

# Semantično modeliranje podatkov z grafnimi bazami za podporo interoperabilnosti v pametnih omrežjih

Amila Dervišević<sup>1</sup>, Matej Zajc<sup>2</sup>, Nermin Suljanović<sup>1,3</sup>

<sup>1</sup> Elektroinštitut Milan Vidmar, Ljubljana, Slovenija

<sup>2</sup> Univerza v Ljubljani, Fakulteta za elektrotehniko, Slovenija

<sup>3</sup> Univerza v Tuzli, Fakulteta za elektrotehniko, Bosna in Hercegovina

E-pošta: amila.dervisevic@eimv.si

**Povzetek.** Postopek digitalizacije dobavne verige z električno energijo in uvedba pametnega omrežja kot koncepta, ki bo olajšal vključitev obnovljivih virov energije in električnih vozil v obstoječe elektroenergetsko omrežje, sta privedla do znatnega povečanja količine podatkov in izmenjave informacij med deležniki. Članek obravnava semantično modeliranje podatkov o električni energiji in shranjevanje podatkov v grafne podatkovne baze za nadaljnje hitro in zanesljivo iskanje informacij. Skupni informacijski model (CIM) omogoča semantično interoperabilnost podatkov o električni energiji, ki se nanašajo na različne procese in časovne skale od realnega časa do dolgoročnega načrtovanja. Ker je okvir za opis virov (RDF) učinkovito sredstvo za serializacijo (pretvorbo) podatkov na semantičnem spletu, in so trenutno veliki podatki o električni energiji pogosto serializirani v tej obliki, si prizadevamo za globlje razumevanje RDF. V naslednjem koraku bomo proučili metodologijo shranjevanja podatkov, serializiranih v formatu RDF/XML, v grafno podatkovno bazo in pristope za iskanje podatkov iz baz podatkov, ki jih zahtevajo drugi poslovni procesi. V okviru praktičnega dela bodo modeli distribucijskih mrež v formatu RDF postavljeni v dve grafni bazi podatkov z različno zasnovano (Neo4j in GraphDB) in ovrednotene bodo njihove zmogljivosti.

**Gljučne besede:** CIM, pametna omrežja, interoperabilnost, grafne baze podatkov, RDF

## Semantic data modelling with graph databases enabling interoperability in smart grids

Digitalization of the power supply chain and introduction of smart power grids to integrate renewable energy sources and electric vehicles into the existing power grid have led to a significant increase in the data and information exchange between stakeholders. The paper studies semantic modeling of the power data and its storing in graph databases to enable its fast and reliable retrieval. A common information model enables semantic interoperability of power data related to different processes and time scales, either real or long-term planning. The resource description framework (RDF) is an effective means to serialize data on the Semantic Web. The paper examines the methodology for storing the data serialized as an RDF/XML file in a graph database and approaches for data retrieval data from databases to be used by other business processes. As part of the practical work of this paper, RDF distribution network models will be placed in two graph databases of different concepts (Neo4j and GraphDB) and their performance will be compared and evaluated.

**Keywords:** CIM, smart grid, interoperability, graph database, RDF

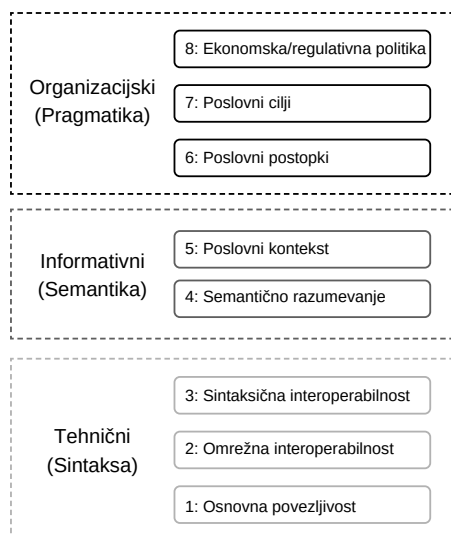
## 1 UVOD

Pametna omrežja temeljijo na podatkih, zato so modeliranje, razumevanje in uporaba podatkov pomembne

naloge za pravilno upravljanje omrežja. Rast števila deležnikov v komunikaciji vodi do rasti količine izmenjanih podatkov. Vsak deležnik na področju električne energije lahko uporablja drugačen pristop modeliranja, kar predstavlja omejitve za interoperabilnost podatkov.

IEEE definira interoperabilnost na sistemski ravni kot *spodobnost dveh ali več sistemov ali komponent za izmenjavo informacij in uporabo informacij, ki so bile izmenjane*, na podatkovni ravni pa je definirana kot *“spodobnost povezovanja, združevanja in obdelave dveh ali več naborov podatkov”* [1]. S problematiko interoperabilnosti podatkov se srečujemo na številnih področjih, en primer je semantični splet (angl. Semantic Web). Skupne semantične modele, kot temelj za pametna omrežja, podpira tudi GridWise Architecture Council (GWAC) [2]. Semantika je osrednji sloj sklada GWAC, ki je sestavljen iz osmih slojev, razvrščenih v tri skupine, kot prikazuje slika 1. Če pogledamo podrobneje, četrta plast govori o pomenskem razumevanju in nam predstavlja izhodišče za izboljšanje interoperabilnosti podatkov. Semantični model je strukturiran opis semantike niza informacij, ker so podatki enolično in natančno opredeljeni, tako kot tudi njihova povezava z drugimi podatki. Semantični pristop lahko pomaga pri obravnavi raznolikosti udeležencev in tipov podatkov, zato poudarja interoperabilnost [3]. Semantična interoperabilnost uvaja metapodatke, ki jih pogosto imenujemo *podatki o*

podatkih, kar zagotavlja pravilno interpretacijo informacij, natančen opis in edinstveno identifikacijo. Pravilna interpretacija omogoča, da podatek postane informacija [4]. Koncept enolične identifikacije fragmentov podatkov je ključnega pomena za ravnanje z velikimi podatki, celotna ideja semantičnega spleta temelji na enotnih identifikatorjih virov (URI-jev). V dobi pametnih omrežij in interneta stvari (IoT) semantika še nikoli ni bila tako pomembna.



Slika 1: Sklad GWAC [2].

Za popolno interoperabilnost in homogenost okolja moramo skladno z GWAC zagotoviti [5]:

- Zelo podroben model, ki opisuje elektroenergetski sistem.
- Format datoteke, ki lahko hrani razširjene podatke, brez vpliva na osnovne podatke.
- Proizvajalci programske opreme in energetska podjetja morajo sprejeti ta podatkovni model ter ga vključiti v svoje poslovanje.

Skupni informacijski model (angl. Common Information Model) (CIM) lahko izpolni prvo zgornjo zahtevo, medtem ko razširljiv označevalni jezik (angl. eXtensible Markup Language) (XML) v kombinaciji z okvirom za opis virov (angl. Resource Description Framework) (RDF) ponuja rešitev za drugo zahtevo. Tretja zahteva se lahko šteje bolj za komercialen in regulativen izziv kot tehnični.

## 2 SKUPNI INFORMACIJSKI MODEL (CIM)

Podatkovni model, ki se uporablja za opis elektroenergetskega sistema, mora podpirati semantiko in biti dovolj podroben, da lahko celovito predstavimo kompleksen sistem. Uveljavljen model, ki izpolnjuje vsa merila, je skupni informacijski model CIM [6]. CIM uvaja semantično interoperabilnost podatkov o električni ener-

giji in opredeljuje vmesnike, ki omogočajo izmenjavo podatkov med različnimi aplikacijami.

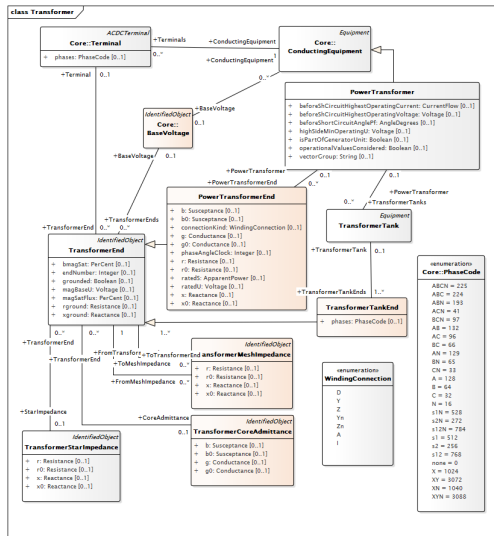
CIM je prvič uporabil NERC (severnoameriški svet za zanesljivost električne energije) kot format izmenjave podatkov med energetske podjetji. Danes ENTSOE (evropsko omrežje operaterjev prenosnih sistemov za električno energijo) razvija sistem CIM v Evropi, ki postaja vse bolj razširjen po vsem svetu [6]. Leta 2005 je bila ustanovljena skupina uporabnikov CIM (CIMug) [7]. CIMug je forum, na katerem lahko uporabniki in svetovalci sodelujejo in uporabljajo mednarodne standarde IEC CIM za izboljšanje interoperabilnosti. Primarni namen je izmenjava tehnoloških osnov, najboljših praks in tehničnih virov ob hkratnem izboljšanju interoperabilnosti [7]. IEC razvija CIM na področju dela tehničnega odbora TC57. Tri delovne skupine so osredotočene na CIM: delovne skupine 13, 14 in 16 [5].

CIM je opisan z uporabo enotnega jezika za modeliranje UML (angl. Unified Modeling Language). UML ponuja konceptualni model ter omogoča razumevanje sistema in povezav med njegovimi moduli. To je piktoGRAFSKI jezik, ki se uporablja za ustvarjanje načrtov programske opreme. UML standardizirata in razvijata skupini za upravljanje objektov (OMG) in ISO/IEC 19501 [6]. Na voljo sta dve glavni vrsti diagramov, diagrami razredov (angl. class diagrams) UML in diagrami paketov (angl. package diagrams) UML. Uporabljajo se za grafično modeliranje različnih sistemov, pomembnih za razvijalce sistemov, modeliranje znanja ali interakcij [6]. UML modelira sistem in ponuja načine grafične predstavitve njegovih elementov, vendar ne daje dodatnih informacij, na primer kako modelirati ali uporabiti dani model [6].

Z uporabo podrobnih opisov elementov lahko sistem modeliramo tako, da je zelo podoben pravemu. Podrobni opisi so lahko v pomoč, vendar tudi v breme, saj prinašajo delo z velikimi nabori podatkov. Za modeliranje transformatorja z uporabo ontologije CIM na primer potrebujemo 11 razredov, 14 relacij in približno 40 atributov, kot je razvidno s slike 2.

## 3 OKVIR ZA OPIS VIROV (RDF)

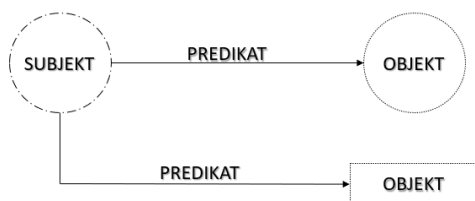
Primarni cilj pametnih omrežij je omogočiti interoperabilnost med številnimi različnimi uporabniki, entitetami, storitvami in napravami, ki imajo pomembno vlogo v sodobnem in razvitem energetskega sistemu. Nužen, a ne zadosten korak je podroben model, ki opisuje elektroenergetski sistem. CIM je bil izbran za izpolnitev te zahteve, vendar je prevelik, da bi ga lahko delili udeleženci v komunikaciji. Zato iščemo format, ki lahko shrani razširjene podatke, ne da bi vplival na osnovne podatke, ki je razumljiv tako ljudem kot strojem, zanesljiv za zelo povezane semantične podatke in v skladu z ontologijo CIM. Idealni format se imenuje Okvir za opis virov (RDF). RDF je idealna rešitev za odpravo pomanjkljivosti modela CIM. Na področju energetike se



Slika 2: Transformator, modeliran s pomočjo razredov CIM.

uporablja za serializacijo podatkov topologije omrežja [6].

RDF ima obliko usmerjenega grafa za predstavitev informacij. Najosnovnejši del modela RDF je *trojček* (angl. triple). Trojčke sestavljajo subjekt (angl. subject), predikat (angl. predicate) in objekt (angl. object). Subjekt opisuje entiteto, o kateri je izjava, predikat se nanaša na določeno lastnost te entitete, vrednost te lastnosti pa je objekt. Na sliki 3 je prikazan trojček kot usmerjen graf, kjer sta subjekt in objekt vozlišči, predikat pa predstavlja povezavo. Na sliki sta dva objekta, eden je predstavljen kot krog, in je lahko del drugega trojčka. Drugi objekt ima pravokotno obliko, in predstavlja konec te relacije.



Slika 3: Trojček kot usmerjen graf.

### 3.1 Model CIM RDF

Da bi lahko modele RDF delili in objavili, jih moramo serializirati v enega od standardiziranih formatov, kot so RDF/XML, N3, Turtle, N-triples itd [8]. V primeru uporabe CIM je najprimernejši format RDF/XML. Jezik CIM XML je aplikacija RDF za CIM, opredeljena s tremi sintaksnimi specifikacijami, to so CIM, shema RDF (RDFS) in RDF [9]. Namen RDF je enolično opisati in povezati informacije. Struktura trojčka, uporabljena v dokumentih RDF, se morda zdi preprosta, vendar je primerna za predstavitev električnega omrežja.

Za natančnejšo razlago modelov CIM RDF je bil izbran realni model omrežja, spodaj je prikazan majhen vzorec:

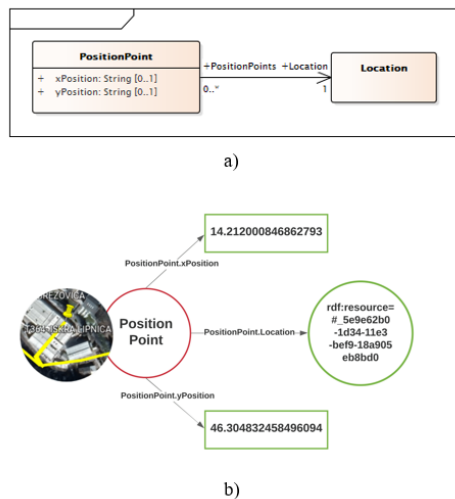
```
<rdf:RDF xmlns:cim="http://iec.ch/TC57/2010/CIM-schema-cim15#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <cim:PositionPoint
    rdf:ID="_5e9e62b1-bef9-18a905eb8bd0">
    <cim:PositionPoint.xPosition>
      14.212000846862793
    </cim:PositionPoint.xPosition>
    <cim:PositionPoint.yPosition>
      46.304832458496094
    </cim:PositionPoint.yPosition>
    <cim:PositionPoint.Location
      rdf:resource="#_5e9e62b0-bef9-18a905eb8bd0"/>
  </cim:PositionPoint>
```

Prikazan je element imenovan *PositionPoint*. Prva vrstica je skupna vsem elementom celotnega RDF/XML. V tej vrstici so definirani URI in njihova skrajšana imena (angl. qnames). Skrajšanje URI-ja omogoča lažje branje dokumenta RDF, zaradi česar je dokument uporabniku bolj prijazen. V drugi vrstici je določeno ime elementa (*PositionPoint*) in njegov ID. ID in ime sta definirana s pomočjo qnames *cim* oziroma *rdf*, da skrajša vnos in olajša branje. Imenski prostor (angl. namespace) *cim* je ključnega pomena in eno od oznak, da vedno prepozna model CIM RDF. To je standardni imenski prostor, definiran z URI *cim*: <http://iec.ch/TC57/2010/CIM-schema-cim15#>. URI se lahko razlikuje, odvisno od uporabljene različice CIM. Naslednji dve vrstici definirata lastnosti elementa. *PositionPoint* ima dve lastnosti o poziciji, *PositionPoint.xPosition* in *PositionPoint.yPosition*. Zadnja vrstica definira povezavo do drugega elementa z ID elementa, podanega kot *rdf:resource = "#\_5e9e62b0-1d34-11e3-bef9-18a905eb8bd0"*.

Slika 4 prikazuje, kako je ta kratek segment videti kot CIM UML in kako je preslikan v CIM RDF. Atributi iz razredov UML so preslikani v dobesedne objekte, medtem ko je ime atributa predikat v RDF. Razredi so preslikani v vozlišča.

## 4 GRAFNE BAZE PODATKOV

Ker je CIM zelo podroben in včasih zapleten model, je lahko ravnanje z velikimi količinami podatkov precej zahtevno za operaterje. Zdi se naivno vedno izmenjati te velike podatkovne modele, namesto da bi jih shranili v bazo podatkov, in ustreznim uporabnikom omogočili dostop le do potrebnih podatkov. Uvedba koncepta baz podatkov, prilagojenih na CIM, bi imela pomemben vpliv na delovanje in obdelavo podatkov elektroenergetskega sistema. Ker je RDF v svoji strukturi graf, smo raziskali različne metodologije za shranjevanje podatkov



Slika 4: Preslikava iz a) CIM UML v b) CIM RDF.

v grafne baze podatkov.

Grafne baze ne vsebujejo shem, struktura se prilagaja potrebam primera uporabe in s tem izboljšuje interoperabilnost [10]. Grafične zbirke podatkov so bolj objektivno usmerjene, kar pomeni, da uporabnik pri pisanju poizvedb [11] deluje z jasno in eksplicitno semantiko. Vpeljava vozlišč in relacij namesto tabel in ključev je blizu načinu povezovanja elementov v resničnem svetu in v pomoč pri predstavitvi električnega omrežja. Namesto sledenja primarnim ključem in iskanja znotraj številnih tabel, je dovolj slediti povezavami, kar poenostavlja tudi poizvedbe.

RDF je preprost matematični graf, ki ima samo vozlišča in povezave. Označeni grafi lastnosti (angl. Labeled property graph) so razširitev navadnega matematičnega grafa ter imajo poleg vozlišč in povezav oznake in lastnosti. Oznake definirajo vlogo vozlišča v določeni domeni, mu dajo natančnejši pomen in določajo omejitve. Eno vozlišče ima lahko več oznak. Lastnosti pa so pari ključ-vrednost, ki vsebujejo informacije o vozlišču ali relaciji.

Grafne baze nimajo predpisane sheme, njihova druga koristna funkcija pa je hitra prilagoditev spremembam. Dodajanje novih vozlišč ali povezav je enostavno in brez vpliva na druge. Največja prednost podatkovnih baz grafov pred drugimi je, da podatki običajno težijo grafom, in ne k tabelam. Če so podatki bolj povezani, lažje jih je predstaviti z grafi. In ne gre samo za podatke, ljudje ponavadi bolje razumejo vse, kar je predstavljeno kot graf.

Obstaja več kot ena vrsta grafnih baz. Razvrstimo jih lahko po različnih kriterijih, kot sta vrsta osnovnega modela ali način shranjevanja in iskanja.

Na trgu je na voljo veliko rešitev, vendar izstopata Neo4j in GraphDB. Gre za grafni bazi, odprtokodni in NoSQL, vendar obstaja tudi veliko razlik, zaradi česar sta najboljša izbira za določene primere uporabe.

Kako delati v obeh grafnih bazah podatkov, je podrobneje pojasnjeno v 4.1.

#### 4.1 Primerjava med Neo4j in GraphDB

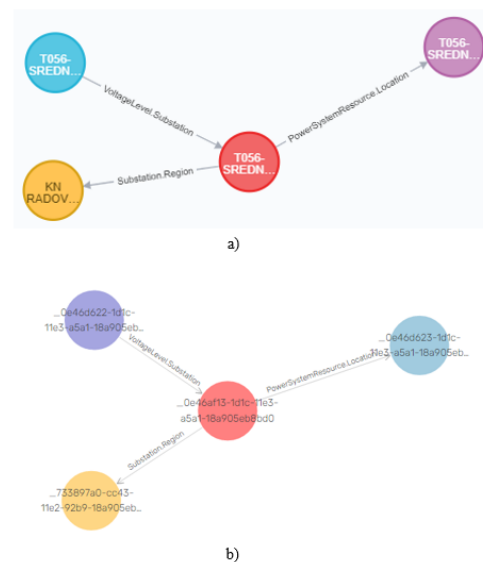
Neo4j je odprto kodna in NoSQL grafna baza, pri čemer je osnovni model graf lastnosti [12]. V Neo4j so podatki shranjeni v vozliščih, povezavah in lastnostih, tako kot jih vizualizira uporabnik.

Po drugi strani je GraphDB večmodelna grafna baza. Večmodelna pomeni, da je GraphDB hkrati grafna baza in baza RDF trojk (angl. triplestore). GraphDB je izvoren za RDF, zato se pogosto reče, da je GraphDB baza podatkov za shranjevanje grafov znanja.

Najpomembnejše razlike med Neo4j in GraphDB so:

- *Osnovni model* - Neo4j graf lastnosti uporablja kot osnovni model, GraphDB pa je večmodelna baza.
- *Iskanje po bazi podatkov* - Neo4j uporablja iskanje brez indeksa, GraphDB pa uporablja indekse za iskanje podatkov.
- *Poizvedovalni jezik* - Neo4j ima svoj poizvedovalni jezik ki se imenuje Cypher, medtem ko GraphDB uporablja standardizirani SPARQL.
- *Delo z RDF* - Neo4j ne zna delati z RDF brez uporabe vtičnikov, medtem ko je GraphDB izvorno okolje za RDF.

Na sliki 5 je prikazano kako je ena razdelilna transformatorska postaja (angl. Substation) shranjena v bazah Neo4j in GraphDB.



Slika 5: Razdelilna transformatorska postaja (RTP) shranjena v a) Neo4j b) GraphDB.

Fokus našega dela je shranjevanje modelov CIM RDF v grafnih bazah s ciljem povečanja stopnje interoperabilnosti.

## 5 REZULTATI

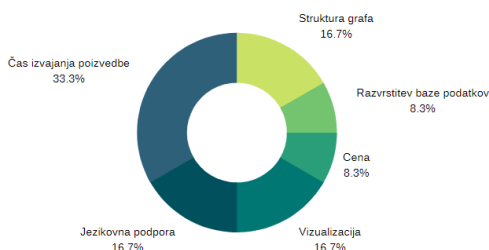
V okviru praktičnega dela raziskovanja smo v dveh grafnih bazah podatkov, Neo4j in GraphDB, razporedili modele distribucijskih mrež v formatu RDF in primerjali njihove zmogljivosti s standardnimi ključnimi kazalniki uspešnosti (angl. Key Performance Indicators) za baze podatkov. Neo4j je izbran kot predstavnik za grafne baze podatkov z označenim grafom lastnosti kot osnovnim modelom, medtem ko GraphDB združuje grafne baze in baze RDF-trojki. Označeni graf lastnosti je bolj kompleksen kot graf RDF, ker vključuje lastnosti za vozlišča in relacije. Shranjevanje modela RDF v Neo4j je bolj zapleteno kot v primeru GraphDB, saj je na primer, vsaka lastnost vozlišča v bazi Neo4j je predstavljena kot vozlišče povezano z drugim vozliščem v RDF. Tako moramo pri Neo4j za vsako povezavo preveriti ali gre za povezavo do novega vozlišča ali za lastnost vozlišča.

Oba pristopa smo ovrednotili z izbranim primerom, ki temelji na realni topologiji distribucijskega omrežja. Uporabili smo dva nabora podatkov, manjšega, s 3.110 elementi in večjega s 5.996.667 elementi.

Prvi korak je uvoz datoteke CIM RDF v vsako bazo podatkov. V primeru GraphDB brez težav uvozimo modele RDF, medtem ko smo pri Neo4j delali na dva načina. Najprej smo uporabili vgrajeni vtičnik *Neosemantics*, nato smo razvili še lastni vtičnik, ki služi kot ontološki adapter za CIM. Za razvoj adapterja in uvoz RDF smo uporabili različne knjižnice za Neo4j in RDF, po uvozu modela Neo4j z RDF dela kot da bi bil označen graf lastnosti. Po uspešnem uvozu lahko z ustreznimi poizvedbami pridobimo informacije o postajah in vodih izbranega primera nabora podatkov.

Neo4j ima lasten jezik za poizvedbe, imenovan Cypher, medtem ko GraphDB uporablja standardizirani jezik za poizvedbe SPARQL. Pridobljene podatke smo zapisali v datoteko KML, ki omogoča vizualizacijo teh elementov v programu Google Earth.

Na podlagi različnih virov smo sestavili seznam KPI, kot je prikazano na sliki 6, ki se uporabljajo za testiranje baz podatkov, kjer je najpogosteje uporabljeni KPI čas izvajanja poizvedbe.



Slika 6: Statistika pogostosti uporabe različnih KPI za testiranje grafnih baz.

Za testiranje smo pretežno uporabljali večji model, s katerim lažje določamo odstopanja v času izvajanja. Statistična analiza dobljenih rezultatov je pomagala določiti

najprimernejši način shranjevanja in upravljanja datotek CIM RDF.

Delali smo iterativno, da bi dosegli najboljše rezultate. Prvi pristop je bil uporabiti izvorni model in poizvedovati z več, manjšimi poizvedbami. V tabeli 1 sta prikazana srednji čas izvedbe in skupni čas izvedbe za večji model, kjer vidimo, da Neo4j potrebuje bistveno več časa kot GraphDB. Srednji čas pri Neo4j je približno 0.25 ms, v primeru večjega modela z okvirno 6 milijoni elementov, pa je skupni čas izvedbe primera več kot 10 ur.

Tabela 1: Rezultati za večji model.

	Neo4j	GraphDB
Srednji čas izvedbe poizvedbe [ms]	0.25082	0.0001574
Skupni čas izvedbe primera	10.1433 ur	5.9634 minut

Prvi pristop je potreboval preveč časa, po dodatnih raziskavah je bilo ugotovljeno, da je potrebna optimizacija modela, pri čemer se poizvedbe ohranijo enake. Odstranili smo prazna vozlišča, vozlišča brez povezav in nekatere napake, ki so nastale med izdelavo modela. Optimizacija je narejena na modelu v bazi z ustreznimi poizvedbami. V tabeli 2 so prikazani rezultati za manjši model. Razvidno je, da se je čas izvedbe pri Neo4j izboljšal, vendar je GraphDB kljub temu deloval bolje. Za večji model smo z Neo4j še vedno potrebovali približno 10 ur za izvedbo primera.

Tabela 2: Skupni čas izvedbe primera za manjši model z optimizacijo.

	Neo4j (brez optimizacije)	Neo4j (z optimizacijo)
Čas izvedbe [ms]	308.7538	234.3566

Zadnji pristop je bil optimizacija modela ob uporabi samo ene podrobnejše poizvedbe. GraphDB je sicer dal boljše rezultate kot Neo4j, vendar je tudi 10 minut predolg čas za izvedbo posameznega primera. Zato smo napisali bolj kompleksno poizvedbo v Cypher in SPARQL, ki analizira celotno omrežje naenkrat. Iz tabele 3 je razvidno, da se je čas izvedbe primera uporabe močno skrajšal ter da sta zdaj časa za izvedbo primera uporabe za obe bazi podatkov primerljiva. Če naredimo primerjavo med tabelama 1 in 3 je razvidno, da se je čas izvedbe v primeru Neo4j skrajšal z več kot 11 ur na le 21 sekund, v primeru GraphDB, pa iz 10 minut na 22 sekund.

## 6 ZAKLJUČEK

Za obe bazi podatkov smo uspešno izvedli primer uporabe za vsak nabor podatkov (manjši in večji). Tako

Tabela 3: Rezultati za večji model z eno poizvedbo.

	Neo4j	GraphDB
Čas izvedbe poizvedbe [ms]	320.3436	0.007436
Skupni čas izvedbe primera [s]	<b>20.3433</b>	<b>21.7652</b>

GraphDB kot Neo4j izpolnjujeta vse tehnične zahteve, pri delu pa smo potrdili njune prednosti in slabosti. Poizvedovalni jezik Cypher (Neo4j) je večinoma bolj učinkovit in zahteva manj kodiranja s strani uporabnika. Ker Neo4j ni prilagojen na RDF, lahko to zahteva dodatno optimizacijo podatkov, kar ni vedno preprosto ali celo dovoljeno. Po drugi strani sta RDF in SPARQL izvorna za GraphDB, zato je poizvedovanje zelo preprosto in intuitivno. Hkrati ima SPARQL manj možnosti kot Cypher, kar včasih zahteva več kodiranja.

S poglobljeno analizo in razvojem lastnega pristopa smo čas izvajanja skrajšali iz več ur na nekaj sekund in s tem dosegli očitno pohitritev. Uspešno smo uvozili model CIM RDF z različnimi pristopi, manipulirali z modelom in izvajali poizvedbe, vse brez spreminjanja originalnega podatkovnega modela.

Shranjevanje modelov CIM RDF v grafni bazi podatkov je zelo priročno, subjekt, ki zahteva podatke, mora imeti le ustrezno pooblastilo za dostop, ki mu omogoča da pregleduje model in poišče željeno informacijo. RDF ni lahko berljiv brez predhodnega poznavanja strukture trojčka, poleg tega moramo razumeti ontologijo CIM. Vse to zahteva veliko učenja in izobraževanja deležnikov, medtem ko ti želijo le preprost in hiter dostop do informacij. Z uporabo baz podatkov in grafičnih uporabniških vmesnikov, ki bodo poizvedovali v ozadju, morajo biti subjekti pooblaščen samo za dostop do baze podatkov in zahtevanje informacij, ki jih potrebujejo, ne da bi podrobno poznali CIM ali RDF.

## ZAHVALA

Raziskave je delno podprla Javna agencija za raziskovalno delo (ARRS) v okviru programa P2-0246.

## LITERATURA

- [1] M. Janssen, E. Estevez and T. Janowski *Interoperability in Big, Open, and Linked Data— Organizational Maturity, Capabilities, and Data Portfolios*, IEEE Computer Society, 2014
- [2] M. Uslar, M. Specht, S. Rohjans, J. Trefke and J.M. González, C. Rosinger and C. Dänekas *Standardization in Smart Grids - Introduction to IT-Related Methodologies, Architectures and Standards*, Springer, 2013
- [3] A. Gyrard, A. Zimmermann and A. Sheth *Building IoT based applications for Smart Cities: How can ontology catalogs help?*, IEEE Internet of Things Journal, 2018
- [4] R. L. Ackoff *From Data to Wisdom*, J. Applied Systems Analysis, 1989
- [5] *EPRI Technical report, "Common Information Model Primer", Third Edition. EPRI, 2015*
- [6] M. Uslar, M. Specht, S. Rohjans, J. Trefke and J.M. González, *The Common Information Model CIM: IEC 61968/61970 and 62325 - A Practical Introduction to the CIM*, Springer, 2012
- [7] CIM Users Group (CIMug) [Online], <https://cimug.ucaiug.org/pages/home.aspx>
- [8] O. Curé and G. Blin, *RDF Database Systems - Triples Storage and SPARQL Query Processing*, Morgan Kaufmann, 2015
- [9] A. deVos, S.E. Widergren and J. Zhu, *XML for CIM model exchange*, Power Industry Computer Applications, 2011
- [10] M. Lal, *Neo4j Graph Data Modeling - Design efficient and flexible databases by optimizing the power of Neo4j*, Packt Publishing, 2015
- [11] *Graph database vs. relational database* [Online], <https://medium.com/@mtbuzzerseol/graph-database-vs-relational-database-e5798281f6ef>
- [12] Neo4j developer [Online], <https://neo4j.com/developer/>

**Amila Dervišević** je magistrirala s področja elektrotehnike na Fakulteti za elektrotehniko, Univerze v Ljubljani. Njeno glavno področje dela so informacijsko-komunikacijske tehnologije (IKT). Med študijem je delala na Elektroinštitutu Milan Vidmar (EIMV) v skupini za pametna omrežja in se ukvarjala z izvajanjem rešitev IKT v elektroenergetskih sistemih. Njeno raziskovalno področje pokriva CIM, pravila modeliranja in orodja CIM, ki se uporabljajo za ustvarjanje profilov. Amila Dervišević je prejela več štipendij za predhodno delo in študij.

**Matej Zajc** je redni profesor na Katedri za IKT, Fakultete za elektrotehniko, Univerze v Ljubljani. Predava na dodiplomskih in podiplomskih študijskih programih IKT in Multimedija. Raziskovalno je aktiven na področju IKT za pametna omrežja.

**Nermin Suljanović** je raziskovalec na Elektroinštitutu Milan Vidmar (EIMV) v Ljubljani in redni profesor na Fakulteti za elektrotehniko Univerze v Tuzli. Njegovi raziskovalni interesi so na področju pametnih omrežij ter modeliranja semantičnih podatkov in komunikacijskih arhitektur.