

UNIVERZA V LJUBLJANI
FAKULTETA ZA ELEKTROTEHNIKO

MATJAŽ KAVČIČ

**Upravljanje pametnih merilnih naprav po
priporočilu NTA 8130**

UNIVERZITETNO DIPLOMSKO DELO

Mentor: doc. dr. Boštjan Murovec

Ljubljana, marec 2010

Zahvala

Najprej se zahvaljujem družini za podporo in spodbude v času celotnega študija. Zahvaljujem se tudi prijateljem, kolegom in sošolcem, ki so mi pomagali in stali ob strani ter poskrbeli da je bil študij bolj zanimiv. Še posebej se zahvaljujem mentorju, doc. dr. Boštjanu Murovcu, ki me je usmerjal pri izdelavi diplomske naloge in pomagal z nasveti in pristopi k reševanju. Prav tako se zahvaljujem vsem sodelavcem v Iskraemecu za pomoč ter spodbudo, še posebno pa mentorju Aleksandru Remarju. Zahvala gre tudi Alešu Glavini za pomoč pri izdelavi zapletenih grafičnih kontrol ter Juretu Germovšku za številne uporabne strokovne nasvete.

Kazalo

1	UVOD	1
2	NIZOZEMSKI STANDARD ZA PAMETNO MERJENJE NTA 8130.....	3
2.1	SPLOŠNO	3
2.1.1	Aplikacije uporabe sistemov pametnih merilnih naprav	3
2.2	NAMEN IN PODROČJE	4
2.3	OPIS RAZLIČNIH FUNKCIJ IN SPECIFIKACIJA ZAHTEV	4
2.3.1	Splošno	4
2.3.2	Podatkovne zahteve	5
2.3.3	Nadzorni ukazi	8
2.3.4	Ukazi za parametriranje.....	9
2.3.5	Komunikacija	9
2.3.6	Predplačniška podpora.....	15
2.3.7	Aplikacije različnih funkcij	16
3	ZAHTEVE ZA NIZOZEMSKES PAMETNE NAPRAVE.....	17
4	SPREMLJEVALNI STANDARD ZA MERJENJE ENERGIJE.....	21
4.1	PODROČJE	21
4.2	NAMEN	22
4.3	VMESNIŠKI RAZREDI COSEM	23
4.3.1	Osnovni principi	23
4.3.2	Podatkovni tipi	24
4.3.3	Pregled.....	26
4.3.1	Primeri	26
4.4	SISTEM IDENTIFIKACIJE OBJEKTOV - OBIS	29
4.4.1	Struktura kode	29
5	ANALIZA ZAHTEV	33
5.1	METERVIEW 4.....	33
5.2	SEP2W SYSTEM	34
5.3	IZBIRA PLATFORME	36
6	NAČRTOVANJE ARHITEKTURE	39
6.1	VTIČNIK ZA SEP2 MANAGER	39
6.2	BAZNI GONILNIK.....	41

6.2.1	Komunikacija.....	42
6.3	MODELIRANJE VMESNIŠKIH RAZREDOV COSEM	43
6.3.1	Podatkovni model	44
6.3.2	Sledenje sprememb	48
6.4	PRIKAZ OBJEKTOV COSEM	49
6.4.1	Grafične kontrole	49
6.4.2	Sistem pogledov.....	50
6.4.3	Iskanje.....	51
6.5	GENERATOR IMEN ZA STANDARDNE KODE OBIS	54
6.6	NAČRT INSTANC.....	56
6.7	TRANSFORMACIJE.....	58
6.7.1	Splošno	58
6.7.2	Enumeracije	59
6.7.3	Bitne maske	60
6.7.4	Octet-string kot ASCII.....	60
6.7.5	Octet-string kot datum in čas	61
6.8	BELEŽENJE DOGODKOV.....	61
6.9	SAMOSTOJNA APLIKACIJA	62
7	ZAKLJUČEK	65
	LITERATURA.....	67

Kazalo tabel

Tabela 1: indeksi za električno energijo	5
Tabela 2: indeksi za plin.....	6
Tabela 3: intervalni merilni rezultati za električno energijo.....	7
Tabela 4: intervalni merilni rezultati za plin	7
Tabela 5: komunikacijski profil vrat P1	10
Tabela 6: komunikacijski profil vrat P4	14
Tabela 7: aplikacije različnih funkcij	16
Tabela 8: podatkovni tipi COSEM	25
Tabela 9: specifikacija atributov in metod za vmesniški razred podatki.....	27
Tabela 10: specifikacija atributov in metod za vmesniški razred nastavitve varnosti.....	27
Tabela 11: vrednosti za skupino A	30
Tabela 12: nekaj vrednosti skupine C za elektriko.....	31
Tabela 13: nekaj vrednosti skupine D za elektriko.....	32

Kazalo slik

Slika 1: komunikacijska vrata merilnega mesta	5
Slika 2: DLMS/COSEM infrastruktura P3.....	18
Slika 3: komunikacijska vrata merilnega mesta z dodatnimi vrati P0.....	19
Slika 4: pristop DLMS/COSEM v treh korakih: modeliranje – sporočanje – prenos	21
Slika 5: vmesniški razred register in njegove instance.....	24
Slika 6: pregled vseh vmesniških razredov COSEM.....	26
Slika 7: struktura kode OBIS	29
Slika 8: razporeditev moči po kvadrantih.....	31
Slika 9: parametriranje naprave DLMS/COSEM z uporabo MeterView 4.....	34
Slika 10: komponente SEP2W System.....	35
Slika 11: izgled vtičnikov v aplikaciji SEP2 Manager	37
Slika 12: razred vtičnika in nekatere njegove lastnosti	39
Slika 13: drevesna struktura vtičnika	40
Slika 14: vmesnik vozlišča gonilnika naprave	41
Slika 15: nekatere lastnosti baznega gonilnika.....	42

Slika 16: lastnosti in metode baznega gonilnika za komunikacijo	43
Slika 17: abstraktni osnovni razred za vmesniške razrede COSEM.....	44
Slika 18: podatkovni model atributov vmesniških razredov.....	45
Slika 19: razredni diagram za preproste COSEM tipe.....	46
Slika 20: razredni diagram za zapletene tipe COSEM.....	48
Slika 21: vmesnik INotifyIsDirty.....	49
Slika 22: vmesnik ICosemSimpleDataTypes.....	49
Slika 23: kontrola za prikaz COSEM datuma in časa.....	50
Slika 24: pogled vmesniškega razreda z izdelanimi kontrolami.....	50
Slika 25: pogled vmesniškega razreda v surovi obliki.....	51
Slika 26: iskanje po delu logičnega imena instance objekta.....	52
Slika 27: razred kontrole za prikazovanje COSEM objektov naprave	53
Slika 28: kontrola za prikazovanje COSEM objektov naprave	54
Slika 29: generator imen in razredi za delo s kodami OBIS	55
Slika 30: testno orodje za generator imen.....	55
Slika 31: XSD shema načrta instanc	57
Slika 32: proces pridobivanja imena in skupine za COSEM objekt.....	58
Slika 33: del XSD sheme načrta instanc za transformacije	59
Slika 34: primer transformacije tipa enumeracija	60
Slika 35: primer transformacije tipa bitna maska za NTA alarme.....	60
Slika 36: primer transformacije tipa octet-string kot ASCII.....	61
Slika 37: primer transformacije tipa octet-string kot datum in čas	61
Slika 38: beleženje dogodkov gonilnika DLMS/COSEM	62
Slika 39: prikaz uporabe programske opreme izdelane v diplomski nalogi	63

Seznam akronimov in izrazov

AMR	(ang.: Automatic Meter Reading) avtomatsko branje merilnih naprav
APDU	(ang.: Application Protocol Data Unit) aplikacijska protokolna podatkovna enota
ASCII	(ang.: American Standard Code for Information Interchange) ameriški standardni nabor za izmenjavo informacij
CAS	(ang.: Central Access Server) centralni strežnik
COSEM	(ang.: COmpanion Specification for Energy Metering) spremljevalni standard za merjenje energije
DC	(ang.: Data Collector) zbiralec podatkov
DLL	(ang.: Dynamic Link Library) dinamično naložena knjižnica
DLMS	(ang.: Device Language Message Specification) specifikacija jezikovnih sporočil za naprave, generaliziran koncept za abstraktno modeliranje komunikacijskih konceptov
DSMR	(ang.: Dutch Smart Meter Requirements) zahteve za nizozemske pametne merilne naprave
ENBIN	(niz.: De Vereniging Energienetbeheerders Nederland) nizozemsko združenje upravljavcev energetske omrežij
IEC	(ang.: International Electrotechnical Commission) mednarodna elektrotehniška komisija
ISO	(ang.: International organization for standardization) mednarodna organizacija za standardizacijo
NEN	(niz.: Nederlands Normalisatie-instituut) nizozemski inštitut za standardizacijo
OBIS	(ang.: OBject Identification System) sistem za identifikacijo objektov
P1	NTA kanal za lokalno komunikacijo
P2	NTA kanal za komunikacijo med merilnim sistemom in merilnimi instrumenti in/ali opremo operaterja
P3	NTA kanal za povezavo med merilno instalacijo in centralnim strežnikom (CAS)
P4	NTA kanal za povezavo med centralnim strežnikom (CAS) in različnimi tržnimi ponudniki
XML	(ang.: Extensible markup language) razširljiv opisni jezik

XSD (ang.: Xml Schema Definition) definicija xml sheme

Povzetek

Potrebe po energiji so iz dneva v dan večje, pri čemer energijska infrastruktura večinoma temelji na fosilnih gorivih, ki niso obnovljivi in zelo slabo vplivajo na okolje. Da bi zmanjšali porabo energije, so na Nizozemskem oblikovali specifikacije za pametne merilne sisteme in jih zapisali v dokument NTA 8130. Ta vsebuje minimalen osnovni nabor funkcij, ki jih morajo izpolnjevati pametne merilne naprave na Nizozemskem, v spremljevalnih dokumentih pa so podane tudi zahteve v zvezi z namestitvijo, vzdrževanjem in komunikacijskimi protokoli.

Za pomoč pri namestitvi ter za nadzor in upravljanje pametnih merilnih naprav, smo v diplomski nalogi izdelali programsko opremo, ki omogoča lokalno in oddaljeno komunikacijo z napravami ter grafično predstavitev in urejanje parametrov merilne naprave. Glavni del predstavlja razvoj gonilnika za aplikacijski nivo odprtega protokola DLMS/COSEM z implementacijo posebnosti po zahtevah za nizozemske pametne merilne naprave.

Rezultat diplomskega dela je programska oprema, ki jo pri namestitvah pametnih merilnih sistemov že uporabljajo v različnih nizozemskih podjetjih.

Ključne besede:

Pametni merilni sistemi, NTA 8130, protokol DLMS/COSEM, objektno programiranje, ogrodje .NET

Abstract

Energy is a key factor in sustained development of modern society. While energy demands increase, our energy infrastructure continues to rely mainly on fossil fuels. This unbalance between modern energy demands is now making itself felt by reaching the physical limits of our environment.

In order to reduce the use of energy, the Nederland Ministry of Economic Affairs has commissioned Nederland standardizations institute (NEN) to formulate and describe specifications for smart metering systems, which are formalized in document NTA 8130. It contains a minimum set of functions for metering of electricity, gas and thermal energy for domestic consumers. Companion documents contain additional requirements, mainly with respect to installation, maintenance, quality, performance, and aim at meter interoperability.

In order to assist installation and to control and manage smart metering systems, we developed software that enables local and remote communication with smart meters and allows graphic presentation and editing of parameters. Main part represents development of driver for application layer of open DLMS/COSEM protocol with implementation of special requirements for Nederland smart meters.

The result of this diploma thesis is software, which is being used for installations of smart metering systems by Nederland largest grid operators, suppliers and independent service providers.

Key words:

Smart metering systems, NTA 8130, DLMS/COSEM protocol, object oriented programming, .NET framework

1 Uvod

Energija je ključni faktor za razvoj moderne družbe. Zahteve po energiji se močno povečujejo, energijska infrastruktura pa večinoma temelji na fosilnih gorivih. Posledično bomo kmalu dosegli fizične omejitve zemljine atmosfere. Za razvoj okolju prijaznih obnovljivih virov energije in njihovo masovno uporabo je potreben čas. Zato je optimizacija uporabe fosilnih goriv samoumevna strategija za zmanjšanje emisij toplogrednih plinov v atmosfero. To pa zahteva energijsko ozaveščenost in kvalitetne informacije o porabi energije na vseh nivojih družbe, od vlade do posameznih prebivalcev.

Najprej moramo razumeti, kako porabljamo energijo, da bi lahko zmanjšali njeno porabo. Za boljše razumevanje pa je potrebno kvalitetno merjenje in sprotno zagotavljanje izmerjenih vrednosti vsem porabnikom. V skladu s tem se razvijajo tudi merilne naprave, ki se s pomočjo modernih tehnologij povezujejo v omrežja. Tako nastaja infrastruktura pametnih merilnih sistemov, ki so upravljani iz centralnih strežnikov. Ti preko omrežja komunicirajo z merilnimi napravami in pridobivajo podatke o porabi, ki so tako na voljo vsem zainteresiranim udeležencem.

Osnovni gradniki pametnih merilnih sistemov so pametne merilne naprave. Te v splošnem merijo porabo energije, podatki o porabi pa so preko pametnih omrežij na voljo centralnim strežnikom. Poleg tega imajo pametne merilne naprave še veliko drugih funkcij, ki pa se zelo razlikujejo po posameznih državah.

Nizozemska vlada je zato naložila inštitutu za standardizacijo (NEN), naj oblikuje *»standardiziran minimalni nabor funkcij za merjenje električne in termalne energije ter plina za gospodinjske uporabnike«* [1, stran 2]. Tako je nastal dokument NTA 8130 [1], ki vsebuje minimalni nabor funkcij, na podlagi tega pa še različni spremljevalni dokumenti [2-5], ki še podrobneje opisujejo zahteve za nizozemske pametne merilne naprave s poudarkom na interoperabilnosti sistemov različnih operaterjev. Glavni dokument [2] vsebuje poslovne primere uporabe, ki jim morajo zadostiti pametni merilni sistemi, ostali pa so spremljevalni standardi, ki opisujejo oz. dopolnjujejo uporabljene protokole.

V diplomski nalogi smo izdelali programsko opremo za upravljanje in nadzor pametnih merilnih naprav, izdelanih po zahtevah NTA 8130 ter zahtevah za nizozemske pametne merilne naprave, ki omogoča tudi izvajanje in prikaz poslovnih primerov uporabe iz [2].

V prvih treh poglavjih takoj za uvodnim predstavimo zahteve za nizozemske pametne merilne sisteme iz dokumenta NTA 8130 in dokumentov DSMR. Opišemo tudi protokol DLMS/COSEM, ki je uporabljen za komunikacijo z napravami ter sistem OBIS za identifikacijo objektov.

V petem poglavju analiziramo zahteve ter pregledamo obstoječo programsko opremo v podjetju Iskraemeco d.d. na področju parametriranja in daljinskega odčitavanja merilnih naprav.

V šestem poglavju opišemo arhitekturo in delovanje programske opreme. Pri razlagi si pomagamo z razrednimi diagrami, XSD shemami, diagrami potekov in s posnetki zaslona.

V zadnjem poglavju predstavimo rezultate in podamo možnosti za prihodnje izboljšave.

2 Nizozemski standard za pametno merjenje NTA 8130

Poglavje je povzeto po viru [1].

2.1 Splošno

Nizozemsko Ministrstvo za gospodarstvo je pooblastilo Nizozemski inštitut za standardizacijo (NEN), da oblikuje minimalni nabor osnovnih funkcij za merjenje in daljinsko odčitavanje električne energije, plina, termalne energije in vode za gospodinjске odjemalce. NEN je ustanovil projektno skupino, ki je osnovala zahteve, katerim morajo ustrezati pametni merilni sistemi. Med procesom so obvezen minimalni nabor funkcij omejili na električno energijo in plin, za vodo in termalno energijo pa so v informativnem dodatku podana priporočila. Proces je bil končan aprila 2007, kot rezultat pa je nastal nizozemski tehnični dogovor (NTA), ki vsebuje »*minimalni nabor funkcij za merjenje električne energije, plina in termalne energije za gospodinjске odjemalce*« [1, stran 2]. Referenčna oznaka dokumenta je NTA 8130.

2.1.1 Aplikacije uporabe sistemov pametnih merilnih naprav

Aplikacije sistemov merilnih naprav so bile ugotovljene v razpravah, ki so vodile k funkcijam, zapisanih v dokumentu NTA 8130. Od sistemov pametnih merilnih naprav je pričakovano da bodo:

- prinesli prispevek k izboljšanju administrativnih procesov z dajanjem točnih merilnih rezultatov periodično in na zahtevo ter možnostjo branja na daljavo,
- omogočali dobavitelju izboljšati porabnikovo ozaveščenost o porabi energije in stimulirati varčevanje z energijo,
- zmožni množično ali posamezno daljinsko priklopiti ali odklopiti elektriko ali plin,
- zmožni množično ali posamezno spremeniti omejitev za električno energijo (npr. za situacije ko pride do pomanjkanja energije zaradi zmanjšane proizvodnje),
- omogočali dobavitelju energije in/ali vode da uporablja različne tarifne sisteme,
- omogočali plačilo pred dobavo energije (predplačilo),
- lahko uporabljeni za nadzor nad distribucijskim omrežjem.

Na osnovi teh predpostavk je projektna skupina naredila seznam potrebnih funkcij, ki naj bi jih bili pametni merilni sistemi sposobni izvršiti, da bi bile zahtevane aplikacije možne. Tako je 27. aprila 2007 nastala trenutno veljavna verzija NTA 8130: 2007.

2.2 Namen in področje

NTA 8130 se osredotoča na sporazume v zvezi dodajanjem minimalnih osnovnih funkcij merilnih mest za plin in električno energijo ter s povezanimi potrebnimi funkcijami za komunikacijo, preklapljanje in nastavljanje omejitev. Bolj preprosto povedano je to minimalni nabor osnovnih funkcij za povezovanje porabnika v energijsko distribucijsko infrastrukturo na Nizozemskem.

Področje uporabe so merilna mesta pri gospodinjskih odjemalcih na Nizozemskem, zadeva pa porabnike, dobavitelje, upravljavce omrežij, podjetja za upravljanje z merilnimi podatki in neodvisne ponudnike storitev, sistemov in opreme.

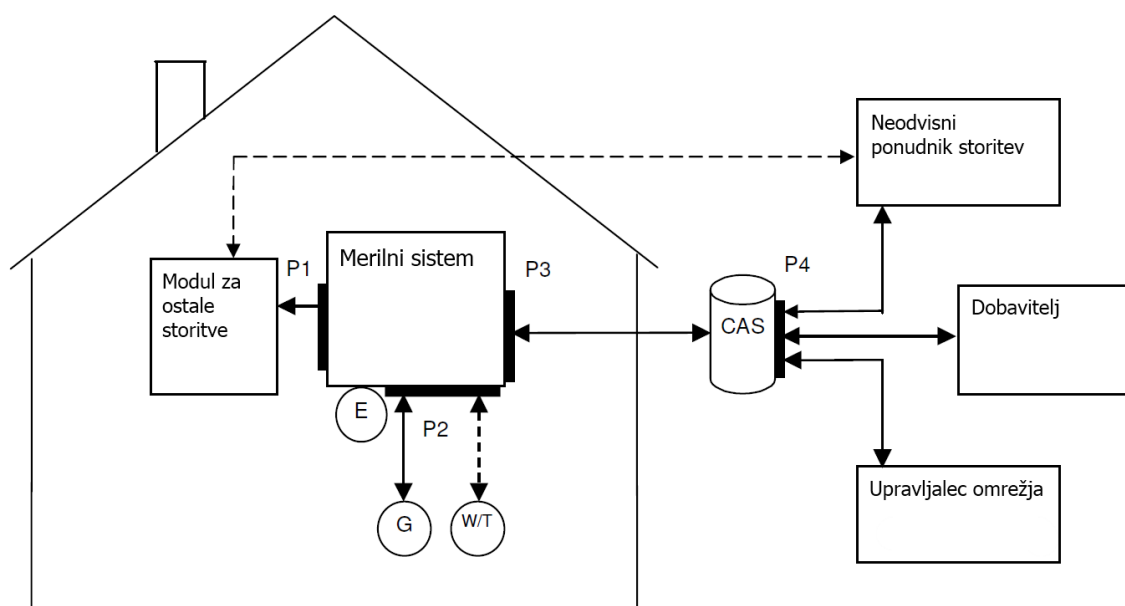
2.3 Opis različnih funkcij in specifikacija zahtev

2.3.1 Splošno

Poleg različnih prikazovalnikov kjer so na voljo merilni podatki ima merilno mesto še naslednja komunikacijska vrata:

- vrata P1 za komunikacijo med merilnim mestom in enim ali večim storitvenim modulom, na voljo samo za branje,
- vrata P2 za komunikacijo med merilnim sistemom in z do štirimi merilnimi inštrumenti in/ali opremo operaterja omrežja,
- vrata P3 za komunikacijo med merilnim mestom in centralnim strežnikom (CAS),
- vrata P4 so vrata na centralnem strežniku (CAS) s katerim lahko neodvisni ponudniki storitev, dobavitelji in operaterji omrežja dostopajo do podatkov ki jih ponuja CAS.

Slika 1 prikazuje komunikacijska vrata merilnega mesta:



Slika 1: komunikacijska vrata merilnega mesta

2.3.2 Podatkovne zahteve

Indeksiranje merilnih podatkov

Dnevni merilni podatki vsakega merilnega mesta so vsak dan shranjeni v dnevnem registru na merilnem mestu, vsak mesec pa so mesečni podatki shranjeni v mesečni register.

Število indeksov

Tabela 1 prikazuje indekse s katerimi je opremljeno merilno mesto za merjenje električne energije:

<i>opis indeksa</i>	<i>enota, prikazana pri komunikaciji preko vrat P3</i>
porabljena električna energija (višja tarifa)	kWh
porabljena električna energija (nižja tarifa)	kWh
dobavljena električna energija (višja tarifa)	kWh
dobavljena električna energija (nižja tarifa)	kWh

Tabela 1: indeksi za električno energijo

Registracija dobave električne energije je dodeljena določenemu indeksu za določeno obdobje (višja ali nižja tarifa). Preklop med indeksi je časovno odvisen. Za indikacijo tarife je uporabljen tarifni indikator.

Tabela 2 prikazuje indekse merilnega mesta za merjenje porabe plina:

<i>opis indeksa</i>	<i>enota prikazana pri komunikaciji preko vrat P3</i>
poraba plina	m ³

Tabela 2: indeksi za plin

Trenutni merilni rezultati na zahtevo preko vrat P3

Merilno mesto je sposobno poslati zadnje znane merilne rezultate na zahtevo preko vrat P3. Za električno energijo to pomeni dejanske merilne rezultate v trenutku ko prispe zahteva. Za plin pa to pomeni zadnje merilne rezultate, ki so na voljo brez dodatne komunikacije. Ti so lahko stari največ 24 ur.

Trenutni merilni rezultati preko vrat P1

- preko vrat P1 so na vsake 10 s na voljo samo trenutni merilni rezultati za električno energijo,
- merilni rezultati za plin skupaj s časovno značko so ponujeni vsaj enkrat na 24 ur,
- podobno kot pri merilnih rezultatih, je tudi trenutna moč poslana z resolucijo 0,01 kW preko vrat P1 vsakih 10 s. Zgodovina o tem se na merilnem mestu ne hrani.

Intervalni merilni rezultati

Tabela 3 prikazuje intervalne merilne rezultate skupaj s časovno značko, ki so na voljo preko vrat P3:

	<i>enota prikazana pri komunikaciji preko vrat P3</i>	<i>največja starost intervalnih merilnih rezultatov</i>	<i>intervalni čas</i>	<i>kapaciteta merilnega mesta</i>
porabljena električna energija	0,001 kWh	ni na voljo	15 min.	960 zapisov
dobavljena električna energija	0,001 kWh	ni na voljo	15 min.	960 zapisov

Tabela 3: intervalni merilni rezultati za električno energijo

Tabela 4 prikazuje intervalne merilne rezultate skupaj s časovno značko za merjenje porabe plina, ki so na voljo preko vrat P1 in P3:

	<i>enota prikazana pri komunikaciji preko vrat P3</i>	<i>največja starost intervalnih merilnih rezultatov</i>	<i>intervalni čas</i>	<i>kapaciteta merilnega mesta</i>
porabljen plin	0,001 m ³	24 ur	1 ura	240 zapisov

Tabela 4: intervalni merilni rezultati za plin

Informacije o stanju

Informacije o stanju ponujajo dodatne informacije o stanju merilnega mesta. Na voljo so preko vrat P1 in na zahtevo preko vrat P3. Dodatno ima operater omrežja možnost prikaza informacij stanja na prikazovalniku merilne naprave s pomočjo standardnih sporočil.

Preko vrat P1 so vsakih 10 sekund na voljo naslednje informacije:

- indikator trenutne tarife (višja/nížja) za električno energijo,
- trenutna pozicija odklopnika za električno energijo,
- trenutno nastavljena omejitev (največja možna poraba toka v A),
- trenutna pozicija ventila za plin.

Ob spremembi pozicije ventila za plin je informacija takoj na voljo preko vrat P1.

Na zahtevo lahko merilno mesto pošlje zadnje stanje preko vrat P3. V statusu se nahajajo enake informacije kot vsakih 10 s preko vrat P1. Razlika je v tem, da so vrednosti stanja vzete ob času prejetja zahteve, za pozicijo ventila za plin pa zadnje znano stanje na voljo brez dodatne komunikacije, ki pa ni staro več kot 24 ur.

Ob spremembi stanja odklopnika ali pozicije ventila za plin je novo stanje avtomatično javljeno preko vrat P3.

Spremljanje

Spremljanje poda informacije o kvaliteti dobavljene električne energije in samem stanju merilnega mesta. Na voljo je na zahtevo preko vrat P3.

Na voljo so:

- števec napetostnih vzponov in padcev (ang.: voltage swells and sags): vzponi in padci napetosti so posneti z uporabo števca,
- števec kratkih izpadov: število izpadov krajših od določenega časa,
- beleženje dolgih izpadov: izpadi daljši kot T so zabeleženi, merilno mesto lahko shrani najmanj zadnjih 10 izpadov. Zabeležita se začetek in konec izpada s časovno značko. V primeru 3-faznega merilnega mesta se zabeležijo tudi izpadi po posameznih fazah,
- sporočila z napakami: vsaj zadnjih 10 sporočil z napakami je shranjenih na merilnem mestu,
- zaznavanje prevar (ang.: fraud detection): ob zaznanju poskusov poškodovanja merilnega mesta ali odprtja pokrova je poskus zabeležen skupaj s časovno značko. Na merilnem mestu je shranjenih najmanj 10 zadnjih poskusov prevar. Poskusi prevar z uporabo zunanjih magnetnih polj se beležijo, če je merilno mesto občutljivo nanje. Prav tako se zabeleži zadnjih 10 poskusov prevar.

2.3.3 Nadzorni ukazi

Priklop/odklop in nastavitev omejitve

Priklop in odklop se za električno energijo izvaja z oddaljeno upravljanim odklopnikom. Lokalno z uporabo odklopnika ni možno izvesti odklopa.

Omejitev za električno energijo lahko nastavimo oddaljeno preko vrat P3. Odklopnik odklopi ko električni tok preseže nastavljeno vrednost in je presežena več kot 30 sekund. Ko odklopnik odklopi zaradi presežene omejitve ga lahko ročno prestavimo v priklopljeno stanje.

Ventil za plin ima tri možna stanja: odprt, zaprt in pripravljen za priklop. Odklop in pripravljenost za priklop lahko izvedemo iz oddaljene lokacije. Ponoven priklop zaradi varnosti lahko izvedemo samo lokalno, razen če je oddaljen priklop lahko izvršen varno.

Vsaj 10 zadnjih nadzornih ukazov s časovno značko je zabeleženih na merilnem mestu.

Informacije

Operaterji omrežij in dobavitelji lahko preko vrat P3 pošljejo standardna sporočila v zvezi z dobavo energije merilnemu mestu. Ta sporočila so prikazana na prikazovalniku merilnega mesta, ponujena pa so tudi preko vrat P1.

Pošiljajo se lahko sporočila, povezana z razlogi za odklop, nastavljenimi omejitvami ter opozorila za nizek kredit (predplačniški način).

Za operaterje omrežja, dobavitelje in neodvisne ponudnike storitev je na voljo pošiljanje poljubnih sporočil merilnemu mestu. Sporočila so brez sprememb posredovana na vrata P1.

2.3.4 Ukazi za parametriranje

Nastavljanje tarifnih preklpov

Registracija dobave električne energije je dodeljena določenemu indeksu za določeno časovno obdobje (tarifo). Preklop tarif je časovno odvisen, merilno mesto pa omogoča nastavitve preklpov z oddaljene lokacije preko vrat P3.

Posodobitve strojne programske opreme

Merilno mesto omogoča spremembo strojne programske opreme (ang.: firmware) preko vrat P3. Posodobitev metroloških lastnosti ni dovoljena.

2.3.5 Komunikacija

Prenos podatkov preko vrat P1

Vsaki 10 sekund je na vratih P1 na voljo:

- trenutni merilni podatki o porabljeni in proizvedeni električni energiji (višja/nížja tarifa),
- trenutna električna moč z resolucijo 10 W,
- indikator tarife (višja/nížja),
- nastavitev omejitve v A,
- stanje odklopnika.

Vsaj enkrat na 24 ur je na vratih P1 na voljo:

- količina dobavljenega plina za prejšnjih 24 ur, na vsako uro ena vrednost skupaj s časovno značko,
- pozicija ventila za plin.

Preko vrat P1 je možno omogočiti še naslednja sporočila:

- opozorilo zaradi prenizkega stanja predplačniškega kredita,
- razlog za izključitev,
- poljubno sporočilo dolžine največ 1024 znakov.

Fizična izvedba, protokoli in objekti vrat P1

Tabela 5 podaja podrobnosti komunikacijskega profila vrat P1:

<i>komunikacijska vrata</i>	<i>nivo</i>	<i>protokoli</i>	<i>zaznamki</i>
P1	fizični	RJ11 električno implementiran optični sklopnik	TTL nivo
	protokol	NEN-EN 62055-21 način D, samo za branje	če je aktiven signal za zahtevo, merilno mesto pošlje vse objekte vsakih 10 sekund
	objektni	OBIS (elektrika: NEN-EN-IEC 62056-61, plin: NEN-EN 13757-1)	

Tabela 5: komunikacijski profil vrat P1

Merilno mesto vsebuje ena vrata P1, ki jih lahko uporabnik svobodno uporablja. Delovanje merilnega mesta ne sme biti odvisno od normalne ali nenormalne uporabe vrat P1.

Prenos podatkov preko vrat P2

Vrata P2 so namenjena komunikaciji med merilnim sistemom in merilnimi inštrumenti ter drugo opremo operaterja omrežja.

Vsaj enkrat na 24 ur preko vrat P2 merilnik porabe plina pošlje urne vrednosti prejšnjih 24 ur za:

- merilne podatke količine porabe pline skupaj s časovno značko,
- pozicijo ventila.

Fizična izvedba, protokoli in objekti vrat P2

Za fizično izvedbo vrat P2 so na voljo naslednje možnosti:

- M-BUS v skladu z NEN-EN 13757-2,
- brezžični M-BUS v skladu z NEN-EN 13757-4.

Izvedba M-BUS protokola je svobodna, vendar mora biti zagotovljena interoperabilnost.

Prenos podatkov preko vrat P3

Preko vrat P3 so na CAS lahko vsakih 24 ur poslane naslednje informacije:

- merilni podatki o porabljeni in proizvedeni električni energiji (višja/nížja tarifa),
- merilni podatki porabe plina skupaj z indikatorjem temperaturne korekcije če je uporabljena,
- intervalni merilni podatki o porabljeni in proizvedeni električni energiji,
- intervalni merilni podatki porabe plina skupaj z indikatorjem temperaturne korekcije če je uporabljena,
- podatki o spremljanju.

Na zahtevo lahko merilno mesto pošlje preko vrat P3 na centralni strežnik naslednje podatke:

- trenutni merilni podatki o porabljeni in proizvedeni električni energiji (višja/nížja tarifa),
- največ 24 ur stare podatke o porabi plina skupaj z indikatorjem temperaturne korekcije če je uporabljena,

- zadnje dnevne merilne rezultate o porabljeni in proizvedeni električni energiji (višja/nížja tarifa),
- intervalni merilni podatki o porabljeni in proizvedeni električni energiji,
- intervalni merilni podatki porabe plina skupaj z indikatorjem temperaturne korekcije če je uporabljena,
- nastavitev preklonih časov za tarife,
- nastavitev omejitve v A,
- stanje odklopnika,
- pozicija ventila za plin,
- beležka o spremembah stanja odklopnika,
- podatki o spremljanju.

Preko vrat P3 lahko CAS pošlje merilnemu mestu naslednje:

- nastavitev omejitve v A,
- nastavitve časovnega preklopa tarif,
- nastavitev ventila za plin,
- opozorilo o nizkem stanju predplačniškega kredita,
- razlog za odklop,
- poljubno sporočilo dolžine največ 1024 znakov.

Fizična izvedba in protokoli vrat P3

Za fizično izvedbo vrat P3 so na voljo:

- PLC,
- GPRS,
- ethernet.

Protokol za komunikacijo med merilnim mestom in centralnim strežnikom zagotavlja interoperabilnost.

Prenos podatkov preko vrat P4

Preko vrat P4 je na voljo:

- trenutni merilni podatki o porabljeni in proizvedeni električni energiji (višja/nížja tarifa),

- največ 24 ur stare podatke o porabi plina skupaj z indikatorjem temperaturne korekcije če je uporabljena,
- zadnje dnevne merilne rezultate o porabljeni in proizvedeni električni energiji (višja/nížja tarifa),
- nastavitve preklonih časov za tarife,
- nastavitve omejitve v A,
- stanje odklopnika,
- pozicija ventila za plin,
- beležka o spremembah stanja odklopnika,
- podatki o spremljanju,
- intervalni merilni podatki o porabljeni in proizvedeni električni energiji,
- intervalni merilni podatki porabe plina skupaj z indikatorjem temperaturne korekcije če je uporabljena.

Preko vrat P4 lahko merilnemu mestu pošljemo:

- nastavitve omejitve v A,
- nastavitve časovnega preklopa tarif,
- nastavitve ventila za plin,
- opozorilo o nizkem stanju predplačniškega kredita,
- razlog za odklop,
- poljubno sporočilo dolžine največ 1024 znakov.

Fizična izvedba, protokoli in objekti vrat P4

Tabela 6 prikazuje fizično implementacijo vrat P4 po nivojih:

<i>komunikacijska vrata</i>	<i>nivo</i>	<i>protokoli</i>	<i>zaznamki</i>
P4	fizični	internet	podrobna odločitev o izvedbi je prepuščena operaterju omrežja, npr. TCP/IP
	protokol	XML spletne storitve	operater omrežja mora podati XSD specifikacijo
	objektni	OBIS (elektrika: NEN-EN-IEC 62056-61, plin: NEN-EN 13757-1) XSD ni definiran	

Tabela 6: komunikacijski profil vrat P4

Dostop in varnost

Merilno mesto je preko vrat P3 dostopno samo operaterju omrežja. V primeru da sta operaterja za električno energijo in plin različna, ima dostop samo operater, ki upravlja z električnim omrežjem.

Operater električnega omrežja je odgovoren za merilno mesto in pravilno podatkovno komunikacijo med merilnim mestom in centralnim strežnikom (CAS).

Operater električnega omrežja zagotovi pravilno izvedbo identifikacije, avtentikacije in avtorizacije v zvezi z merilnim mestom in šifriranje (ang.: encryption) podatkov med merilnim mestom in centralnim strežnikom (CAS), neodvisno od uporabljenega prenosnega medija.

Tržni akterji (dobavitelji, operater plinovoda in neodvisni ponudniki storitev) imajo dostop do merilnih podatkov in nadzornih ukazov preko vrat P4 centralnega strežnika (CAS).

Operater omrežja zagotovi tako identifikacijo, avtentikacijo in avtorizacijo tržnih akterjev na centralnem strežniku, da oseba s posrednim dostopom do merilnega mesta lahko uporablja samo funkcije za katere je pooblaščen.

2.3.6 Predplačniška podpora

Sistem opisan v dokumentu NTA 8130 je uporaben za podporo tarifnih struktur osnovanih na plačilu vnaprej (predplačilo). Osnovna predpostavka je, da je stanje kredita shranjeno centralno, prav tako pa je centralno upravljan odklopnik. To pomeni, da sistem ni t.i. avtonomna predplačniška merilna naprava.

Sledenje spremembam plačilne bilance je na centralnem nivoju podprto z dnevnim branjem registrov merilne naprave. Poleg tega lahko dobavitelj zahteva branje kadar misli da je to potrebno.

2.3.7 Aplikacije različnih funkcij

Tabela 7 [1, stran 20] prikazuje funkcije, ki so na voljo akterjem na določenih vratih. Oznake v oklepaju so originalne oznake poglavja v [1], ki podrobneje opisuje funkcijo:

<i>funkcija</i>	<i>vrata P3/P4</i>				<i>vrata P1</i>	<i>prikaz-ovalnik</i>
	operater	dobavitelj	neodvisni	ponudnik	povezana stranka	povezana stranka
dnevni merilni rezultati (5.2.1)	x	x	x			
mesečni merilni rezultati (5.2.1)	x	x	x			
trenutne vredn. za elektriko na zahtevo (5.2.4, 5.2.5)	x	x	x		x	x
trenutne vrednosti za plin na zahtevo (5.2.4, 5.2.5)	x	x	x		x	x
trenutna moč (5.2.5)					x	
intervalni rezultati preko množične zahteve (5.2.6)	x	x	x			
intervalni rezultati za elektriko na zahtevo (5.2.6.1)	x	x	x			
intervalni rezultati za plin na zahtevo (5.2.6.2)	x	x	x		x	
informacije o stanju (5.2.7)	x	x	x		x	
spremljanje (5.2.8)	x					
priklop/odklop in nastavljanje omejitve (5.3.1)	x	x				
beleženje nadzornih ukazov (5.3.1.3)	x	x				
pošiljanje standardnih sporočil (5.3.2.1)	x	x			x ¹	x ¹
pošiljanje sporočil (5.3.2.2)	x	x	x		x ¹	
nastavljanje tarifnih preklopov (5.4.1)	x	x			x ¹	
branje tarifnih preklopov (5.4.1)	x ¹	x ¹			x ¹	
indikator tarife (5.2.2)					x ¹	
posodobitev strojne programske opreme (5.4.2)	x ²					
dostop in varnost (5.5.5)	x ²					

Tabela 7: aplikacije različnih funkcij

¹ Viden je samo rezultat akcije

² Posodobitve strojne programske opreme ter dostop in varnost izvaja upravljavec električnega omrežja

3 Zahteve za nizozemske pametne naprave

V prejšnjem poglavju smo podali pregled minimalnega osnovnega nabora funkcij, ki jih morajo vsebovati nizozemski pametni merilni sistemi in so zapisani v dokumentu NTA 8130 [1]. V njem so zahteve napisane splošno, nekatere potrebne stvari pa tudi manjkajo. Zato ga je združenje nizozemskih operaterjev energetskih omrežij (ENBIN) izpopolnilo s ciljem interoperabilnosti med sistemi različnih operaterjev:

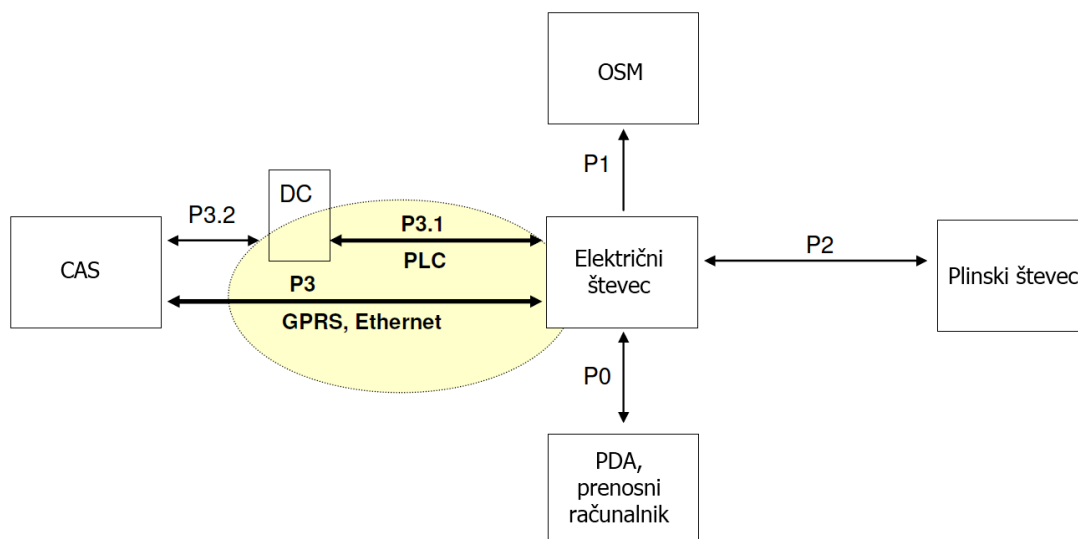
- dodali so nove funkcionalnosti, ki niso definirane v NTA 8130,
- definirali dodatne zahteve v zvezi z namestitvijo in vzdrževanjem,
- definirali kriterije za kvaliteto in zmogljivosti,
- definirali spremljevalne standarde za P1 [3], P2 [4] in P3 [5].

Spremljevalni standard za P1 vsebuje fizične karakteristike in definicije protokola za vmesnik. Je preprost, varen, jasno določen nizko-cenovni vmesnik, ki ščiti merilni sistem in podatke pred zlorabami.

Spremljevalni standard P2 se osredotoča na vmesnik za merilnike za plin, toploto ter vodo in je osnovan na M-BUS (EN 13757) standardu. Vsebuje samo razlike, razjasnitve in dodatke standardu.

Cilj spremljevalnega standarda P3 je odprt standard na osnovi odprtega DLMS/COSEM (IEC 62056) standarda za komunikacijo preko vrat P3 (med merilnim mestom in centralnim strežnikom z uporabo GRPS ali ethernet) in P3.1 (med merilnim mestom in centralnim strežnikom z uporabo PLC preko zbiralnika podatkov - DC). Vsebuje samo razlike, razjasnitve in dodatke obstoječim ustreznim standardom.

Slika 2 prikazuje DLMS/COSEM infrastrukturo vrat P3:



Slika 2: DLMS/COSEM infrastruktura P3

Kot že omenjeno je v NTA 8130 opisana minimalna zahtevana funkcionalnost pametnih merilnih naprav, v praksi pa so se pokazale še dodatne potrebe. Dodali so jih v obliki definicij poslovnih primerov uporabe (ang.: business use case), ki jih morajo podpirati pametne merilne naprave.

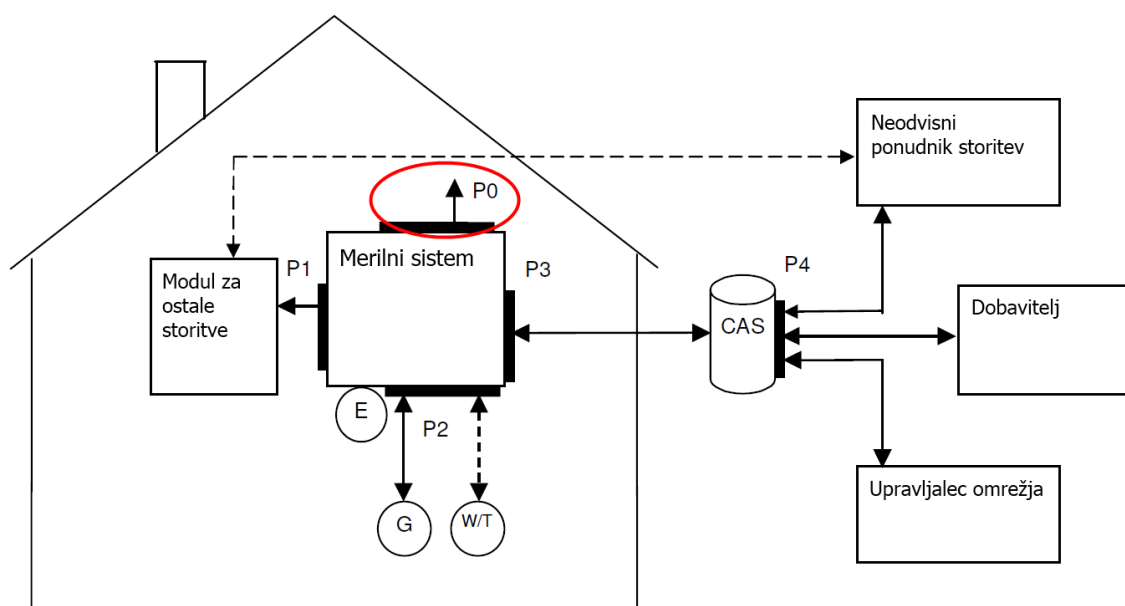
Primeri dodatnih funkcionalnosti so npr. merjenje povprečne napetosti, omejitev moči merilnega mesta, zahteve v zvezi s pričakovano življenjsko dobo, ...

Dodanih je bilo tudi nekaj kriterijev v zvezi s kvaliteto in učinkovitostjo. Ti kriteriji določajo največji dovoljeni čas za pridobitev podatkov in ostale odzivne čase merilnega sistema.

Dodali so tudi celoten nabor zahtev za namestitev in vzdrževanje. Namen teh zahtev je zagotoviti hitro, varno, brezhibno namestitev in nastavitev opreme ter omogočiti oddaljeno vzdrževanje. Podane so na način, da osebje, ki namešča in vzdržuje ne potrebuje posebnih usposabljanj. Vključujejo fizične karakteristike in funkcionalnost za konfiguracijo opreme.

Kot del zahtev so dodali tudi vrata P0, ki sicer niso del specifikacij NTA 8130. Namenjene so za potrebe lokalne komunikacije z zunanji napravami (PDA, prenosni računalnik, ...) pri namestitvi in vzdrževanju.

Slika 3 prikazuje celotno sliko merilnega sistema z vključenimi vrati P0.



Slika 3: komunikacijska vrata merilnega mesta z dodatnimi vrati P0

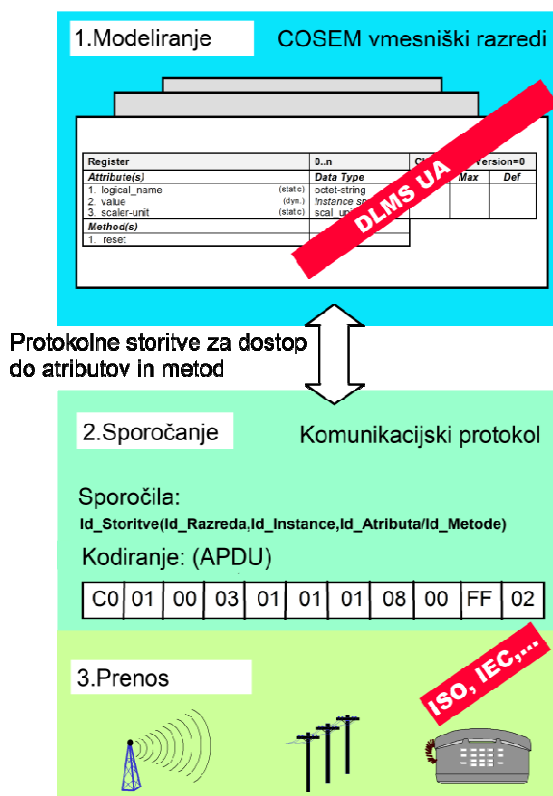
Zahteve za vzdrževanje se osredotočajo na omogočanje oddaljenega vzdrževanja. Oprema podpira oddaljeno vzdrževanje preko funkcionalnosti za:

- avtomatsko odkrivanje napak,
- zbiranje diagnostičnih podatkov,
- konfiguracijo merilnega mesta (kot celote in posameznih komponent),
- zbiranje podatkov o stanju merilnega mesta (parametrov).

4 Spremljevalni standard za merjenje energije

4.1 Področje

Specifikacija DLMS/COSEM opisuje podatkovni model in komunikacijske protokole za izmenjavo podatkov med merilno opremo. Slika 4 prikazuje zgradbo v treh korakih [6, stran 10]:



Slika 4: pristop DLMS/COSEM v treh korakih: modeliranje – sporočanje – prenos

Korak 1, modeliranje: Pokriva podatkovni model merilne opreme ter pravila za identifikacijo podatkov. Podatkovni model omogoča pregled funkcionalnosti merilne naprave, ki je na voljo na vmesniku.

Korak 2, sporočanje: Pokriva komunikacijske storitve in protokole za povezovanje elementov podatkovnega modela z aplikacijskimi protokolnimi podatkovnimi enotami (APDU).

Korak 3, prenos: Pokriva storitve in protokole za prenos sporočil po komunikacijskih kanalih

Korak 1 je opisan v modri knjigi [6] in vsebuje specifikacijo vmesniških razredov COSEM (ang.: COSEM interface classes), OBIS sistem identifikacije objektov in uporabo objektov za modeliranje raznovrstnih funkcij merilne opreme.

Koraka 2 in 3 sta opisana v zeleni knjigi [7]. V njej so opisani komunikacijski profili ter pripadajoči protokolni nivoji za raznovrstne komunikacijske prenosnike. Zgornji nivo v vsakem profilu je COSEM aplikacijski nivo, ki predpisuje storitve za vzpostavitev logičnih povezav med odjemalcem in strežnikom. Predpisuje tudi sporočilne storitve za dostop do atributov in metod vmesniških razredov COSEM. Spodnji nivo je odvisen od izbranega komunikacijskega profila in je namenjen prenosu podatkov.

4.2 Namen

Zaradi želja po optimizaciji poslovnih procesov postajajo merilne naprave vedno bolj integrirane v merilne in obračunske sisteme. V preteklosti je bila vrednost merilne naprave določena z možnostjo zbiranja podatkov in procesnimi sposobnostmi, danes pa so ključni problemi komunikacijske zmožnosti, sistemska integracija in interoperabilnost.

COSEM se na te izzive odziva z gledanjem na merilno napravo kot integriran del poslovnih procesov, ki se začnejo z merjenjem dobavljene energije in končajo z zbiranjem dohodkov.

Merilna naprava je določena z njenim obnašanjem kot je vidno s stališča poslovnih procesov. Formalna specifikacija obnašanja je osnovana na tehniki modeliranja objektov. Specifikacija teh objektov tvori večinski del specifikacije COSEM.

COSEM strežniški model predstavlja samo elemente merilne naprave, ki so vidni od zunaj. Uporabljajo ga odjemalske aplikacije, ki podpirajo poslovne procese. Merilne naprave ponujajo dostop do strukturnega modela (seznam objektov vidnih preko vmesnika) in ponujajo kontroliran dostop do atributov in metod teh objektov.

Proizvajalci lahko sestavijo (modelirajo) različne produkte s pomočjo standardizirane knjižnice različnih vmesniških razredov. Elementi so načrtovani tako, da pokrivajo celoten obseg produktov (gospodinjske, komercialne, industrijske in prenosne ter distribucijske aplikacije). Izbira vmesniških razredov uporabljenih v merilnih napravah, instanciranje in implementacija je del načrtovanja izdelka in je zato prepuščena proizvajalcem. Koncept standardizirane knjižnice vmesniških razredov nudi različnim uporabnikom in proizvajalcem največjo možno raznolikost brez žrtvovanja interoperabilnosti.

Za nadaljnjo analizo informacij za namene obračuna, upravljanja obremenitve, strank ter pogodb je potrebna edinstvena identifikacija vseh podatkov, ki je neodvisna od proizvajalca. Definicija take identifikacije je podana v standardu OBIS [8], opisana pa je tudi v poglavju 4.4 na strani 29.

4.3 Vmesniški razredi COSEM

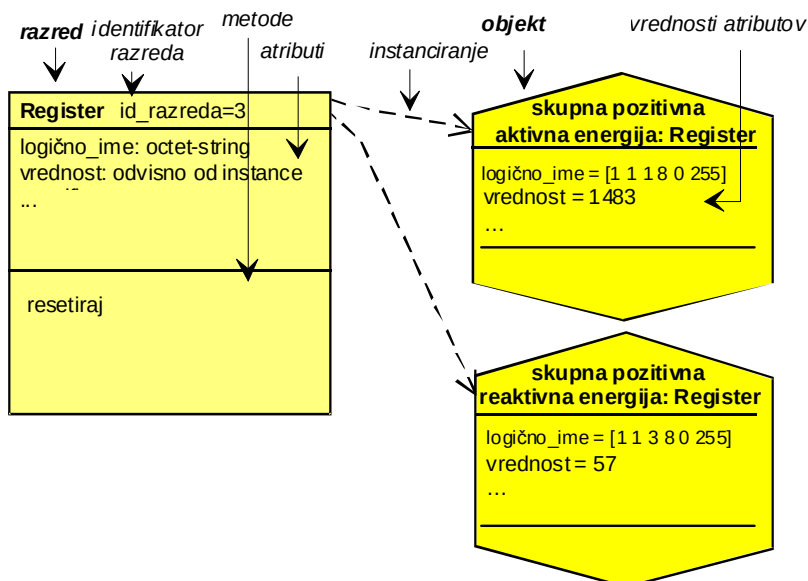
4.3.1 Osnovni principi

V tem poglavju so opisani osnovni principi na katerih temeljijo vmesniški razredi COSEM. Podan je tudi kratek pregled o tem kako so vmesniški objekti oz. instance vmesniških razredov uporabljeni za komunikacijske namene. Merilna oprema različnih proizvajalcev, ki sledi tem specifikacijam lahko izmenjuje podatke na interoperabilen način.

Objekt je zbirka atributov in metod. Atribut predstavlja karakteristike objekta, njegova vrednost lahko vpliva na obnašanje objekta. Prvi atribut vsakega objekta je logično ime, ki predstavlja del identifikacije objekta. Objekt ima lahko različno število metod, ki lahko spreminjajo vrednosti atributov.

Objekti z enakimi karakteristikami so posplošeni kot vmesniški razredi, identificirani so z razrednim identifikatorjem (ang.: class id). Za določen vmesniški razred so skupne karakteristike (atributi in metode) opisane enkrat za vse objekte. Instancam vmesniških razredov pravimo objekti COSEM.

Slika 5 prikazuje primer vmesniškega razreda in njegovih instanc:



Slika 5: vmesniški razred register in njegove instance

Vmesniški razred *Register* je sestavljen z združenjem potrebnih lastnosti za modeliranje posplošenega registra (vsebuje merilne in ostale podatke) kot je viden s strani odjemalca. Vsebina registra je identificirana z atributom logično ime (ang.: logical name), ki vsebuje identifikator OBIS. Dejanska dinamična vrednost registra je shranjena v atributu vrednost (ang.: value).

Definiranje specifične merilne naprave pomeni določanje več specifičnih objektov. Slika 5 prikazuje primer, ko merilna naprava vsebuje dva registra. To pomeni, da sta instancirani dve instanci vmesniškega razreda *Register*. Preko instanciranja prvi objekt postane register za skupno pozitivno aktivno energijo, drugi pa register za skupno pozitivno reaktivno energijo.

4.3.2 Podatkovni tipi

Tabela 8 prikazuje podatkovne tipe, ki so na voljo za attribute objektov COSEM [6, stran 22]:

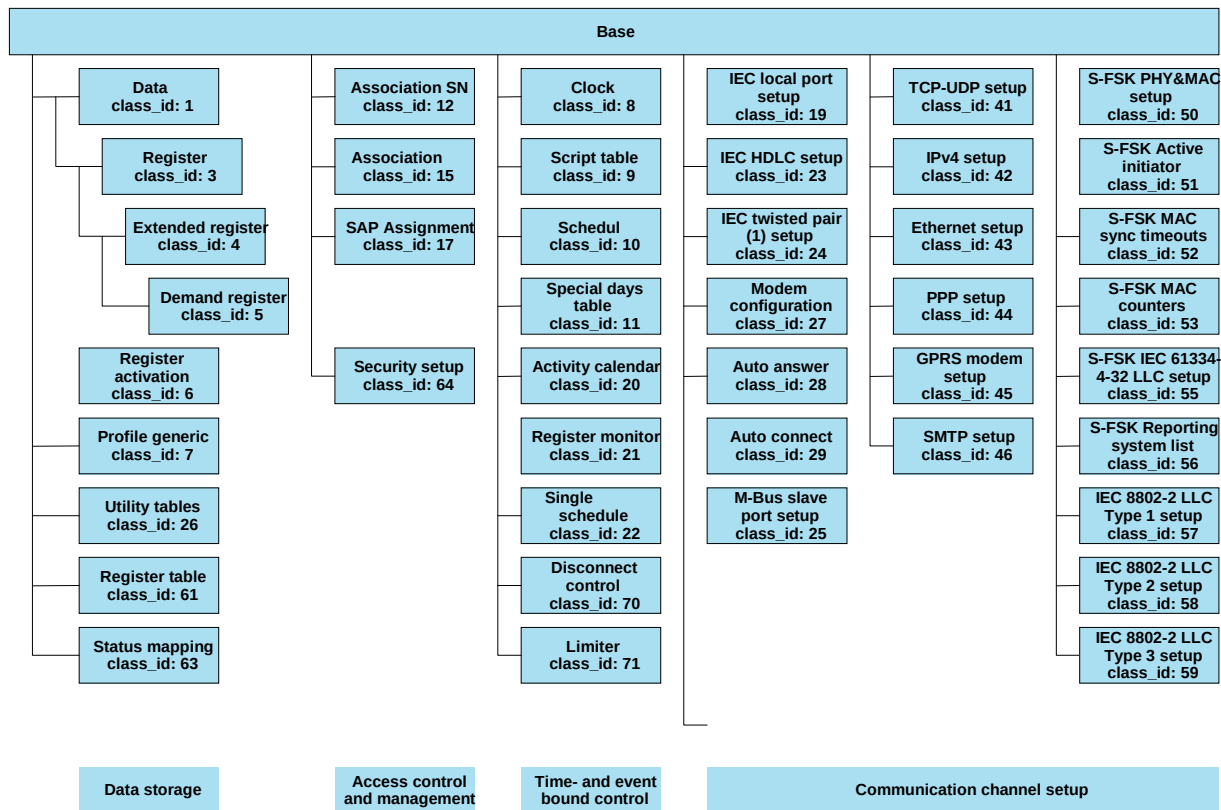
opis tipa	definicija	obseg vrednosti
-- preprosti		
null-data		
boolean	boolova vrednost	RESNIČNO ali NERESNIČNO
bit-string	urejena vrsta Boolovih vrednosti	
double-long	Predznačeno 32-bitno število	-2 147 483 648...

opis tipa	definicija	obseg vrednosti
		2 147 483 647
double-long-unsigned	nepredznačeno 32-bitno število	0...4 294 967 295
octet-string	urejena vrsta oktetov (8 bitnih zlogov)	
visible-string	urejena vrsta ASCII znakov	
bcd	binarno kodirano število (ang.: binary coded decimal)	
integer	predznačeno 8-bitno število	-128...127
long	predznačeno 16-bitno število	-32 768...32 767
unsigned	nepredznačeno 8-bitno število	0...255
long-unsigned	nepredznačeno 16-bitno število	0...65 535
long64	predznačeno 64-bitno število	$-2^{63} \dots 2^{63} - 1$
long64-unsigned	nepredznačeno 64-bitno število	$0 \dots 2^{64} - 1$
enum	elementi naštevnege tipa so podani v opisu atributa v specifikaciji vmesniškega razreda	
float32	octet-string (VELIKOST(4))	
float64	octet-string (VELIKOST(8))	
date_time	octet-string (VELIKOST(12))	
date	octet-string (VELIKOST(5))	
time	octet-string (VELIKOST(4))	
-- zapleteni		
array	zbirka elementov, njihov tip je podan v opisu atributa v specifikaciji vmesniškega razreda	
structure	struktura različnih elementov, elementi strukture in njihov tip so podani v opisu atributa v specifikaciji vmesniškega razreda	
compact array	zgoščena zbirka elementov, njihov tip je podan v opisu atributa v specifikaciji vmesniškega razreda	
-- IZBIRA	za nekatere attribute COSEM vmesniških razredov je lahko podatkovni tip znan šele ob nastanku instance v implementacijski fazi strežnika COSEM. Strežnik vedno pošlje podatek o tipu in vrednosti atributa. Seznam možnih tipov je definiran v opisu atributa v specifikaciji vmesniškega razreda.	

Tabela 8: podatkovni tipi COSEM

4.3.3 Pregled

Natančne definicije atributov in metod vseh vmesniških razredov COSEM so na voljo v modri knjigi [6]. Slika 6 prikazuje pregled vseh trenutno definiranih vmesniških razredov in odnosov med njimi [6, str. 30]:



Slika 6: pregled vseh vmesniških razredov COSEM

4.3.1 Primeri

Za lažjo predstavo v tem podpoglavju navajamo nekaj primerov specifikacije vmesniških razredov COSEM. Vsi ostali so na voljo v modri knjigi [6].

Podatki (id razreda: 1, verzija: 0)

Namenjen je za modeliranje raznovrstnih podatkov kot so konfiguracijski podatki in parametri. Pomen podatkov je določen v vrednostjo atributa *logično_ime*.

Tabela 9 prikazuje specifikacijo atributov in metod:

podatki	0...n	id_razreda = 1, verzija = 0			
<i>atributi</i>	<i>podatkovni tip</i>	<i>min.</i>	<i>maks.</i>	<i>predn.</i>	<i>kratko ime</i>
1. logično_ime	octet-string				x
2. vrednost	IZBIRA				x + 0x08
<i>specifikacija metod</i>	<i>m/o</i>				

Tabela 9: specifikacija atributov in metod za vmesniški razred podatki

Opis atributov:

logično_ime	Identificira instance vmesniškega razreda <i>podatki</i> .
vrednost	Vsebuje podatke. Tip podatka je <i>CHOICE</i> , kar pomeni da je lahko atribut <i>vrednost</i> katerikoli COSEM podatkovni tip (Tabela 8). Dejanski tip je znan šele ko preberemo vrednost atributa iz merilne naprave.

Nastavitve varnosti (id razreda: 64, verzija: 0)

Instance vsebujejo vse potrebne podatke o varnostni politiki in tipu uporabljenega varnostnega mehanizma med dvema sistemoma identificiranima z nazivom odjemalskega in strežniškega sistema. Na voljo imajo tudi metode, ki lahko povečajo nivo varnosti in prenašajo globalne ključe. Tabela 10 prikazuje specifikacijo atributov in metod:

nastavitve varnosti	0...n	id_razreda = 64, verzija = 0			
<i>atributi</i>	<i>podatkovni tip</i>	<i>min.</i>	<i>maks.</i>	<i>predn.</i>	<i>kratko ime</i>
1. logično_ime	octet-string				x
2. varnostna_politika	enum				x + 0x08
3. varnostni_mehanizem	enum				x + 0x10
4. naziv_odjemalskega_sistema	octet-string				x + 0x18
5. naziv_strežniškega_sistema	octet-string				x + 0x20
<i>specifikacija metod</i>	<i>m/o</i>				
1. aktivacija_varnosti	o				x + 0x28
2. prenos_globalnih_ključev	o				x + 0x30

Tabela 10: specifikacija atributov in metod za vmesniški razred nastavitve varnosti

Opis atributov:

logično_ime	Identificira instance vmesniškega razreda <i>nastavitve varnosti</i> .
varnostna_politika	Uveljavi algoritem za overjanje in/ali šifriranje. enum: (0) brez, (1) vsa sporočila naj bodo overjena, (2) vsa sporočila naj bodo šifrirana, (3) vsa sporočila naj bodo overjena in šifrirana
varnostni_mehanizem	Specificira algoritem za overjanje, šifriranje in šifriranje ključa (ang.: key wrapping). enum: (0) AES-GCM-128 za overjeno šifriranje in AES-128 za šifriranje ključa
naziv_odjemalskega_sistema	Vsebuje trenutni naziv odjemalca.
naziv_strežniškega_sistema	Vsebuje trenutni naziv strežnika. Atribut mora biti na voljo samo za branje.

Opis metod:

aktivacija_varnosti (podatki)	Aktivira in poostri varnostno politiko. Parameter podatki mora biti enakega tipa kor atribut varnostna_politika.
prenos_globalnih_ključev (podatki)	Posodobi enega ali več globalnih ključev. Parameter podatki vsebuje šifrirane ključe. Ima naslednjo obliko: array podatek_ključa podatek_ključa ::= structure { id_ključa: enum (0) globalni ključ za šifriranje (ang.: global unicast encryption key), (1) globalni ključ za množično šifriranje (ang.: global broadcast encryption key), (2) ključ za overjanje, šifriran_ključ: octet-string }

Pri opisu metode *prenos_globalnih_ključev* so iz osnovnih podatkovnih tipov definirani dodatni tipi. Parameter metode je namreč polje elementov (ang.: array) tipa podatek_ključa

(nova definicija tipa). Definiran je kot struktura (ang.: structure), ki vsebuje dva elementa. Naziv prvega elementa je `id_ključa` in je tipa enum s podanimi vrednostmi. Drugi element je `šifriran_ključ` in je preprost COSEM podatkovni tip octet-string.

S tem primerom smo želeli pokazati kako so za parametre metod lahko določeni novi (izpeljani) podatkovni tipi. Enako velja tudi za attribute, globina izpeljav pa ni omejena (npr. element strukture je lahko struktura, ki ima za element strukturo...).

4.4 Sistem identifikacije objektov - OBIS

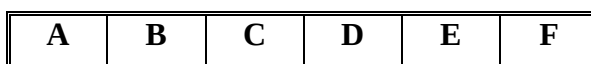
Sistem identifikacije objektov (OBIS) definira identifikacijske kode za objekte, ki so pogosto uporabljeni v merilnih napravah. Predpisuje unikatno identifikacijo za vse podatke v merilni opremi. Poleg podatkov o izmerjenih vrednostih tudi za abstraktne vrednosti za konfiguracijo ali pridobivanje informacij o obnašanju merilne opreme. Kode OBIS so uporabljene za identifikacijo:

- logičnih imen različnih instanc vmesniških razredov,
- podatkov prenesenih po komunikacijskih poteh,
- podatkov prikazanih na merilni opremi.

OBIS je veljaven za vse tipe merilne opreme, za pokritje vseh tipov energije v kombinaciji z večimi fizičnimi kanali pa vpelje koncept prenosnega medija in kanalov. Tako je možno identificirati merilne podatke različnih izvorov.

4.4.1 Struktura kode

Kode OBIS identificirajo podatke v merilni opremi za merjenje energije. Imajo hierarhično strukturo v obliki šestih vrednostnih skupin od A do F. Slika 7 prikazuje strukturo kode OBIS:



Slika 7: struktura kode OBIS

Skupina A

Vrednost skupine A identificira prenosni medij (tip energije) na katerega se nanašajo meritve. Vsi podatki, ki niso vezani na prenosni medij so obravnavani kor abstraktni podatki (A=0).

Tabela 11 podaja vrednosti za skupino A [6, stran 177]:

skupina A	
0	abstraktni objekti
1	objekti povezani z elektriko
4	objekti povezani s toplotno energijo (ang.: heat cost allocators)
5	objekti povezani s hlajenjem
6	objekti povezani s toploto
7	objekti povezani s plinom
8	objekti povezani z mrzlo vodo
9	objekti povezani s toplo vodo
vse ostalo	rezervirano

Tabela 11: vrednosti za skupino A

Skupina B

Skupina B se uporablja za identifikacijo številke kanala. Kanal je številka vhoda merilne opreme, ki ima več vhodov za merjenje energije enakih ali različnih tipov. Tako je možna identifikacija podatkov iz različnih izvorov. Definicije za to skupino so odvisne od skupine A.

Skupina C

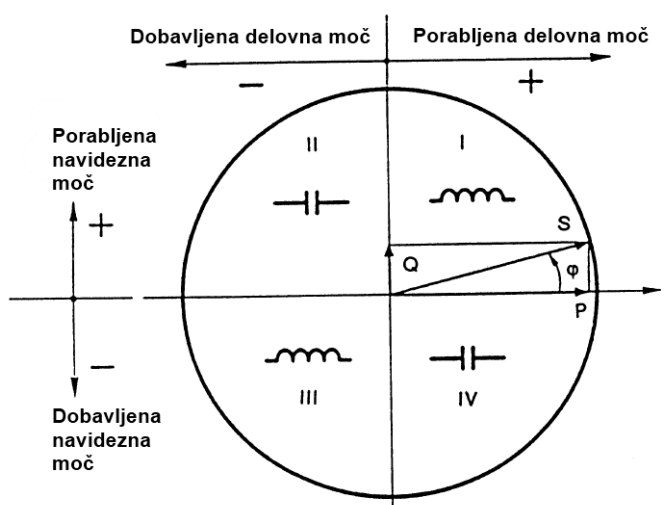
Vrednost skupine C označuje abstraktne ali fizične podatke povezane s tipom izvora, npr. tok, napetost, moč, prostornina, temperatura. Definicije so odvisne od vrednosti v skupini A. Tabela 12 prikazuje pomen za nekaj prvih vrednosti pri A=1 [6, str. 185]:

skupina C – elektrika (A = 1)	
1	ΣL_i delovna moč+ (QI+QIV)
2	ΣL_i delovna moč – (QII+QIII)
3	ΣL_i jalova moč+ (QI+QII)
4	ΣL_i jalova moč– (QIII+QIV)
5	ΣL_i jalova moč QI
6	ΣL_i jalova moč QII
7	ΣL_i jalova moč QIII
8	ΣL_i jalova moč QIV
9	ΣL_i navidezna moč+ (QI+QIV)
10	ΣL_i navidezna moč– (QII+QIII)
11	tok : katerakoli faza
12	napetost : katerakoli faza
13	ΣL_i faktor moči–

skupina C – elektrika (A = 1)	
14	frekvenca
15	ΣL_i delovna moč ($\text{abs}(QI+QIV)+(\text{abs}(QII+QIII))$)
16	ΣL_i delovna moč ($\text{abs}(QI+QIV)-\text{abs}(QII+QIII)$)
17	ΣL_i delovna moč QI
18	ΣL_i delovna moč QII
19	ΣL_i delovna moč $QIII$
20	ΣL_i delovna moč QIV

Tabela 12: nekaj vrednosti skupine C za elektriko

Slika 8 [6, str. 187] prikazuje uporabljene oznake in razporeditev moči po kvadrantih za lažje razumevanje oznak:



Slika 8: razporeditev moči po kvadrantih

Nadaljnje procesiranje, klasifikacija in metode shranjevanja so opisane v skupinah D, E in F.

Za abstraktne podatke vrednostne skupine D do F omogočajo nadaljnjo klasifikacijo podatkov identificiranih z vrednostnimi skupinami A do C.

Skupina D

Označuje tip ali rezultat procesiranja fizičnih veličin, identificiranih z vrednostnimi skupinami od A do C, z raznovrstnimi specifičnimi algoritmi.

Tabela 13 prikazuje nekaj vrednosti [6, stran 187]:

skupina D – elektrika (A = 1, C <> 0, 93, 94, 96, 97, 98, 99)	
0	povprečje obračunske periode (od zadnjega resetiranja)
1	kumulativni minimum 1
2	kumulativni maksimum 1
3	minimum 1
4	trenutno povprečje 1
5	prejšnje povprečje 1
6	maksimum 1
7	trenutna vrednost
8	časovni integral 1
9	časovni integral 2
10	časovni integral 3

Tabela 13: nekaj vrednosti skupine D za elektriko

Skupina E

Identificira nadaljnje procesiranje ali klasifikacijo veličin podanih z vrednostmi skupin A do D.

Skupina F

Uporabljena je za identifikacijo zgodovinskih podatkov, podanih z vrednostnimi skupinami A do E, glede na različne obračunske intervale. Kjer to ne pride v poštev, je lahko skupina F uporabljena za nadaljnjo klasifikacijo.

5 Analiza zahtev

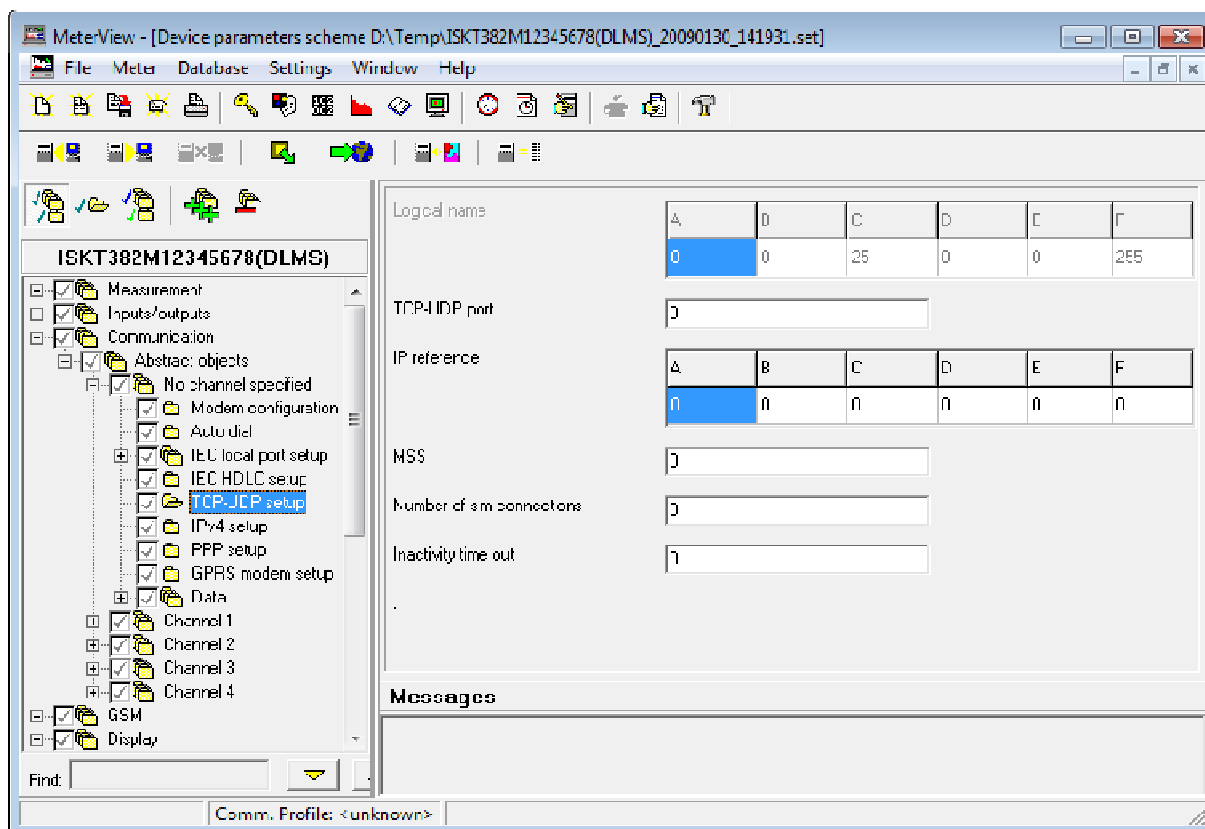
V tem poglavju bomo na podlagi zahtev opisanih v prejšnjih poglavjih in danih možnosti izbrali platformo za izdelavo programske opreme.

Programska oprema mora omogočati lokalni in oddaljen nadzor ter upravljanje merilnih naprav izdelanih po specifikacijah NTA 8130 in dodatnih zahtevah za nizozemske pametne merilne naprave. Iz zahtev lahko ugotovimo, da je to možno s komunikacijo preko vrat P0 in P3 merilnega mesta. Za protokol komunikacije preko vrat P3 je v [5, stran 8] določen protokol DLMS/COSEM ki smo ga na kratko opisali tudi v poglavju 0. Z implementacijo tega protokola lahko dostopamo do instanc objektov COSEM, ki jih odjemalcem ponuja merilna naprava, s tem pa je zagotovljeno upravljanje in nadzor merilne naprave.

Implementacija celotnega protokola DLMS/COSEM na vseh nivojih [6, str. 10] bi vzela veliko preveč časa, zato smo poskušali najti rešitev, ki bi omogočala uporabo obstoječih komunikacijskih knjižnic in programske opreme. V nadaljnjih poglavjih bomo zato opisali obstoječo programsko opremo, ki je na voljo v podjetju Iskraemeco.

5.1 MeterView 4

MeterView 4 je programska oprema, ki je bila sprva namenjena za nadzor in upravljanje merilnih naprav po standardu IEC 1107 in vsebuje komunikacijsko knjižnico za lokalno ter oddaljeno komunikacijo. Napisan je v okolju Borland C++ builder 6, prva verzija je nastala že pred več kot 10 leti. Dodajanje podpore za nove naprave je podprto s sistemom gonilnikov, pred nekaj leti pa je bil dodan tudi gonilnik za protokol DLMS/COSEM. Slika 9 prikazuje parametriranje naprave po protokolu DLMS/COSEM z uporabo MeterView 4:

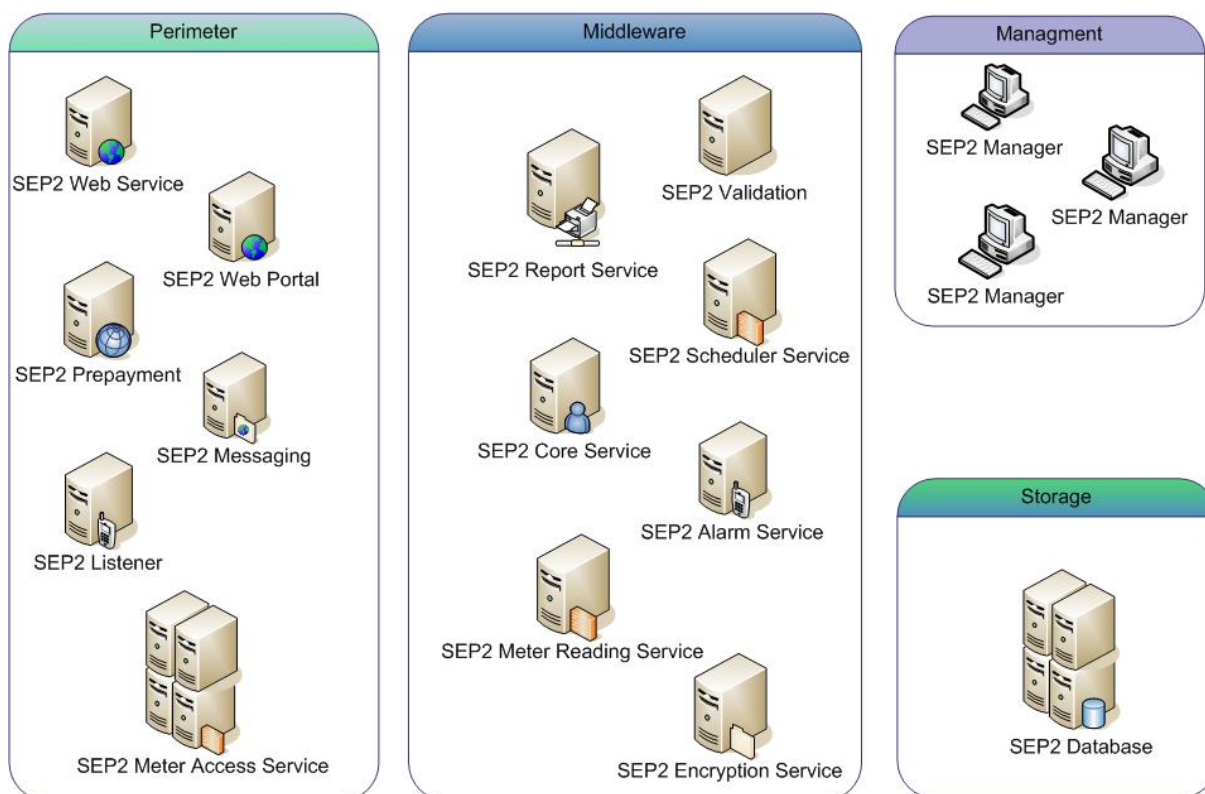


Slika 9: parametriranje naprave DLMS/COSEM z uporabo MeterView 4

5.2 SEP2W System

SEP2W System je odprta in prilagodljiva programska oprema za avtomatsko branje merilnih naprav (AMR). Sestavljen je iz različnih komponent, večina je napisana s pomočjo integriranega razvojnega orodja Microsoft Visual Studio v programskem jeziku C# na ogrodju Microsoft .NET.

Slika 10 prikazuje komponente sistema SEP2W System:



Slika 10: komponente SEP2W System

Večina komponent je izdelana v obliki Windows storitev, sistem pa vsebuje še eno ali več podatkovnih baz ter Windows aplikacijo za nadzor nad sistemom. Kratak opis komponent [9, stran 5]:

- SEP2 Database – za shranjevanje podatkov, trenutno so podprte podatkovne baze Microsoft SQL Server, IBM DB2, Oracle,
- SEP2 Manager – Windows aplikacija za konfiguracijo in nadzor sistema,
- SEP2 Core Service – nadzor in upravljanje licenc, upravljanje uporabnikov in ostale skupne funkcije,
- SEP2 Scheduler Service – omogoča avtomatsko izvrševanje nalog z uporabo diagramov poteka,
- SEP2 Meter Reading Service – nadzira branje merilnih naprav in shranjuje zbrane podatke,
- SEP2 Validation Service – omogoča preverjanje veljavnosti zbranih podatkov,
- SEP2 Report Service – za izdelavo poročil, podpira različne izračune in manipulacijo podatkov,

- SEP2 Alarm Service – shranjuje alarme merilnih naprav in omogoča obvestila o alarmih,
- SEP2 Meter Access Service – omogoča dostop do merilnih naprav,
- SEP2 Listener Service – posluša za alarme merilnih naprav na različnih kanalih,
- SEP2 Messaging Service – omogoča avtomatsko uvažanje in izvažanje podatkov,
- SEP2 Web Service – omogoča oddaljen dostop do sistema z uporabo SOAP in XML tehnologij,
- SEP2 Web Portal – omogoča oddaljen dostop do sistema s pomočjo HTML,
- SEP2 Prepayment Service – omogoča predplačniško funkcijo,
- SEP2 Integration – omogoča integracijo SEP2 System.

5.3 Izbira platforme

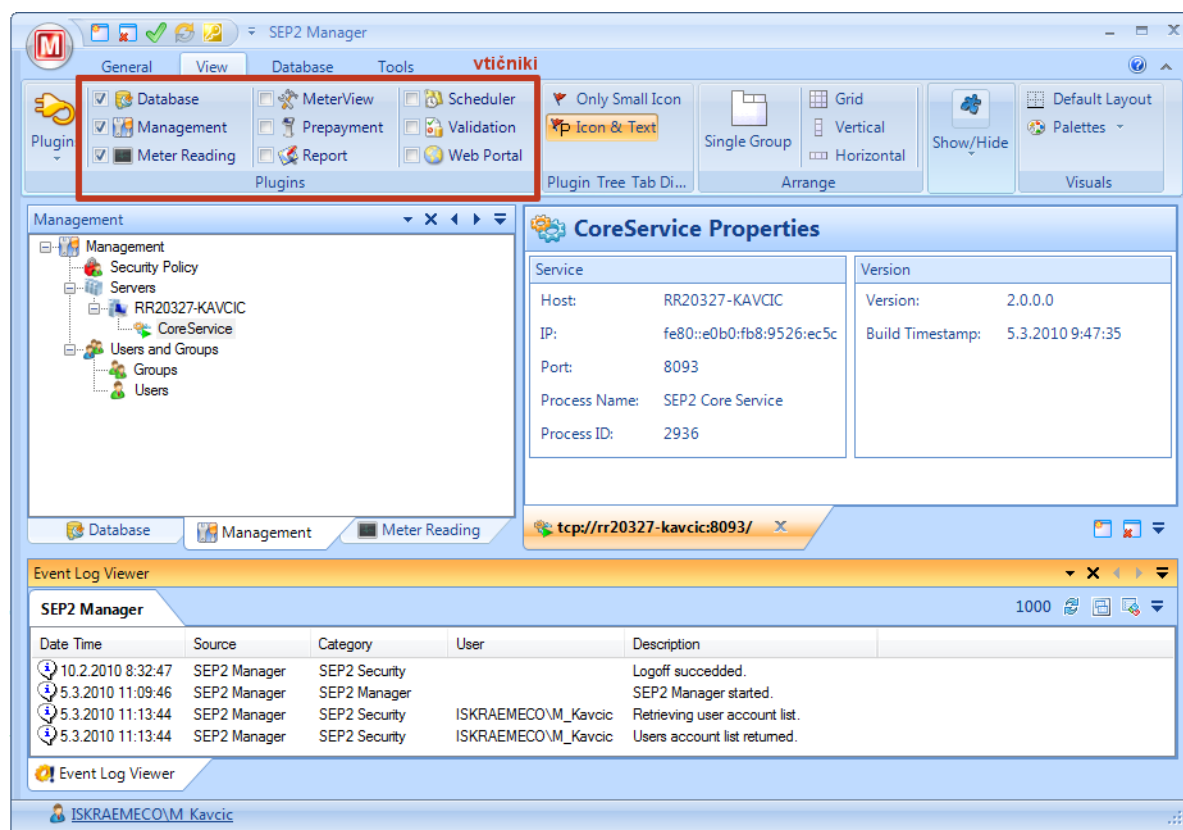
MeterView 4 vsebuje komunikacijske knjižnice, ima pa tudi že gonilnik za protokol DLMS/COSEM. Po podrobnem pregledu smo ugotovili naslednje pomanjkljivosti:

- komunikacijska knjižnica je nestabilna, izvorna koda je zapletena in je zato vzdrževanje ter dodajanje dodatnih komunikacijskih protokolov zelo zapleteno,
- vmesnik za gonilnike je zelo zaprt, saj je bil prvotno narejen za IEC 1107 gonilnike,
- kot posledica zaprtosti vmesnika je gonilnik za protokol DLMS/COSEM narejen tako, da podpira točno določene instance vmesniških razredov COSEM, kar pomeni, da je za vsako novo instanco potrebna sprememba izvorne kode,
- dodajanje novih vmesniških razredov je zaradi arhitekture gonilnika DLMS/COSEM zapleteno in počasno,
- podatki med MeterView 4 in gonilniki se prenašajo v tekstovni obliki, kar hitro pripelje do napak,
- omogoča hkratno delo največ z eno napravo, kar dostikrat predstavlja oviro.

Zaradi teh pomanjkljivosti lahko zaključimo, da MeterView 4 ni ustrezen kot osnova za izdelavo zahtevane programske opreme.

SEP2W System vsebuje odlično komunikacijsko knjižnico, ki podpira tudi nižje nivoje protokola DLMS/COSEM (sporočanje, prenos, glej poglavje 4.1). Ima tudi sistem za beleženje dogodkov, ki lahko dogodke vpisuje med drugim tudi v dnevnik dogodkov operacijskega sistema Windows (ang.: Windows event log) in je preprost za uporabo.

Aplikacija za nadzor sistema SEP2 Manager je zgrajena v obliki vtičnikov (ang.: plugins). Ti imajo dostop do vseh podatkov sistema in so namenjeni za nadzor in upravljanje celotnega sistema. Slika 11 prikazuje izgled vtičnikov v aplikaciji SEP2 Manager:



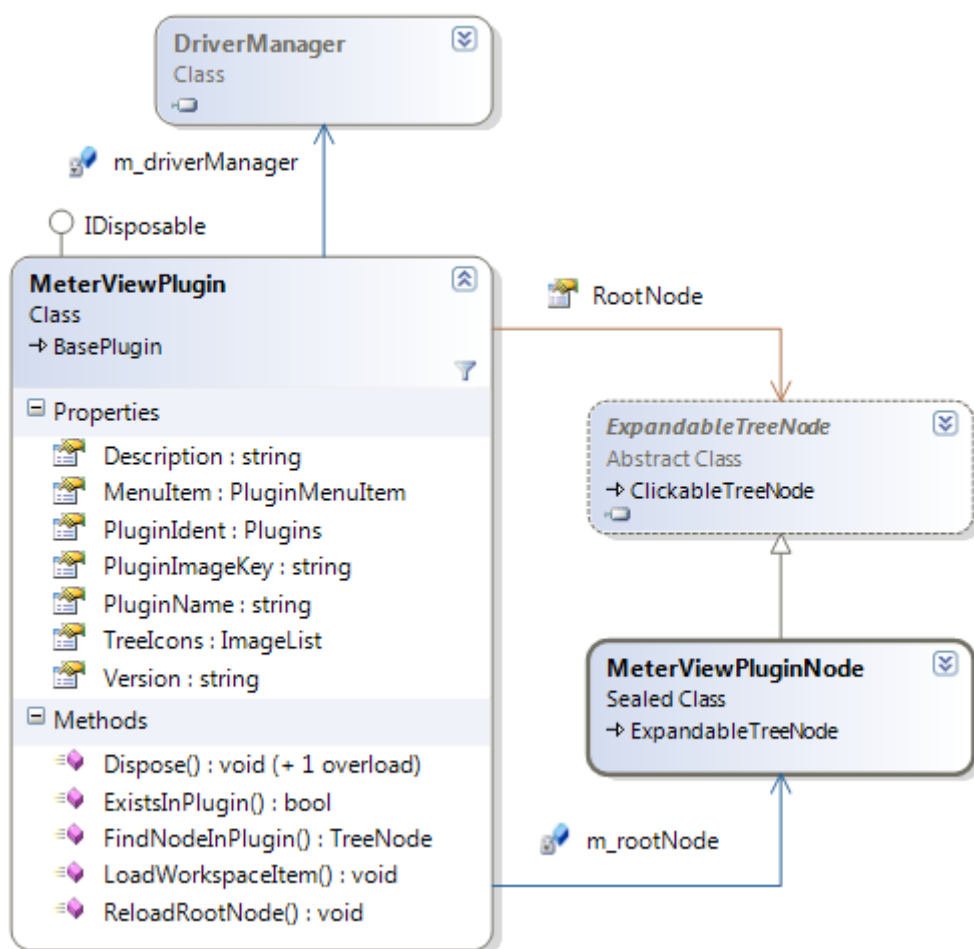
Slika 11: izgled vtičnikov v aplikaciji SEP2 Manager

Zaradi vseh opisanih prednosti smo se odločili, da zahtevano programsko opremo izdelamo kot vtičnik v SEP2 Manager. S tem smo prihranili veliko časa in se izognili podvajanju dela (skupne kontrole, skupna komunikacijska knjižnica).

6 Načrtovanje arhitekture

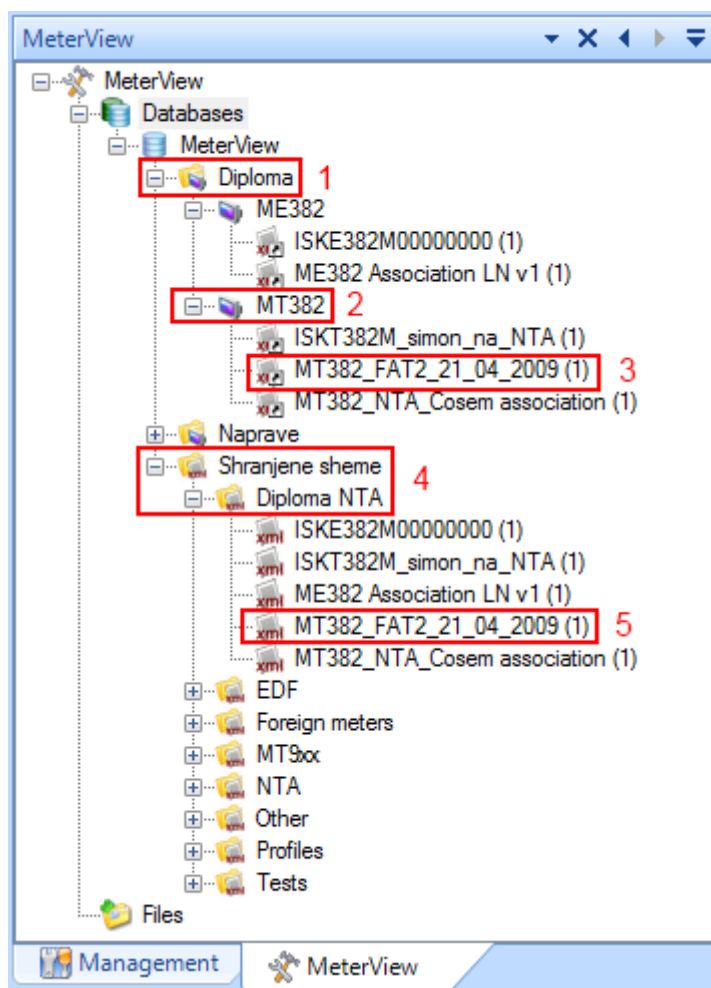
6.1 Vtičnik za SEP2 Manager

Da bi omogočili uporabo vseh prednosti, ki jih ponuja SEP2 System, smo izdelali vtičnik za SEP2 Manager. Vtičnik se imenuje *MeterView*, razred vtičnika *MeterViewPlugin* pa je dedovan iz abstraktnega razreda *BasePlugin*, ki predstavlja vmesnik za vse vtičnike. Slednje je tudi osnovni pogoj, da je razred prepoznan kot veljaven vtičnik za SEP2 Manager. Glavna lastnost vmesnika vtičnika je, da lahko vtičnik vrne svoje vozlišče (ang: node), ki je potem prikazano v drevesu vtičnika (razred *MeterViewPluginNode*). Zaradi razširitve možnosti uporabe vtičnika za upravljanje in nadzor tudi ostalih naprav, smo zasnovali sistem gonilnikov. Vtičnik zato vsebuje še instanco razreda *DriverManager*, ki skrbi za nalaganje gonilnikov. Za vsako prikazano napravo je ob prvi uporabi naložena natanko ena instanca gonilnika, tip gonilnika pa je določen v atributih naprave v podatkovni bazi. Slika 12 prikazuje omenjene razrede in odnose med njimi:



Slika 12: razred vtičnika in nekatere njegove lastnosti

Vtičnik v drevesni strukturi prikazuje podatke iz ene ali večih podatkovnih baz. Na najvišjem nivoju so prikazane podatkovne baze, potem pa sledijo skupine v katerih so prikazani elementi. Vsaka skupina ima lahko poljubno podskupin enakega tipa (Slika 13, oznaka 4), elementi pa so lahko naprave (Slika 13, oznaka 2) ali xml dokumenti (Slika 13, oznaka 5), iz podatkovne baze.



Slika 13: drevesna struktura vtičnika

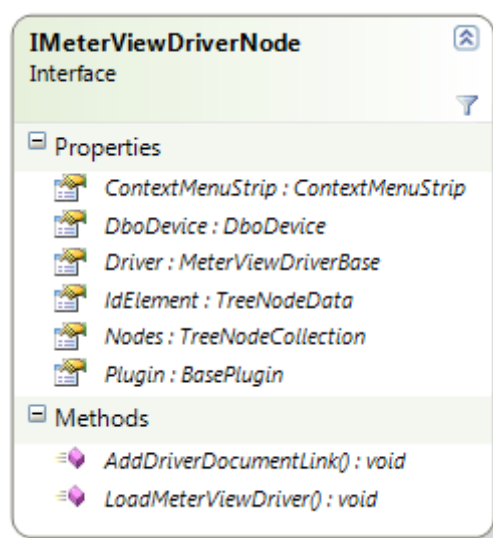
Na Slika 13 številčne oznake pomenijo:

- 1 – skupina naprav,
- 2 – naprava,
- 3 – bližnjica do xml dokumenta,
- 4 – skupina xml dokumentov,
- 5 – xml dokument.

Slika 13 pod oznako 3 prikazuje bližnjico do xml dokumenta pod oznako 5. Princip bližnjic do dokumentov omogoča, da imamo pod napravo bližnjice do dokumentov, ki pripadajo

določeni napravi. Ob odprtju dokumenta preko bližnjice se samodejno izbere instanca gonilnika naprave pod katero se nahaja bližnjica.

Razred vozlišča naprave implementira vmesnik *IMeterViewDriverNode*, ki vsebuje lastnosti in metode za pravilno delovanje nalaganja gonilnikov, bližnjic do dokumentov, prikaz dokumentov, itd. Slika 14 prikazuje lastnosti in metode vmesnika *IMeterViewDriverNode*:



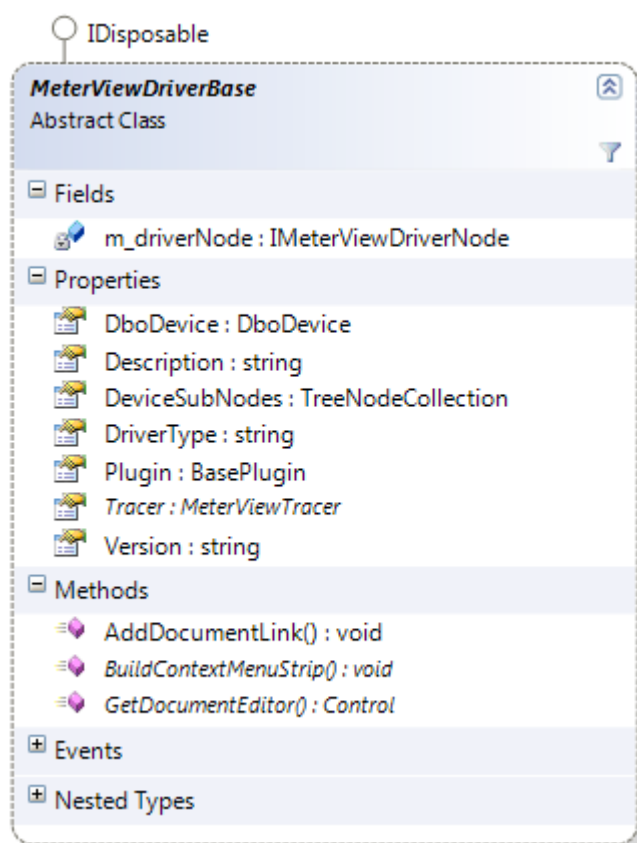
Slika 14: vmesnik vozlišča gonilnika naprave

6.2 Bazni gonilnik

Da bi omogočili dodajanje gonilnikov za poljubne naprave, smo izdelali bazni gonilnik, v katerem smo implementirali skupne funkcije vseh gonilnikov. Vmesnik gonilnika je načrtovan zelo odprto, saj je tako omogočeno dodajanje gonilnikov tudi za naprave, ki niso nujno merilne (npr. zbiralniki podatkov - DC).

Bazni gonilnik omogoča gonilnikom dodajanje vozlišč pod napravo, dodajanje menijev v kontekstnem meniju vozlišča naprave, dodajanje zahtev komunikacijski knjižnici ter možnost shranjevanja in prikazovanja poljubnih dokumentov direktno in preko bližnjic.

Slika 15 prikazuje nekatere lastnosti in metode razreda baznega gonilnika:



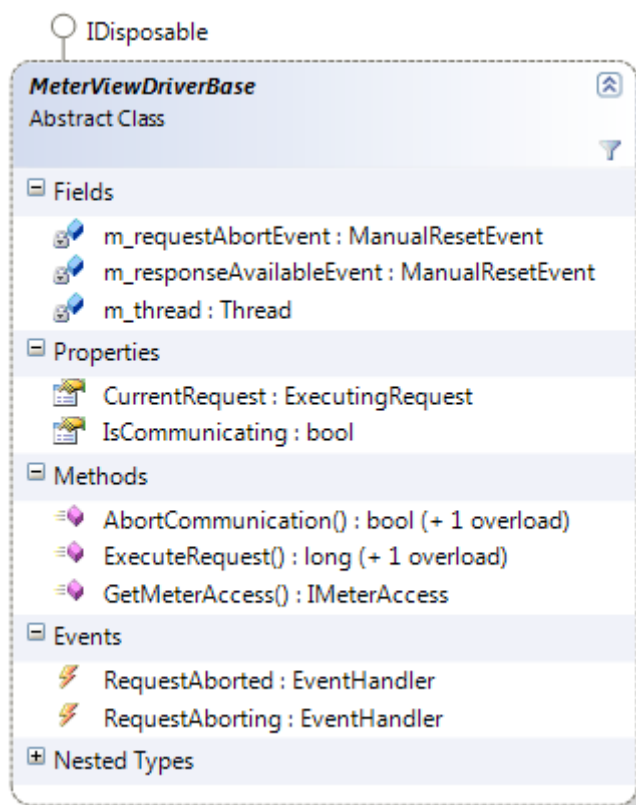
Slika 15: nekatere lastnosti baznega gonilnika

V nadaljevanju bomo prikazali lastnosti in metode baznega gonilnika za komunikacijo.

6.2.1 Komunikacija

Dejanska komunikacija z napravami je izvedena s pomočjo komunikacijske knjižnice SEP2 Meter Access Service [10]. Ta deluje preko .NET remoting [11], kar pomeni, da imamo na voljo objekt, ki se obnaša kot da bi bil lokalni, čeprav je v resnici lahko tudi na kakšnem oddaljenem računalniku. Objekt je tipa `IMeterAccess` in vsebuje metode, ki omogočajo dodajanje in spremljanje zahtev za komunikacijo. Da bi načrtovalcem gonilnikov prihranili težave pri dodajanju zahtev in spremljanju (večnost – možnost napak), smo v baznem gonilniku implementirali nekaj metod, ki omogočajo preprosto dodajanje zahtev, njihovo prekinitev in obveščanje o prekinitvi izvajanja zahteve.

Slika 16 prikazuje metode in lastnosti baznega gonilnika, ki to omogočajo:



Slika 16: lastnosti in metode baznega gonilnika za komunikacijo

Gonilniki najprej dodajo zahtevo v izvajanje s pomočjo metode *ExecuteRequest*. Kot parameter metoda sprejme tudi delegat [12, strani 177 - 194], ki je klican, ko je na voljo rezultat. Bazni gonilnik najprej doda zahtevo komunikacijski knjižnici nato pa v novi niti (ang.: thread) čaka na odgovor komunikacijske knjižnice ali prekinitve s pomočjo metode *AbortCommunication*. Čakanje na odziv in upoštevanje prekinitve je izvedeno s pomočjo instanc razreda *ManualResetEvent*, ki je del ogrodja .NET in se uporablja za obveščanje čakajočih niti o dogodkih [11]. Ko je zahteva zaključena, bazni gonilnik poda rezultat s klicem povratne metode, podane ob klicu metode *ExecuteRequest*. Lastnost *CurrentRequest* je uporabljena za sledenje stanja zahteve na grafičnih kontrolah.

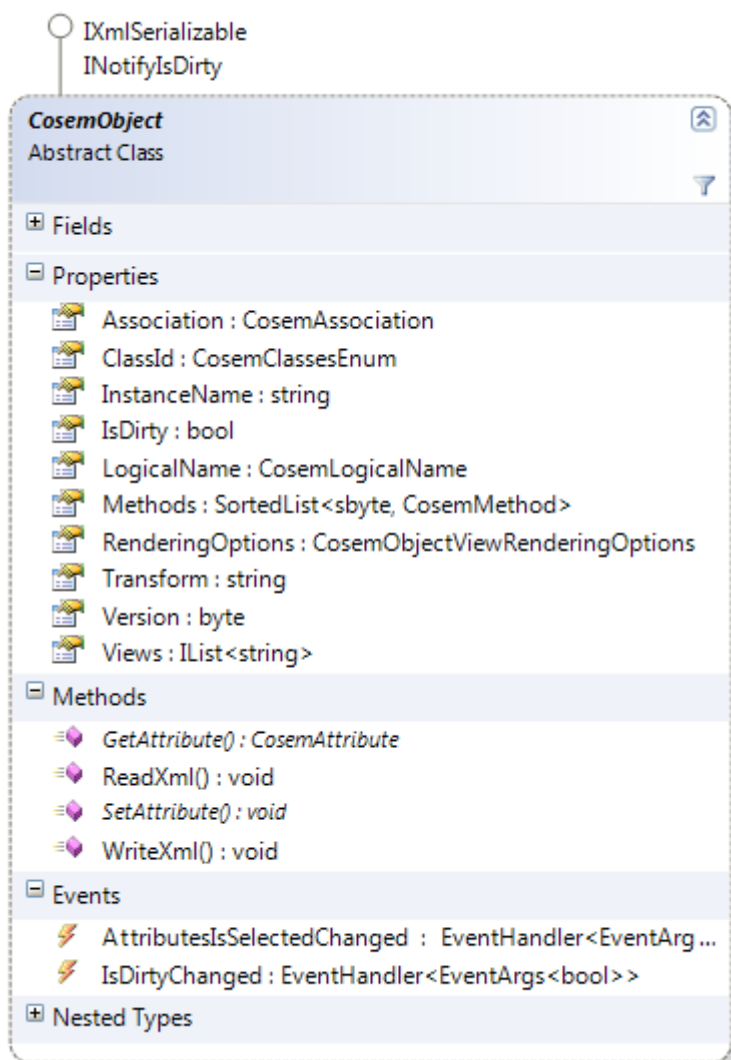
6.3 Modeliranje vmesniških razredov COSEM

V poglavju 5 smo ugotovili, da za rešitev problema potrebujemo gonilnik DLMS/COSEM, zato smo izdelali gonilnik, katerega večinski del predstavlja modeliranje vmesniških razredov COSEM in izdelava grafičnih kontrol za prikaz. V naslednjih podpoglavjih bomo pokazali kako smo modelirali vmesniške razrede COSEM:

6.3.1 Podatkovni model

Naprave COSEM lahko vsebujejo več različnih instanc vmesniških razredov COSEM, zato smo izdelali razred *CosemAssociation*, katerega lastnost *Objects* vsebuje celoten ali delen nabor objektov naprave. Implementira tudi vmesnike *IMeterViewDriverDocument* (deduje iz *IXmlSerializable*), *IMeterViewDriverChangeable*, ter *INotifyIsDirty*. Prvi omogoča shranjevanje v xml obliki, drugi je potreben za menjavanje instanc gonilnikov, zadnji pa zagotavlja obveščenost o spremembah vrednosti atributov.

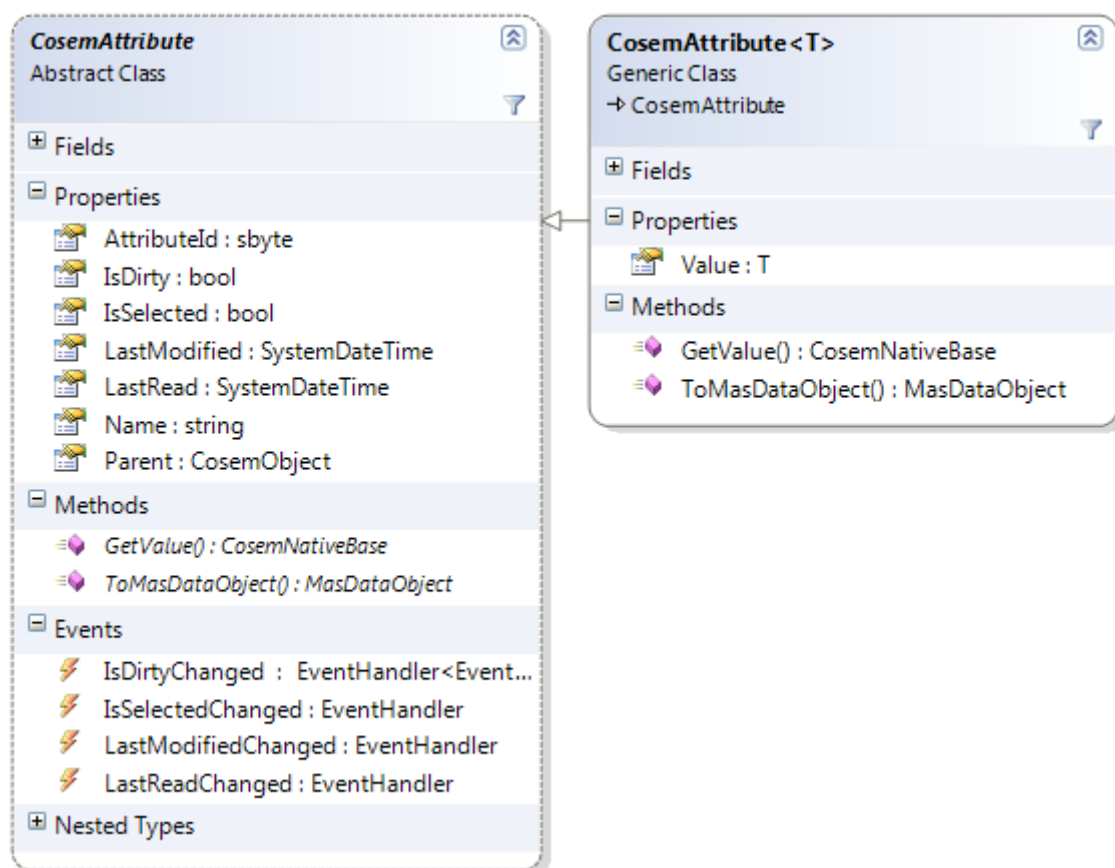
Za modeliranje vmesniških razredov COSEM smo naredili abstraktni razred, katerega dedujejo dejanski vmesniški razredi. Slika 17 prikazuje nekatere pomembnejše lastnosti osnovnega abstraktnega razreda za vmesniške razrede COSEM:



Slika 17: abstraktni osnovni razred za vmesniške razrede COSEM

Lastnosti *ClassId*, *LogicalName* in *Version* so lastnosti iz [6], ostale so pomožne. Kot je opisano v [6, stran 18] imajo vmesniški razredi lahko različno število atributov in metod, ki so v napravah dostopne po številkah. Abstraktni metodi *GetAttribute* in *SetAttribute* sta namenjeni pridobivanju in nastavljanju atributov oz. njihovih vrednosti, lastnost *Methods* pa vsebuje seznam vseh metod.

Slika 18 prikazuje podatkovni model atributov vmesniških razredov COSEM:

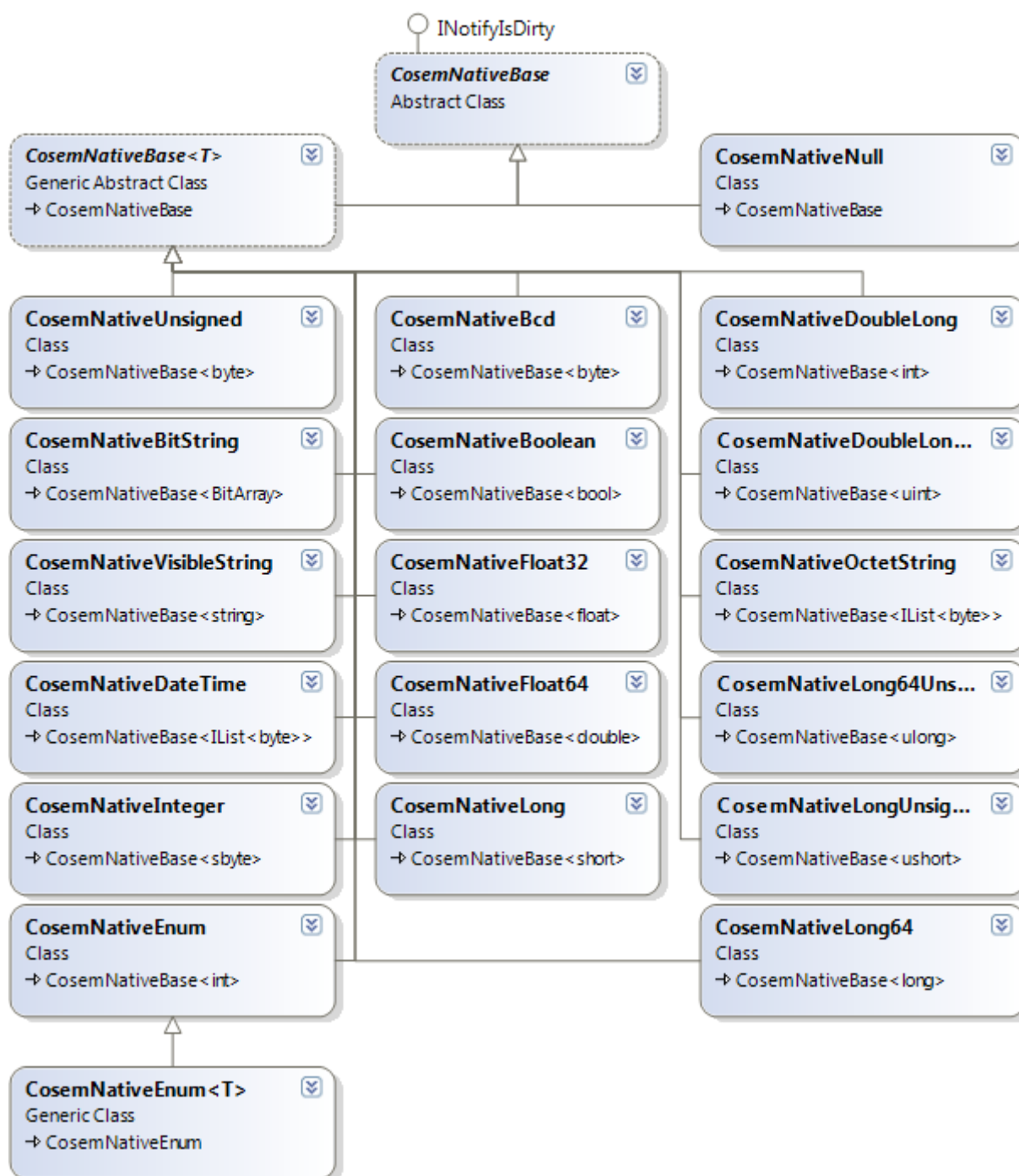


Slika 18: podatkovni model atributov vmesniških razredov

Generični razred *CosemAttribute<T>* je definiran kot *public class CosemAttribute<T> : CosemAttribute where T : CosemNativeBase*, kar pomeni, da je T lahko katerikoli podatkovni tip, ki deduje po razredu *CosemNativeBase*. *CosemNativeBase* predstavlja osnovni razred za vse podatkovne tipe, definirane v [6], zato bomo v nadaljevanju opisali modeliranje teh tipov.

Za vse preproste podatkovne tipe COSEM (poglavje 4.3.2) smo s pomočjo dedovanja [12, strani 101-120] in generičnih tipov [12, strani 223-247] izdelali podatkovni model.

Slika 19 prikazuje razredni diagram podatkovnega modela za preproste tipe COSEM:

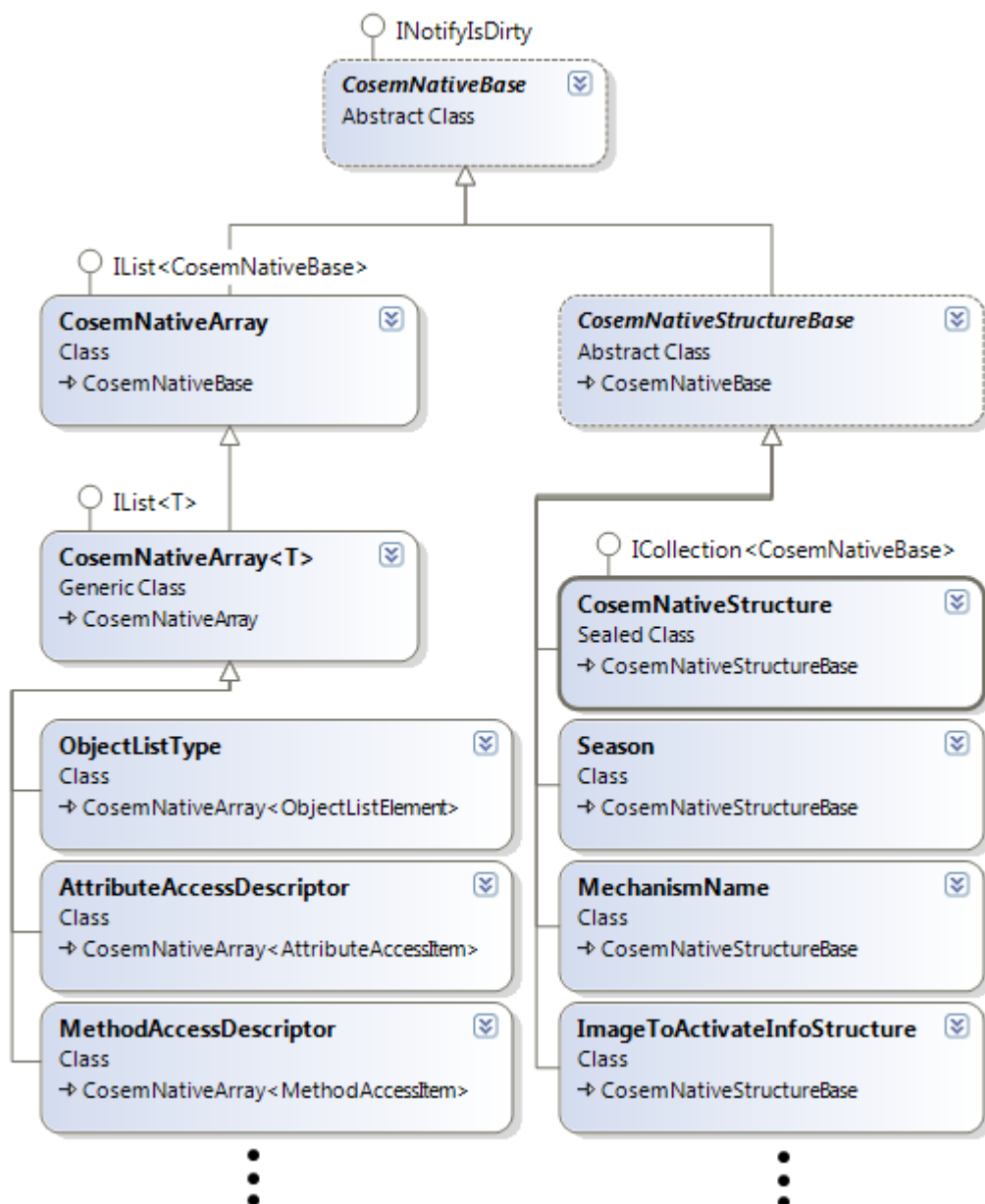


Slika 19: razredni diagram za preproste COSEM tipe

Osnovni razred *CosemNativeBase* je abstrakten [12, stran 106] in implementira vmesnik *INotifyIsDirty*, ki je potreben za delovanje sledenja sprememb. Uporabljen je kot osnovni razred iz katerega dedujejo vsi podatkovni tipi COSEM, s pomočjo abstraktnih metod [12, stran 106] pa tudi zahteva, da vsi podpirajo pretvorbo v in iz SEP2 Meter Access Service tipov (v nadaljevanje MAS tipi) [13].

Iz osnovnega razreda smo s pomočjo generičnega tipa *CosemNativeBase*<*T*> naredili osnovni razred za večino preprostih tipov. Ta vsebuje lastnost *Value*, ki vsebuje dejansko vrednost. Tipi dejanskih vrednosti so ustrezni C# vrednostni tipi, ki ustrezajo podatkovnemu tipu posameznega preprostega tipa COSEM (npr. zlog oz. bajt je ustrezen C# vrednostni tip za preprost tip unsigned). Vsem razredom smo prekrili operatorje (ang.: operator overloading) [12, strani 155-164] za enakost in neenakost, prekrili pa smo tudi metodo za primerjavo *Equals*. S tem smo dosegli primerjavo dejanske vrednosti v lastnosti *Value* namesto primerjave referenc objektov [12, strani 153-154].

Za zapletene tipe smo, podobno kot za preproste, uporabili za osnovni razred *CosemNativeBase*. Slika 20 prikazuje razredni diagram podatkovnega modela za zapletene COSEM tipe. Na sliki smo s tremi pikami nakazali, da ne prikazujemo celotnega modela, saj je celoten model precej obsežen. Generični razred *CosemNativeArray*<*T*> omogoča preprosto modeliranje zapletenega tipa COSEM array z enakimi elementi. Pretvorba iz MAS tipov je za posamezne elemente narejena s pomočjo uporabe refleksije [12, strani 357-377]. Za strukture smo pripravili ogrodje za izdelavo konkretnih struktur z dedovanjem abstraktnega razreda *CosemNativeStructureBase*. Razred *CosemNativeStructure* je namenjen za strukture, ki še niso podprte. Podobno je *CosemNativeArray* namenjen za polja z neznanim tipom elementov.



Slika 20: razredni diagram za zapletene tipe COSEM

6.3.2 Sledenje sprememb

Ko uporabnik preko kontrole spremeni katerokoli vrednost v podatkovnem modelu, mora biti v kontroli za prikaz vidno, da se je nekaj spremenilo. Zato smo izdelali sledenje sprememb. Deluje tako, da vsi tipi in vmesniški razredi COSEM, atributi ter razred *CosemAssociation* implementirajo vmesnik *INotifyIsDirty*.

Slika 21 prikazuje vmesnik za sledenje sprememb:



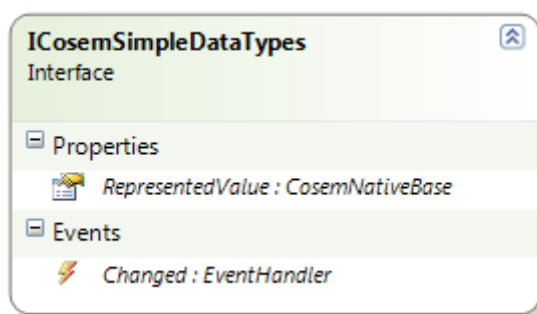
Slika 21: vmesnik *INotifyIsDirty*

Spremembe se prenašajo po hierarhiji objektov navzgor in navzdol s pomočjo lastnosti oče (ang: parent) za prenašanje navzgor oz. seznamom članov (ang: childs) za navzdol.

6.4 Prikaz objektov COSEM

6.4.1 Grafične kontrole

Za prikaz in urejanje podatkovnega modela oziroma objektov COSEM, smo izdelali grafične kontrole, ki jim damo direkten dostop do delov podatkovnega modela. Uporabljene so večinoma standardne kontrole ogrodja .NET [14] in nekaj kontrol iz knjižnic sistema SEP2W System, ki pa dodatno še implementirajo vmesnik *ICosemSimpleDataTypes*. Slika 22 prikazuje vmesnik za grafične kontrole, ki prikazujejo podatkovni model COSEM:



Slika 22: vmesnik *ICosemSimpleDataTypes*

Ob nastavitvi lastnosti *RepresentedValue* kontrola prikaže vsebino, potem pa ob uporabnikovi spremembi vrednosti v kontroli, posodobi vrednost v podatkovnem modelu.

Za lažje razumevanje podajamo primer kontrole za prikazovanje podatkovnih tipov za datum in čas, ki so definirani v [6, 23-25]. Kontrola podpira podatkovne tipe TIME (octet-string dolžine 4 zloge), DATE (octet-string dolžine 5 zlogov) in DATETIME (octet-string dolžine

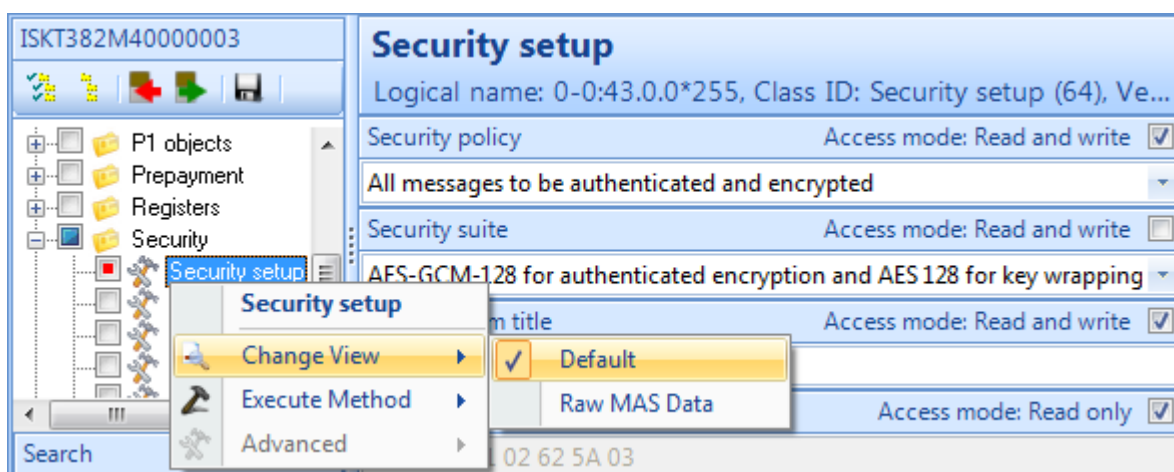
12 zlogov). Slika 23 prikazuje kontrolo za prikaz podatkovnega tipa DATETIME, vidna je tudi vrednost iz podatkovnega modela, ki jo prikazuje kontrola (FF FF 03 FE 00 02 00 00 00 80 00 FF).

Slika 23: kontrola za prikaz COSEM datuma in časa

Izdelali smo generične kontrole za vse preproste podatkovne tipe ter za nekatere zapletene. Pri tem smo veliko uporabljali refleksijo. Te kontrole smo potem uporabili za izdelavo večine ostalih bolj zapletenih kontrol.

6.4.2 Sistem pogledov

Z namenom omogočanja različnih pogledov vmesniških razredov smo izdelali sistem pogledov. Tako je za vsak vmesniški razred možnih več pogledov, informacija o vseh možnih pogledih za določen vmesniški razred pa je shranjena v samem podatkovnem modelu vmesniških razredov (Slika 17, lastnost Views). Vsi vmesniški razredi imajo na voljo vsaj surovi pogled (ang.: Raw MAS data), ki pokaže vrednosti atributov v xml obliki in je namenjen predvsem razvijalcem naprav COSEM, saj prikaže tudi attribute vmesniških razredov, ki sploh še niso podprti ali pa zanje še ni na voljo ustrezne kontrole. Slika 24 prikazuje pogled vmesniškega razreda *SecuritySetup* z izdelanimi kontrolami:



Slika 24: pogled vmesniškega razreda z izdelanimi kontrolami

Slika 24 prikazuje tudi pomen lastnosti *IsSelected* (Slika 18), katere namen je označevanje izbranosti atributov. Izbranost atributa vpliva na vse operacije, ki zajemajo attribute (npr. branje in pisanje v/iz naprave, izvoz podatkov, ...).

Slika 25 prikazuje surovi pogled enakih podatkov:

Security setup	
Logical name: 0-0:43.0.0*255, Class ID: Security setup (64), Version: 0	
Security policy	Access mode: Read and write ☒
<code><enum NativeType="Enum">3</enum></code>	
View Edit Reset	
Security suite	Access mode: Read and write ☐
<code><enum NativeType="Enum">0</enum></code>	
View Edit Reset	
Client system title	Access mode: Read and write ☒
<code><hexBinary NativeType="OctetString"/></code>	
View Edit Reset	
Server system title	Access mode: Read only ☒
<code><hexBinary NativeType="OctetString">49534B0102625A03</hexBinary></code>	
View Edit Reset	

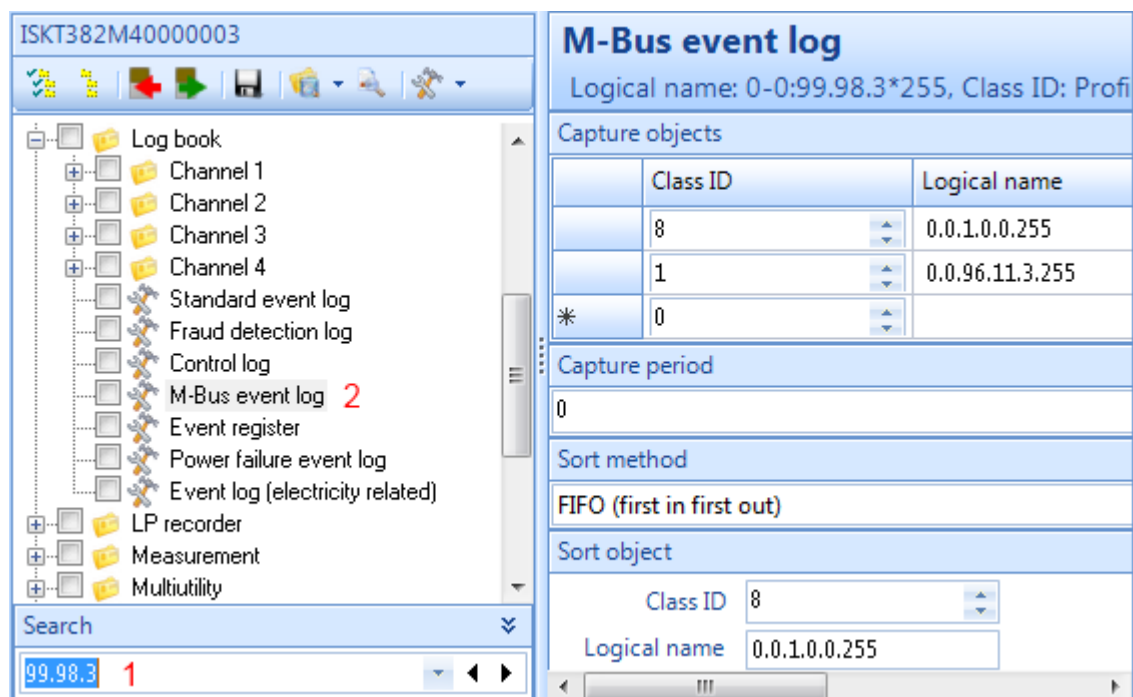
Slika 25: pogled vmesniškega razreda v surovi obliki

6.4.3 Iskanje

Naprave lahko vsebujejo veliko število objektov COSEM, zato smo izdelali možnost iskanja po vseh objektih naprave. Iskanje deluje po drevesni strukturi objektov. Vsa vozlišča so najprej linearizirana, potem pa iskanje lahko poteka naprej ali nazaj po naslednjih kriterijih:

- celotnem ali delnem imenu instance objekta,
- celotnem ali delnem logičnem imenu instance objekta,
- celotnem ali delnem imenu vmesniškega razreda.

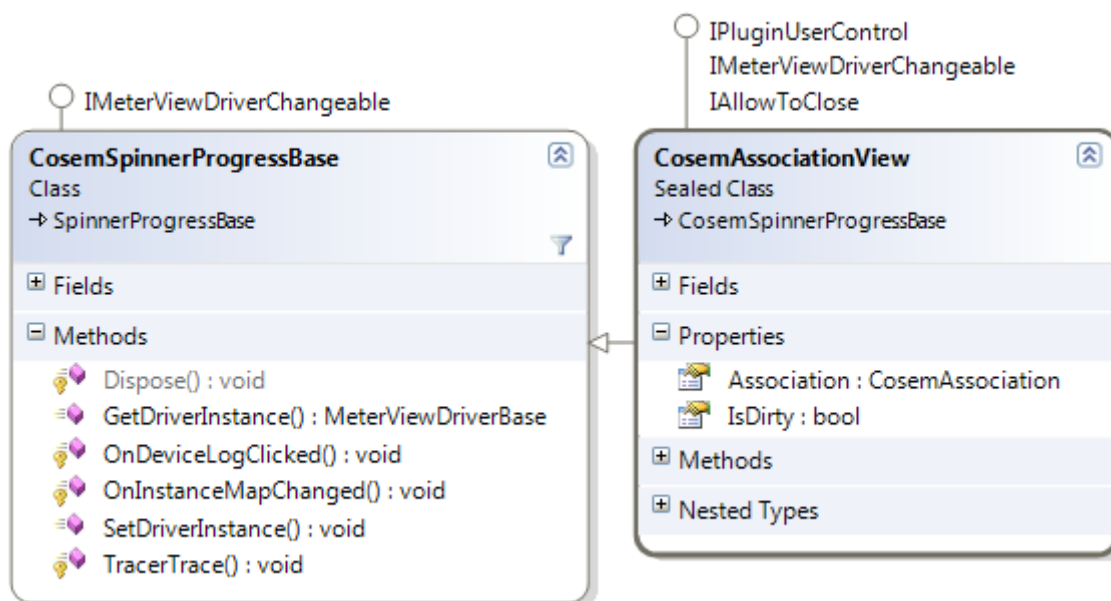
Slika 26 prikazuje izgled iskanja po delu logičnega imena instance objekta, z oznako 1 je označen prostor za vpis iskalnega niza, z oznako 2 pa rezultat iskanja:



Slika 26: iskanje po delu logičnega imena instance objekta

Za prikaz COSEM objektov naprave ter prikazovanje poteka komunikacije smo izdelali kontrolo, ki prikazuje objekte naprave v drevesni strukturi po skupinah (poglavje 6.3.1). Omogoča tudi branje iz naprave, vpisovanje, shranjevanje ter različne izvoze podatkov. Razred kontrole se imenuje *CosemAssociationView* in deduje iz bazne kontrole *CosemSpinnerProgressBase*, ki omogoča sledenje poteka komunikacije.

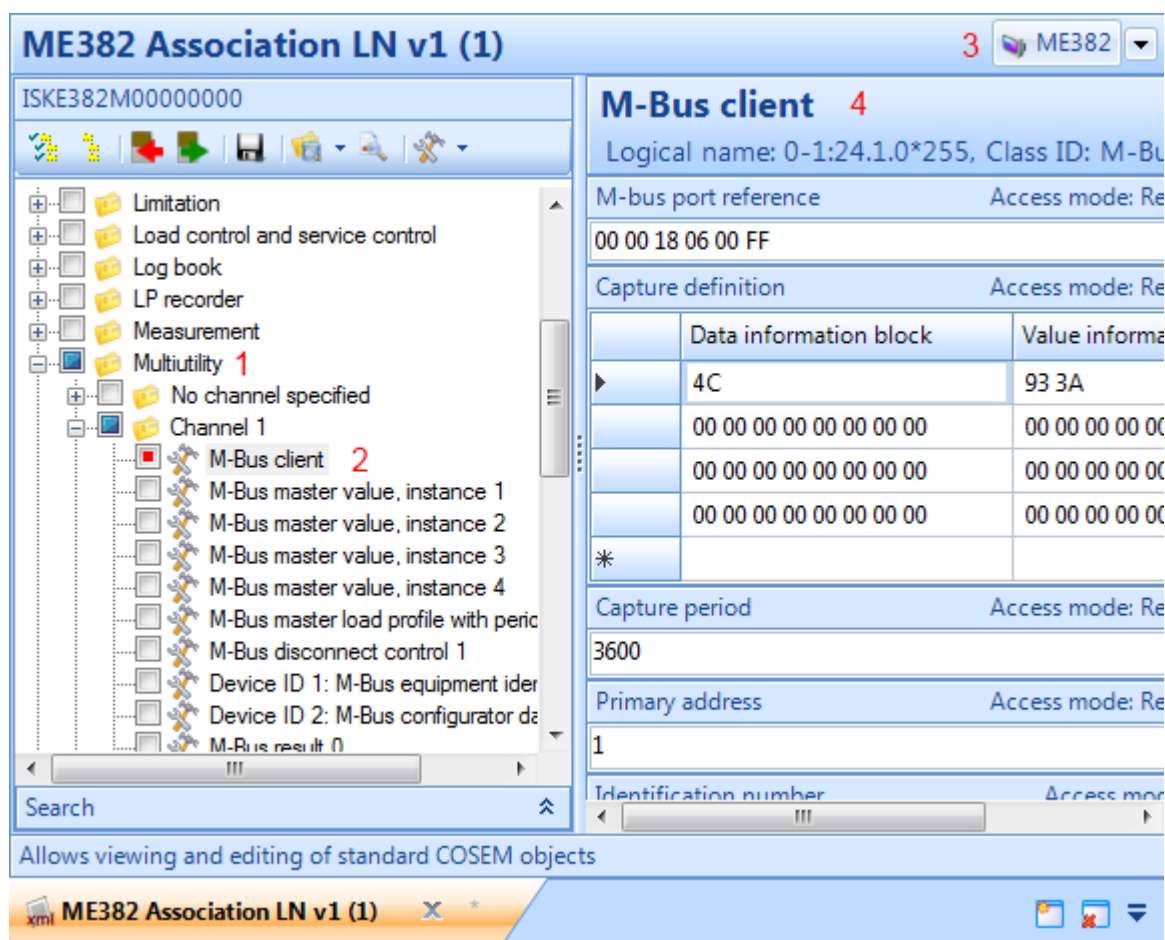
Slika 27 prikazuje razreda omenjenih kontrol:



Slika 27: razred kontrole za prikazovanje COSEM objektov naprave

Kontrola *CosemAssociationView* implementira vmesnik *IPluginUserControl*, kar ji omogoča da je prikazana kot kontrola za SEP2 Manager vtičnik in ji da možnost, da se odziva na zahteve iz glavnega menija kot so npr. osveži, uporabi, itd. Slika 28 prikazuje izgled kontrole za prikazovanje COSEM objektov naprave, oznake pa pomenijo:

- 1 – vozlišče skupine, ki vsebuje podskupine,
- 2 – vozlišče ki predstavlja objekt COSEM,
- 3 – možnost izbora naprave za komunikacijo,
- 4 – grafični prikaz izbranega objekta COSEM.

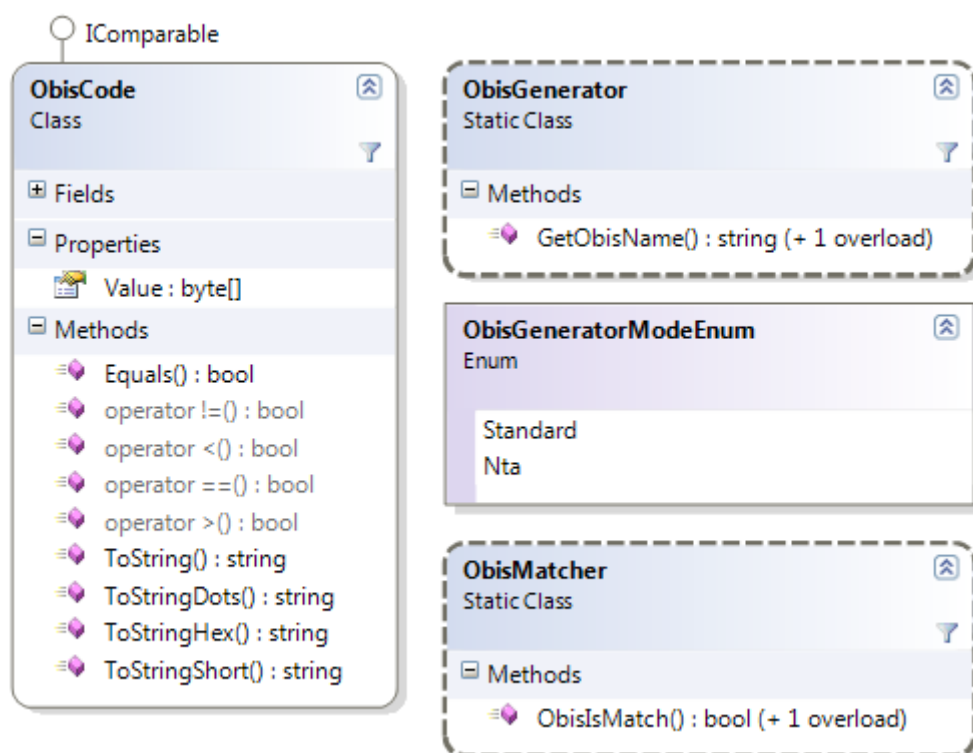


Slika 28: kontrola za prikazovanje COSEM objektov naprave

6.5 Generator imen za standardne kode OBIS

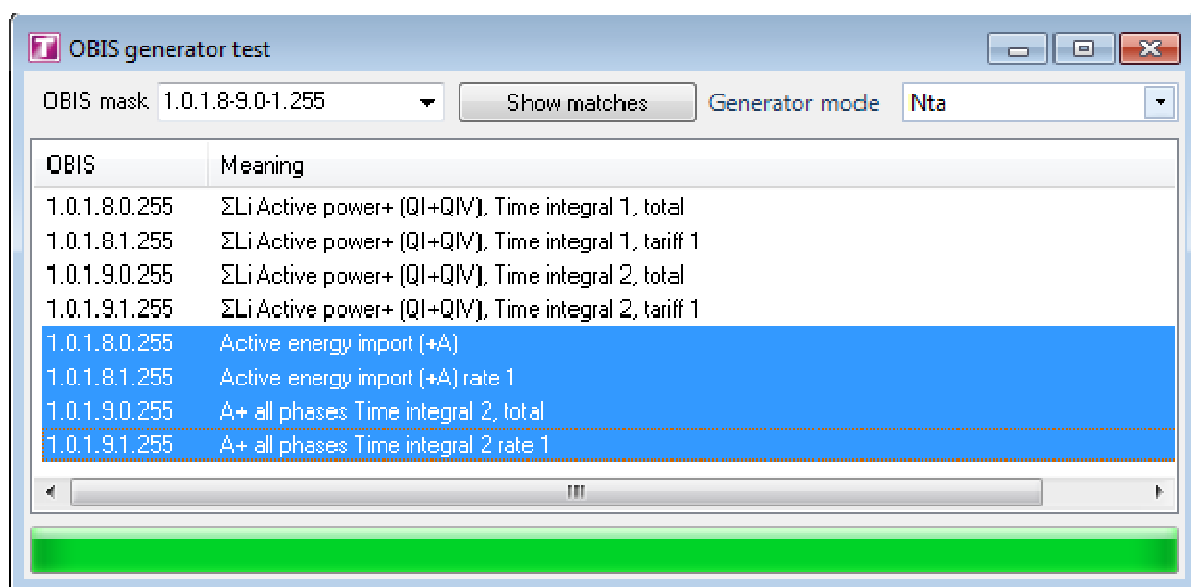
Za poimenovanje objektov COSEM, ki imajo ime standardno določeno v [5, strani 173-244], smo izdelali generator imen. Za predstavitev kode OBIS smo najprej izdelali razred *ObisCode*, ki vsebuje kodo OBIS in metode za delo z njo. Generator imen smo izdelali v obliki statičnega razreda *ObisGenerator*, ki ima na zunaj vidno samo metodo *GetObisName*. Ta metoda kot parameter sprejme instanco razreda *ObisCode*, vrne pa lokalizirano ime objekta COSEM. V zahtevah za vrata P3 [5] se nekatera imena objektov razlikujejo od standardnih, zato smo metodi dodali še parameter tipa *ObisGeneratorModeEnum*, s katerim smo omogočili različne načine delovanja generatorja.

Slika 29 prikazuje omenjene razrede:



Slika 29: generator imen in razredi za delo s kodami OBIS

Za testiranje generatorja imen smo izdelali tudi testno orodje, ki omogoča generiranje imen z masko. Kot pomožni razred smo zato izdelali *ObisMatcher*, ki vrne ustreznost kode OBIS podani maski. Slika 30 prikazuje testno orodje in različne načine delovanja. Prve 4 vrednosti smo generirali v standardnem, ostale (označeni z modro) pa v NTA načinu.



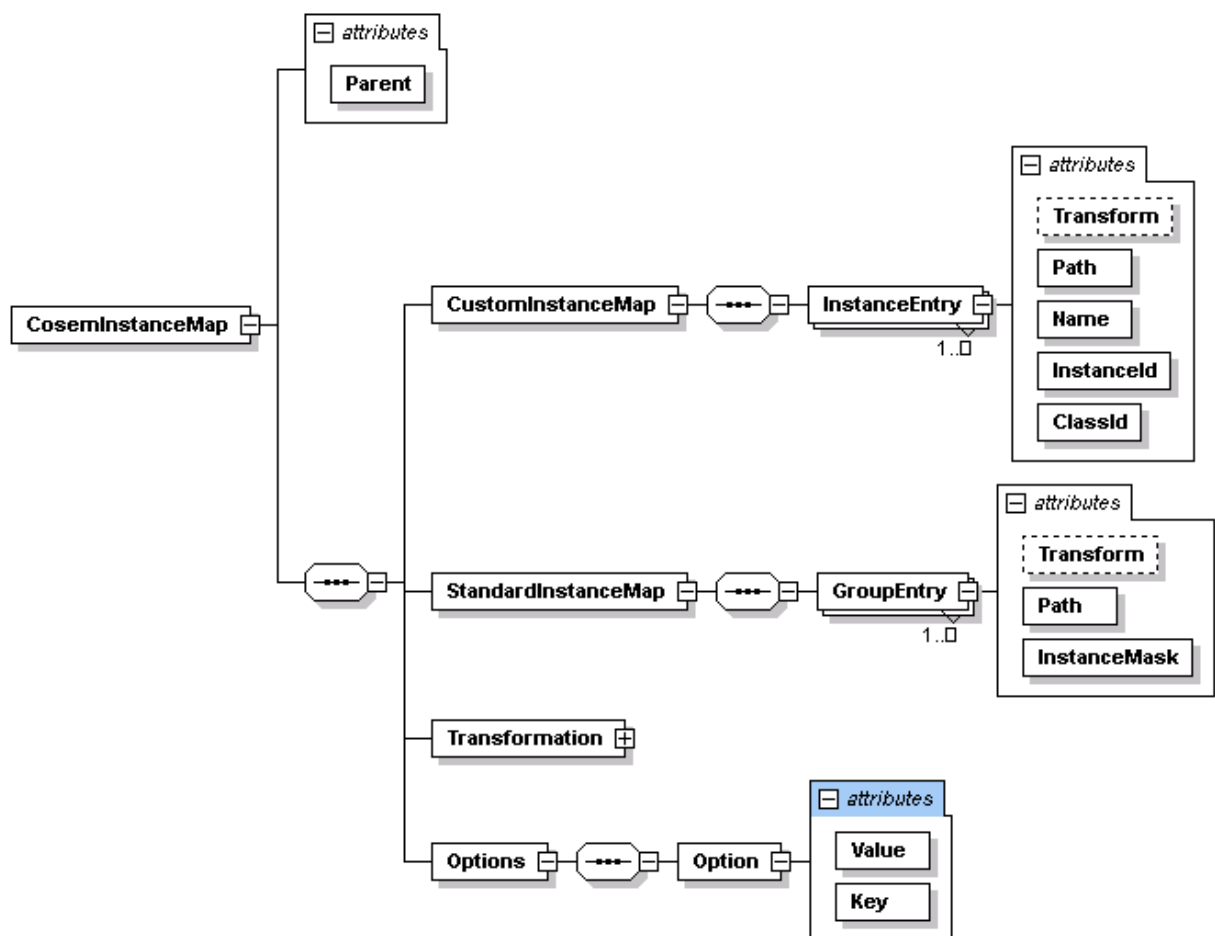
Slika 30: testno orodje za generator imen

6.6 Načrt instanc

V prejšnjem poglavju smo opisali generator imen za standardne objekte. Merilne naprave pa vsebujejo tudi objekte COSEM, ki nimajo standardnih imen. Da bi zagotovili način za poimenovanje takih objektov, poleg tega pa še razvrščanje vseh objektov v drevesno strukturo in ostale, za napravo specifične lastnosti, smo izdelali načrt instanc (ang.: instance map).

Načrt instanc je xml datoteka, v kateri lahko prekrijemo katerakoli imena objektov in dodamo imena nestandardnim objektom COSEM. Omogoča tudi razvrščanje objektov COSEM v skupine, ki so prikazane v drevesni strukturi. Vsebuje odprt sistem opcij za različne nastavitve. S pomočjo lastnosti *Parent* je možno zaporedno nizati več datotek in jim dodajati elemente, ki še bolj podrobno opisujejo napravo (npr. ko imamo osnovno datoteko, ki vsebuje vse kar je skupno za celo družino naprav COSEM, potem pa za posamezne naprave naredimo še eno datoteko, ki ima lastnost *Parent* nastavljeno na osnovno datoteko in lahko prekrije ali doda nekaj elementov). Na ta način se izognemo podvajanju podatkov, uporabnikom pa omogoča tudi, da si prilagodijo razporeditev skupin brez da bi spremembe izgubili ob namestitvi nove verzije programske opreme.

Slika 31 prikazuje XSD shemo xml datoteke v grafični obliki:

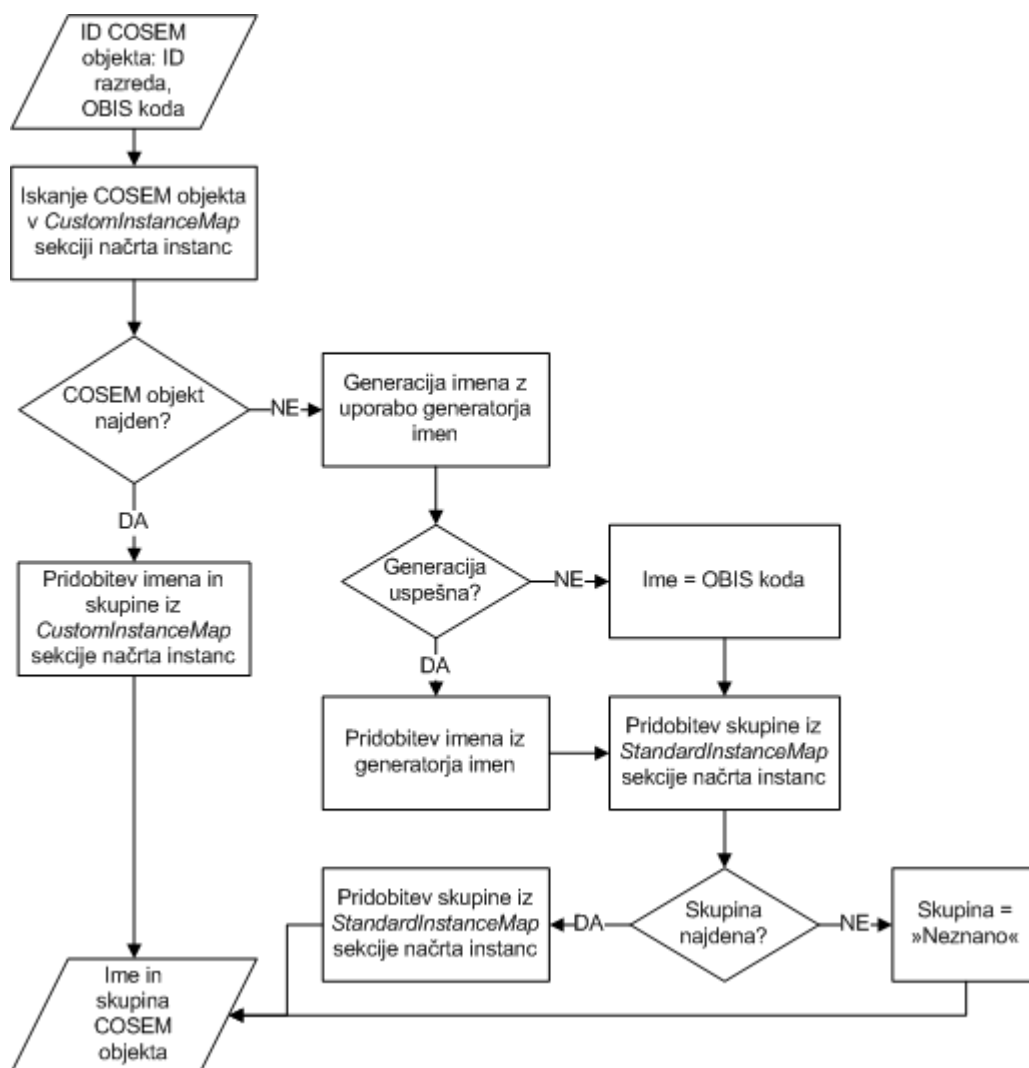


Slika 31: XSD shema načrta instanc

Celotno sekcijo *Transformation* bomo razložili v naslednjem poglavju o transformacijah.

Ko gradimo drevesno strukturo COSEM objektov naprave, uporabimo načrt instanc ter generator kod OBIS za pridobitev imena in skupine posameznega objekta COSEM iz ID razreda in kode OBIS.

Slika 32 prikazuje proces pridobivanja imena in skupine za objekt COSEM:



Slika 32: proces pridobivanja imena in skupine za COSEM objekt

6.7 Transformacije

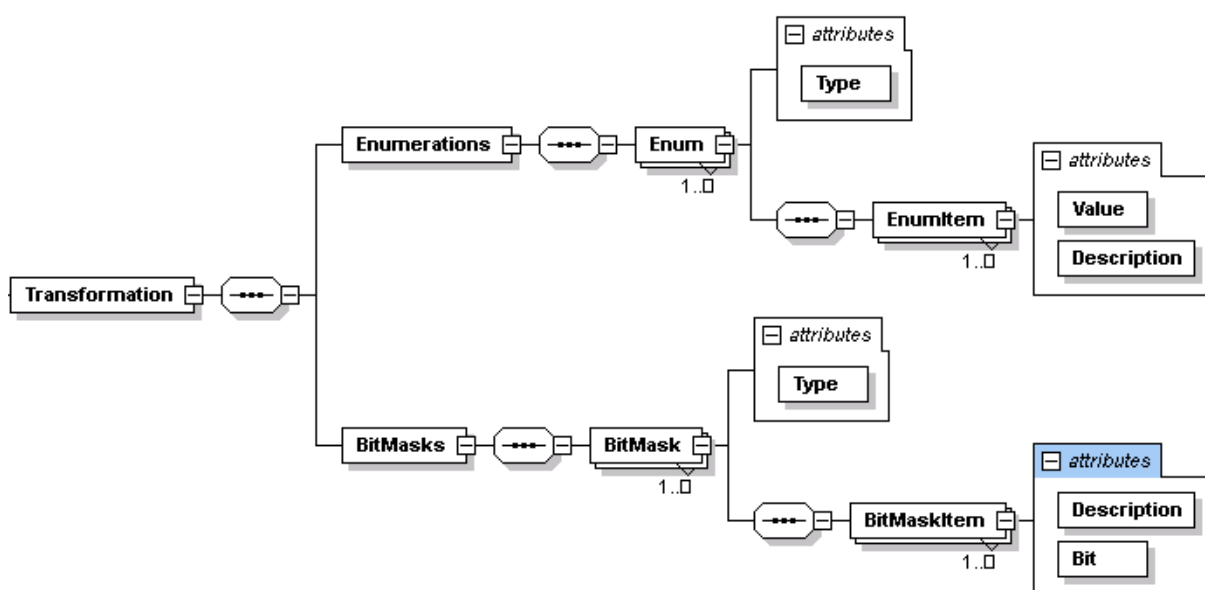
6.7.1 Splošno

Nekateri vmesniški razredi COSEM so že po svoji naravi definirani tako, da dopuščajo različne interpretacije podatkov. Tak je npr. vmesniški razred *Data* [6, stran 31], ki vsebuje samo en atribut kateregakoli podatkovnega tipa COSEM. Izbira je prepuščena proizvajalcu, ki lahko uporabi poljuben podatkovni tip COSEM in ga interpretira po svoje. Nekateri tipi interpretacije so zelo pogosti, zato smo uvedli sistem transformacij. Transformacije omogočajo, da v načrtu instanc (poglavje 6.6) za posamezen objekt COSEM (Slika 31, element *InstanceEntry*) ali skupino objektov (Slika 31, element *GroupEntry*) določimo kako

naj se transformira, vsebuje pa tudi podatke za nekatere transformacije. Različne transformacije bomo opisali v naslednjih poglavjih.

6.7.2 Enumeracije

Kadar je v napravi objekt COSEM vmesniškega razreda *Data*, ki ima atribut tipa enum ali katerikoli številski (unsigned, double-long-unsigned,...), vsaka vrednost pa ima določen pomen, lahko za transformacijo uporabimo enumeracijo. Deluje tako, da v načrtu instanc vpišemo v atribut *Transform* (Slika 31) tip enumeracije, sam tip pa v obliki opisov pomena za posamezne vrednosti dodamo v sekcijo *Enumerations*. Slika 33 prikazuje del XSD sheme načrta instanc za transformacije:



Slika 33: del XSD sheme načrta instanc za transformacije

Slika 34 prikazuje primer transformacije. Na sliki je vidno, da je tip atributa *unsigned* (oznaka 1), pa je kljub temu zaradi transformacije prikazan kot enumeracija (oznaka 2), pri kateri ima vsaka številka določen pomen.

Load control mode
 Logical name: 0-0:128.30.0*255, Class ID: Data (1), Version: 0

Value : Unsigned **1** Access mode: Read and write ☐

0 - Normal **2**

0 - Normal
 1 - Switch on delayed
 2 - Switch on random delayed
 3 - Switch on delayed with power on delay
 4 - Switch on delayed with power on random delay
5 - Switch on random delayed with power on delay
 6 - Switch on random delayed with power on random delay
 255 - Not defined

Slika 34: primer transformacije tipa enumeracija

6.7.3 Bitne maske

Bitne maske se uporabljajo za attribute številskega tipa, pri katerih imajo pomen posamezni biti v številu. Slika 33 prikazuje del XSD sheme za bitne maske. Slika 35 prikazuje primer transformacije za NTA alarme [5, stran 29]. Na sliki vidimo, da je tip atributa double-long-unsigned (oznaka 1), s transformacijo pa je prikazan pomen posameznih bitov.

Alarm register object 1
 Logical name: 0-0:97.98.0*255, Class ID: Data (1), Version: 0

Value : DoubleLongUnsigned **1** Access mode: Read and write ☐

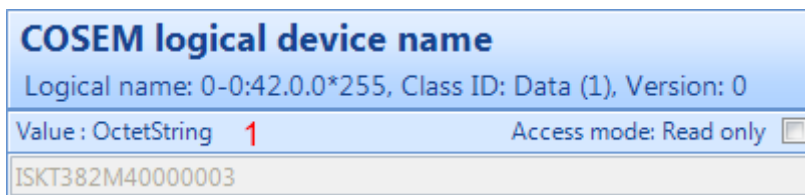
	Description	ON/OFF
B8	Program memory error	☐
B9	RAM error	☐
B10	NV memory error	☐
B11	Measurement system error	☐
B12	Watchdog error	☐
B13	Fraud attempt	☐
B14	Unused	☐
B15	Unused	☐

Slika 35: primer transformacije tipa bitna maska za NTA alarme

6.7.4 Octet-string kot ASCII

Podatkovni tip octet-string je definiran kot urejena vrsta zlogov [6, stran 22], v napravah pa je včasih uporabljen tudi za prikaz ASCII znakov. Za prikaz podatkovnega tipa octet-string z ASCII znaki lahko uporabimo transformacijo, ki namesto urejene vrste oktetov prikaže ASCII

znake. Slika 36 prikazuje primer takega podatka, kjer je z oznako 1 označen tip atributa octet-string, spodaj pa vidimo prikazane ASCII znake:

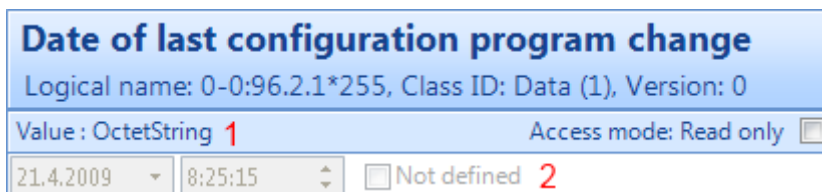


COSEM logical device name
 Logical name: 0-0:42.0.0*255, Class ID: Data (1), Version: 0
 Value : OctetString 1 Access mode: Read only ☐
 ISKT382M40000003

Slika 36: primer transformacije tipa octet-string kot ASCII

6.7.5 Octet-string kot datum in čas

V napravah je octet-string lahko uporabljen tudi za predstavitev datumov in časov, kot je to definirano v [6, stran 23]. Da bi omogočili prikaz in urejanje takih podatkov, smo izdelali transformacijo, ki prikazuje octet-string kot datum in čas. Slika 37 prikazuje primer, z oznako 1 smo označili podatkovni tip, z oznako 2 pa kontrolo za prikaz:



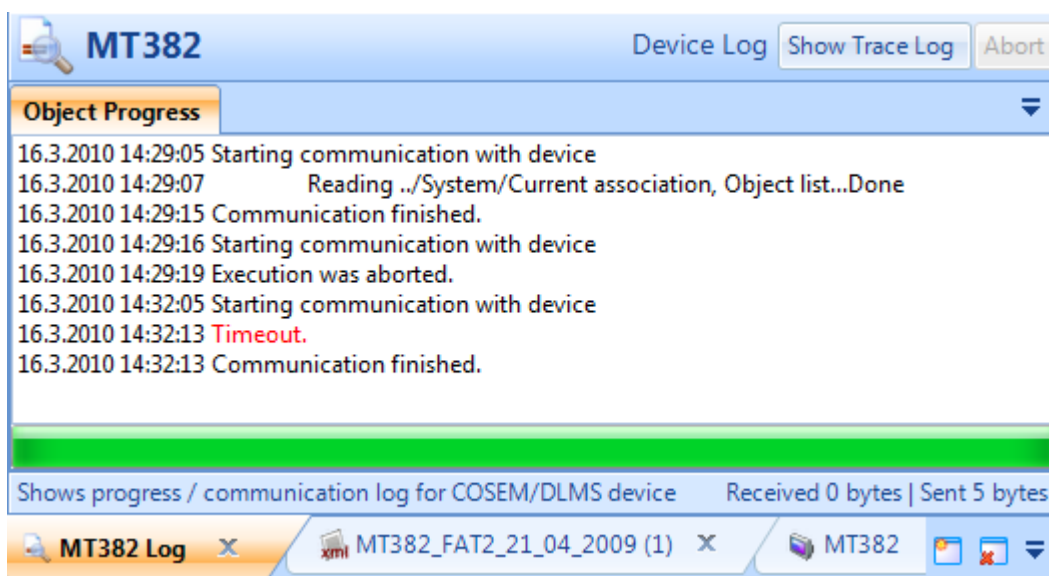
Date of last configuration program change
 Logical name: 0-0:96.2.1*255, Class ID: Data (1), Version: 0
 Value : OctetString 1 Access mode: Read only ☐
 21.4.2009 8:25:15 ☐ Not defined 2

Slika 37: primer transformacije tipa octet-string kot datum in čas

6.8 Beleženje dogodkov

SEP2W System ima v okviru beleženja dogodkov (ang.: event logging) izdelan tudi sistem za sledenje (ang.: tracing) s pomočjo sledljivih akcij. S pomočjo sledenja smo izdelali beleženje dogodkov, ki jih beleži SEP2 Meter Access Service, na nivoju gonilnika DLMS/COSEM.

Slika 38 prikazuje kontrolo za beleženje:

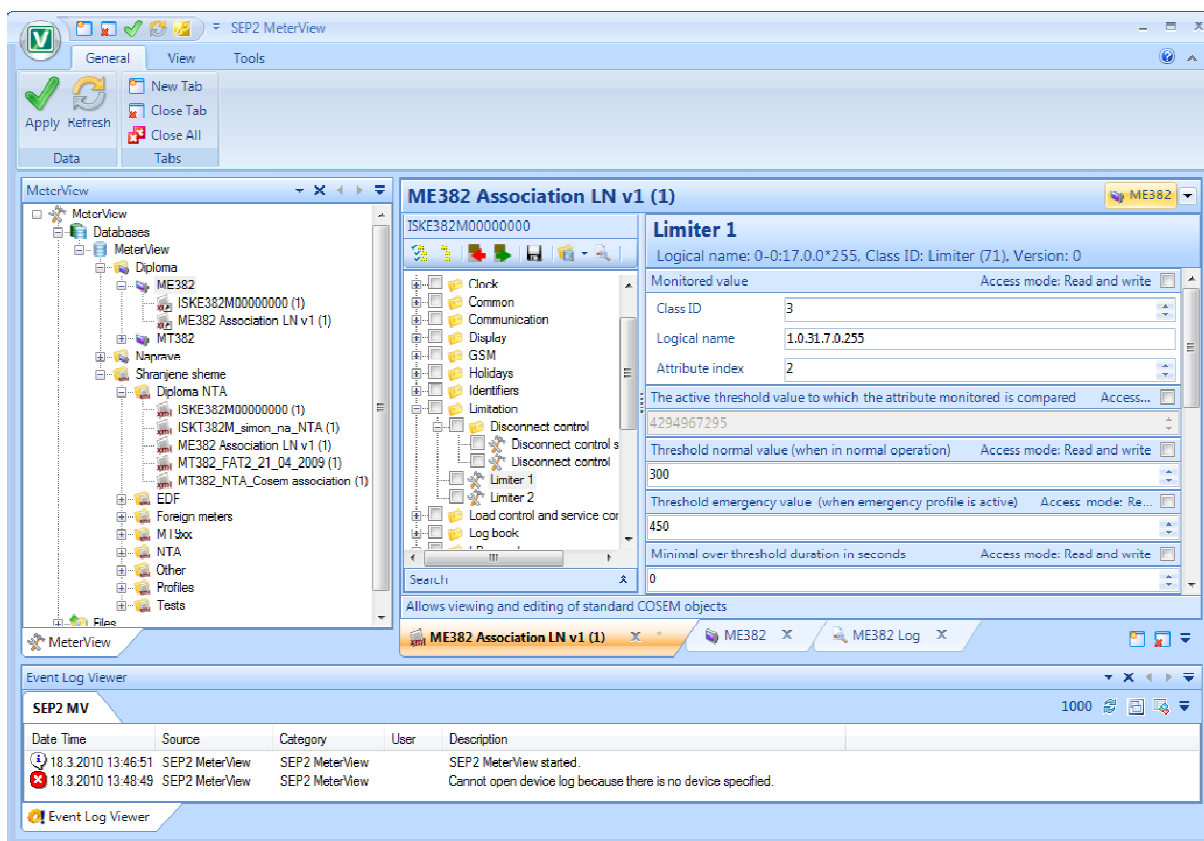


Slika 38: beleženje dogodkov gonilnika DLMS/COSEM

6.9 Samostojna aplikacija

Zaradi zapletene namestitve celotnega programskega paketa SEPW System skupaj z vsemi potrebnimi storitvami, smo aplikacijo SEP2 Manager razdelili na aplikacijsko ogrodje in knjižnico, ki vsebuje večino funkcionalnosti. Vse potrebne storitve za delovanje smo zbrali v knjižnice .NET (ang. .NET assemblies) in tako je nastala aplikacija SEP2 MeterView, ki omogoča enostavno namestitev in vsebuje samo vtičnik MeterView, opisan v diplomskem delu. S tem smo dosegli, da je vtičnik na voljo kot del sistema SEP2 System, v nespremenjeni obliki pa tudi kot samostojna aplikacija.

Slika 39 prikazuje uporabo aplikacije SEP2 MeterView za nastavljanje parametrov naprave izdelane po zahtevah za nizozemski trg:



Slika 39: prikaz uporabe programske opreme izdelane v diplomski nalogi

7 Zaključek

Na Nizozemskem so v želji po zmanjšanju porabe energije izdelali specifikacije za pametne merilne sisteme. V diplomski nalogi smo izdelali programsko opremo za nadzor in upravljanje pametnih merilnih naprav izdelanih po teh specifikacijah.

Zaradi zelo splošnega pristopa programska oprema omogoča nadzor in upravljanje vseh naprav izdelanih po specifikaciji DLMS/COSEM. S pomočjo načrtov instanc lahko z malo truda in v zelo kratkem času dodajamo naprave. Dodajanje novih vmesniških razredov je preprosto in pregledno, saj je večinoma potrebno narediti samo podatkovni model ter prikaz s pomočjo pripravljenih kontrol.

Že med razvojem smo dobili kar nekaj idej za izboljšave. Ena izmed njih je izboljšava povezav med podatkovnim modelom COSEM in kontrolami za prikaz, ki je sedaj narejena tako, da ima vsak model vmesniškega razreda COSEM seznam kontrol, ki so primerne za prikaz. Za lažjo izdelavo novih pogledov za razrede COSEM tudi za zunanje uporabnike, bi bilo možno izdelati vmesniški sistem, ki bi možne kontrole za poglede objektov COSEM pridobival iz DLL knjižnic s pomočjo refleksije.

Beleženje dogodkov bi bilo možno prestaviti v bazni gonilnik, tako da bi bile vse napake in skupni dogodki beleženi na enem mestu za vse gonilnike, gonilniki pa bi dodali specifične vnose.

Programska oprema SEP2 MeterView, ki smo jo izdelali v diplomskem delu se uporablja v razvojnem oddelku podjetja Iskraemeco, kot trenutno edino orodje za delo z napravami izdelanimi po zahtevah za nizozemski trg, pa jo uporabljajo tudi največji nizozemski distributerji in upravljavci omrežij (Alliander, Enexis, Eneco, Delta, Edinet, Cogas) ter v združenju nizozemskih elektro distribucij (Netbeheer). Kot pomoč pri izdelavi programskih rešitev za daljinsko zbiranje podatkov jo uporablja tudi podjetje Netinium.

Literatura

- [1] NTA 8130:2007, Minimum set of functions for metering of electricity, gas and thermal energy for domestic customers, Nederlands Normalisatie-instituut, 2007, www.nen.nl/cmsprod/groups/public/documents/bestand/244558.pdf, dostopnost preverjena 2010.
- [2] Dutch Smart Meter Requirements v2.31 final Main.
- [3] Dutch Smart Meter Requirements v2.31 final P1.
- [4] Dutch Smart Meter Requirements v2.31 final P2.
- [5] Dutch Smart Meter Requirements v2.31 final P3.
- [6] DLMS UA, 1000-1 Blue book, 9. izdaja, osnutek 0.7, 2009.
- [7] DLMS UA, 1000-2 Green book, 3. izdaja, 2002
- [8] IEC 62053-61: Object identification system (OBIS), IEC standard, 2006.
- [9] SWS_SEP2WSystem.doc, interna specifikacija podjetja Iskraemeco d.d.
- [10] SWS_SEP2MeterAccess.doc, interna specifikacija podjetja Iskraemeco d.d.
- [11] Microsoft, MSDN Library. Dostopno na: <http://msdn.microsoft.com/library>
- [12] C. Nagel, B. Evjen, J. Glynn, M. Skinner, K. Watson, Professional C# 2008, Wiley Publishing, Inc., Indianapolis, 2008
- [13] MAS Classes Description, interna specifikacija podjetja Iskraemeco d.d.
- [14] Microsoft, .NET ogrodje. Dostopno na http://en.wikipedia.org/wiki/.NET_Framework

Izjava

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Boštjana Murovca. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Matjaž Kavčič

Dražgoše, 30. 3. 2010