Zapisovanje v XML dokument se preneha pri napravi. V dokument se zapišejo le osnovni atributi naprave (ime, opis, tip...). Konfiguracija ter nastavitve naprave, diagnostični podatki ipd. pa se hranijo v datotekah, ki so locirane znotraj projektnega poddirektorija, ki pripada napravi.

V dokumentu so uporabljena enaka imena za poimenovanje elementov, kot se uporabljajo za poimenovanje objektov ter njihovih atributov. Atributi, ki nimajo definirane vrednosti, se ne shranijo v dokument.

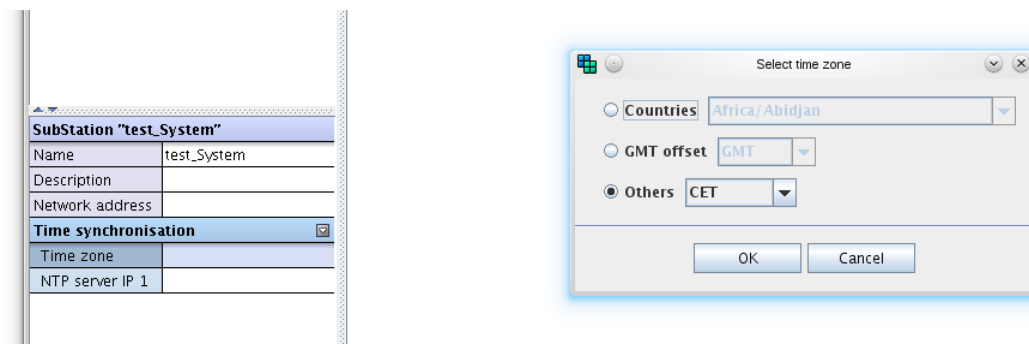
Poenostavljen primer XML dokumenta s podatkovno strukturo projekta:

```
<Project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SubStation>
    <Name>Test station</Name>
    <Description>This is just an example</Description>
    <TimeZone>GMT offset/GMT</TimeZone>
    <VoltageLevel>
      <Name>20 kV</Name>
      <Bay>
        <Name>Bay 1</Name>
        <Device>
          <Name>Device 1</Name>
          <IP>10.0.1.71</IP>
          <Type>FPC 680</Type>
        </Device>
        <Device>
          <Name>Device 2</Name>
          <IP>10.0.1.72</IP>
          <Type>FPC 680</Type>
        </Device>
      </Bay>
    </VoltageLevel>
  </SubStation>
</Project>
```

V zgornji strukturi XML dokumenta je zapisana vrstica, ki programu pove katera časovna datoteka je nastavljena. Grafični vmesnik za izbiro časovne datoteke ta podatek uporabi za začetni izpis v lastnostih postaje. Če podatka ni ali pa oznaka nima definirane vrednosti, se uporabniku prikaže prazno polje. Seveda pa prikaz časovne datoteke v lastnostih postaje ni odvisen samo od tega XML dokumenta, ampak tudi od zoneinfo direktorija, ki se nahaja na inštalacijskem direktoriju. Več o prikazu in prenosu časovnih datotek v naslednjih 2 poglavjih.

4 GRAFIČNI VMESNIK ZA IZBIRO ČASOVNE DATOTEKE

Kot smo že omenili v 3. poglavju, je zoneinfo direktorij razdeljen na več podmap. Vsaka mapa združuje datoteke po njihovih lastnostih. Zato smo se odločili, da bo bil tudi grafični vmesnik za izbiro želene časovne datoteke razdeljen na 3 segmente (Slika 8).



Slika 8: Prikaz grafičnega vmesnika za izbiro časovne datoteke

Prvi segment je namenjen časovnim datotekam, ki so ločene po geografski lokaciji. Drugi segment je specifično namenjen samo GMT datotekam. Tretji pa je namenjen vsem ostalim časovnim datotekam, ki se nahajajo na glavnem direktoriju. Tu naj še omenimo, da GUI ne prikaže map `right` in `posix` in datotek na glavnem nivoju, ki imajo v imenu GMT. Prav tako ne prikaže vseh pomožnih datotek, npr. `iso3166.tab`, povezava `localtime`, `posixrules`, `zone.tab`.

Pri implementaciji tega okna v PSM smo naleteli na težavo. V lastnostih objektu ni bila predvidena uporaba dialogov. Tako so v delu za lastnosti, bila vsebovana vnosno polje, spustna lista in polje z logično vrednostjo. Sicer bi lahko za izdelavo uporabili dve spustni listi, pri kateri bi zgornja spustna lista določala prikazane vrednosti spodnje spustne liste. Za to opcijo se nismo odločili, saj smo mnenja, da bi lahko bil uporabnik zmeden, saj ne bi vedel zakaj mora izbirati iz dveh spustnih list, ko pa določa samo en parameter. Mogoče ne bi niti opazil razlike, ki bi se zgodila v drugi spustni listi, itd. Tudi ni najbolj uporabniku prijazno, ko se sredi izvajanja zamenja vrednosti spustnih list.

Zato smo se odločili, da bomo za to polje uporabili nov tip, ki bo na uporabnikov klik odprl novo okno, v katerem bomo lahko prikazali te tri skupine.

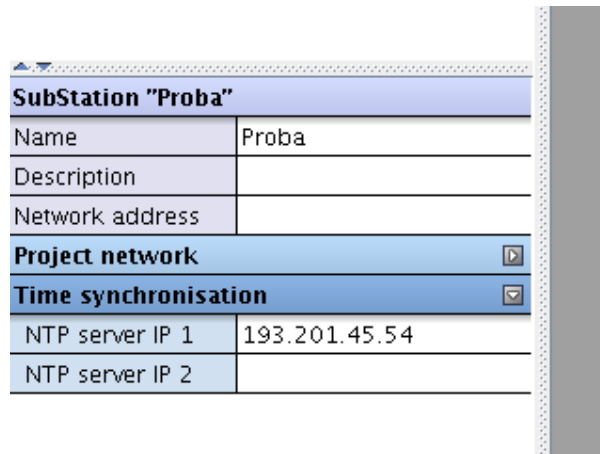
Grafični vmesnik ima na voljo 3 spustne liste, med katerimi je naenkrat aktivna samo ena. Med temi listami uporabnik izbira s pomočjo izbirnih gumbov (ang. radio buttons).

Okno se zapre na 3 načine. Ob pritisku na gumb okvirja „Zapri“ ali gumba „Confirm“ ali „Cancel“. Vrednost se zapiše v del okna, kjer so lastnosti (ang. properties) samo, ko uporabnik potrdi izbiro s klikom na gumb „Confirm“.

4.1 Uvod v objekt `Device_SystemSettings_TimeZone`

Ob zagonu PSM aplikacije, najprej preverimo ali na lokanlem direktoriju obstaja mapa `zoneinfo`. Le ta vsebuje vse časovne datoteke. Če direktorij obstaja, v lastnosti izpišemo

možnost za menjavo časovne datoteke. Nato iz `Project.xml` v elementu `timeZone` preberemo, katera časovna datoteka je že shranjena na postaji in vseh njenih pripadajočih napravah. V primeru, da `Project.xml` ne vsebuje elementa `timeZone`, je opcija prazna. Drugače vanjo zapišemo relativno pot datoteke od `zoneinfo` direktorija naprej. Uporabnik spremeni časovno datoteko s klikom na lastnost `Time zone` v lastnostih postaje. S tem sproži prikaz dialoga tipa `Device_SystemSettings_TimeZone`. Ob potrditvi se pod možnost `Time zone` v lastnostih postaje zapiše nova vrednost. Drugače se ohrani stara vrednost.



SubStation "Proba"	
Name	Proba
Description	
Network address	
Project network	
Time synchronisation	
NTP server IP 1	193.201.45.54
NTP server IP 2	

Slika 9: Prikaz lastnosti, ko aplikacija ne vsebuje `zoneinfo` direktorija

4.2 Seznam spremenjenih oz dodanih datotek

Za potrebe vmesnika za izbiro časovne datoteke, smo morali spremeniti ali dodati naslednje razrede:

- Spremenjeni razredi:
 - `Property_Field`
 - `PropertyInput`
 - `SubStaion`
- Dodani razredi:
 - `Device_SystemSettings_TimeZone`
 - `GuiNoTimeZoneFolder`
 - `GuiChooseTZ2`

4.3 Dodane lastnosti razredu `Property_Field`

Razred `Property_Field` je tisti razred, ki poskrbi za inicializacijo in izris lastnosti postaje, oz. naprav. Glede na to katere spremenljivke razreda uporabiš pri inicializaciji, PSM izriše vrstico za lastnost s 3, oz. 4 tipi uporabniških vmesnikov. Na prvi pogled so gradniki med sabo vsi enaki. Ko pa uporabnik želi spremeniti vrednost določene lastnosti, se ustrezno glede na to katere spremenljivke ima podane, izrišejo grafični vmesniki. Tako se lahko ob kliku na lastnost pojavi kombinirana lista, vnosno polje, dialog ali pa se ne zgodi nič, saj lastnosti ni mogoče spremeniti.

V razred `Property_Field.java` smo dodali naslednje tri globalne spremenljivke:

- `Object Dialog`
- `Object extendedClassOfBase`

Vse so tipa `Object` z namenom, da jih lahko uporabljamo za različne tipe dialogov.

Spremenljivka `Dialog` je namenjena hranjenju celotnega objekta. V našem primeru je to `Device_SystemSettings_TimeZone`. Ta je neke vrste vmesnik s katerim se srečujejo ostali objekti.

Spremenljivka `extendedClassOfBase` je namenjena, da kaže na tisto spremenljivko objekta `Device_SystemSettings_TimeZone`, ki je tipa `Base`. S pomočjo tega razreda prikažemo dialog na zaslon.

4.4 Dodane lastnosti razredu `PropertyInput`

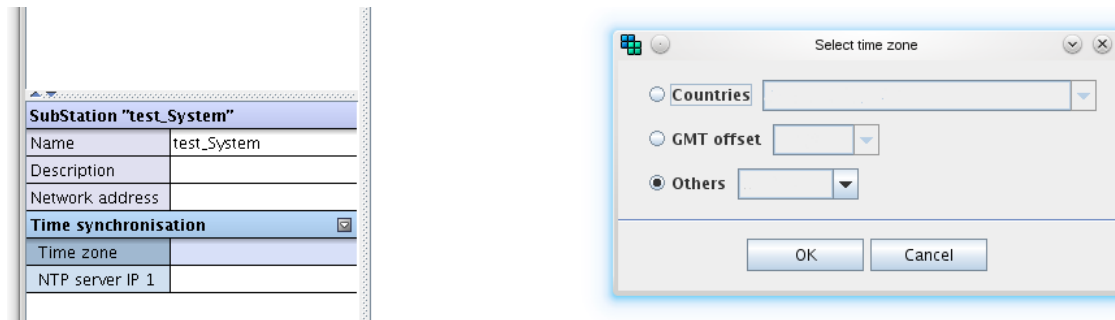
Kot že samo ime razreda pove, je ta razred namenjen izrisu ustreznega vnosnega polja, ko uporabnik klikne na lastnost.

V razredu smo spremenili metodo `protected boolean Edit(MouseEvent e)`. Dodali smo pogoj za element tipa `TYPE_DIALOG`. Tukaj se najprej prikaže okno dialoga. Če uporabnik potrdi, se v spremenljivko `File chosenFile` v razredu `Device_SystemSettings_TimeZone` zapiše izbrana časovna datoteka. Pri potrditvi se dobi absolutna pot izbrane časovne datoteke, ki se jo nato s pomočjo `zoneinfo` direktorija skrajša na relativno. Relativna pot bo izpisana na zaslonu v sekciji lastnosti postaje. Potrebno je omeniti, da se datotekam izbranim v spustni listi GMT pretvori njihovo ime. S tem mislimo na datoteke GMT, ki imajo ime GMT+ ali GMT- neko število. Te datoteke imajo ravno nasproten predznak odmika od GMT, kot pa je ime datoteke. Tako na primer datoteka GMT+1 prehiteva GMT za 1 uro in ne zaostaja tako kot nakazuje ime datoteke. Ob potrditvi se kliče metoda `public boolean Element_Change(Element element, String property, Object oldValue, Object newValue)`, ki poskrbi da se sprememba zapiše v lastnosti postaje in shrani za nadaljnje manipuliranje s spremembami v PSM programu. Klic te metode je potreben saj se mora sprememba zabeležiti v objekt `Change`. V tej vrsti se hranijo vse spremembe, ki so se zgodile med uporabo PSM vmesnika. Nastavitvene možnosti, ki so tipa `TYPE_DIALOG`, nimajo nikjer implementirane metode `Validate()`, saj se ne ve kakšne vse tipe bo vračal dialog, ki si ga bo izmislil naslednji programer. Zato mora programer sam poskrbeti za validacijo vrnjene vrednosti.

4.5 Dodane lastnosti razredu `SubStation`

V tem razredu se generirajo vse lastnosti postaje. Tu sta tudi metodi, ki inicializirata in spremenita vrednosti posameznim lastnostim. Med drugimi je tu še metoda, ki prebere začetne vrednosti iz `Project.xml` in metoda, ki shrani vrednosti v `Project.xml`.

Dodali smo globalni spremenljivki `Device_SystemSettings_TimeZone` `timeZoneWindow` in `String timeZone`. V konstruktorju tega razreda inicializiramo spremenljivko `Device_SystemSettings_TimeZone` `timeZoneWindow`. Metoda `public Properties getProperties()` poskrbi, da se `timezone` polje doda v `Properties` objekt. Polje za spremembo časovne datoteke se doda samo ob pogoju, da program vsebuje `zoneinfo` direktorij. Tukaj moram poudariti, da ta pogoj za prikazovanje polja glede na direktorij ni najboljši. Za primer lahko vzamemo prazen `zoneinfo` direktorij. Aplikacija bo prikazala polje v lastnostih postaje in tudi pogovorno okno se bo odprlo in ponudilo uporabniku na izbiro prazne sezname.



Slika 10: Prikaz grafičnega vmesnika, če uporabnik izbriše datoteke in mape v `zoneinfo`

Spremenljivka `window` v objektu `Device_SystemSettings_TimeZone` je lahko tipa `GuiNoTimeZoneFolder`, če `zoneinfo` direktorij ne obstaja, ali tipa `GuiChooseTZ2`, če `zoneinfo` direktorij obstaja. Ko je objekt tipa `SubStation` generiran kličemo metodo `public boolean Load(org.w3c.dom.Node elem)`, kjer iz `Project.xml` preberem zadnjo shranjeno datoteko na postaji in vseh njenih napravah. Relativno pot te datoteke zapišem tudi v spremenljivko `String timeZone`, ki se prikaže v lastnostih postaje. Metoda `SetProperty(String id, Object oldValue, Object newValue, Change change)` poskrbi, da se ob spremembi zapiše nova vrednost v ustrezno polje tabele lastnosti. Hkrati se shrani stara vrednost za pomoč `undo/redo` funkcionalnosti.

4.6 Dodan razred `Device_SystemSettings_TimeZone`

Razred `Device_SystemSettings_TimeZone` služi kot vmesnik med grafičnim prikazom in ostalim svetom PSM aplikacije. Tako bodoči programer, ki bi želel ustvariti novo polje za izbiro časovne datoteke, kliče samo konstruktor tega tipa in ga priredi spremenljivki `Dialog` objekta `Property_Field`.

`Device_SystemSettings_TimeZone` razred ima tri spremenljivke: `Object window`, `String dir` in `JFrame fatherFrame`.

S konstruktorjem `Device_SystemSettings_TimeZone(JFrame fatherFrame, boolean isVisible)` se inicializirajo lokalne spremenljivke in hkrati se s spremenljivko `isVisible` nastavi ali se bo dialog prikazal takoj ob kreiranju ali ne. Če obstaja `zoneinfo` direktorij se kliče konstruktor razreda `GuiChooseTZ2`, drugače se kliče konstruktor razreda `GuiNoTimeZoneFolder`.

Razred vsebuje tudi metode `Object getWindow()`, `File getChosenFile()` in `String getDir()`, ki vračajo vrednosti lokalnih spremenljivk. V metodi `Object getWindow()` se vrnjena vrednost primerno casta v dejanski tip objekta `window`.

4.7 Dodan razred `GuiNoTimeZoneFolder`

Razred je bil prvotno namenjen prikazu dialoga, če uporabnik ne bi imel nameščenega `zoneinfo` direktorija. S spremembo funkcionalnosti je potreba po tem dialogu odpadla. Dorečeno je bilo, da se polje „Time zone“ v lastnostih postaje prikaže samo, ko ima uporabnik nameščen `zoneinfo` direktorij.

Kadar ni `zoneinfo` direktorija se tako kreira objekt tega tipa, katerega referenca je shranjena v `window` spremenljivki razreda `Device_SystemSettings_TimeZone`.

4.8 Dodan razred `GuiChooseTZ2`

`GuiChooseTZ2` razred je namenjen izrisu dialoga ter prikazu časovnih datotek v spustni listi. Razred razširja `Base` razred. Za izbiro časovnih datotek so na voljo tri spustne liste. Ker so časovne datoteke razdeljene na tri nivoje, časovne datoteke izražene geografsko (prvo ime do znaka / predstavlja celino, nato pa je ime mesta), GMT časovne datoteke in ostale, je dialog zgrajen s pomočjo izbirnih gumbov (ang `radiobutton`), ki omogočijo, oz. onemogočijo delo s spustno listo. Uporabnik potrdi svojo izbiro s pritiskom na gumb `Confirm` ali zavrne spremembe z zaprtjem okna ali pritiskom na gumb `Cancel`.

Konstruktor `GuiChooseTZ2(JFrame frame, String dir, boolean isVisible)` je zadolžen za sestavo okna, prikaz tega na zaslonu, če je tako določeno s spremenljivko `isVisible`, ter pozicioniranje dialoga relativno na njegovega prednika, ki je podan s spremenljivko `JFrame frame`.

V metodi `void Initialize()` sestavimo dialog z vsemi komponentami. Za izpis možnosti v spustni listi, se najprej pridobi vse potrebne datoteke, nato pa se jih s pomočjo objekta `Comparator` sortira in v takšnem vrstnem redu zapiše v ustrezen `JComboBox`. V enakem vrstnem redu se zapišejo datoteke tudi v tabelo datotek `File[]`. Časovne datoteke, ki so predstavljene geografsko, se sortirajo naraščajoče v abecednem vrstnem redu. Pri GMT datotekah najprej izpišemo GMT datoteko, nato ji sledi GMT0 in nato ji sledita 2 skupini datotek (plus in minus skupina), ki so sortirani po velikosti odmika. Ostale datoteke se sortirajo po abecedi in se nahajajo v `Others` skupini. Za izpis GMT datotek na zaslon ustrezno spremenimo predznake, zato da ime predstavlja dejanski odmik od GMT. Ob inicializaciji za privzeto datoteko izberemo CET datoteko v skupini `Others`.

Metoda `void actionPerformed(ActionEvent e)` glede na izbrano opcijo izbirnega gumba onemogoči izbiro iz ostalih dveh spustnih list. Metoda `void Update()` poskrbi, da se ustrezno posodobi spremenljivka `File chosenFile`. Metoda `void Confirmed()` najprej kliče metodo `void Update()` in nato `void Confirmed()` metodo nadrazreda. Namen metode `void Canceled()` je, da nastavi spremenljivko `File`

`chosenFile` na `null` vrednost, tako da razred, ki ga bo klical, ve, da uporabnik ni izbral nobene datoteke. `JPanel AddRow(String title, int width)` metoda je modificirana, tako da se ne upošteva naslov in ustvari samo prazen `JPanel` na katerega se lahko dodajajo ostali poljubni elementi. Dodali smo metodo `JRadioButton AddRadio(String title)`, ki ustvari nov izbirni gumb in ga doda v skupino `ButtonGroup buttonGroup`.

`static boolean isTimeZoneFile(File file)` je metoda, ki dobi za argument datoteko in preveri ali je časovna datoteka. To naredi tako, da preveri prvih 28 bajtov datoteke. Prvih 5 bajtov predstavlja magic word `TZif` in številko verzije. V našem primeru je to verzija številka 2. Nato sledi 15 bajtov, ki morajo biti enaki 0 in so rezervirani za nadaljnjo uporabo. Temu sledijo dvakrat 4 bajti, ki predstavljajo isto celo število, ki ne sme imeti 4. bajta enakega 0.

Za listo vseh časovnih datotek, ki se nahajajo v določenem direktoriju in vseh njegovih poddirektorijih smo napisali metodo `static void listTimeZoneFilesThatAreInDirs(File dir, ArrayList<File> tmp, File parent)`, ki dobi podano prazno listo `tmp` in jo napolni. Metoda začne iskanje v mapi `File dir` in potem rekurzivno pregleda vse direktorije, ki niso skriti ali `right`, `posix` in `Etc`. Iz rezultata izključi časovne datoteke, ki se nahajajo na prvem nivoju, saj se te uporabijo le v `others` listi. Najdene časovne datoteke so zapisane v list `tmp`.

5 PRENOS IN POMNENJE ČASOVNE DATOTEKE

Prenos izbrane časovne datoteke se izvrši samo, ko uporabnik to eksplicitno zahteva. Prenos izbrane časovne datoteke se izvrši med akcijo izvoza sistemskih nastavitvev na napravo. Ker se sistemske nastavitve naprav razlikujejo, je potrebno tudi za vsako napravo posebej izvoziti njene nastavitve.

Na napravo se lahko izvozi nastavitve, samo takrat, kadar le ta spada v družino NEO3000 in je vzpostavljena povezava. Če ta dva pogoja nista izpolnjena, na napravo ali iz naprave ni mogoče naložiti ali pridobiti nastavitvev.

Metoda `boolean Save(org.w3c.dom.Node parentElem)` v razredu `Substation.java` shrani trenutno izbrano vrednost časovne datoteke, s pomočjo metode `boolean Save(SubStation subStation)` objekta `timeZoneWindow`. V `Project.xml` se tudi zapiše pot izbrane relativne časovne datoteke. Zapiše se kot vrednost v element `TimeZone`, ki je pod elementom `SubStation`.

5.1 Pomnjenje časovne datoteke

Zaradi preglednosti kode je v razredu `Device_SystemSettings_TimeZone` definirana tudi metoda `boolean Save(SubStation subStation)`. V tej metodi se popravi datoteka `neo3000.conf` pod projektnim direktorijem `etc`, ki se nahaja na nivoju naprave. Če datoteka ne obstaja, program nadaljuje izvajanje in ne opozori uporabnika. Datoteka obstaja samo takrat, kadar je to zapisano v predlogah naprave. Prejšnja `neo3000.conf` datoteka se preimenuje v `neo3000.con~` in na novo se ustvari datoteka `neo3000.conf`. Relativna pot z ustrezno besedo spredaj, ki predstavlja skupino iz katere je bila datoteka izbrana, se shrani v spremenljivko `LOCALTIME` na PC platformi. Če ime časovne datoteke vsebuje GMT+ ali GMT- število, se spremeni predznak. Tako se shrani dejanska relativna pot datoteke. Na napravi s platformo CEP se dodatno shrani še spremenljivka `CEP_LOCALTIME` z vrednostjo `localtime`. Če med shranjevanjem pride do napake, se na novo ustvarjena `neo3000.conf` datoteka izbriše. Datoteka `neo3000.con~` pa se preimenuje nazaj v `neo3000.conf`. Napaka pa se zapiše v statično spremenljivko `errorDescription` razreda `Application`.

5.2 Shranjevanje časovne datoteke na napravo

V metodi `boolean Upload(Device device)` razreda `Device_SystemSettings_TimeZone`, se prenese izbrana časovna datoteka na napravo. `/NEO3000/etc/localtime` je pot na katero na napravi shranimo časovno datoteko. Na začetku metode preverimo ali obstaja predstavitevni niz izbrane časovne datoteke. Ob uspehu reprezentaciji izbrane časovne datoteke zamenjamo poševnico s separatorjem značilnim za uporabnikov sistem. Nato spremenimo ime izbrane datoteke. Na primer iz GMT offset/GMT dobimo Etc/GMT, ki predstavlja relativno pot do izbrane datoteke. GMT datotekam spremenimo predznak, če datoteka le tega vsebuje, saj so zaradi jasnosti znaki zamenjani, da dejansko predstavljajo odmik od GMT. Prav tako spremenimo String reprezentacijo `Others/<datoteka>` v `<datoteka>`, ki predstavlja relativno pot

od glavnega zoneinfo direktorija naprej. Datotekam, ki spadajo v skupino `Countries` ni potrebno spreminjati ničesar, saj imena že predstavljajo relativno pot.

Ustvarimo objekt `saveFile`, ki kaže na izbrano datoteko, od njega izvemo tudi velikost datoteke, katero uporabimo za inicializiranje tabele `byte[] content`. V `content` tabelo s pomočjo toka `FileInputStream fis` in njegove metode `read(content)` preberemo vsebino datoteke.

Ob napaki sporočilo napake zapišemo v lokalno spremenljivko `String errorDescription`.

Na koncu pokličemo metodo `FileSystem_Upload(byte[] content, String filePath, boolean createBackup)` objekta `iServer`, ki vrne `true`, če je prenos datoteke uspel in `false` ob neuspehu. Če je spremenljivka `errorDescription` enaka `null` in je metoda `FileSystem_Upload` uspela, vrnemo boolean vrednost `true`.

Če preverjanje `String` reprezentacija izbrane časovne datoteke ne uspe, se preveri kakšnega tipa je spremenljivka `window`. Če je `window` tipa `GuiNoTimeZoneFolder` ali če je `timeZone`, ki je spremenljivka objekta `SubStation`, enaka `null` ali pa je predstavljena kot prazen string, se vrne boolean vrednost `true`, saj ni prišlo do napake med izvajanjem. V tem bloku kode se ne prenese nobena datoteka, saj uporabnik niti nima na voljo izbire časovne datoteke, oz. ne vidi polja z izbrano časovno datoteko ali pa ni nastavil nobene časovne datoteke.

V primeru, da noben od zgornjih pogojev ne uspe, metoda vrne vrednost `false`.

6 SKLEP

V diplomski nalogi smo poskušali bralcu približati časovne datoteke, ki jih uporabljajo Unix sistemi. Predstavili smo mu sestavo zoneinfo direktorija in vrste izvornih datotek. Na kratko smo opisali zic program in njegova stikala. Napisali smo tudi vzrok za nastanek časovnih datotek.

Nadaljevali smo z opisom PSM sistema, v katerega smo vgradili grafični vmesnik za izbiro časovne datoteke. Z vgradnjo smo imeli kar nekaj težav, saj smo si izbrali pot, ki se je razlikovala od sedaj ustaljenih poti. Je bila pa to poučna izkušnja, ki nam je omogočila, da malo bolje razumemo sistem, ki ga razvijamo.

Na koncu smo podrobno opisali spremembe, ki smo jih naredil na posameznih izvornih datotekah. Opisali smo tudi na novo ustvarjene objekte in njihovo uporabo.

Namen tega dela je, da uporabniku omogočimo grafični vmesnik, s katerim bo lahko učinkovito parametriral postaje ali naprave. Razlog, zakaj je bilo treba vključiti časovne datoteke tiči predvsem v tem, da so Iskra Sistemi globalno podjetje in ima kupce z različnih delov sveta. Ko pride do napake, bi bilo zelo neprijetno, če bi naprava zabeležila napako z napačnim časovnim žigom. Namen vsakega grafičnega vmesnika je, da uporabniku olajša upravljanje z napravo. Veliko lažje je spremeniti časovno datoteko preko sistema PSM, kot če bi to isto stvar počeli v terminalu in ročno prepisovali datoteko.

V PSM sistemu je sprememba časovne datoteke rešena s tremi kliki. Ne moremo pa si predstavljati, kako bi to počeli v terminalu, saj bi si uporabnik moral najprej priskrbeti prave časovne datoteke. Nato pa bi moral prenesti datoteko na pravo mesto, da ne govorimo, da bi moral imeti tudi ustrezne pravice za delo z napravo.

V diplomski nalogi smo pokazali, da ustvarjeni grafični vmesnik sovпада z že narejenim sistemom. Grafični vmesnik, prenos in zapis časovnih datoteke deluje in je skladen z ostalimi funkcionalnostmi aplikacije. Prestal je tudi interna testiranja.

Sam grafični vmesnik bi bilo mogoče precej izboljšati. Mogoče dodati zemljevid sveta in uporabniku omogočiti oznako zelene lokacije na tem zemljevidu. Lahko bi imel vnosno polje, kjer bi uporabnik vpisal želeno lokacijo. Pri teh rešitvah bi se pojavilo veliko novega dela, za kar pa žal ni časa, prav tako ni bilo to predvideno v diplomski nalogi.

Tu je še vprašanje uporabnikovega upravljanja s sistemskim direktorijem PSM sistema. Problem lahko nastane, če uporabnik pobriše datoteke ali pa cel direktorij zoneinfo. Problem je predvsem v tem, da aplikacija ne preverja ali se je direktorij spremenil. Tako lahko uporabnik izbere časovno datoteko, pobriše zoneinfo direktorij in potem zahteva od aplikacije, da naloži datoteko. Bolj napreden uporabnik pa lahko celo zapolni disk naprave s preveliko datoteko, ki se predstavlja kot časovna datoteka. Še lažje bi zamenjal imeni 2 ali več datotek in tako povsem zmešal časovne pasove.

Te stvari kažejo, da sistem PSM še ni v celoti izdelan in se lahko zahteve zanj spremenijo.

KAZALO SLIK

Slika 1: Časovni pasovi v celinski ZDA	10
Slika 2: Prikaz Posix notacije	11
Slika 3: Struktura postaje	20
Slika 4: Primer povezovanja različnih modulov naprave	24
Slika 5: Prikaz diagnostike naprave	27
Slika 6: Prenos datotek na napravo	30
Slika 7: Prikaz relacij objektov	33
Slika 8: Prikaz grafičnega vmesnika za izbiro časovne datoteke	36
Slika 9: Prikaz lastnosti, ko aplikacija ne vsebuje zoneinfo direktorija	37
Slika 10: Prikaz grafičnega vmesnika, če uporabnik izbriše datoteke in mape v zoneinfo	39

LITERATURA

- [1] B. Grandovec, I. Kobal, P. Kastelic, Orodje za konfiguracijo posameznih naprav v sistemu in izdelavo komunikacijskega modela, Ljubljana 2011
- [2] (2011) iServer. Dostopno na (interna internetna stran):
<http://neopedia.zve.iskrasistemi.si/index.php/IServer>
- [3] (2011) izvorna koda časovnih datotek. Dostopno na:
<ftp://elsie.nci.nih.gov/pub/tzdata2011g.tar.gz>
- [4] (2011) Iskra Sistemi. Dostopno na:
<http://www.iskrasistemi.si/>
- [5] (2011) FreeBSD Man Pages. Dostopno na:
<http://www.gsp.com/cgi-bin/man.cgi?section=8&topic=zic>
- [6] (2011) Time zone. Dostopno na:
<http://en.wikipedia.org/wiki/Timezone>
- [7] (2011) Solar time. Dostopno na:
http://en.wikipedia.org/wiki/Solar_time
- [8] (2011) Nautical time. Dostopno na:
http://en.wikipedia.org/wiki/Nautical_time
- [9] (2011) Extreme points of the United States. Dostopno na:
http://en.wikipedia.org/wiki/Extreme_points_of_the_United_States
- [10] (2011) tzset. Dostopno na:
<http://www.gsp.com/cgi-bin/man.cgi?section=3&topic=tzset>
- [11] (2011) UTC. Dostopno na:
http://en.wikipedia.org/wiki/Coordinated_Universal_Time
- [12] (2011) TAI. Dostopno na:
http://en.wikipedia.org/wiki/International_Atomic_Time