

Податочно воден пристап за
подобрување на решенијата кај
хардверските кола и системи

Владимир Здравески

Ментор: Проф. Димитар Трајанов

Факултет за информатички науки и компјутерско инженерство
Универзитет св. Кирил и Методиј во Скопје, Р. Македонија

Докторска дисертација

Мај 2017

Апстракт

Податочно водениот пристап се применува на хардверски кола и системи, насочувајќи се кон мали системи, какви што се електронските кола, големи хомогени системи, како електричните мрежи и големите хетерогени системи какви што се интегрирани софтверски решенија за паметен град. Во областа на мали системи ги презентираме онтологијата на доменот, која го поедноставува процесот на размена на мета-податоци за компонентите и поддржува размена и повторна употреба на информации меѓу софтверски агенти, интегрирани во алатките за автоматизација на електронски дизајн. За да се покаже корисноста и употребливоста на овој пристап, ние создадовме прототип на интелигентен софтверски агент и веб апликација базирана на онтологијата, кои нудат споделување, пронаоѓање, повторна употреба и композиција на компоненти. Во насока на големи хомогени системи предлагаме иновативен, многу едноставен алгоритам за динамичко интелигентно балансирање на товар, што ги намалува загубите на моќност во мрежата на дистрибуција на електрична енергија. Нашето решение лесно се применува на секој дел од дистрибутивната мрежа, без суштински измени на инсталацијата на мрежата. Концепт воден од ISO 37120 стандардот за услуги и квалитетот на животот во еден град, е предложен како унифицирана рамка за паметни интегрирани софтверски решенија во полето на големи хетерогени системи. Врз основа на предложената архитектура е развиена прототип апликација за градот Скопје, Македонија.

Содржина

Апстракт	iii
Листа на слики	vii
Листа на табели	xi
1 Вовед	1
2 Податочно воден пристап за подобрување на решенијата кај хардверските кола и системи	7
3 Податочно воден пристап во областа на мали системи	13
3.1 Онтологии за хардвер и систем во чип	13
3.1.1 Достигнувања во областа	15
3.1.2 Примена на семантички технологии кај јазиците за опис на хардвер	18
3.1.3 Онтологии за опис на хардвер	22
3.1.4 Прототип апликација според предложениот концепт	29
3.1.5 Отворени можности	30
3.1.6 Пример за семантички опис	32
3.2 Класификација на IP јадра	35
3.3 Вклопување на тест-компоненти	37
3.3.1 HDL IP Cores - преглед на системот	39
3.3.2 Модул за пронаоѓање на тест-компонента	41
3.3.3 Conclusion and future work	45
3.4 Карактеристика со помош на логичката функција	46
3.4.1 Достигнувања во областа	47
3.4.2 Процес за извлекување на логичката функција	48
3.4.3 Отворени можности	51

4	Податочно воден пристап во областа на големи хомогени системи	53
4.1	Алгоритам за балансирање на потрошувачите	53
4.1.1	Опис на архитектурата	55
4.1.2	Преглед на алгоритмот за оптимизација	57
4.1.3	Мерења и резултати од симулацијата	61
4.1.4	Паметна дистрибутивна мрежа со DILB	64
4.1.5	Заклучок	67
4.2	Алгоритам за дистрибуирана компензација	67
4.2.1	Општа архитектура	69
4.2.2	Балансирање на товар	70
4.2.3	Компензација на реактивна моќност	71
4.2.4	Прототип клиент	73
4.2.5	Анализа на промената на фаза	75
4.2.6	Заклучок	77
4.3	Онтологија за електроенергетски систем	77
4.3.1	Онтологија за електроенергетски систем	80
4.3.2	Реални примери	83
4.3.3	Заклучок и идна работа	88
4.3.4	SPARQL прашања и извештаи	89
5	Податочно воден пристап во областа на големи хетерогени системи	97
5.1	ISO-стандардизирана архитектура на платформа за паметен град .	97
5.1.1	Достигнувања во досегашниот развој на платформи	99
5.1.2	Архитектура на платформата	103
5.1.3	Прототип	105
5.1.4	Заклучок	108
6	Заклучок и идна работа	111
	Литература	115

Листа на слики

- 3.1 Општиот концепт на поврзани компоненти. Компонентите „C1“, „C2“ and „C3“ само се поврзуваат со оригиналните изворни датотеки, правејќи го процесор на искористување лесен за одржување. 20
- 3.2 Онтологии за дизајн на хардвер. Онтологијата SoC реискористува неколку онтологии кои се дел од облакот на поврзани податоци и дава опис на воопштен модел на хардвер. 22
- 3.3 SoC домен онтологијата (SoC.owl). Елипсите претставуваат класи и стрелките претставуваат својства (релации). Онтологијата може да се користи како шема за опишување на хардверска архитектура на високо ниво, независно од јазикот за опис на хардвер. 23
- 3.4 Насочен ацикличен граф, формиран од од аотации на својството „soc:hasPart“. Претставено е знаење вградено во SoC онтологијата, кое се користи во процесот на одредување на типот на компонентата. Сите типови на компоненти се поткласи на класата „soc:Architecture“. 25
- 3.5 Мал подграф на SoC онтологијата, што го илустрира својството „hasPart“ со кое се наведува дека „FlipFlop“, „Port“ или „Latch“ може да се инстанцира во внатрешноста на „Register“. 26
- 3.6 Еден дел од знаењето за можните поврзување на компоненти вградени во онтологијата, што покажува дека „Processor“ може да биде поврзан со „Memory“, „Gate“ или „Testbench“ компонента. 26
- 3.7 VHDL онтологијата (VHDL.owl). Сивите елипси и означени стрелки ја сочинуваат VHDL онтологијата. Останатите класи и својства припаѓаат домен онтологијата и онтологијата за апликацијата, кои се реискористуваат во онтологијата за VHDL. 27
- 3.8 Едноставни текстуални веб страници во споредба со веб ресурси. . . 28

3.9	Прелистувач на проект кој графички го претставува HDL проектот, нуди прозорец со детали и се отвора специфичната архитектура со еден клик. Типовите на компоненти се означени рачно и со користење на тековно развиениот модул за класификација (кој е во идна работа и е надвор од доменот на овој документ).	31
3.10	Архитектура на модулот за класификација.	36
3.11	HDL IP Cores - архитектура на системот.	38
3.12	Функционалности на клиентската алатка.	42
3.13	Процес на совпаѓање.	43
3.14	Симулатор на клиентска страна.	44
3.15	Симулатор на серверска страна.	45
3.16	Анотација на логичката функција.	48
3.17	Дијаграм на алгоритмот. Алгоритмот се повикува за секоја излезна порта (a) и за секој излез рекурзивно ја наоѓа инстанцата и нејзините влезови, се додека стигне до влезовите на архитектурата (b).	49
3.18	Ниво на абстракција на синтаксата. VHDL кодовите на двете компоненти се различни, но логичките функции се еквивалентни.	50
4.1	Поедноставена архитектура на дистрибутивна мрежа, формирана од еден родител јазел и неговите деца.	56
4.2	Генерализирана архитектура на дистрибутивна мрежа, проширена со DILB модули. Јазелот корен претставува електрична централа или графостаница, додека јазлите листови ги означуваат електричните броила со вградени ротациони склопки.	57
4.3	Дијаграм на DILB алгоритмот: a) алгоритам за дистрибутивна мрежа што се активира во јазелот-корен, b) Рекурзивен DILB алгоритам за еден јазел.	60
4.4	Средниот дисбаланс на товарот како функција од временскиот интервал на промена на ротационите склопки. Нивото на дисбаланс е пропорционално со нивото на активните загуби на моќност.	63
4.5	Систем Архитектура и комуникациски детали. Серверот е поставен на Интернет или во локална мрежа. Клиентите се поврзани со (или вградени во) електричните броила. Секој клиент ги користи услуги на серверот 1/2/3, испраќа свои мерења и добива оптимална ротациона состојба и состојба на дистрибуираниот кондензатор.	69

4.6	Серверот прикажува листа на поврзани уреди и пресметува оптимално s . Достапен е со <code>http/https</code> на пр. <code>https://dilb.com</code> (10.42.0.1 во нашиот случај).	70
4.7	Дистрибуирана компензација на реактивна моќност. Вкупната капацитивност (nC) на кондензаторската батерија додадена во корен-јазелот (лево) е еквивалентна на сумата на n дистрибуирани кондензатори (C) поставени во секое електрично броило (десно).	71
4.8	Клиентот овозможува конфигурациска форма и го прикажува товарот по фаза во системот. Достапен е преку неговата локална адреса (на пр. 10.42.0.15). Адресата на серверот треба да се додаде во конфигурацијата на клиентот (<code>http://10.42.0.1/</code> означува <code>https://dilb.com</code> во нашиот пример).	73
4.9	Шематски приказ на клиентската плоча, поврзана со електричното броило и панелот со осигурувачи (лево) и фотографија од имплементираниот прототип систем (десно). Кондензаторот е со капацитивност од $60 \mu F$	74
4.10	Напон на потрошувачот во моментот кога релето се вклучува (лево) и кога се исклучува (десно). Сигналот што предизвикува промена на релето е со сина боја, додека напонскиот сигнал е со црвена боја.	75
4.11	Единечна ќелија на една излезна фаза од модулот за промена на фазите.	76
4.12	Онтологијата за електроенергетски систем (PGO) обезбедува модел на податоци од доменот на електроенергетски систем (како мрежна топологија и континуирани и периодични мерења).	80
4.13	Мрежа од паметни јазли.	81
4.14	Мапа на моќност.	82
4.15	Известувања за паметен телефон.	83
4.16	Електрични централи во Германија.	84
4.17	Критични врски во Германија.	85
4.18	SPARQL сервиси. Еден дел од пребарувањето со SPARQL се извршува на PGO складиштето, а другиот дел се доставува до DBPedia. Потоа, податоците добиени од DBPedia се спојуваат со податоците од PGO и дополнително се филтрираат пред да бидат испратени на клиентот.	85
5.1	Дијаграм на текот на податоците на платформата.	101
5.2	Архитектура на апликацискиот слој.	102

- 5.3 3D простор на ISO индикатори. Резолуцијата на ISO индикаторите се зголемува со луѓе (население), простор (област) и време, што резултира со прилагодени информации за корисниците. 104
- 5.4 Имплементација со програмската околина на Јава. 106
- 5.5 ISO стандардизирана платформа за паметен град, која ги прикажува категориите на индикатори (горе-лево), филтер според оските на 3D просторот со висока резолуција (горе) и податоците за градот Скопје. 107

Листа на табели

3.1	Веб портали на складишта со компоненти и број на анотирани IP јадра. Ресурс претставува веб-ресурс (датотека или компресирана архива), проект се однесува на HDL проект, IP јадро означува HDL датотека и архитектура е хардверска архитектура.	32
4.1	Минимизација на активните и реактивните загуби на моќност. . . .	66
4.2	Вкупен товар по фази измерен во корен-јазелот.	66
4.3	Робустност на прототипот - време на одговор.	75
4.4	Незабележлива промена на фаза - тестирани уреди.	77
4.5	Топ 10 операторs според должината на врските.	84
4.6	Градови и жители на радиус од 20 км од електрична централа. . . .	86
4.7	Топ десет градови (со над 100 000 жители) подредени според бројот на жители по трафостаница.	87
4.8	Последните десет градови (со над 100 000 жители) подредени според бројот на жители по трафостаница.	87
4.9	Градови (со над 100 000 жители) подредени според бројот на оператори.	88
4.10	Резултати: Листа од сите електрични централи.	90

Глава 1

Вовед

Науката пред околу илјада години била емпириска и фокусирана кон опишување на природните појави и феномени. Во последните стотина години научниците се ориентирани кон теоретски пристап, развивање на модели и нивна генерализација. Развојот на компјутерските системи од пред неколку децении ја пренасочил науката кон пресметување и симулација. Денес науката во голем дел се сведува на искористување на податоци, кои потоа се користат и за експерименти и за симулации. Податоците се добиваат од мерни инструменти или симулатори, софтверски се процесираат и знаењето/информациите се зачувуваат во некој компјутерски формат. Научниците постојано анализираат бази на податоци и датотеки, на некој начин управувајќи со податоците и користејќи научни методи. Поимот е-Наука често се дефинира како спојна точка помеѓу информатичките алатки и науката и научниците, а практично многу од научните процеси водени и насочувани од достапните податоци во дадениот домен [1].

Податочното водениот пристап подразбира дека извршувањето на одреден процес или активност зависи од податоците, наместо од други правила, дефиниции или лични искуства [2]. Основниот циклус во искористувањето на податочното водениот пристап се состои од:

- Собирање на податоците
- Индексирање и структурирање
- Анализа и носење заклучоци

Еден начин на интерпретација на податочното водениот пристап е со помош на моделот на пирамида, во чија основа се податоците, кои се само некакви измерени вредности или други нумерички или текстуални записи. Кога на податоците ќе им

се даде некаков контекст, тогаш од нив може да се извлече одредена информација, која е корисна, организирана и структурирана. Штом ќе се даде некакво значење на информациите се добива знаење, кое е синтетизирано и има одреден контекст. И на крај, на врвот на пирамидата кога веќе имаме увид во некакво знаење, всушност може да зборуваме за мудрост или нешто, што е разбирливо, интегрирано и кое може да послужи за носење одлуки и преземање акции.

Податочно водениот пристап веќе се применува во повеќе домени и резултира со различни иновативни модели и подобрувања на постоечки системи. Податочно воден пристап се користи за моделирање и симулација на патниот сообраќај во реално време во [3], а заклучоците се во насока на донесување на одлуки за системите за поддршка и просторно планирање. Во Шведска се применува податочно водениот пристап за проценка на количината на водите во реките, користејќи ги регионалните климатски модели [4], а може да се примени и за проценка на потенцијалот на подземните води [5]. Во [6] со помош на податочно водениот пристап се развиваат индикатори за животната средина и достапноста на податоците е централен критериум. Податочно водениот пристап се користи и во медицината и тоа за проценка на хетерогеноста на паркинсоновата болест во рана фаза [7] и когнитивни пореметувања [8], а исто така се предлагаат и нови податочно водени критериуми за класификација на Сјоргеновиот синдром [9]. Примената е можна и во анализа на симптомите на пациентите кај кои се регистрирана Шизофренија и ментално растројство [10]. Преглед на податочно водените решенија кон решавање на проблемот на проценка на преостанато време на живот на единка е даден во [11]. Може да се користи и за предвидување на ефектите и интеракцијата на лековите [12] и во хемијата за медицински цели во полето на обработка на големо количество податоци [13].

Податочно водениот пристап веќе се насочува и кон индустријата, како на пример во [14], каде се користи за мониторирање на процесот на сериско производство и анализа на динамиката на процесот и стохастичките нарушувања, што тековно се случуваат. Во [15] и [16] се предлага користење на податочно водена методологија за мониторирање на комплексни индустриски процеси и дијагностика на падови на системот, а во [17] се применува во насока на идентификување на процесните променливи и карактеристики. Во софтверското инженерство податочно водениот пристап отвора можност за интегрирање на дополнителни тековно достапни податоци во постоечки апликации [18], како и подлабоко поврзување на периодично достапните податоци и податоците кои се генерираат континуирано [19]. Исто така, податочно водениот пристап се користи и во полето на композиција на веб сервиси

во напредните софтверски решенија [20]. И во образованието постојат обиди за искористување на податочно водениот пристап и тоа за создавање на материјали потребни во наставниот процес [21] и за изучување на граматика и вокабулар [22].

Поимот „податоци“ како парадигма ја наметнува целта за определување и продлабочување на нивното семантичкото значење, што покажува дека семантичките технологии во некоја мера мора да се присутни во податочно водените системи и апликации. Применатана семантичките технологии во неколку други области, како здравство и хемиска индустрија, создаде многу успешни проекти, предизвикувајќи голем чекор напред кон автоматизација на многу процеси кои вклучуваат податоци информации. Големината на податочното множество достапно во формат на Семантичкиот веб постојано расте, создавајќи огромен Веб од ресурси, познат како Поврзани податоци (анг. „Linked Data“) [23]. Тоа претставува напреден концепт на поврзување и објавување на податоците, кој овозможува нови кориснички сценарија, невозможни со неповрзаните податочни множества.

Примената на семантичките технологии веќе има произведено многу успешни апликации [24], како NextBio ¹, што е семантичка платформа за агрегирање и интерпретација на големи количества генетски податоци за истражување и клинички апликации. Data.gov.uk е уште еден пример на објавување на владини податоци во структуриран формат, мотивирајќи ги развивачите на софтвер да создадат многу корисни апликации ². Powerset ³ (сега му припаѓа на Binq) е алатка за пребарување на некој од природните јазици, базирана на семантички технологии. Една од водечките електронски продавници во САД, BestBuy ⁴, користи семантички технологии за зголемување на видливоста на нејзините продукти и услуги. BestBuy ги означува податоците (како име на продавница, адреса, работно време и гео-локација) со помош на RDFa, со што алатките за пребарување се во можност да ги идентификуваат тие податоци полесно и да ги стават во контекст, зголемувајќи го сообраќајот на Веб и подобрувајќи ги услугите за клиентите. Откако го откупи Ellerdale (компанија за семантичка анализа во реално време), Flipboard ⁵ понуди многу интелигентни функционалности за категоризација и преглед на дневни вести и стана лидер во полето на „персонализирани социјални весници“. Дури и Facebookима искористено семантички технологии во неговата популарна платформа за пребарување Open Graph ⁶.

¹NextBio, <http://www.nextbio.com/>

²Data.gov.uk apps, <http://data.gov.uk/apps>

³Powerset, <http://www.powerset.com/>

⁴BestBuy, <http://www.bestbuy.com/>

⁵Flipboard, <https://flipboard.com/>

⁶The open graph protocol, <http://ogp.me/>

Обидувајќи се да ги обединат различните шеми на податоци, Google, Microsoft, Yahoo и Yan-dex, заедно со поширокото Веб општество, неодамна го иницираа Schema.org ⁷, што претставува проект со мисија за создавање, одржување и промоција на универзална шема за структурирани податоци достапни на Интернет, на веб страниците, во електронските пораки и тн. Главната идеја е да се објавуваат, меѓусебно да се поврзуваат и да се реискористуваат податочните шеми во единствен вокабулар, бидејќи универзалниот вокабулар е поедноставен за користење и овозможува максимални придобивки [25]. Но, се уште во тој вокабулар не постојат класи и релации поврзани со хардвер, иако развојот на хардверски кола и системи во последните неколку години значително се базира на примена на информатичките технологии, алатки за поврзување на податоци податочни водените техники, и тоа во насока на олеснување на процесот на дизајн на хардвер и подобрување на функционалноста на постоечките електрични и електронски системи.

Алатките за дизајн на хардвер стануваат се поедноставни за користење и скоро секој кој сака, може за разумно време да успее да дизајнира и симулира сопствен хардвер. Постојат многу интелегентни уредувачи на код, паметни синтетизери, симулатори и дури и претварачи на код од повисоките програмски јазици во јазици за опис на хардвер [26]. Некои од современите интегрирани развојни околинис нудат и можности за развој без пишување на код, што го прави дизајнот на хардверски кола привлечен и интересен и за почетници [27] [28], но и за поискусни дизајнери. Слободниот пристап до развојни алатки резултираше со големи количини отворен код достапен на Веб. Компаниите кои работат со јазици за опис на хардвер исто така имаат сопствени складишта, каде се чува нивниот развоен код. Но, без разлика на големата количина на код достапен на Интернет или во компанијата, се појавуваат потешкотии кога дизајнерите сакаат да пронајдат и реискористат дел од тој код во некој нов дизајн. Процесот одзема време бидејќи корисникот мора да пребарува низ кодот и рачно да ги идентификува деловите кои му се потребни.

Постојат неколку обиди за развој на софтверско решение кое поддржува споделување и реискористување на HDL код [29]. Проектот VALBOA и VALBOA јазикот за опишување на интерфејс (BIDL) [30] иницијално се базирале врз идејата за користење на Tck/Tk за создавање на скриптирачки јазици кои ќе овозможат опишувањена костурот на архитектурата на систем-во-чип, но потоа била дефинирана XML шема за спецификација на IP интерфејс. Конзорциумот SPIRIT ⁸ го објавил

⁷Schema.org Vocabulary, <http://schema.org/>

⁸Spirit Consortium, <http://www.spiritconsortium.org/>

стандардот IP-XACT, кој понудува проверка на комплетноста на XML податоци за опис на електронски систем. Тој содржи мета-податчни форми и правила за семантичка конзистентност. IP-XACT овозможува забрзување на процесот на дизајн, пренесување, валидација, документација и користење на електронските IP јадра и опфаќа компоненти, дизајни, интерфејс и тн. Исто така забележливи се и пристапите фокусирани на дефинирање на параметризирани компоненти [31] [32] [33] и насочени кон создавање на бази со генерички компоненти, кои овозможуваат лесна и брза интеграција во идни проекти.

Од друга страна и системите, кои секојдневно ги користиме се почесто може да се анализираат како множества или потоци на податоци и да се искористи податочно воден пристап за нивно подобрување. На пример електричните дистрибутивни мрежи постојано се соочуваат со постојано растечка оптовареност и брзи промени од низок кон висок степен на оптовареност. Иако за време на инсталација на дистрибутивните мрежи потрошувачите се балансираат на трите фази, сепак дневните промени на потрошувачката резултираат во небалансирана состојба. Потребата од балансираност на товарот и минимизација на загубите поттикнале различни научни обиди за подобрување на балансираноста и временско распоредување на товарот. Постојат два главни методи за балансирање и тоа: реконфигурација на напојните јазли на системско ниво [34] [35] [36] и промена на распоредот фазите во јазлите.

Промената на распоредот на фазите од поодамна е актуелна истражувачка тема. Неколку алгоритми се представени од Ванг, Скиена и Робертази [37], вклучувајќи ги и сеопфатно пребарување, алчен алгоритам, генетски алгоритам и др. Во светот постојат и дистрибутивни мрежи кај кои вработените во дистрибутивната компанија ги реконфигурираат трансформаторите, во текот на периодите за одржување, кога се приклучува нов корисник или ако процентот на дисбалансираност надмине одредена граница. Три фактори се разгледуваат при одлучувањето за ре-балансирање на јазелот: монетарната вредност на промената на склопките, очекуваното подобрување во балансираноста на јазелот и привременiot прекин на напојувањето на клиентот. Моментално ре-балансирањето е комплицирано да се врши рачно со застарените електрични броила, кои немаат можност за вградување на интелигенција или за поврзување во логичка (компјутерска) мрежа и затоа е непрактично, скапо и одзема многу време.

Податочно водениот пристап веќе се користи за решавање на некои од проблемите во дистрибутивните мрежи. Во [38] се предлага искористување на историските податоци за управување на енергетскиот систем за правење на добра проценка

на состојбата на системот во иднина. Според [39] искористувањето на податочна водениот пристап може да се користи и за визуелизација на енергетските системи, со помош на алгоритми за обработка на податоците и создавање на визуелизација на карактеристиките на системот. Податочна водено моделирање на потрошувачите се применува во [40], а добиените модели се евалуираат со помош на податоци од реални мерења направени во областа на Тексас. Иновативен динамички податочна воден пристап за прилагодување на операцискиот план на дистрибутивните мрежи во реално време е предложен во [41]. Во [42] е прикажан податочна воден модел на енергетските потреби за полнење на електрични возила на географскиот простор на Велика Британија, а проценка на добивките од паметно полнење на електричните возила во ненаселените локации е прикажано во [43].

Податочна водена детекција и справување со падови кај турбините на ветер е претставена во [44], а податочна водениот пристап се користи и за споредба на перформансите на ветерниците во [45] и за предвидување на моќноста на ветерот [46]. Шема базирана на машинско учење, која овозможува податочна водена дијагностика на падови во опремата на електричните дистрибутивни мрежи е развиена во [47], а детекција на падови, нивна идентификација и локализација се прави со помош на податочна водени пресметковни модели во [48]. Партиционирање на дистрибутивните мрежи со помош на податочна воден алгоритам е претставено во [49], а алгоритам за ефикасна проценка на топологијата на дистрибутивна мрежа е прикажан во [50].

Глава 2

Податочно воден пристап за подобрување на решенијата кај хардверските кола и системи

Истражувањето во рамките на докторската дисертација се одвиваше во насока на искористување на податочно водениот пристап и издвојување на најдобри практики на негова примена во областа на хардверските кола и системи. Во научното истражување најпрво беа применети аналитички процеси, и тоа:

- (1) преглед на искористувањето на податочно водениот пристап во повеќе области,
- (2) искористување на податочно водениот пристап во областа на мали системи, какви што електронски кола,
- (3) искористување на податочно водениот пристап во областа на големи хомогени системи, какви што се електрични дистрибутивни мрежи и
- (4) искористување на податочно водениот пристап во областа на големи хетерогени системи, каков што е интегрираниот софтверски систем на паметен град.

Техниките и алатките, што произлегоа од процесот (1) беа надмножество над техниките и алатките, што произлегоа од процесите (2), (3) и (4). Во рамките на (2) беше искористен методот за компарација на резултатите добиени со податочно водениот пристап во однос на резултатите со постоечките алатки, а во (3) и (4) беше искористен методот на индукција за да се обопшти очекуваната придобивка

од примената на паметните додатоци во електричните мрежи и системите во рамките на еден паметен град.

Пресекоот помеѓу (2), (3) и (4) е сублимат за добри практики за искористување на податочно водениот пристап во областа на хардверски кола и системи, т.е. применувајќи го методот на синтеза беа дефинирани спецификите и можностите за искористување на податочно водениот пристап во областа на хардверски кола и системи.

Истражувањето во полето на мали системи (2) беше фокусирано на хардверските кола и алатките за дизајн на хардвер, каде следејќи го и применувајќи го концептот за споделен вокабулар во полето на јазици за опис на хардвер, ја продолживме идејата на стандардот Семантички веб, издаден од W3C9, кој промовира објавување на податоци и поврзани ресурси [51] [52], наместо објавување на документи и веб страници. Документите и веб страниците се примарно наменети за користење и интерпретација од страна на луѓето, а не за компјутери и софтверски агенти, затоа не се добра основа за дополнителна автоматска интеграција и анализа на податоците [53]. Семантичкиот веб дефинира стандарди занедвосмислени формати за размена на податоци, кои се доволно блиски до човечките јазици, но и разбирливи за компјутерите. Тие формати и алатки нудат сосем нова перспектива за зачувување и пребарување на податоците, со подобрени механизми за споделување, што е добра појдовна точка за подлабока анализа и интеграција на податоците достапни на Веб.

За да се искористат придобивките од стандардите, алатките и рамките за развој на семантички апликации, потребно е да се дизајнира онтологија за моделирање на доменот на податоците, која потоа ќе се користи за нивна анотација. Бидејќи јазиците за опис на хардвер следат предефинирана синтакса, процесот на анотација беше автоматизиран со користење на HDL-во-RDF мапирање [54]. Процесот на анотација резултира со RDF ресурси, кои ги опишуваат компонентите пронајдени во HDL датотеките или проектите. Секоја HDL датотека може да биде референцирана од повеќе ресурси, опишувајќи различни компоненти или делови од компоненти дефинирани во домен-онтологијата. На ваков начин е можно да се објават и комерцијални компоненти и проекти, бидејќи објавувањето на RDF описот, не се објавува оригиналниот HDL код, туку само мета-податоците. Структурата на SoC онтологијата не само што ќе овозможи пребарување на компоненти според нивните својства, туку ќе овозможи и пронаоѓање на меѓусебно слични и компатибилни компоненти.

Лесно може да се забележи дека постоечките напредни алатки и софтверски решенија во областа на дизајн на хардверски кола се директно базирани врз одредени податочни множества, т.е. складишта со кодови во некој од јазиците за опис на хардвер, што наведува на идејата искористување на податочно воден пристап во насока на подобрување на алатките за дизајн на хардверски кола. Следејќи го успехот на податочно водениот пристап семантичките алатки во домените каде веќе се искористени, се покажа дека нивната примена во полето на дизајн на хардверски кола максимално ќе го искористи структурираниот формат на јазиците за опис на хардвер за автоматска анотација и ќе ги претвори HDL компонентите со отворен код во поврзани HDL компоненти, овозможувајќи платформа за развој на напредни алатки и нови кориснички сценарија во насока на споделување, реискористување и автоматска композиција на компоненти.

Истражувањето поврзано со примена на податочно водениот пристап за подобрување на постоечките големи хомогени системи (3) се одвиваше во доменот на електричните дистрибутивни мрежи. И во доменот на електричните мрежи може да се забележи дека постојано се споменуваат контролни мерења, измерени вредности или со еден збор големо количество на податоци, кои тековно се генерираат. Тоа покажува дека и во овој домен може да се размислува за податочно воден пристап во насока на подобрување на стабилноста и намалување на загубите во електричните мрежи. Можноста за искористување на податочно водениот пристап во електричните мрежи се отвора со тековното зголемувањето на бројот на извори на податоци од тековни мерења на различни параметри на мрежата. Меѓу најзначајните се појавата на мерни инструменти со мали димензии, кои се поинтензивно се поставуваат во форма на контролни мерења, со развојот на паметни приклучоци, што постојано ја мерат потрошувачката на приклучениот уред и со новите функционалности и предности на дигиталните електрични броила, кои поседуваат пресметковни и комуникациски модули, а некои од нив дури и поврзани на компјутерска мрежа и достапни преку Интернет. Мрежата од паметни електрични броила е добра основа за додавање на дополнителна интелигенција и напредни динамички алгоритми, кои ќе овозможат постабилна и поекономична мрежа.

Идентификувајќи ги проблемите на балансирање на товарот помеѓу фазите во 3-фазните системи, имајќи ги в предвид тековните податоци кои веќе постојат во дистрибутивните мрежи и познавајќи ги можностите на податочно водениот пристап, како дел од истражувањето беше имплементиран едноставен иновативен алгоритам за динамичко, интелигентно балансирање на товарот (од англ. „Dynamic

Intelligent Load Balancing“ -DILB)[55]. Некои делови на предложената архитектура, се базираат на постоечкиот концепт на хиерархиско адресирање на уредите (клиент-сервер). Презентирани се алгоритмот и проширената архитектура на електрична мрежа, кои се едноставно применливи во било кој дел на електричните мрежи, без потреба од есенцијални промени на на постоечките електрични инсталации. Примената на алгоритмот беше илустрирана со симулација врз унифицираните добро познатите мрежи IEEE 34-Bus, 37-Bus и 123-Bus, каде се покажа очекуваното намалување на активните загуби на електрична енергија.

Истражувањето се одвиваше и во насока на надградување на паметни електрични уреди со податочно водени интелигентни контролно-комуникациски модули, кои овозможуваат подобри карактеристики на системите и намалување на потрошувачката на енергија. Искористувајќи ја меѓусебната поврзаност на податочно водените модули во комуникациска мрежа, беше овозможена нивна меѓусебна директна или индиректна комуникација и размена на измерените вредности во реално време. Конкретните чекори за примена на споменатиот пристап и придобивките од неговото користење беа илустрирани на мрежа од паметни електрични броила, преку нивно надградување со мал податочно воден сензорско-актуаторски подсистем, кој може да ги ротира фазите за да постигне балансираност на товарот и соодветно да вклучува или исклучува релативно мал по големина капацитивен елемент за да ја намали реактивната моќност [56]. Искористувајќи го податочно водениот пристап беше овозможено, со процесирање на измерените вредности во реално време и преземање на соодветни акции, да се добие оптимална конфигурација (во поглед на балансираност на товар, намалена реактивна моќност и подобрени карактеристики на дистрибутивната мрежа) на податочно водениот сензорско-актуаторски модул на секое броило во дистрибутивната мрежа.

Беше предложен и концепт за замена на еден многу моќен централизиран модул на хардверско коло или систем со мрежа од помалку моќни модули управувани со помош на податочно воден пристап. Евалуацијата на предложениот концепт беше извршена со помош на електрична дистрибутивна мрежа и тоа со цел вкупната капацитивност на една централизирана кондензаторска батерија поврзана во јадрото на мрежата (електрична централа или трансформатор) да се постигне со дистрибуирани парови на податочно водени сензорско-актуаторски модули и мали кондензатори, вградени во паметните електрични броила. Со цел да се потврди податочно-водениот концепт во областа на електрични мрежи, беше дизајниран и имплементиран прототип хардвер и беше иницирана апликација за патент [57] [58].

Истражувањето во доменот на големи хетерогени системи (4) беше во насока на создавање на архитектура за интегриран софтверски систем за паметен град. Беше развиен концепт на платформа која ги надополнува бавните ISO индикатори со ниска резолуција (годишни, квартални или месечни) со знаење со висока резолуција добиено од сензори и социјалните медиуми и лични индикатори, кои може да обезбедат податоци во реално време и предвидувања на индикаторите за паметен град. Затоа, софтверско решение ќе обезбеди методи за набљудување и мерење на феномени од заеднички интерес (на пример, условите во сообраќајот, загадувањето на воздухот, бучавата во урбаните средини), на големи географски области, експлотирајќи ја мобилноста на сензорските уреди. Со комбинирање со податоците од социјалните мрежи, вести, блогови и други извори на податоци, се создава генеричка слика за градот, кој може да ги насочи решенија и технологиите кон концептот на паметен град.

Со изградба на алатки за анализа на податоци и анализа на поточни извори како вестите, онлајн податоци, и социјалните медиуми, беше обезбеден сигурен податочно богат пристап кон прогнозирање на индикатори за паметен град. Платформата опфаќа собирање и обработка на податоци, заклучување на знаење, и трансформација на податоците во лесно разбирливи информации со што е првата платформа со индикатори за паметен град која вклучува IoT/BigData/Cloud и индикатори на ISO 37120. Поконкретно, можните показатели на градот, предвидени од страна на стандардот ISO 37120 за паметни градови, се извлекуваат како влез во предложената платформа, а потоа, овие најчесто застарени податоци со ниска резолуција се дополнети со податоци од граѓаните добиени преку мобилни и сензорски уреди и/или собрани од социјалните мрежи, вести и блогови.

Истражувањето во рамките на докторската дисертација имаше за цел да покаже дека податочно водениот пристап може да отвори нов поглед и нови можности во областа на хардверски кола и системи, но исто така да создаде множество од најдобри практики за искористување на податочно водениот пристап во рамките на хардверски ориентираните системи и апликации.

Имајќи ги в предвид претходно споменатите факти, главната хипотеза на докторската дисертација гласеше:

- Податочно водениот пристап овозможува напредни решенија за подобрување на хардверските кола и системи.

Главната хипотеза беше претставена со следните поспецифични хипотези:

- 1.Податочно водениот пристап овозможува развој на софтверски платформи со проширени и нови функционалности на алатките за дизајн на хардвер, во насока на споделување и реискористување на кодот, како и забрзување на процесот на дизајн на хардвер.
- 2.Податочно водениот пристап овозможува развој на интелигентни додатоци за динамичко подобрување на стабилноста на електричните мрежи и намалување на загубата на енергија во постоечките системи за дистрибуција на електрична енергија.

За потврда на ефектот од искористувањето на податочно водениот пристап и издвојување најдобри практики, податочно водениот пристап беше применет (1) во алатките за автоматизација на дизајнот на електронски кола и (2) во развој на интелигентни додатоци за динамичко подобрување на електричните мрежи. Целта на искористувањето на податочно водениот пристап во (1) беше да се наметне нов поглед кон развојот на кола, кој предизвика помасовно споделување и реискористување на код. Со тоа, се подобрија постоечките функционалности на алатките за развој на чипови и се отворија нови можности и кориснички сценарија, невозможни без податочно водениот пристап.

Искористувањето на податочно водениот пристап во (2) имаше за цел да наметне нов концепт на подобрување на стабилноста на електричните мрежи, ставајќи ги во прв план податоците, добиени од тековните дистрибуирани мерења на параметрите во мрежата. Исто така, искористувајќи ја процесирачката моќ на дистрибуираните модули (паметните електрични броила) истражувањето имаше за цел да предложи различни оптимизации на електричната мрежа со помош на оркестрација на нејзините дистрибуирани елементи.

Искористувањето на податочно водениот пристап во доменот на електричните мрежи резултираше со иновативни методи за подобрување на карактеристиките и стабилноста на дистрибутивните мрежи. Истражувањето ги опфати проблемите на динамичко балансирање на товарот помеѓу фазите и дистрибуирана компензација на реактивната моќност, но и отвори можност и за слични решенија за останатите познати проблеми во електричните мрежи.

Крајната цел беше да се сублимираат најдобри практики и техники за примена на податочно водениот пристап во областите поврзани со хардвер, т.е. е полето на хардверски кола и системи.

Глава 3

Податочно воден пристап во областа на мали системи

3.1 Онтологии за хардвер и систем во чип

Забавувањето на напредокот на транзисторите и крајот на ерата на Муровиот закон го одвлекува вниманието на програмерите од чипови за општа намена и ги насочува кон пишување сопствени хардверски архитектури, со цел да се постигне оптимална конфигурација според енергија-цена-перформанси [59]. Овој факт го прави полето програмабилни компоненти (од Анг. „Field Programmable Gate Arrays“ -FPGAs) популарни и исплатливи, а градењето на сопствена архитектура во програмабилни чипови е овозможува скоро секој да може да дизајнира ново парче на хардвер. Во последните неколку години, јазиците за опис на хардвер (од Анг. „Hardware Description Languages“ - HDLs) беа широко користени од страна на развивачите на хардвер и софтвер во процесот на дизајн на чип [60] [61], во процесот на косимулација кај вградливи компјутерски системи, валидација на хардвер и софтвер [62] и во продукциски верзии на вградливи системи [63].

Уште еден тренд во последно време е споделување на кодот на различни електронски портали прифатени од стручната заедница и тој тренд е прифатен од страна на дизајнерите на HDL. Постојат многу складишта (Open Cores¹, Design-Reuse², Asic world³, IP supermarket⁴, Lattice⁵, Chip estimate⁶ и др.), каде што

¹Open Cores, <http://opencores.org>

²Design-Reuse, <http://www.design-reuse.com>

³Asic world, <http://www.asic-world.com>

⁴IP supermarket, <http://www.ipsupermarket.com/index.php>

⁵Lattice, <http://www.latticesemi.com>

⁶Chip estimate, <http://http://www.chipestimate.com>

HDL компоненти и проекти се објавени под лиценци со отворен код. Формат на објавување варира низ складиштата, но вообичаено едноставните HDL компоненти се објавени како датотеки, а посложените проекти се објавени како архиви.

Пребарувањето и повторната употреба на соодветна компонента обично се прави со преземање и анализа на содржина (изворниот код и документација) на датотеки или архиви од складиштата, додека не се најде вистинската компонента. Овие складишта нудат само тривијална категоризација на компоненти/проекти, но не обезбедуваат флексибилно пребарување. Избор на јазик за опис на хардвер (VHDL, Verilog, и др.) е често задолжително поле во филтрите за пребарување, што значи дека дизајнерите не може да пребаруваат со помош на „описи на високо ниво“, туку мора да пребаруваат низ множества од компоненти, групирани според јазикот или категорија на проектот.

Процесот на пребарување на HDL компонента одзема многу време и го одвлекува вниманието на дизајнерите од нивната главна цел. Тие често преферираат да создадат една компонента повторно, наместо да пребаруваат, и покрај тоа што се свесни за големата веројатност дека тоа би можело да се најде во некои од складиштата [64]. Еден од најважните проблеми со системите за складирање е нивната пристап на индексирање на HDL датотеките базиран на индексирање на текстуална датотека, што значи дека не нудат структурирани податочни складишта соодветни за создавање на интелегентни модули за пребарување и алгоритми за композиција.

Со цел да се создаде можност да дизајнирање на софтверски агент кој ќе го направи автоматска споредба и спојување на IP јадра, треба да постојат описни мета-податоци за компонентите. Процесот рачно внесување на мета-податоци бара дополнителни напори од дизајнерите, а од друга страна големо множество со IP јадра е веќе достапно на Интернет како чист HDL (без мета-податоци). Бидејќи процесот на рачна анотација е комплициран и одзема многу време, потребно е одредено ниво на автоматизација. Јазичите за опис на хардвер се структурирани јазичи и често дизајнерите дури и ги следат парадигмите на структуриран дизајн, па автоматска софтверска алатка би била во можност да направи многу повеќе од индексирање на едноставна текстуална датотека. Ние предлагаме семантички описи базирани на онтологија, кои ќе обезбедат значајни описи компонентата (со минимално или без рачно аотирање) и ќе им овозможи напредно пребарување на HDL компоненти.

Обидувајќи се да ги обединат различните шеми на податоци, Google, Microsoft, Yahoo и Yandex, заедно со зголемено учество на општата веб-заедница, неодамна

го создадоа Schema.org ⁷ што претставува активност на заедницата со мисија да се создаде, одржува и промовира заедничка шема за структурирани податоци на Интернет, на веб страни, во електронски пораки, но и пошироко. Основната идеја е да се објавуваат, да се поврзуваат и реискористуваат податочните шеми во еден вокабулар, бидејќи заедничкиот вокабулар е поедноставен за програмерите да донесат одлука за шема, што ќе ја користат и да се добие максимална корист [65]. Сепак, во моментот не постојат ентитети и релации од областа на хардвер.

Следејќи го и применувајќи го концептот на споделен вокабулар во доменот на јазици за опис на хардвер, ние ја применуваме идејата на стандардот Семантички веб, објавен од страна на W3C ⁸, за објавување на податоци и меѓусебно поврзани ресурси [51] [66], наместо објавување на документи и веб страници. Документи и веб страници се примарно наменети за луѓето, а не за компјутери и софтверски агенти, а со тоа не нудат основа за било каква дополнителна автоматска интеграција и анализа на податоци [67]. Семантичкиот веб дефинира стандарди за еднозначни формати за размена на податоци, кои се доволно блиску до природниот јазик на луѓето и се разбирливи за компјутерите. Овие алатки и формати нудат сосема поинаква перспектива за складирање на податоци и пребарување, со подобрени механизми за споделување, што е добра појдовна точка за подлабока анализа и интеграција податоците достапни на Веб.

Овој дел е организирана на следниот начин. Во поглавјето 3.1.1 ги претставуваме досегашните достигнувања во областа. Поглавјето 3.1.2 ги објаснува предностите на Семантичкиот веб и мотивација за користење на онтологијата за анотација на HDL компоненти. Во поглавјето 3.1.3 ја презентираме онтологија која ги интегрира различните HDL формати во шема на податоци за опис на хардвер на високо ниво. Поглавјето 3.1.4 го илустрира прототип системот и примери за полноважност и употребливост на онтологијата, а поглавјето 3.1.5 ги наведува главните придобивки од предложениот концепт. Додатокот 3.1.6 претставува едноставен пример на семантички анотирана компонента.

3.1.1 Достигнувања во областа

Алатките за дизајн на хардвер стануваат се поедноставни за користење и скоро секој кој сака, може за разумно време да успее да дизајнира и симулира сопствен хардвер. Постојат многу интелегентни уредувачи на код, паметни синтетизери, симулатори и дури и претварачи на код од повисоките програмски јазици во јазици

⁷Schema.org вокабулар, <http://schema.org/>

⁸Семантички веб стандард, <https://www.w3.org/standards/semanticweb/>

за опис на хардвер [26]. Некои од современите интегрирани развојни околина нудат и можности за развој без пишување на код, што го прави дизајнот на хардверски кола привлечен и интересен и за почетници [27] [28], но и за поискусни дизајнери. Слободниот пристап до развојни алатки резултираше со големи количини отворен код достапен на Веб. Компаниите кои работат со јазици за опис на хардвер исто така имаат сопствени складишта, каде се чува нивниот развоен код. Но, без разлика на големата количина на код достапен на Интернет или во компанијата, се појавуваат потешкотии кога дизајнерите сакаат да пронајдат и реискористат дел од тој код во некој нов дизајн. Процесот одзема време бидејќи корисникот мора да пребарува низ кодот и рачно да ги идентификува деловите кои му се потребни.

Постојат неколку обиди за развој на софтверско решение кое поддржува сподедување и реискористување на HDL код [29]. Проектот VALBOA и VALBOA јазикот за опишување на интерфејс (BIDL) [30] иницијално се базирале врз идејата за користење на Tsk/Tk за создавање на скриптирачки јазици кои ќе овозможат опишувањена костурот на архитектурата на систем-во-чип, но потоа била дефинирана XML шема за спецификација на IP интерфејс. Конзорциумот SPIRIT⁹ го објавил стандардот IP-XACT, кој понудува проверка на комплетноста на XML податоци за опис на електронски систем. Тој содржи мета-податчни форми и правила за семантичка конзистентност. IP-XACT овозможува забрзување на процесот на дизајн, пренесување, валидација, документација и користење на електронските IP јадра и опфаќа компоненти, дизајни, интерфејс и тн. Исто така забележливи се и пристапите фокусирани на дефинирање на параметризирани компоненти [31] [32] [33] и насочени кон создавање на бази со генерички компоненти, кои овозможуваат лесна и брза интеграција во идни проекти.

Неколку други пристапи, исто така, се важни, а некои од нив се фокусираат на дефиниција на параметризирани компоненти [68] [69] [70] и имаат за цел да создадат голема база на генерички дефинирани компоненти, со што нудат лесно прилагодување и интеграција во идните проекти. Други се фокусирани на код за верификација [71], што е секако многу важно прашање во автоматската композиција на компоненти од различни автори, со системски добиена гаранција на квалитетот и точноста. Покрај тоа, постојат системи и алатки кои користат групи на правила и граматика, со цел да овозможат автоматски дизајн хардвер [72] [73].

⁹Spirit Consortium, <http://www.spiritconsortium.org/>

Поради комплексноста на јазиците за опис на хардвер, многу предизвикувачка и атрактивна тема се чини дека е на интеграцијата помеѓу јазиците за опис на хардвер (најчесто SystemC) и повисоки програмски јазици (како на пример C или C++) што ќе им овозможи на помалку искусните дизајнери на хардвер, или дури и програмери кои немаат било каков контакт јазик за опис на хардвер, да синтетизираат некои специфични архитектури во програмабилен чип. Автоматски Алатка AutoPilot на AutoESL во комбинација со платформа имплементација Xilinx ја покажува ефикасноста на најдобрите алатки наменети за C-во-FPGA синтеза [74]. Алатката LegUp [75] е уште една алатка која автоматски ги компајлира C програмите во хибридна архитектура, која содржи FPGA-базиран процесор и сопствени хардверски оптимизации.

Квалитетот и постигнатите резултати на овие алатки и проектите не смее да се занемари, но најголем проблем се чини е што бараат дополнително внесување на податоци од страна на дизајнерите, што секогаш е пречка. Исто така, еден потенцијален проблем е недостатокот на централно складиште, соодветна шема анотација и надградлив концептот за создавање на мета-податоци. Како решение за овој проблем, ние предлагаме да се применуваат постојните Семантички веб стандарди, практики за анотација и алатки за управување со мета-податоци.

Употребата на Семантички веб технологии и во неколку други области, како што се здравството и хемијата резултираше со многу успешни приказни, што претставува голем чекор напред во автоматизација на многу процеси кои вклучуваат податоци и информации. Количината на податоци претставени во семантички веб-формат, постојано расте, формирајќи голема мрежа на ресурси, позната како множество на поврзани податоци (од англ. „Linked Data“) [76]. Тоа е значително подобрување на концептот на поврзување и објавување на податоци, што овозможува нови кориснички сценарија, кои се достапни со изолирани складишта на податоци.

Примената на семантичките технологии веќе има произведено многу успешни апликации [24], како NextBio ¹⁰, што е семантичка платформа за агрегирање и интерпретација на големи количества генетски податоци за истражување и клинички апликации. Data.gov.uk е уште еден пример на објавување на владини податоци во структуриран формат, мотивирајќи ги развивачите на софтвер да создадат многу корисни апликации ¹¹. Powerset ¹²(сега му припаѓа на Binq) е алатка за пребарување на некој од природните јазици, базирана на семантички

¹⁰NextBio, <http://www.nextbio.com/>

¹¹Data.gov.uk apps, <http://data.gov.uk/apps>

¹²Powerset, <http://www.powerset.com/>

технологии. Една од водечките електронски продавници во САД, BestBuy ¹³, користи семантички технологии за зголемување на видливоста на нејзините продукти и услуги. BestBuy ги означува податоците (како име на продавница, адреса, работно време и гео-локација) со помош на RDFa, со што алатките за пребарување се во можност да ги идентификуваат тие податоци полесно и да ги стават во контекст, зголемувајќи го сообраќајот на Веб и подобрувајќи ги услугите за клиентите. Откако го откупи Ellerdale (компанија за семантичка анализа во реално време), Flipboard ¹⁴ понуди многу интелегентни функционалности за категоризација и преглед на дневни вести и стана лидер во полето на „персонализирани социјални весници“. Дури и Facebook има искористено семантички технологии во неговата популарна платформа за пребарување Open Graph ¹⁵.

Врз основа на успехот на споменатите апликации и придобивките кои тие ги нудат на крајните корисници, ние ги применуваме Семантичките веб технологии во областа на дизајн и повторна употреба на хардвер, искористувајќи го структурираниот формат на јазиците за опис на хардвер за автоматско создавање на семантички описи, и претварање на отворен код во јазиците за опис на хардвер во поврзани податоци.

3.1.2 Примена на семантички технологии кај јазиците за опис на хардвер

Семантички веб технологии се изградени гради врз база на рамката за опишување на ресурси (од Англ. „Resource Description Framework“ - RDF) ¹⁶, кој им овозможува на опис на ресурси со користење тројки, со структура слична на едноставните реченици во еден јазик. Секоја тројка додава одредено знаење за ресурсот. Првиот елемент, наведен како субјект, претставува ресурс што се опишува. Вториот елемент од тројката е карактеристика или својство, а третиот елемент е одредена вредност за тоа својство. Ресурсите и својства се претставени со единствен идентификатор на ресурси (од Англ. „Uniform Resource Identifier - URI). Секој ресурс има свој единствен идентификатор на глобално ниво и може да се користи и во многу други системи, без никакви загуби на информации. Идентификаторот, исто така, ни овозможува означиме и ресурси кои постојат надвор од нашиот систем, односно некаде на Интернет.

¹³BestBuy, <http://www.bestbuy.com/>

¹⁴Flipboard, <https://flipboard.com/>

¹⁵The open graph protocol, <http://ogp.me/>

¹⁶Рамка за опишување на ресурси, <http://www.w3.org/RDF/>

Постојните Семантички веб формати овозможуваат структуриран опис на ресурси кој може да се користи и референцирани од страна на неограничен број на надворешни системи. Облакот на поврзани отворени податоци (од Англ. „Linked Open Data“ - LOD) [77] е водечки потфат во поврзаноста, објавување и повторната употреба на ресурси, својства и класи од различен домен, меѓу голем број на хетерогени платформи и услуги.

Во секој домен, важно е споделувањето на податоци да се врши во формат, кој овозможува флексибилно пребарување во процесот на откривање, без промена на оригиналната структура на податоците. RDF ги исполнува сите овие барања, бидејќи RDF ресурсите само покажуваат кон оригиналните податоци, без да ја менуваат нивната структура или содржина. Оригиналните податоци можат да бидат откриени и извадени од пребарувањето на содржината RDF со SPARQL јазик за пребарување [78]. Бидејќи RDF ресурсите само покажуваат кон оригиналните податоци, тие можат да се складираат, анализираат и истражуваат независно од самите податоци. Користење на современите семантички алатки и рамки, како Virtuoso¹⁷, Jena API¹⁸ и Sesame¹⁹ овозможуваат напредни функционалности.

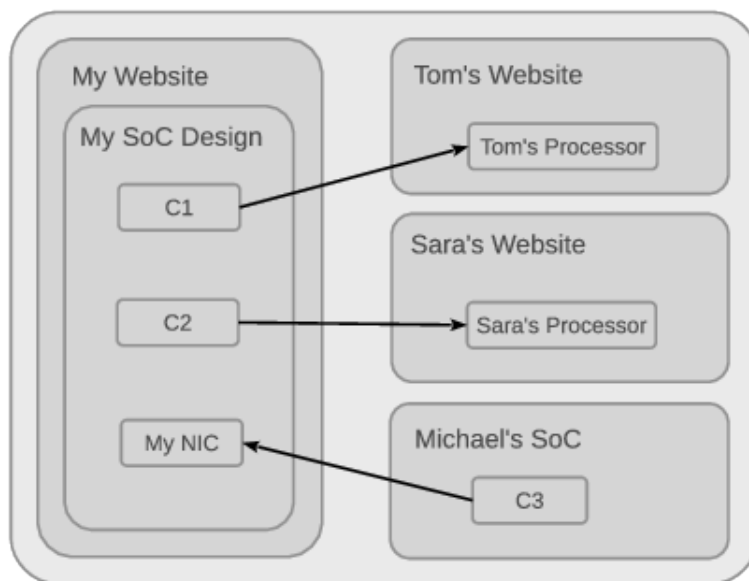
За да се искористат придобивките од стандардите, алатките и рамките за развој на семантички апликации, потребно е да се дизајнира онтологија за моделирање на доменот на податоците, која потоа ќе се користи за нивна анотација. Бидејќи јазиците за опис на хардвер следат предефинирана синтакса, процесот на анотација беше автоматизиран со користење на HDL-во-RDF мапирање [54]. Процесот на анотација резултира со RDF ресурси, кои ги опишуваат компонентите пронајдени во HDL датотеките или проектите. Секоја HDL датотека може да биде референцирана од повеќе ресурси, опишувајќи различни компоненти или делови од компоненти дефинирани во домен-онтологијата. На ваков начин е можно да се објават и комерцијални компоненти и проекти, бидејќи објавувањето на RDF описот, не се објавува оригиналниот HDL код, туку само мета-податоците. Структурата на SoC онтологијата не само што ќе овозможи пребарување на компоненти според нивните својства, туку ќе овозможи и пронаоѓање на меѓусебно слични и компатибилни компоненти.

Еден пример за откривање, упатување и повторна употреба на компоненти со описи од различни складишта е прикажано на Слика 3.1. Имаме сценарио каде што компонентите „C1“ и „C2“ од проектот „My SoC Design“ се користат во други складишта („Tom’s Website“ и „Sara’s Website“), а нашите „My NIC“ се користи

¹⁷Virtuoso, <http://virtuoso.openlinksw.com>

¹⁸Jena API, <https://jena.apache.org>

¹⁹Sesame, <http://www.openrdf.org>



Слика 3.1 Општиот концепт на поврзани компоненти. Компонентите „C1“, „C2“ and „C3“ само се поврзуваат со оригиналните изворни датотеки, правејќи го процесор на искористување лесен за одржување.

во проектот „Michael’s SoC“. Во овој случај, компонентите „C1“, „C2“ и „C3“ со помош на нивните идентификатори, а нивниот RDF опис може да се добие со де-референцирање на нивните идентификатори. Со ова, ние обезбедуваме следење на повторната употреба на компонентите, со што индиректно се утврдува валидноста и квалитетот на компонентите, така што повеќе референци ја прават компонента посигурна за други корисници. Покрај тоа, кога референцирани компонента се менува, складиштето може да го извести секој корисник на компонентата користи преку протоколот за Семантички нотификации [79].

Овој концепт дава начин на напредна функционалност на автоматизирана композиција на компоненти, која ќе се овозможи со семантичката анотација. Постои видлив напредок во областа композиција на Семантички веб сервиси [80] во насока на автоматизација на корпоративни процеси [81].

Следејќи ја парадигмата на повторно искористување на онтологии, ние реискористуваме онтологии во процесот на изработка на нашата онтологија за опис на хардвер. Бидејќи секоја HDL компонента, всушност, е документ или збир на поврзани документи, може да се опише во детали со помош на Dublin Core онтологија и нејзините својства. Тие претставуваат вокабулар што може да се користи за опишување на документи, различни видови на веб-ресурси, како што се видео ресурси, слики, веб-страниците, физички ресурси, како што се книги

и уметнички дела. Dublin Core онтологијата може да се користи со едноставен опис на ресурсите, како и за да се комбинираат лексикони на различни стандарди и за обезбедување на интероперабилност помеѓу онтологиите во Облакот ²⁰ и Семантички веб имплементации. Со користење на Dublin Core онтологијата, ние може да се опише на фактот дека една компонента е дел од друга, може да го дефинираме нејзиното име, автор, јазик, кодирање итн. Сите описи ќе ја користи истата шема како и постојните податоци од облакот на поврзани податоци, и ќе бидат достапни за семантички пребарувачи и корисници, и тоа во отворен и стандарден формат.

Бидејќи компонентите најчесто се групираат и се објавуваат во форма на проекти, од суштинско значење е да се обезбеди анотација на проектите како целина. Онтологијата за опис на проекти (од Англ. „Description of a Project“ - DOAP) е RDF Шема и XML вокабулар соодветен за опишување на софтверски проекти. Создаден е за да овозможи на семантичка анотација на софтверски проекти со отворен код ²¹.

Тековни проекти често се развиваат на јавни или приватни складишта достапни на Интернет, па семантичкиот опис, исто така треба да содржи информации за фазата на развој, следење на верзии, промени, програмери, итн. OpenLink неодамна ја објави онтологија за GitHub ²². Онтологијата состои од речник за опишување на односите меѓу проектот и ентитетите поврзани со него.

Дизајнерите на хардвер се личности поврзани едни со други преку проекти и организации, па нашите онтологии за хардвер исто така, треба да овозможуваат опишување на овие односи меѓу авторите на компоненти. Ова ќе обезбеди целосно опремен семантички облак на податоци за дизајн на хардвер. Онтологијата FOAF (од Англ. „Friend of a Friend“) е машински-разбирлива онтологијата ²³ и се користи за да се опишат луѓе и нивните односи со другите луѓе и објекти. Бидејќи онтологијата FOAF е јавно достапна, секој може да ја користат за да се опише себеси или да опише група на луѓе или социјална мрежа. Главната улога на онтологијата FOAF во онтологиите за опис на хардвер е добар опис на авторите, кои се поврзани со проектот.

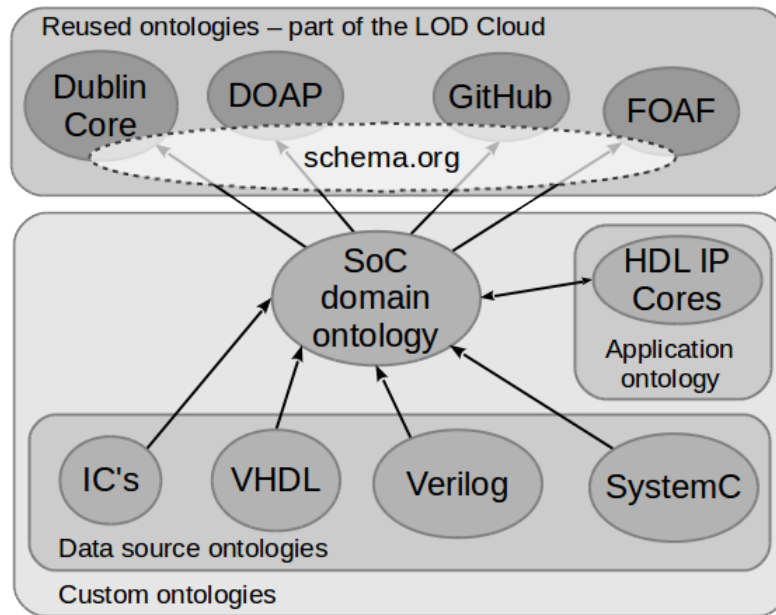
Schema.org ги обединува претходно споменатите онтологии и многу други популарни онтологии, па тоа е шема на податоците која треба да се користи со

²⁰Облак на поврзани отворени податоци, <http://lod-cloud.net/>

²¹DOAP онтологијата, <http://usefulinc.com/ns/doap>

²²GitHub онтологијата,
http://www.openlinksw.com/data/turtle/cartridge_ontologies/github.ttl

²³FOAF онтологијата, <http://xmlns.com/foaf/0.1/>



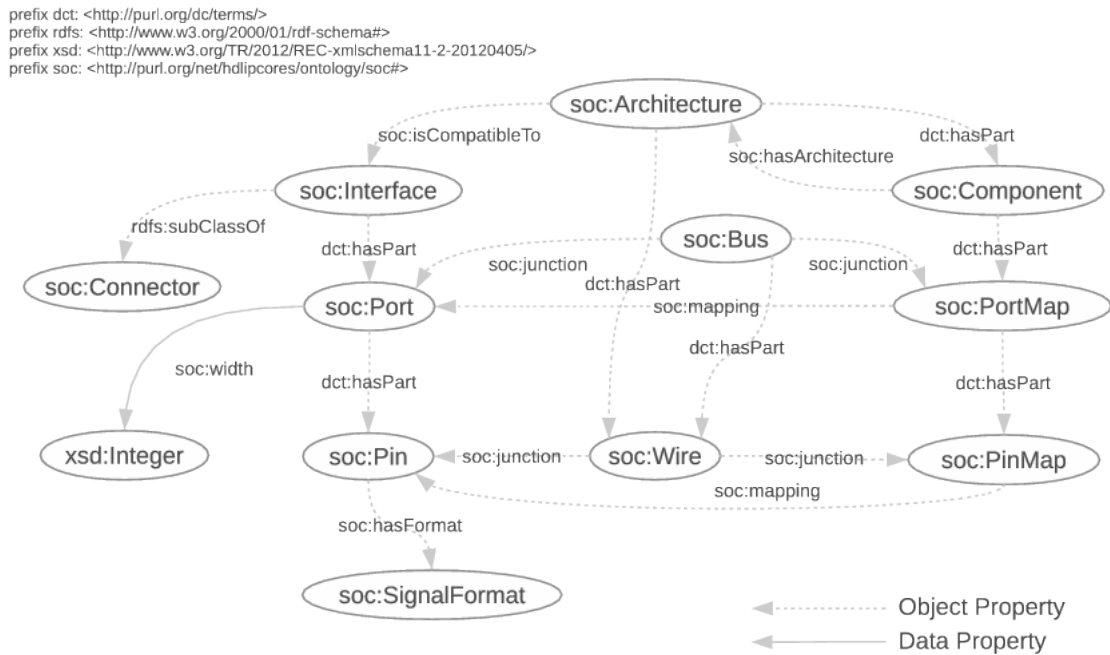
Слика 3.2 Онтологиите за дизајн на хардвер. Онтологијата SoC реискористува неколку онтологии кои се дел од облакот на поврзани податоци и дава опис на воопштен модел на хардвер.

цел да постои генерализирана шема на податоци. Сепак, со оглед на тоа што веќе се поставени еквивалентни мапирања, нема разлика дали ќе се користи на пример својството „hasPart“ на онтологијата Dublin Core или на Schema.org.

3.1.3 Онтологии за опис на хардвер

Најдобрите практики на поврзани Податоци [76] и концептот на развој на семантичките системи, разгледани во [82], заклучуваат дека за било кој семантички систем мора да постои домен онтологијата, која ќе го опишува знаењето во даден домен, онтологија на изворот на податоци, тесно поврзана со типот на извор на податоци, и онтологија за апликација, која ги опишува ентитетите и релациите. Според оваа парадигма, ние предлагаме множество на онтологии, прикажано на Слика 3.2.

Онтологиите на врвот на облакот на Слика 3.2 веќе беа споменати, а онтологијата во средината (SoC) е нашата домен онтологија. Таа служи како основна шема за опис на хардверска архитектура, доволна за да се аотира постоечки систем во чип. На долниот дел се прикажани ICs, VHDL, Verilog и SystemC онтологии на различните извори на податоци. Секоја од нив ги наследува главните концепти за хардвер од SoC онтологијата, но обезбедува и дополнителна шема специфична



Слика 3.3 SoC домен онтологијата (SoC.owl). Елипсите претставуваат класи и стрелките претставуваат својства (релации). Онтологијата може да се користи како шема за опишување на хардверска архитектура на високо ниво, независно од јазикот за опис на хардвер.

на јазикот за опис на хардвер. На десната страна на Слика 3.2, е прикажана апликациската онтологија на HDL IP Cores системот (HIPC.owl). Таа ги опфаќа сите класи и релации потребни за имплементација на функционална апликација, а со тоа нуди клиентски можности со иновативен концепт на чување и пребарување на HDL IP јадра.

Бидејќи процесот на семантички анотација може да се направи целосно автоматски, RDF тројките може да бидат генерирани и на клиентска страна, веднаш што кодот е подготвен за споделување или објавување. Исто така може да бидат генерирани во позадина, додека на дизајнерот го пишува кодот. Со оваа функционалност, дизајнерот не треба да направи ништо дополнително или да внесува дополнителни податоци, но сепак процесот на пребарување и на прибирањето на HDL компоненти може да биде дизајниран на сосема поинаков начин, со цел да ги користи меѓусебно поврзани податоци од RDF тројките.

SoC - домен онтологија

Главната цел на SoC домен онтологијата, прикажана на Слика 3.3, е да се обезбеди шема за анотација на хардверски архитектури²⁴. Секое интегрирано коло (IC) има интерфејс (црна кутија со пинови) и архитектура (структура во внатрешноста на црната кутија), така што архитектурата треба да се поврзе со интерфејсот.

Во SoC онтологијата има поткласи на хардверска архитектура (како порта, мултиплексер, процесор, меморија, итн.), кои се најчесто користените видови на компоненти. Исто така постои и едно множество на ограничувања, односно правила, кои ги дефинираат условите кои еден ресурс мора да ги задоволи за да се биде член на некоја поткласа.

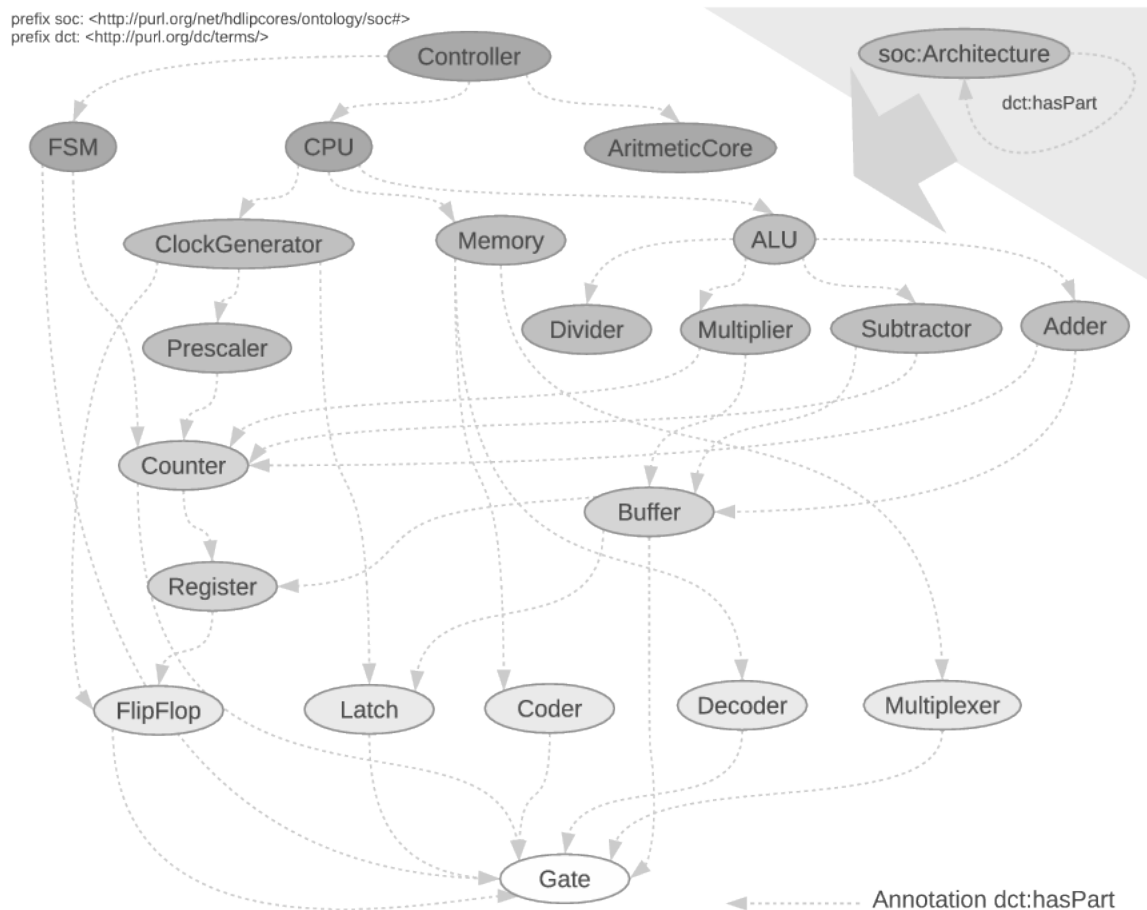
Интерфејсот има порти и секоја порта има еден или повеќе пинви. Пиновите се опишани со нивниот формат на сигналот. Интерфејсот може да има специфично пакување (конектор), на пример, DIP20, ZIP20, итн. Од друга страна, архитектурата може да содржи инстанци на компоненти (со различни архитектури), што треба да бидат мапирани во тековната архитектура со жици. Сите овие правила се вклучени во дефиницијата на SoC онтологијата, прикажана на Слика 3.3.

Својствата „hasPart“ (што е прикажано во Слика 3.4) се аксиоми кои дефинираат хиерархија помеѓу типовите на компонента во онтологијата. Хиерархијата започнува со комплексни видови, како што се контролор и конечен автомат, а завршува со едноставни типови, како што е порта. Својството „hasPart“ е дефинирано како асиметрично. На Слика 3.5 е прикажан пример на зависноста помеѓу порта и регистар.

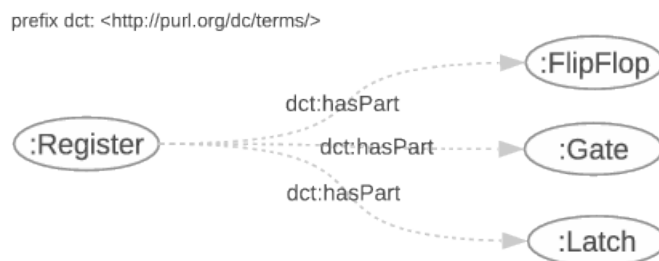
Структурата од Слика 3.4 е погодна за автоматско одредување на тип на компонента. Ако модулот за класификација на HDL IP Cores системот успее да генерираж множество на предвидените типови на компонента (не само еден), типот исто така може да се изведе од ацикличниот граф, со примена на некои од познатите алгоритми за најмлад заеднички предок (од англ. „Lowest Common Ancestor“ - LCA). Постојат одредени алгоритми, кои најпрво ја претвараат граф-структурата во стебло и потоа наоѓаат заеднички предок на дадено множество на јазли [83].

Друго својство во онтологијата е „isConnectableTo“, симетрична релација. Нејзината цел е да се обезбеди знаење за можни поврзувања, што претставува чекор напред кон автоматска композиција на компонентите. Едноставен пример за анотација во која се наведува дека процесор може да се поврзе со меморија, порта и тест-компонента е прикажана на Слика 3.6.

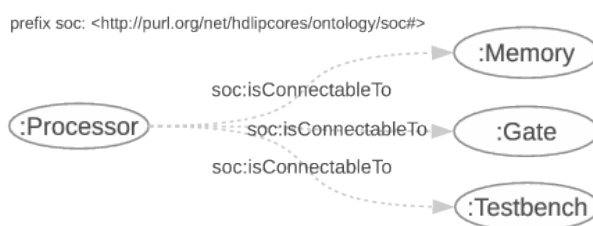
²⁴SoC онтологија, <http://purl.org/net/hdlipcores/ontology/soc>



Слика 3.4 Насочен ацикличен граф, формиран од од аотации на својството „soc:hasPart“. Претставено е знаење вградено во SoC онтологијата, кое се користи во процесот на одредување на типот на компонентата. Сите типови на компоненти се поткласи на класата „soc:Architecture“.



Слика 3.5 Мал подграф на SoC онтологијата, што го илустрира својството „hasPart“ со кое се наведува дека „FlipFlop“, „Port“ или „Latch“ може да се инстанцира во внатрешноста на „Register“.

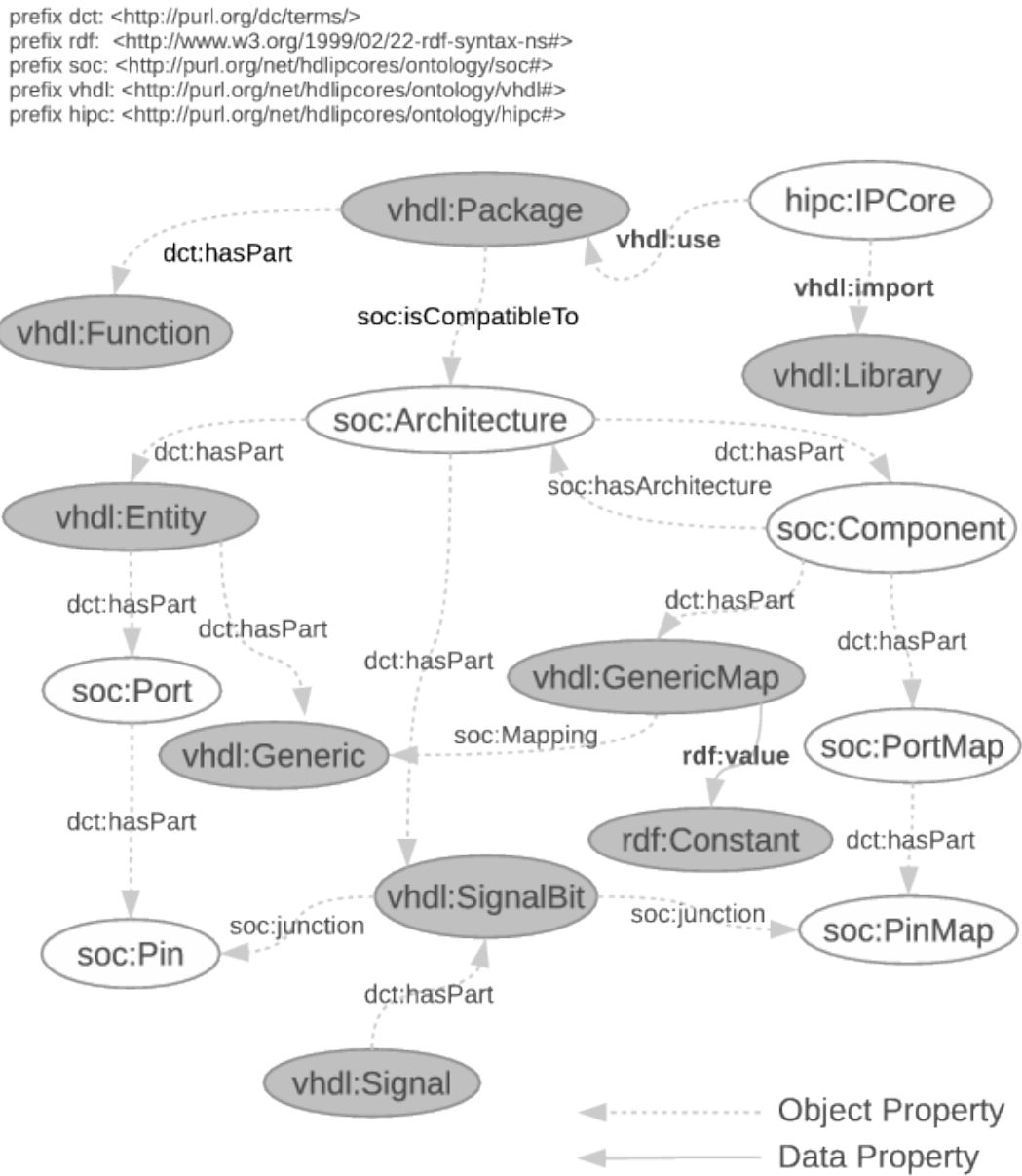


Слика 3.6 Еден дел од знаењето за можните поврзување на компоненти вградени во онтологијата, што покажува дека „Processor“ може да биде поврзан со „Memory“, „Gate“ или „Testbench“ компонента.

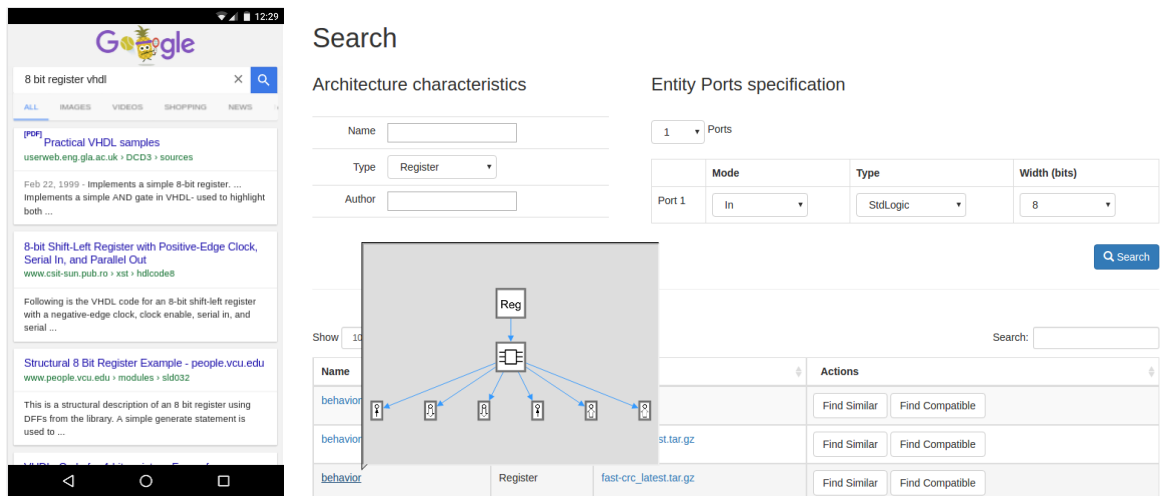
Користејќи го ова знаење и семантичките описи на архитектурите, апликација може автоматски да го процени нивото на поврзаност меѓу компонентите и да дава предлози и специфични пребарувања низ складиштето со компоненти.

Една од главните парадигми во дизајнот на хардвер е апстракција на ниво, каде што покомплексните компоненти се имплементираат со инстанцирање на повеќе едноставни компоненти. Нашата онтологија овозможува споделување на компоненти со нивно инстанцирање, што е директна последица на можноста за пребарување и пронаоѓање во анотираниот изворниот код. Од друга страна, овие вгнездени компоненти се важни за главната компонента за да се одреди нејзиниот тип или други нејзини карактеристики. Основната идеја на овој дел од онтологијата е да се овозможи анотација на инстанците во главното јадро, пребарување и пронаоѓање во складиштето, одредување на нивниот тип и одредување на типот на главното јадро.

За многу дизајнери, вреди да се нагласи важноста на создавање на тест-компоненти, заради што додадовме класа „Testbench“, како подкласа на класата „Architecture“. Тест-компонентите најчесто немаат надворешни порти и тоа е еден начин како може да се распознаваат, но од друга страна тие целосно подлежат на останатите парадигми за моделирање на архитектурата. Тест-компонентите да се



Слика 3.7 VHDL онтологијата (VHDL.owl). Сивите елипси и означени стрелки ја сочинуваат VHDL онтологијата. Останатите класи и својства припаѓаат домен онтологијата и онтологијата за апликацијата, кои се реискористуваат во онтологијата за VHDL.



(a)
Веб страници

(b) Прелистувач на ресурси со прозорец во кој се прикажува графот на компонентата.

Слика 3.8 Едноставни текстуални веб страници во споредба со веб ресурси.

гледаат и како обични компоненти, па алгоритмите за пребарување и композиција ќе ги третираат рамноправно со сите други видови на компоненти.

Со цел да се обезбеди развојна околина базирана на шаблони, имплементиравме концепт за шаблон на систем, кој е корисна алатка за почетници, во насока на брзо првично составување на систем во чип. Но, тоа е корисно и за поiskusни и корпоративни корисници, бидејќи овозможува дефинирање на шаблони на проекти. Шаблонот на еден систем содржи име и листа на типови на компоненти кои мора да се инстанцираат, со цел да се создаде одреден тип на систем. На пример, минимална околина за извршувањето на асемблер може да се изгради со употреба на процесор и меморија. Уште повеќе, ние може да ги означиме и условите на меморијата со услови за инстанцирање на нејзините делови со помош на релацијата „hasPart“ итн.

Онтологија за VHDL

Нашата онтологија за VHDL²⁵ ги содржи основните VHDL конструкции [84], како влезна и излезни типови на порта, типови на податоци и карактеристики, кои се потребни за претставување на VHDL ентитет како логички блок. Онтологијата се користи за да се класифицираат и аотираат VHDL компоненти со цел да се зачуваат деталите на корисничкиот изворен код во системот. Дополнителни

²⁵VHDL онтологија, <http://purl.org/net/hdlipecores/ontology/vhdl>

информации во врска со примери за употреба на онтологијата може да се најдат во [85].

онтологијата на VHDL е изведена од SoC онтологијата. Таа користи и класи на апликациската онтологија. Новите релации, специфични за VHDL онтологијата, се означени со задебелени букви, а новите класи се исполнети со сива боја на Слика 3.7. Основните модули, како што се „Generic“-блокот и „Signal“-блокот се поддржани. За овозможување на покомплексни VHDL IP јадра, концептите на пакет (од англ. „Package“), функција (од англ. „Function“) и библиотека (од англ. „Library“) се исто така додадени.

Со цел да ги следиме W3C принципи за објавување на поврзани податоци ²⁶, ги објавивме SoC и VHDL онтологиите со користење на постојани идентификатори ²⁷. Ова овозможува дереференцирање на идентификаторите додадени во процесот на анотација, така што дефинициите на класи и релации кои се користат во RDF тројките во било кој анотиран систем може да бидат откриени и искористени за валидација на податоците и изведување на ново знаење.

3.1.4 Прототип апликација според предложениот концепт

За да се евалуира концептот на примена на семантички веб технологии доменот на дизајн на хардвер, имплементираме веб апликација, HDL IP Cores ²⁸ која управува со семантички описи на HDL IP компоненти, нуди едноставен кориснички прелистувач (Слика 3.9), овозможува флексибилно пребарување на специфични компоненти врз основа на недвосмислен семантички опис и работи со постоечки HDL IP јадра. Главната цел на системот е да се обезбеди пребарување базирано на веб ресурси (Слика 3.8b), што веќе постои за лица или други објекти. Пребарувајќи за личноста "Алберт Ајнштајн на пример, се добива веб ресурс со множество на предефинирани својства (родното место и датум итн.), додека ако пребараме „8-битен регистер во VHDL“ се добиваат неколку резултати со чист текст или веб страници, кои содржат делови од пребарувањето според клучен збор, како што е прикажано на Слика 3.8a.

Системот има сеопфатна софтверска архитектура и содржи веб робот, кој пронаоѓа јавно достапни HDL IP јадра на Веб и модул за анотација кој го претвора во отворен HDL код поврзани HDL податоци, со користење на онтологиите опишани погоре. За да се докаже валидноста на предложениот концепт за анотација, веб

²⁶Најдобри практики за објавување на поврзани податоци, <http://www.w3.org/TR/ld-bp/>

²⁷Постојан единствен идентификатор на ресурс, <http://purl.org>

²⁸HDL IP Cores систем, <http://hdlipcores.finki.ukim.mk>

роботот периодично ги изминува најпопуларните складишта на глобално ниво, прикажани во Табела 3.1, и успешно аотира 416 проекти, 6590 IP јадра и 7709 архитектури.

Складиштето на RDF податоци е објавено во согласност со најдобрите практики за објавување на поврзани податоци ²⁹ и е достапен онлајн ³⁰. Податочното множество може да се користи од сите заинтересираните страни од заедницата и индустријата, како појдовна точка за градење на напредни HDL алатки за дизајн, кои ги искористуваат семантичките карактеристики на податочното множество.

Исто така, беше имплементиран и клиентски додаток (од англ. „plugin“) за развојната околина Eclipse [86]. Ако се користи заедно со било кој едитор за HDL, корисникот може да пристапи до сите функции на веб платформата директно од својата развојна околина. Додадокот се состои од неколку команди на контекстното мени. Секоја од нив испраќа специфични, контекстно зависни кориснички барања до серверот, а откако серверот ќе врати одговор, клиентот може директно да ја преземе HDL датотеката директно во својот проект.

Специфичните функционалности овозможени од нашиот систем се (1) специфично пребарување според порти и тип на компонента, (2) брзо и лесно прелистување на резултатите, (3) пребарување на компатибилни компоненти, (4) контекстно зависно пребарување, врз основа на претходно пронајдена компонента или локална компонента на корисникот и (5) системот е централна собирна точка, односно нуди виртуелна централизиран пристап кон најчесто користените HDL складишта.

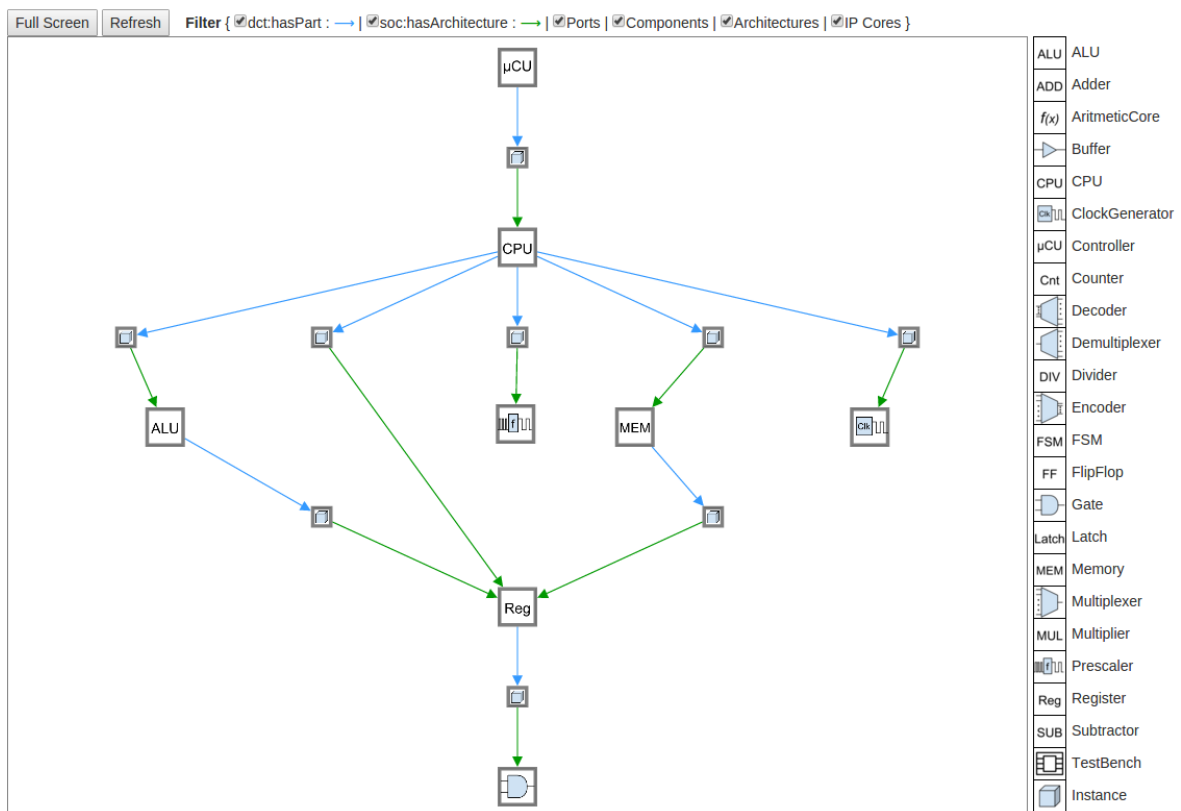
3.1.5 Отворени можности

Предложените онтологии за дизајн на хардвер отвораат широк спектар на можности за реискористување на кодот и нудат лесна автоматска аотација на постоечките множества на IP јадра, што е многу важно за дизајнерите на хардвер, како и за компаниите кои поседуваат големи складишта. За да се провери концептот и неговата применливост, ние развивме веб-базиран систем кој прави автоматска семантичка аотација на IP јадра достапни на Веб и го поддржува споделувањето компонента, откривање, повторна употреба и класификација кон автоматска композиција на HDL IP јадра.

Со користење на развиените онтологии, кои се дизајнирани со помош на класи и релации од најчесто користените онтологии и вокабулари од LOD Облакот,

²⁹Најдобри практики за објавување на поврзани податоци, <https://www.w3.org/TR/ld-bp/>

³⁰Складиште на податоци, <http://hdlipcores.finki.ukim.mk/hdlipcores/data>



Слика 3.9 Прелистувач на проект кој графички го претставува HDL проектот, нуди прозорец со детали и се отвора специфичната архитектура со еден клик. Типовите на компоненти се означени рачно и се користење на тековно развиениот модул за класификација (кој е во идна работа и е надвор од доменот на овој документ).

Табела 3.1 Веб портали на складишта со компоненти и број на анотирани IP јадра. Ресурс претставува веб-ресурс (датотека или компресирана архива), проект се однесува на HDL проект, IP јадро означува HDL датотека и архитектура е хардверска архитектура.

Бр.	Веб портал	Ресурси	Проекти	IP јадра	Архитектури
1	http://opencores.org	861	296	4006	5480
2	http://www.csee.umbc.edu	654	2	774	628
3	http://www.ht-lab.com	41	35	361	330
4	http://members.optusnet.com.au	38	16	201	190
5	http://www.ece.msstate.edu	98	28	288	186
6	http://www.eda.org	32	2	149	172
7	http://esd.cs.ucr.edu	97	0	88	132
8	http://www.asic-world.com	303	0	106	112
9	http://www.vhdl.org	638	2	111	96
10	http://tams-www.informatik.uni-hamburg.de	14	2	84	80
...
-	Вкупно	2841	416	6590	7709

ние сме во состојба да ги поврземе овие податочни множества со податоци веќе објавените на LOD Облакот. Ова поврзување овозможува хардверските компоненти да се трансформираат во отворени поврзани податоци со 5-свезди ³¹, и обезбедува поддршка за нови кориснички сценарија кои се недостапни со изолирани бази на податоци.

Бидејќи нашиот пристап во овој труд се базира на дефиницијата на компонентите и описи во општ јазик, односно во мета-јазик, IP јадрата напишани во различни јазици може да се споредуваат. Дополнително, опишаната инфраструктура може да се користи за подобрување на алатки кои го претвараат кодот од еден во друг јазик.

Во иднина, нашата работа планираме да го примениме концептот на микроподатоци и RDFa, за вградување на семантиката во документите. Семантичкиот опис, односно RDF тројките, ќе бидат во внатрешноста на изворната HDL датотека, така што автономни софтверски агенти ќе бидат во можност автоматски да ги разбираат и реискористуваат, врз основа на кориснички дефинирани контекст.

3.1.6 Пример за семантички опис

Со Користење на онтологијата, ние генерира соодветни RDF код за опишување на HDL компонентите. Како што беше наведено погоре, онтологијата опфаќа интерфејс на компонентите и семантички опис на типот на компонента. Исто така,

³¹5 Star Open Data, <http://5stardata.info/>

може да се дефинира семантички тип на портите, со што би се класифицирале портите како податочни, контролни итн.

Да се илустрира на процесот на анотација ќе биде прикажан едноставен пример со „НИ“ порта, напишан во VHDL, претставени во листата 3.1. Нашиот систем го парсира VHDL кодот и автоматски генерираниот RDF опис (во RDF/XML формат) е претставен во листата 3.2 (заради обемноста, некои делови се изоставени).

Листа 3.1 NAND_GATE.vhd

```

library ieee;
use ieee.std_logic_1164.all;

entity NANDGATE2 is
    port(x : in STD_LOGIC;
         y : in STD_LOGIC;
         z : out STD_LOGIC);
end NANDGATE2;

architecture NANDGATE2 of NANDGATE2 is
    signal c: std_logic;
    component NOTGATE1
        port(n_in : in STD_LOGIC;
             n_out : out STD_LOGIC);
    end component;
    component ANDGATE2
        port(a_in1, a_in2 : in STD_LOGIC;
             a_out : out STD_LOGIC);
    end component;

begin
    N0: ANDGATE2 port map(x, y, c);
    N1: NOTGATE1 port map(c, z);
end NANDGATE2;

```

Листа 3.2 NANDGATE.rdf - системски генериран RDF опис за NANDGATE.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dct="http://purl.org/dc/terms/"
  xmlns:soc="http://purl.org/net/hdlipcores/ontology/soc#"
  xmlns:vhdl="http://purl.org/net/hdlipcores/ontology/vhdl#"

```

```

xmlns:hipc="http://purl.org/net/hdlipcores/ontology/hipc#"
xmlns:datarepo="http://purl.org/net/hdlipcores/ontology/data#"
xml:base="http://purl.org/net/hdlipcores/ontology/"
<hipc:IPCore rdf:about="data#ipcore20140721122935344">
  <soc:name>NAND_GATE.vhd</soc:name>
  <dct:hasPart>
    <vhdl:Architecture rdf:about="data#ipcore20140721122935344A0">
      <soc:name>NANDGATE2</soc:name>
      <dct:hasPart>
        <soc:Component
          rdf:about="data#ipcore20140721122935344A0C0">
          <soc:name>NOTGATE1</soc:name>
          <dct:hasPart>
            <soc:Port
              rdf:about="data#ipcore20140721122935344A0C0P0">
              <soc:name>n_in</soc:name>
              <vhdl:hasMode rdf:resource="vhdl#In"/>
              <vhdl:hasDataType rdf:resource="vhdl#StdLogic"/>
            </soc:Port>
          </dct:hasPart>
          <dct:hasPart>
            <soc:Port
              rdf:about="data#ipcore20140721122935344A0C0P1">
              <soc:name>n_out</soc:name>
              <vhdl:hasMode rdf:resource="vhdl#Out"/>
              <vhdl:hasDataType rdf:resource="vhdl#StdLogic"/>
            </soc:Port>
          </dct:hasPart>
        </soc:Component>
      </dct:hasPart>
      ... Omitted description for ANDGATE2 component ...
    </dct:hasPart>
    <vhdl:Signal
      rdf:about="data#ipcore20140721122935344A0S0">
      <soc:junction
        rdf:resource="data#ipcore20140721122935344A0C0P0"/>
      <soc:junction
        rdf:resource="data#ipcore20140721122935344A0C1P2"/>
      <soc:name>c</soc:name>
    </vhdl:Signal>
  </dct:hasPart>
  <soc:isCompatibleTo>
    <vhdl:Entity
      rdf:about="data#ipcore20140721122935344A0E0">

```

```

    <soc:name>NANDGATE2</soc:name>
    <dct:hasPart>
      <soc:Port
        rdf:about="data#ipcore20140721122935344A0EOP2">
        <vhdl:hasDataType rdf:resource="vhdl#StdLogic"/>
        <vhdl:hasMode rdf:resource="vhdl#Out"/>
        <soc:name>z</soc:name>
      </soc:Port>
    </dct:hasPart>
    ... Omitted description for other ports ...
  </vhdl:Entity>
</soc:isCompatibleTo>
</vhdl:Architecture>
</dct:hasPart>
</hipc:IPCore>
</rdf:RDF>

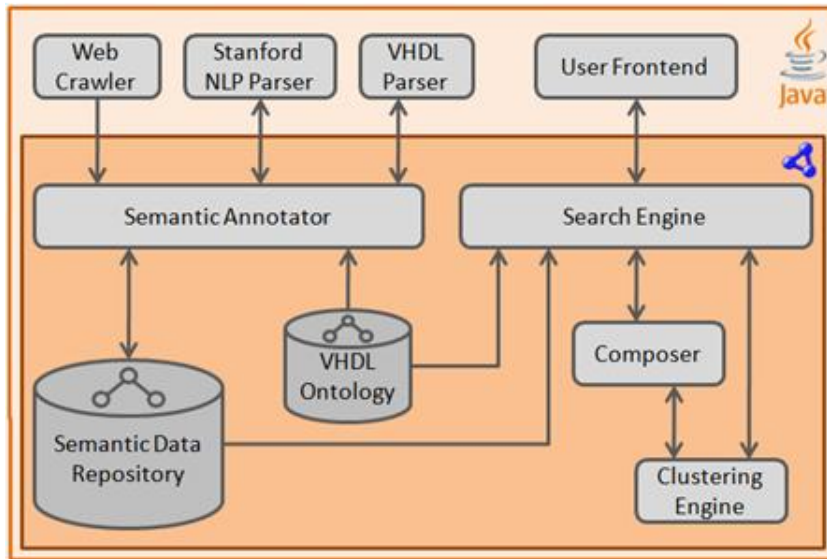
```

3.2 Класификација на IP јадра

Идејата е да се одреди типот на VHDL компонентата, со користење на композитни алгоритам, базиран врз правила. Алгоритмот се состои од (1) анализа елементите од VHDL кодот, (2) правила на заклучување, (3) евалуација на вгнездените компоненти и (4) кластерирање на граф од компоненти, со модули прикажани на Слика 3.10.

Елементите на VHDL кодот се превземаат со употреба на прилагоден VHDL парсер. Покрај VHDL кодот, коментарите и имињата на портите, ентитетите и сигналите, исто така, се значајни за семантичкиот опис. Бидејќи коментарите не се во структуриран формат и обично се во слободна форма на текст, напишан од страна на дизајнерите, ние го користиме парсер за процесирање на природни јазици („Stanford NLP parser API“) за да ги разделиме именките од другите текст блокови. Анализата е од клучно значење во овој чекор, со цел да се направи разлика на некоја имиња/типови на компоненти, како што се "CAN" контролер наспроти „can“ како глагол, на „AND“-порта и „and“ како сврзник и многу други.

Вториот дел од алгоритам е модул за изведување на заклучоци, кој се состои од збир на логички правила. Општо гледано, за секој тип на компонента, постојат неколку правила кои го опишуваат интерфејсот на компонентата. Правилата се сопствени и беа дефинирани со помош на подлабока анализа на многу примери на различни видови на компоненти.



Слика 3.10 Архитектура на модулот за класификација.

Со цел да се прошири на множеството на правила и да се додаде дополнителна семантиката на алгоритмот, се анализираат и вгнездените компоненти. Најпрво, сите вгнездени компоненти се идентификуваат и се анотираат, а потоа се заклучува типот на компонентата, што ги содржи.

Последниот дел од алгоритам е модулот за кластерирање. Овој концепт треба да го направи конечното групирање на компонентите, според сличноста на анализираната компонента со некое од множествата на компоненти. На пример, ако алгоритмот се двоуми помеѓу типот „меморија“ и типот „процесор“ и компонентата е групирани со процесори, тогаш ќе биде класифицирана како „процесор“.

Секој од чекорите на алгоритмот се извршува независно, резултирајќи со низа на предлози за типот на компонентата. На крајот, на пресекот на сите четири низи се пресметува и се зема како резултантна листа на типови на компонентата кои се дадени од страна на композитниот алгоритам. Ако на листата има само еден елемент, процесот е завршен, во спротивно се применува алгоритам за најмлад заеднички предок на членовите на листата, односно тип на компонента која може да содржи компоненти од типовите присутни во листата на резултати.

Во тековната имплементација, сите чекори на алгоритмот се смета како еднакво важни, со што учествуваат во конечниот резултат со коефициент 1. Ова е само едноставно пример-сценарио, што треба да се продолжи со посложени и поприлагодени формули, а дополнително може се прошири и со некои од принципите на машинското учење.

3.3 Вклопување на тест-компоненти

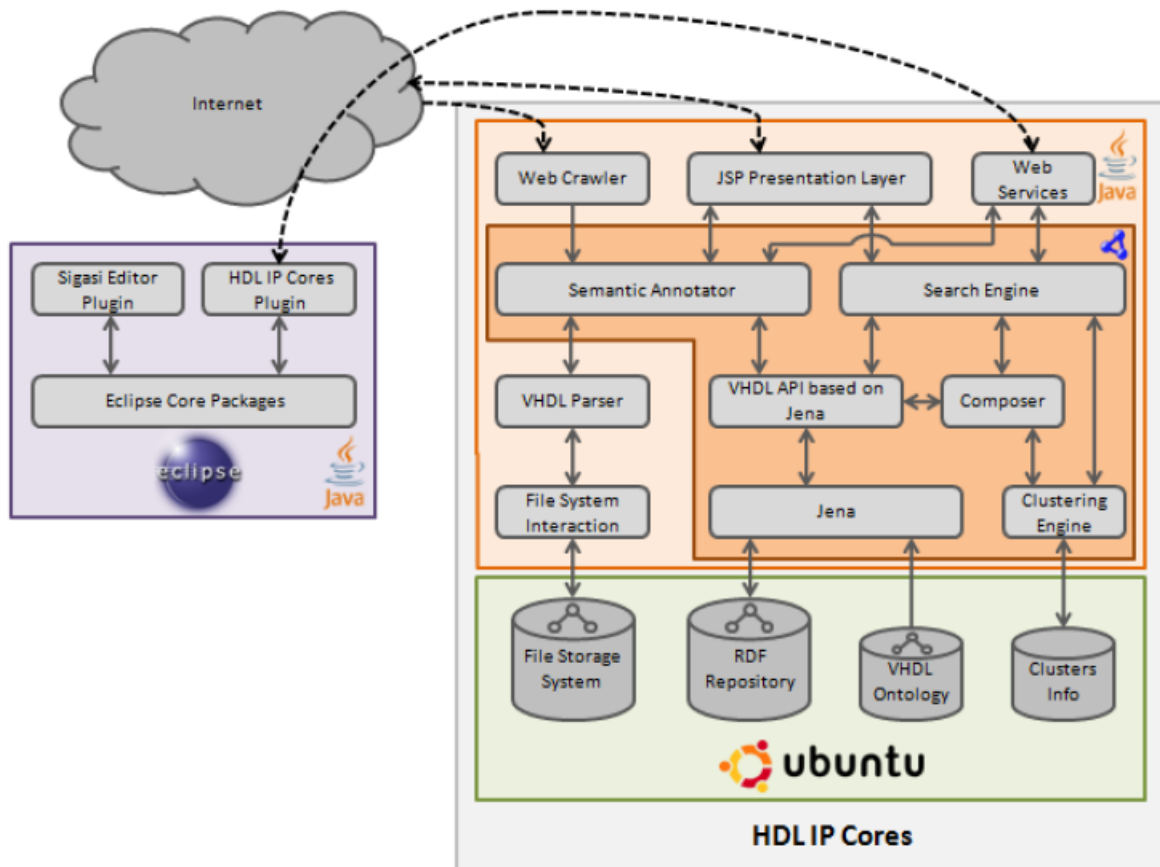
Честата употреба на програмабилни чипови во продукциските системи бара сигурност и стабилност, така што целиот процес на HDL дизајн мора да се одржува на корпоративно ниво и да се гарантираат задоволство и сигурност на корисниците. Компаниите, свесни за овој факт, се многу заинтересирани за подобрување на методите за тестирање и проверка на нивните производи [87] [88]. Некои од нив имаат и посебни одделенија за тестирање, целосно посветени на верификација на IP јадра [89] [90] и, исто така, имаат напредно тестирање на работните процеси, што во голема мера ја намалува веројатноста за грешка [91].

Ерата на развој на софтвер со отворен код овозможува значителен напредок во развојот на апликации. Надвор од комерцијалниот свет, има големо множество на отворен HDL код и многу тест-компоненти. Постојат неколку веб портали, групи и онлајн заедници ³². Овие складишта имаат огромен број проекти во кои е многу тешко да се најдат потребните компоненти и исто така не постои централен отвор за да се поврзат сите портали заедно, така што дизајнерот треба да ги посети еден по еден. Сепак, многу луѓе низ целиот свет работат на слични проекти и дизајнираат тесно поврзани IP јадра и тест-компоненти, но за жал не се свесни еден за друг, ги прават истите компоненти и непотребно трошат време.

Исто така, постојат голем број на тест-компоненти со отворен на Интернет и корпоративниот развој им овозможува и ја стимулира повторната употреба на кодот, што е од особено значење за заштеда на време. Но, многу тешко да се добие вистинската компонента, поради големиот раст на множествата на постоечки тест-компоненти достапни на Интернет, кои треба да се пребараат вклучително и постапката за валидација, пред искористувањето во новите проекти.

Големите компании имаат и свои огромни складишта на тест-компоненти, што се користеле во минатото (целосно верифицирани), и се соочуваат со истиот проблем на недостаток на алатки за автоматизација во своите граници, кои ќе го поддржат реискористувањето на кодот и ќе го забрзаат процесот на развој. Процесот на тестирање и дебагирање одзема прилично многу време [92] и е комерцијално неефикасен.

³² Open Cores – web portal, <http://opencores.org/>
Java optimized processor - group, <http://tech.groups.yahoo.com/group/java-processor/>
IP supermarket – web portal, <http://www.ipsupermarket.com/index.php>
Infineon – web portal, <http://www.ipsupermarket.com/index.php>
Lattice – web portal, <http://www.latticesemi.com/>
Chip Estimate – web portal, <http://www.chipestimate.com/>
Design & Reuse – web portal, <http://www.design-reuse.com/>



Слика 3.11 HDL IP Cores - архитектура на системот.

Дизајнот на HDL, без сомнеж, зависи од тестирање и симулација, кои понекогаш одземаат многу време [93]. Многу често, времето потребно за дизајнирање на тест-компонента може да се спореди со времето потребно да се подготви и основната компонента за која е наменето тестирањето, што е не е добредојдено за менаџерските тимови на компаниите.

Во таа насока, го претставуваме концептот за пронаоѓање на тест-компонента и лесна интеграција во работната околина на дизајнерите. Користејќи го на нашиот постоечки систем HDL IP Cores [85] и интегрирајќи ги клиентските алатки алатки и симулатори, успеавме да направиме чекор напред и да им обезбедиме на корисниците автоматизирана функционалност за пребарување на тест-компонента и можност да ја преземат директно во нивните развојни околина.

3.3.1 HDL IP Cores - преглед на системот

Системот, Слика 3.11, се состои од апликација на серверска страна и клиентски додаток (анг. „plug-in“) за развојната околина Eclipse. Апликацијата на серверот содржи веб робот, кој ги презема компонентите достапни на Интернет, и потоа секоја компонента минува низ процесот на семантичка анотација. Мета-податоците заедно со изворниот код на компонентата се чуваат во RDF складиштето. HDL IP Cores системот овозможува подлабока семантичка анотација и рангирање на компонентите според сличност и компатибилност со одредена компонента.

Вториот дел од нашиот систем е клиентски додаток за Eclipse, кој може да пристапи до јадрото на системот преку интерфејс со веб сервиси. Основната идеја е да им обезбеди на корисниците функционалност на пребарување и преземање на IP јадрата и семантичките описи достапни во складиштето на серверот. Може да се користиме „Sigasi“ уредувачот ³³ за овозможување на HDL функционалностите на клиентската страна, но сите други додатоци, кои обезбедуваат уредување на HDL, може да се користат. Семантичката анотација се базира на знаење сместено во нашата сопствена онтологија [94].

Функционалности на постоечките симулатори

Verissimo System Verilog Testbench Linter ³⁴ ја проверува компатибилноста на методологијата за верификација, што им овозможува на инженерите да извршат дополнителна ревизија на нивните тест-компоненти. Со оваа алатка, дизајнери можат да проверат дали нивниот код е без замки и потенцијални семантички ризици за грешка и дали е во согласност со соодветните методологии. Verissimo може да се прилагоди за да се проверат насоките за кодирање на одредена група или корпоративни заедница за да се обезбедат конзистентност и најдобрите практики за развој на код. На пример, можноста за спроведување на истата функционалност во неколку начини може да влијае врз симулацијата на перформансите или да доведе до неочекувано однесување.

VCS' Native Testbench (NTB) ³⁵ обезбедува интегрирана поддршка за сеопфатен дизајн на тест-компоненти во Verilog и OpenVera, вклучувајќи објектно-ориентирано, и функционална покриеност на можностите. VCS понатаму ги про-

³³Sigasi - интелигентен HDL уредувач за Eclipse,
<http://www.sigasi.com/>

³⁴Verissimo System Verilog Testbench Linter,
http://www.dvteclipse.com/Verissimo_SystemVerilog_Testbench_Linter.html

³⁵VCS' Native Testbench (NTB)
<http://www.synopsys.com/Tools/Verification/FunctionalVerification/Documents/vcs-ds.pdf>

пирува своите можности со „Echo“-технолозијата за конвергенција на изрази и ограничувања. Echo автоматски ефикасно го покрива просторот за ограничувања на тест-компонентите, значително намалувајќи го напорот потребен за верификација на големо множество на функционални сценарија. Echo корисен за сите тимови, што користат тест-компоненти во Verilog со случаен избор на ограничувања.

Напредниот симулатор на Cadence ³⁶ ги поддржува сите IEEE стандардни јазици, Методологија на отворена верификација (OVM), универзална методологија за верификација на Accellera е (UVM) и методологијата за реискористување (ERM), што прави брзо и лесно да се интегрира со постоечките процеси за верификација. Функционалноста на симулаторот обезбедува тесна поврзаност помеѓу тест-компонентата и компонентата, која треба да се тестира. Ова овозможува автоматска верификација на софтвер за вградливи системи, третирајќи го како дел од компонентата која се тестира. Денес, напредниот симулатор форсира автоматизација на тест-компонентите, повторна употреба и анализа за проверка на дизајните на системско ниво, ниво на пренос на регистри (од англ. „Register Transfer Level“ - RTL) и на ниво на логички порти.

Симулаторот Xilinx ISE (ISim)³⁷ нуди целосен HDL симулатор со сеопфатни можности интегриран во развојната програмска околина Xilinx ISE. Симулацијата може да биде дури и повеќе фундаментален чекор во рамките на дизајнот со тесна интеграција со ISim во рамките на развојната околина. Алатките на Xilinx автоматски генерираат VHDL код во датотеката на тест-компонентата за да овозможат дефиниција на сигналите на колото и дефинирање на влезови и излези. Симулаторот има и неколку други алатки со цел да се стартува, пазира или прекине симулацијата.

Покрај споменатите имплементации на симулатори и нивната основна логика [95], исто така има и корисни идеи за концептуално подобрување [96] [97]. Сите претходно споменати симулатори имаат различни карактеристики кои ја имплементираат симулацијата (верификацијата) на тест-компонентите. Но, доаѓаат до израз откако дизајнерот рачно ќе ја инстанцира тест-компонентата во симулаторот и потоа ќе ги користи расположливите алатки. Ниту еден од претходно споменатите симулатори не нуди автоматско пребарување на соодветна тест-компонента на Интернет и лесно реискористување на кодот.

³⁶Cadence's Enterprise Simulator,
http://www.cadence.com/products/fv/enterprise_simulator/pages/default.aspx

³⁷Симулатор Xilinx ISE (ISim),
<https://www.xilinx.com/products/design-tools/isim.html>

3.3.2 Модул за пронаоѓање на тест-компонента

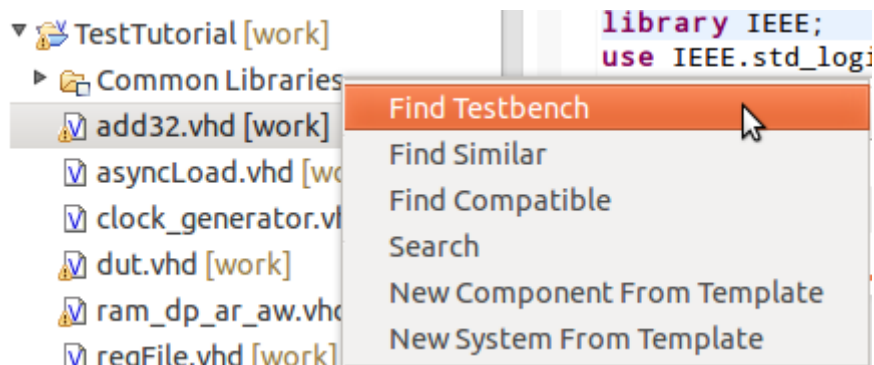
Нашиот пристап се фокусира кон контекстно зависна алатка за пребарување на тест-компоненти, која употребува база на знаење изградена врз онтологија. Системот користи семантички анотирани податоци за да ја најде соодветната тест-компонента и да ја интегрира во развојната околина, во нашиот случај Eclipse. HDL IP Cores системот ќе се користи за да се олесни генерирањето на листата со резултати од пребарувањето на тест-компоненти. Но, исто така, на серверска страна е потребна проверка на компатибилноста помеѓу тест-компонентата и тестираната компонента. Системот треба да изврши верификација, т.е. автоматска симулација на тест-компонентата, создавајќи ранг-листа на резултати со соодветни тест-компоненти. Симулацијата, исто така, може да се направи на клиентската страна, со помош на симулатор, кој е вграден во нашиот клиентски додаток или со користење на други податоци за симулација, веќе инсталирани во развојната околина на клиентот.

Процес на пронаоѓање на тест-компонента

Во нашиот систем, од пребарувањето на тест-компонента е направено со користење на модул за пребарување, базиран на домен-онтологија и RDF складиште [85]. Бидејќи HDL датотеките имаат добро дефинирана структура, анотацијата се прави автоматски, со користење на онтологијата како шема на податоци. И иако процесот не бара дополнителни податоци од страна на крајните корисници, сепак претставува чекор напред кон подобрување на пребарувачите на HDL код.

Нашата алатка му овозможува на дизајнерот да пребара директно од неговиот работен простор, односно да ја искористи алатката, Слика 3.12, повикувајќи ја со десен клик на датотеката на неговата компонента. тогаш компонентата на корисникот се испраќа до серверот HDL IP Cores, се генерира семантички опис за компонентата и се користи како влез во пребарувачот на компатибилни тест-компоненти.

Пребарувањето се прави со совпаѓање на семантичките концепти наведени во барањето на корисникот и семантичките анотации на компонентите кои се на располагање во RDF складиштето на системот, Слика 3.13. Исто така, многу важен дел од листата на резултати е мапирањето на портите помеѓу компонентата на корисникот и тест-компонентата (компонентата која се наоѓа во тест-компонентата во складиштето, Слика 3.13), што овозможува клиентската алатка да ја директно



Слика 3.12 Функционалности на клиентската алатка.

да ја инстанцира компонентата во тест-компонентата и да ја стартува симулацијата автоматски.

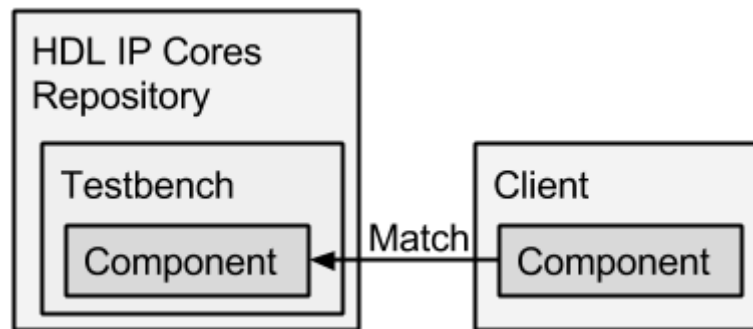
Тест-компонентите соодветни на пребарувањето се прикажуваат во Нов прозорец. Ова е особено важно за еден дизајнер, бидејќи тој ќе добие тест-компонента компатибилна со неговата компонента директно во рамките на својот проект, без прелистување на Интернет и преземање на стотици датотеки и архиви.

Системот обезбедува преглед на тест-компонентата и по изборот на корисникот, тест-компонентата ќе биде веднаш преземена во проектот и го води корисникот на чекор на извршувањето на симулација.

Треба да напоменеме дека автоматското преземање на датотеки се прави во време на извршување, директно од веб локацијата на оригиналната датотека. Не се зачувува HDL код во складиштето на HDL IP Cores, поддржувајќи ја интелектуалната сопственост и парадигмата на лиценцирање и можните промени на условите за лиценцирање во иднина. Директно преземање е достапно само за IP јадрата отворени со слободна лиценца, а ако јадрото е објавено под друг тип на лиценца, системот ќе го пренасочи корисникот на оригиналната веб страница, каде е објавена компонентата.

Верификација на компатибилноста на тест-компонентите

Тест-компонентата генерира влезни податоци за компонентата која се тестира и ги проверува излезните податоци, што мора да се направи претпазливо за да се најдат грешки. Моделите се опишани со одреден формализам, кој може да биде регистриран со модел специфичен на јазикот за опис на хардвер. Штом VHDL описот на однесувањето е напишан и множеството на тест-вектори е утврдено, се започнува функционална симулација.



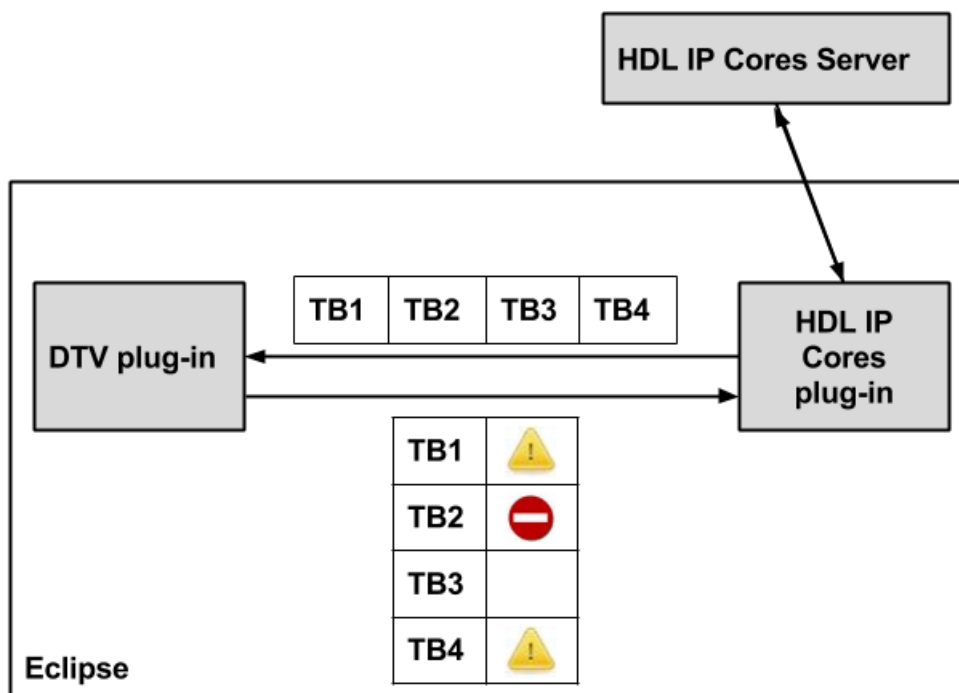
Слика 3.13 Процес на совпаѓање.

За да се изврши симулација, тест-компонентата мора да биде компатибилна со архитектурата или предупредувања и грешки ќе се појават во излезот на симулаторот. Ние предлагаме да се инстанцира тест-компонентата, да се стартува симулаторот автоматски и да се искористи излезот од симулаторот во процесот на рангирање на резултантните тест-компоненти. Покрај тоа, листата на резултати може да содржи предупредувања и грешки на секоја тест-компонента за да го известат клиентот дека доколку се одлучи за таа тест-компонента, тој ќе мора да ги прегледа и поправи предупредувањата и/или грешките пред да продолжи со процесот на симулација. Во нашиот систем постојат две можни решенија за верификација, на клиентската страна на серверската страна.

Првото сценарио е симулација на клиентската страна, која вклучува симулација на тековната тест-компонента. Постојат неколку различни алатки за симулација. Една од нив е DVT Eclipse, кој е додаток за Eclipse ³⁸, кој овозможува задоволителна развојна околина за симулација на тест-компонента. DVT беспрекорно се интегрира со сите водечки симулатори за хардвер за да се овозможи поедноставна анализа на симулацијата. Дизајнерот може да ја изврши симулацијата директно во неговиот проект. Со користење на можностите на DVT и интеграцијата со нашиот систем ќе овозможи автоматска проверка компатибилност на листата на тест-компоненти добиени од серверот на HDL IP Cores.

Алатката овозможува верификација на семантички анотирани тест-компоненти за избраната компонента, и му ги прикажува на дизајнерот тест-компонентите кои се најкомпатибилни со архитектурата на неговата компонента. Симулатор е интегриран на клиентска страна, како дел од пакетот Eclipse. Тоа ќе го забрза процесот на наоѓање тест-компоненти и ќе заштеди време во споредба со рачна

³⁸DVT Eclipse - интеграција на клиентската страна,
http://www.dvteclipse.com/?gclid=CM3Ex_G4_7oCFc1V3godKA4Abw



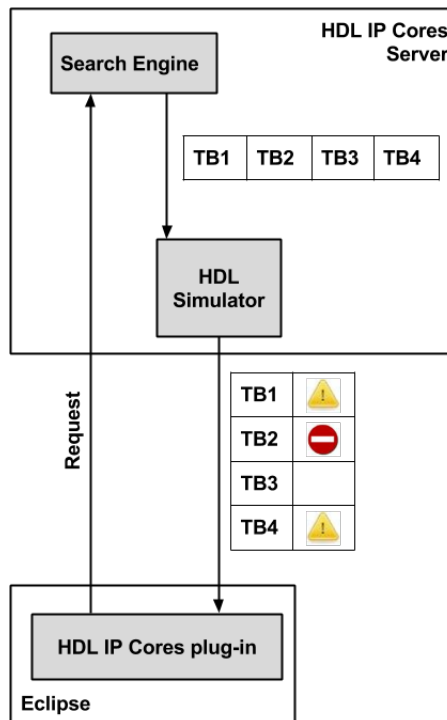
Слика 3.14 Симулатор на клиентска страна.

симулација, Слика 3.14, бидејќи тест-компонентата веќе ќе биде во проектот на дизајнерот и ќе ги опфати синтаксички и семантички проверки со означени грешки и иницијално мапирани порти.

Бидејќи проверката на компатибилноста е направена на клиентската страна, може дополнително да се обезбеди интеграција со неколку други симулатори достапни за Eclipse и корисникот ќе биде во можност да избере помеѓу нив и го користи неговиот омилен заедно со нашиот систем.

Второто можно решение, Слика 3.15, е да се постави симулаторот ³⁹ на страната на серверот. Неговата улога повторно ќе биде да се утврди компатибилноста помеѓу компонентите и тест-компонентите. Интеграцијата на симулатор на страната на серверот ќе генерира дополнителни податоци и ќе предизвика дополнително искористување на серверот поради верификацијата (симулацијата). Доцнењето ќе зависи од големината и типот на компонента и тест-компонента, но клиентот ќе ја добие конечната листа и HDL дизајнерот нема да мора да инсталира дополнителни алатки на клиентска страна, освен додатокот за HDL IP Cores системот.

³⁹Cadence симулатор - интеграција на серверска страна,
<http://www.cadence.com/ip/vip/pages/Default.aspx>



Слика 3.15 Симулатор на серверска страна.

3.3.3 Conclusion and future work

Со цел да се забрза процесот на генерирање на тест-компоненти го опишавме модулот за пребарување на тест-компонента на нашиот постоечки систем и два можни концепти за подобрување со користење на HDL симулатор. Архитектурата на системот и функционалност страна на клиентот се опишани во детали, презентирајќи глобална слика на целиот концепт. Следниот чекор ќе биде да се спроведат и двете сценарија и да се направат тестови и евалуација со цел да се добијат повратни информации од корисниците.

Главната придобивка за нашиот предлог концепт е што тест-компонентите ќе бидат означени во согласност со централна онтологија (во една компанија или во целиот свет) и корисниците ќе можат да најдат и да преземат тест-компоненти побрзо, многу лесно и директно преку својата работна околина, а без да треба да отвораат прелистувачот и да посетуваат десетици веб страници.

Во комерцијална околина е можно да се користи локална инстанца на HDL IP Cores системот и да се интегрира со складиштето на компанијата, искористувајќи ги информациите на корисниците, правилата за пристап и привилегиите. На овој начин ќе се обезбедат опишаните функционалности за HDL дизајнерите,

зачувувајќи ги нивните кодови внатре во компанијата, што е од суштинско значење во една корпоративна средина.

3.4 Карактеристика со помош на логичката функција

Јазиците за опис на хардвер (HDLs) најчесто се користат во процесот на производство на чипови [98] [60] [61], но во последно време се застапени и во продукциските верзии на системите [63], како периферни адаптери на податоци или конфигурирани како ко-процесори кои ја дистрибуираат процесирачката моќ на за да се намали товарот на централниот процесор [99] [100]. Овој нов период започнува нова ера за програмерите на повисоко ниво да ја имплементираат истата логика во програмабилен чип, доведувајќи ги многу поблиску до перспективата на HDL, што би било многу сложено без постоење на некаква рамка, која ќе го забрза развојниот процес.

Исто така голем пакет на HDL код е веќе објавен на многу јавни веб складишта. Постојат повеќе веб портали кои содржат проекти достапни под различни отворени лиценци (GPL, Apache, итн.). Иако порталите нудат основна класификација на проекти, сепак, наоѓањето на одредена компонента се уште е многу тешко и бавно. Порталите немаат можност за флексибилно пребарување (филтер според порти, тип на компонента и сл.).

Причините за недостаток на електронски алатки за автоматизација на дизајнот на хардвер и за споделување и повторна употреба на HDL код може да се идентификуваат во комплексноста на проблемот, што главно произлегува од точката каде што софтверска алатка треба да ја утврди внатрешната архитектура на компонентата. Применувајќи ја главната идеја на Семантичкиот веб [51] [52] [53], предлагаме нов пристап кон подлабоко означувањето на компонентите и повисоко ниво на семантичко знаење [67]. Јазиците за опис на хардвер имаат одредена структура, така што е можно да се направи автоматска семантичка анотација со користење на онтологијата и софтвер кој ќе генерира семантички ресурси за дигитални кола.

Идентификувајќи ја можноста за примена семантичките алатки за дизајн на дигитални системи [85], имплементиравме првична верзија на HDL IP Cores системот⁴⁰. Апликацијата извршува автоматска семантичка анотација само на интерфејсот на компонентите. Системот се состои од веб апликација и клиентски

⁴⁰HDL IP Cores, <http://hdlipcores.finki.ukim.mk>

додаток ⁴¹ за Eclipse. HDL IP Cores обезбедува автоматизација во собирање на податоци и функционалноста на пребарување, но иницијално имаше недостаток на подлабоко разбирање на типот на архитектура, што е од суштинско значење за секое напредно пребарување и композиција на компоненти. Решавајќи го овој проблем, имплементираме нов модул, кој рекурзивно ги аотира вгездените компоненти се до нивото на основни логички блокови и ја извлекува логичката функција на дадена архитектура. Буловиот израз за дадена архитектура ќе биде уште една карактеристика за споредување на компонентите и определување на сличноста, што доведува до подобрување на аалтките за споделување и повторна употреба на HDL кодот.

3.4.1 Достигнувања во областа

Семантичка анализа на логички кола е исклучително важна кај симулаторите и повеќето од нив нудат многу напредни функционалности. Меѓу нив, достапна е работната оклина Xilinx ISE со вграден симулатор ISim ⁴², кој им овозможува на многу прецизно и временски точно извршување време на симулации, генерирање на повеќе начини на симулација (рачно, графички, преку терминал) и голем број на формати во кои можете да ги зачувате резултатите од симулацијата. Симулаторот на Cadence ⁴³ овозможува верификација, според познатите методологии (од англ. „Open Verification Methodology“ - OVM и „Unified Verification Methodology“ - UVM), овозможува брза и лесна интеграција со различни процеси за верификација, како и симулација на податочно ниво (RTL), симулација на однесувањето, симулација со намалена потрошувачка итн. Некои симулатори ⁴⁴ овозможуваат автоматско генерирање на тест-компоненти, со користење на генерички подготвени библиотеки, со што значително го забрзува процесот на тестирање.

HADES⁴⁵ симулаторот најчесто се користи во академска средина, и се смета како алатка за почетници. Можноста на HADES се прилично отворени, бидејќи е модуларен и обезбедува отворен програмски интерфејс (од англ. „Application Programming Interface“ - API) во Јава, што е многу добра можност да се искористи

⁴¹Eclipse plug-in, <http://hdlipcores.finki.ukim.mk/plugin>

⁴²[13] Xilinx ISE Симулатор (Isim), <http://www.xilinx.com/tools/isim.htm>

⁴³Cadence's Enterprise Simulator, <http://www.cadence.com>

⁴⁴Verissimo System Verilog Test bench Linter,

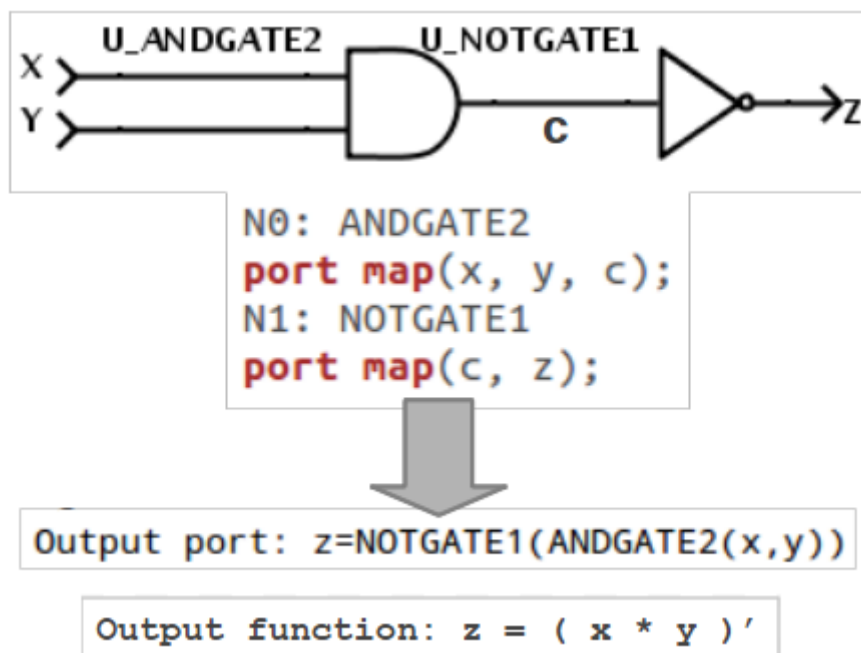
http://www.dvteclipse.com/Verissimo_SystemVerilog_Testbench_Linter.html

VCS' Native Test bench (NTB),

<http://www.synopsys.com/Tools/Verification/FunctionalVerification>

⁴⁵HADES,

<http://tams-www.informatik.uni-hamburg.de/applets/hades>



Слика 3.16 Анотација на логичката функција.

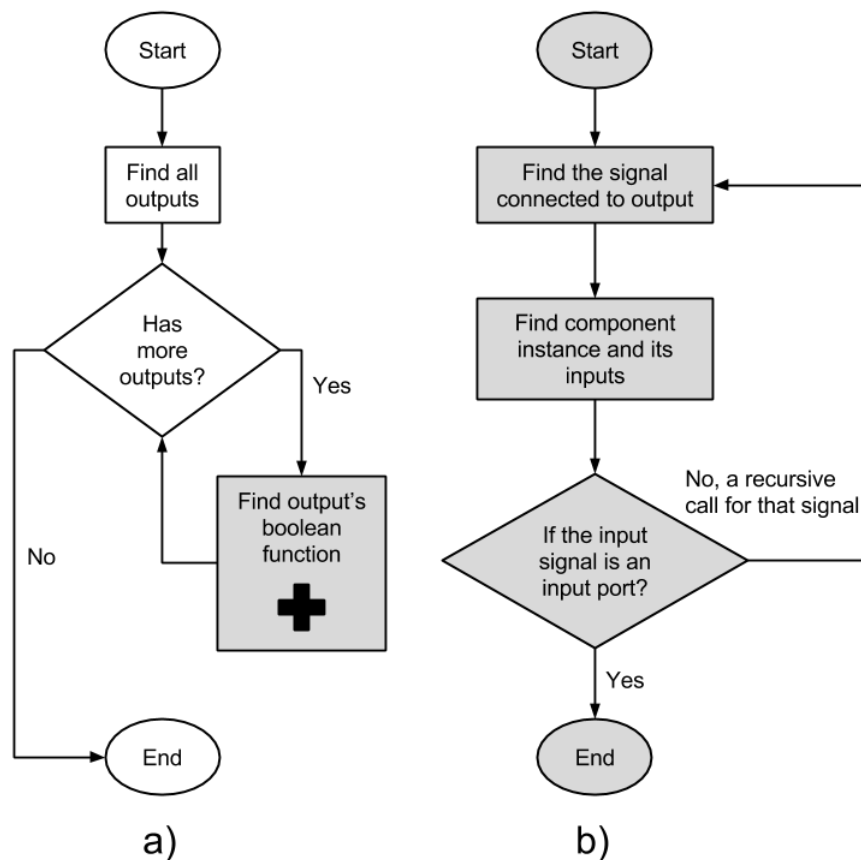
неговата внатрешна логика за семантичко извршување на симулации. Покрај тоа, содржи и модул за интеграција со VHDL.

Она што е важно е да се напомене е дека симулаторите се првенствено развиени со цел да се обезбеди временска точност и прецизно извршување на дадените модели отколку во насока на автоматско определување на типот на компонента и/или автоматска композиција и повторна употреба на системи. Сепак, некои од нивните функции е можно да се искористат за подобрување на семантичката анотација на архитектурата на компонентите.

3.4.2 Процес за извлекување на логичката функција

Автоматската анотација на интерфејсот на компонентите овозможува ниско ниво на семантичка анализа. Со цел да се овозможи целосно автоматско пребарување и автоматска композиција на системи, софтверкиот агент мора да ја земе в предвид внатрешната архитектура на компонентите.

Процесот на извлекување на логичката функција може да се одвои во две фази. Во првата фаза е автоматската семантичка анотација на внатрешната архитектура, а во втората фаза е претворање на добиените семантички ресурси во недвосмилена логичка функција, која ја опишува архитектурата. Двете фази

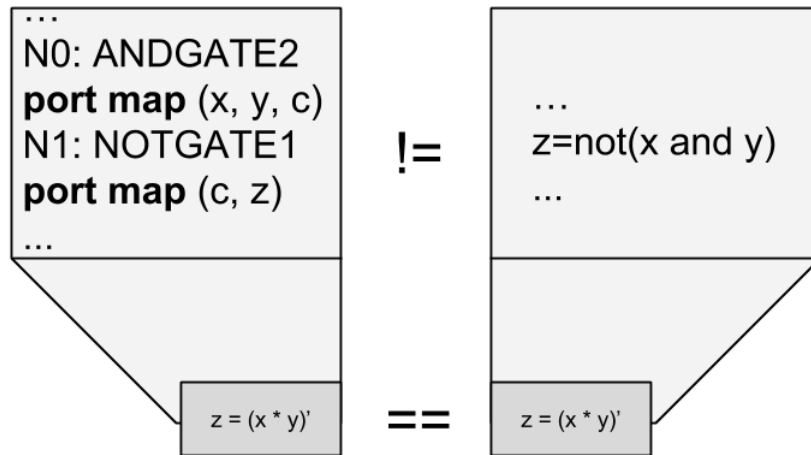


Слика 3.17 Дијаграм на алгоритмот. Алгоритмот се повикува за секоја излезна порта (a) и за секој излез рекурзивно ја наоѓа инстанцата и нејзините влезови, се додека стигне до влезовите на архитектурата (b).

се, всушност, меѓусебно поврзани, така што првата фаза е алатка за втората фаза, која го дефинира резултатот.

Во нашиот случај, првата фаза е направено од страна на модлот за анотација HDL IP Cores системот и внатрешната архитектура на компонентата е моделирана како меѓусебно поврзани инстанце на поедноставни архитектури. Оваа методологија за развој на нови архитектури е прифатен во парадигмата на структурно претставување на архитектура во рамките на јазиците за опис на хардвер, како што VHDL.

Во втората фаза ќе се извлече логичка функција со користење на семантички ресурси со кои е анотирана архитектура. Процесот на извлекување на логичка функција на архитектура, дефинирана во VHDL, е прикажан на Слика 3.16. Архитектура има интерфејс со два влезови (X и Y) и еден излез (Z). Таа содржи пример на „И“-порта („U_ANDGATE2“) и на пример на логичка инвертер



Слика 3.18 Ниво на абстракција на синтаксата. VHDL кодовите на двете компоненти се различни, но логичките функции се еквивалентни.

(„U_NOTGATE1“), меѓусебно поврзани со сигнал „C“. Делот од кодот кој ги инстанцира компонентите прикажани на Слика 3.16, додека целосна содржината на датотеката е прикажано во Листа 3.1.

Крајната цел е да се добие на логичка функција на излезна порта „Z“, односно да се најдат логички функции за сите излезни порти на архитектурата. Предложениот алгоритам за одредување на логичка функција на излезна порта има за цел да ги најде влезните сигнали од кои зависи функцијата, рекурзивно поврзувајќи ги внатрешните инстанци и сигналите поврзани со нив. Резултатот на алгоритмот е логичка функција на архитектурата во Пролог синтакса. Алгоритмот започнува од секоја излезна порта и е рекурзивно ги наоѓа инстанците и нивните влезови, додека се најдат влезовите архитектурата, Слика 3.17.

Описот на архитектурата во форма на логичка функција ќе биде добра основа за подобрување на алгоритмите и системи за автоматска класификација и композиција на компоненти, бидејќи ќе им овозможи недвосмислена споредба на две архитектури, според логичката функција за пребарување и соодветно рангирање резултатите според меѓусебната сличност и компатибилност.

На ваков начин се воведува ниво на апстракција на синтаксата, како што е прикажано на Слика 3.18 и различни компоненти (дури и во различни јазици за опис на хардверх) може да се споредат и еквивалентираат.

Како што веќе споменавме, предложениот алгоритам работи само за комбинациони кола, чија архитектури се опишани со структурна парадигма. Во случај кога се користат процеси (функции) за дефинирање на архитектурата на системите,

постапката за утврдување на логичка функција на архитектурата е значително посложена, бидејќи процесот ќе треба да се моделира како црна кутија со сопствен интерфејс, а потоа да се утврди тип на архитектура. Оваа постапка, не е можна без автоматско извршување на симулацијата, освен во наједноставни сценарија. Постапката е исто така тешко да се изврши за секвенцијални логички кола, па дури и употребата на симулатор не секогаш може да се очекува да доведе до недвосмислена определба на архитектурата.

Определување на процесот на логичката функција на комбинациони дигитални кола опишани во јазик за опис на хардвер, очигледно е дека како излез се добива логичката функција на излезната порта „Z“, од каде што може да лесно да се добие математичка функција на излезот „Z“. Таа може да се претстави како негација на производот на влезовите „x“ и „y“. Оваа операција сигурно припаѓа на класата на процеси на инверзното инженерство, бидејќи од постоечки HDL код се извлекува математичка функција (израз), која може да се користи, разгледува и анализира во различен контекст.

3.4.3 Отворени можности

Беше илустриран процесот на добивање логичка функција за комбинациони логички кола опишани со структурна парадигма, што е прв чекор кон автоматска семантичка анотација на внатрешната архитектура на дигиталните системи.

Следен чекор е да се прошири постапката со цел да се опфатат логичките кола што се дефинирани со процеси (функции). За таа цел, ние може да се искористи програмскиот интерфејс на симулаторот HADES. Покрај тоа, во него е достапна библиотека со класифицирани компоненти (порти, мултиплексери, бројачи итн.), што ќе биде добра основа за тренирање на алгоритам автоматско генерирање на правила за определување на типот на архитектурата на нови компоненти.

Потоа, недвосмислено или со користење на хеuristicки алгоритми ќе биде неопходно да се опише архитектурата на комплексни секвенцијални кола и дигитални системи. Овој концепт може да ги опфати повеќето од достапните компоненти и да се обезбеди напредни функционалности во рамките на системот HDL IP Cores.

Глава 4

Податочно воден пристап во областа на големи хомогени системи

4.1 Алгоритам за балансирање на потрошувачите

Електричните дистрибутивни мрежи се соочуваат со постојано растечки побарувања и/или промени од ниско/високо на високо/ниско ниво на потрошувачка. Иако потрошувачите вообичаено се избалансирани во три-фазните дистрибутивни системи за време на инсталацијата, растот на товарот како и промените во текот на денот резултираат во небалансирана состојба. Што јазелот е избалансиран првично ќе бидат во рамнотежа, но потоа балансирањето се нарушува со текот на времето. Потребата балансирање на товарот и минимизирање на активните загуби иницирале широк спектар на истражување на балансирање и распоредување на товар. Постојат два главни методи за балансирање на фазите: реконфигурација на јазлите на системско ниво [101–103] и замена на фазите на јазлите.

Замената на фазите отсекогаш била многу активна истражувачка тема. Неколку алгоритми, вклучувајќи сеопфатно пребарување, алгоритам на следење во минатото, алчен алгоритам, алгоритам со симулирано жарење, генетски алгоритам и динамичко програмирање, применети врз проблемот на балансирање на фазите се неодамна дискутирани и разгледувани во [104]. Ќе прикажеме краток преглед на достигнувањата поврзани со проблемот на замена на фазите.

Авторите на [105] развиваат алгоритам за хевристичко пребарување за балансирање на товарот во услови на постојано и променливо на оптоварување, која била тестирана во областа на опслужување на јужниот дел на Тајпеј, потврдувајќи ги практичноста на алгоритмот и неговото позитивно влијание. Шеми за оптимизација на замената на фазите се предложени во [106]: авторите прикажуваат решение

со илустративен пример и ги адресираат подобрените карактеристики на избалансираниот систем. Алгоритмот на симулирано жарење (анг. „Simulated Annealing“ - SA) беше воведен како ефикасен метод за решавање на проблемот на балансирање на фазите во [107]; авторите го формулираат проблемот на балансирање на фазите како пристап со комбинирано целобројно програмирање (од анг. „Mixed-Integer Programming“) и потоа се применува методот SA, споредувајќи ги резултатите со неколку други алгоритми и илустрирајќи глобално оптимално решение со примери. Авторот на [108] развива алгоритам за балансирање со акцент на практичните аспекти и имплементација, што овозможува да се укаже на ограничување на бројот на операции за промена на фазите и исклучување на одредени области, тешко достапни за екипажот. Трифазно балансирање на дистрибутивната мрежа со користење на имунолошкиот алгоритам кој гарантира брза конвергенција е предложен во [109], се применува и во системите за дистрибуција во Тајван (Taipower), со цел да се одговори на проблемите во снабдувањето предизвикани од големата струја во неутралниот проводник. Авторите на [110] со користење на можностите на постоечките географски информациски системи (од анг. „Geographic Information Systems“ - GIS), претставуваат програма за балансирање која се базира на автоматизирано мапирање, управување со објекти и географски информациски системи. Решавајќи го проблемот на балансирање на фазите, авторите на [111] го претставуваат новиот комбинациски метод, врз основа на алгоритам за хранење на бактерии (BF-NM), а постигнуваат брз пристап до најдоброто решение и брза конвергенција.

Периодично екипажот ги ребалансира јазлите, за време на одржување или обновување, кога нов клиент треба да биде поврзан или ако процентот на дебаланс надминува одредено ниво и кога моменталната конфигурација станува значително небалансирана. Три фактори се земаат во предвид во донесувањето на одлука повторно да се балансираат јазлите: материјалните трошоци за потребните промени на склопките, очекувањето подобрување на нивото на балансираност и привремениот прекин на електрична енергија на клиентите. Тековно ребалансот на фазите е комплицирано да се одржува рачно со старите електрични броила кои немаат можност за вградување на интелигенција, или да се поврзат во една логичка (компјутерска) мрежа и, според тоа е непрактичен, скап и одзема време.

Во овој труд ќе ги искористиме предностите на новите дигитални електрични броила кои имаат компјутерски и комуникациски можности и се веќе ставени во инсталациските панели, а некои се веќе и поврзани со локалната компјутерска мрежа или се одржуваат преку Интернет. Мрежата на паметни електрични брои-

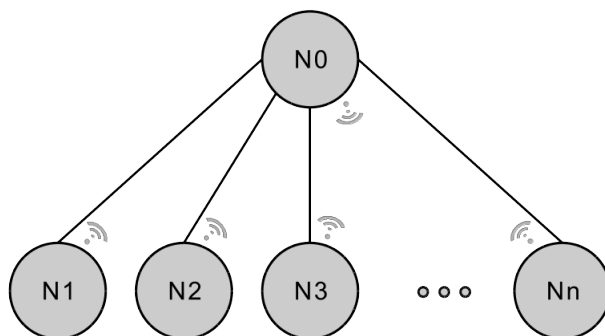
ла е добра основа за вклучување на дополнителна интелигенција и динамични напредни алгоритми кои ќе овозможат постабилна и поекономична дистрибутивна мрежа. Идентификувајќи го проблемот на балансирање на товар помеѓу фази во три-фазните системи системите и искористувајќи ги постоечките паметна мрежа на електрични броила, предлагаме едноставен иновативен алгоритам за динамичко, интелигентно балансирање на товар (од англ. „Dynamic Intelligent Load Balancing“ - DILB). Некои делови од предлог архитектура се базираат на постоечките хиерархиски, клиент-сервер концепти на адресирање. Претставени се алгоритмот и архитектурата на дистрибутивната мрежа надградена со DILB, кои се лесно применливи во секој дел од дистрибутивната мрежа, без потреба од суштински промени во клиентскиот дел од инсталацијата. Алгоритмот DILB ги намалува активните загуби на моќност во мрежата, што е докажано преку симулација со познатите IEEE 34-Bus, 37-Bus, и 123-Bus мрежи, така што очекуваното ниво на минимизација на активните загубите на моќност е потврдено.

The outline of the paper is as follows. Section 4.1.1 explores the main idea of the concept and the algorithm itself is defined in Section 4.1.2. The real data measurements and simulation results are presented in Section 4.1.3. Section 4.1.4 describes the architecture extension of an existing PDN that is simulated on the IEEE 34-Bus, 37-Bus and 123-Bus networks. Section 4.1.5 addresses the main benefits of the proposed concept.

Прегледот на поглавјето е следниот. Дел 4.1.1 ја прикажува главната идеја на концептот, а самиот алгоритам е дефиниран во Дел 4.1.2. Мерењата на реални податоци и резултатите од симулацијата се претставени во Дел 4.1.3. Дел 4.1.4 опишува надградена архитектура на постоечките дистрибутивни мрежи, која е симулирана на мрежите IEEE 34-Bus, 37-Bus и 123-Bus. Дел 4.1.5 се однесува на главните придобивки од предложениот концепт.

4.1.1 Опис на архитектурата

Со цел да се покаже основната идеја на нашиот иновативен концепт, ќе ја поедноставиме дистрибутивната мрежа на дрво со мала големина, кое се состои од еден родител јазел и n деца јазли, што е прикажано на Слика 4.1. Тоа е всушност мал дел од вистинска дистрибутивна мрежа, па дополнителни ограничувања нема да бидат дефинирани, со исклучок на малиот број на јазли. Ако би сакале да клиентската страна на електричната инсталација, тогаш јазелот N_0 , може да се смета како електрично броило во една зграда и јазлите N_1 до N_n како електрични броила во секој стан. Ако се поместиме чекор нагоре кон централниот дел на дис-



Слика 4.1 Поедноставена архитектура на дистрибутивна мрежа, формирана од еден родител јазел и неговите деца.

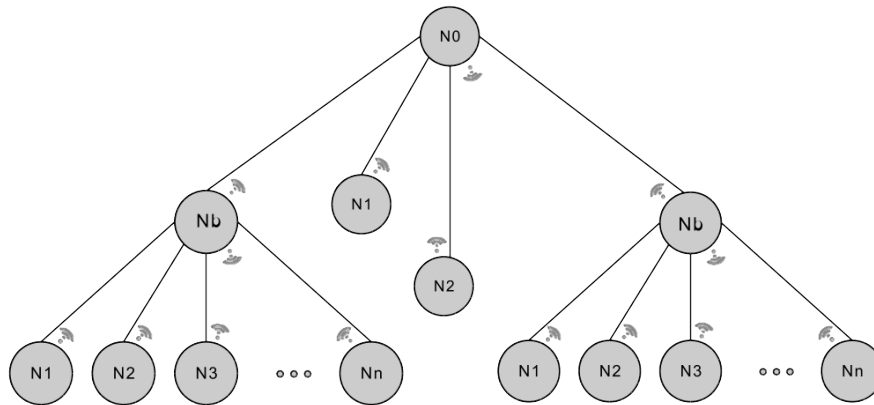
трибутивната мрежа, тогаш може да јазелот N_0 да го гледаме како трафостаница и јазлите од N_1 до N_n како објекти. Треба да се забележи дека во овој труд се смета факторите на моќност на потрошувачите се еднакви.

Предлагаме да се додадат интелигентни ротациони (гребенасти) склопки во секој јазел (N_0 до N_n) и исто така микроконтролер кој ќе ги управува склопките. Покрај тоа, контролорите ќе бидат меѓусебно поврзани со користење на некои од широко користени комуникациски технологии, како што се WiFi, Bluetooth или модулирани сигнали преку жиците на електричната мрежа.

Секој контролер континуирано (на предефиниран временски интервал) ќе ги мери напонот и струјата (со цел да се пресмета моќност). Тогаш сите јазли ќе се синхронизираат меѓу себе и ќе ги постават склопките во оптимални позиции, што ќе резултираат во максимална рамнотежа на товарот во коренот на стеблото (јазелот N_0 во нашиот случај).

Во пооптимално сценарио (за брз почетна имплементација и потврда на концептот), може всушност да се отстрани јазелот корен N_0 и да формира мрежа од јазлите N_1 до N_n и самите да се синхронизираат меѓу себе. Во двата случаи, треба да се обрне внимание на големината на мрежата, со што се обезбедува доволно пропусен опсег на каналот за комуникација и процесирачка моќ на контролерите во јазлите.

Насочувајќи се кон скалабилна и корпоративна архитектура, може да се примени парадигма за апстракција на ниво и да се користи поедноставената мрежа од Слика 4.1 како градбен блок на покомплексни дистрибутивни мрежи. Следејќи го концептот на мрежата од мрежи, резултатот ќе биде мрежата на истиот тип како на Слика 4.1, освен што јазли (N_1 до N_N), исто така, ќе бидат мрежи. Процесот



Слика 4.2 Генерализирана архитектура на дистрибутивна мрежа, проширена со DILB модули. Јазелот корен претставува електрична централа или трафостаница, додека јазлите листови ги означуваат електричните броила со вградени ротациони склопки.

може рекурзивно да се прошири, што ќе резултира со голема мрежа со генеричка топологија прикажано на Слика 4.2.

Треба да се забележи дека граничните јазли (N_b), кои имаат два комуникациски интерфејси, го контролираат нивото на апстракција, па јазелот N_0 на Слика 4.2 нема да биде свесен за јазлите N_1 до N_n , во левото и десното потстебло. Исто така, треба да се забележи дека јазли со исти идентификатори може да постојат во различни приватни мрежи, што е адресна шема слична на Интернет адресирањето во компјутерските мрежи и концептот на преведување на мрежните адреси (NAT). Јазлите означени како N_b , граничните јазли, ќе ги добијат своите идентификатори динамички од јазелот родител (N_0 во нашиот случај). Концептот е скалабилен, па N_0 исто така може да биде граничен јазел во друга поголема дистрибутивна мрежа и така натаму.

4.1.2 Преглед на алгоритмот за оптимизација

Секој јазел ќе ја пресмета моќноста на секоја фаза, па дефинираме P_j , вектор на моќностите по фаза во јазелот N_j , како

$$P_j = \begin{bmatrix} P_{rj} & P_{sj} & P_{tj} \end{bmatrix}^T, \quad (4.1)$$

каде T означува транспонирано. Ги комбинираме сите вектори во матрица:

$$P_{rst} = \begin{bmatrix} P_{r1} & P_{r2} & \dots & P_{rn} \\ P_{s1} & P_{s2} & \dots & P_{sn} \\ P_{t1} & P_{t2} & \dots & P_{tn} \end{bmatrix}. \quad (4.2)$$

Главната цел во остварувањето на оптимална балансираност е да се минимизира разликата помеѓу товарот по фаза, d , т.е. да се минимизира изразот:

$$d = 100 * \frac{|P_r - P_s| + |P_s - P_t| + |P_t - P_r|}{2 * (P_r + P_s + P_t)}, \quad (4.3)$$

каде

$$P_r = \sum_{j=1}^n P_{rj} \quad (4.4)$$

означува сума на редицата на првата редица на матрицата P_{rst} , што е вкупниот товар на фаза r измерено во јазелот N_0 . Аналогно, P_s и P_t го претставуваат вкупниот товар на фазите s и t соодветно и се дефинирани како

$$P_s = \sum_{j=1}^n P_{sj} \text{ and } P_t = \sum_{j=1}^n P_{tj}. \quad (4.5)$$

Изразот во (4.3) содржи 2 во именителот со цел да се задржи во рангот на вредности помеѓу 0 (оптимален случај) и 100 % (граничен случај со целиот товар е на една фаза и нема товари на другите две фази).

За да се конфигурира состојбата на деца јазлите и да се постигне минимално d (d_{min}) за дадени вредности на P_{rst} би значело да се постават ротационите прекинувачи во состојба која ќе резултира со оптимално балансирање на товарот по фаза, измерено во N_0 . Бидејќи насока на три-фазниот систем не смее да се менува (со цел да се избегне обратна ротација во трофазните мотори), единствените можни потези во системот може да бидат ротирање во насока на стрелките на часовникот или обратно, без промена на редоследот на фазите (на тој начин $RST \rightarrow TRS$ е во ред, но $RST \rightarrow SRT$ не е дозволено). Во однос на матрицата P_{rst} , може да се ротира секоја колона со чекор 1 (-2) или 2 (-1) или може да се остави во моменталната позиција.

Ќе дефинираме позитивна ротација како

$$\begin{bmatrix} P_{rj} & P_{sj} & P_{tj} \end{bmatrix}^T \rightarrow \begin{bmatrix} P_{tj} & P_{rj} & P_{sj} \end{bmatrix}^T \quad (4.6)$$

и негативна ротација како

$$\begin{bmatrix} P_{rj} & P_{sj} & P_{tj} \end{bmatrix}^T \rightarrow \begin{bmatrix} P_{sj} & P_{tj} & P_{rj} \end{bmatrix}^T. \quad (4.7)$$

Ако користиме $+1$, -1 или 0 да ги опишеме премините на ротационата склопка, тогаш решението на проблемот на минимизација ќе биде вектор

$$S = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & -1 & \dots & 1 \end{bmatrix}^T \quad (4.8)$$

таков што вредноста s_j ќе го опишува преминот на ротационата склопка j .

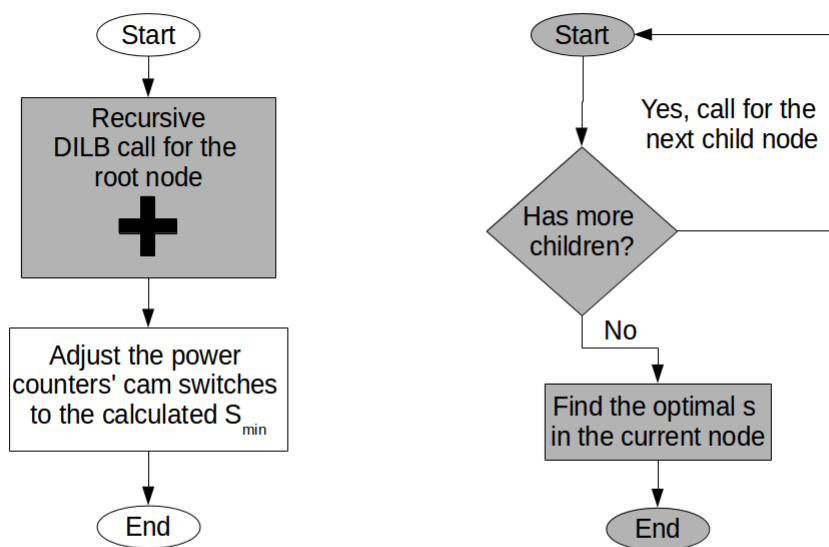
Јасно е дека било какво непотребно движење на вртливиот преклопник треба да се избегнува (нулите S се добредојдени) и, исто така, се препорачува високо оптоварените прекинувачи да се ротираат помалку од лесно оптоварените. Минимизацијата со сигурност ќе биде избор помеѓу бројот на потези и разлика во оптовареноста на фазите. Минималната разлика во оптоварувањето на фазите може да зависи од одредени дистрибутивни мрежи и кориснички/организациски цели и приоритети, па ќе биде додадена како аргумент на функцијата на минимизација (d_{min}) и ќе се пресмета минимален вектор S_{min} што резултира во $d \leq d_{min}$, оптимизирајќи го бројот на потези или се пресмета вектор S_{min} што резултира со што е можно помало d (кога $d > d_{min}$), без ограничувања во однос на бројот на потези.

Дијаграмот на алгоритмот е прикажан на Слика 4.3 и направена е едноставна илустрација, со помош на мрежа со 7 јазли. Иницијалната матрица на мерењата на јазлите е

$$P_{rst} = \begin{bmatrix} 51.934 & 42.959 & 38.607 & 2.275 & 13.698 & 43.092 & 77.282 \\ 76.908 & 40.716 & 40.860 & 43.187 & 13.692 & 47.691 & 79.660 \\ 65.656 & 38.539 & 40.932 & 9.120 & 9.156 & 45.620 & 80.010 \end{bmatrix}.$$

P_{rst} е влезна матрица на алгоритмот, а исто така d_{min} се добива како влезен параметар, така што функцијата е $DILB(P_{rst}, d_{min})$. Трансформираната матрица е

$$P_{min} = \begin{bmatrix} 65.656 & 42.959 & 40.860 & 2.275 & 13.698 & 45.620 & 77.282 \\ 51.934 & 40.716 & 40.932 & 43.187 & 13.692 & 43.092 & 79.660 \\ 76.908 & 38.539 & 38.607 & 9.120 & 9.156 & 47.691 & 80.010 \end{bmatrix},$$



a) DILB algorithm for a PDN

b) Recursive DILB algorithm for a node

Слика 4.3 Дијаграм на DILB алгоритмот: а) алгоритам за дистрибутивна мрежа што се активира во јазелот-корен, б) Рекурзивен DILB алгоритам за еден јазел.

додека векторот на состојби на ротационите склопки е

$$S_{min} = [-1 \ 0 \ 1 \ 0 \ 0 \ -1 \ 0]^T.$$

Нашата имплементација на алгоритмот е рекурзивна. На почетокот колоните на матрицата се подредени во растечки редослед според сумата на колоната, а потоа и минимизирањето започнува од ниско оптоварените јазли (левата колона) со цел да се избегне движење на високо оптоварените прекинувачи. Пресметувајќи со пристап по длабочина (од англ. „debt-first“), постапката завршува кога ќе се најде $d \leq d_{min}$.

4.1.3 Мерења и резултати од симулацијата

Алгоритмот беше симулиран со користење датотеки со податоци од 9 електрични броила, кои содржат записи за напон и струја мерени на интервали од 15 минути. Секоја датотека за евиденција содржи мерења за еден месец (август 2013 година), со почеток во 01.08 00:00 до 31.08 23:45. Датотеки се случајно избрани од достапните податоци¹ на индустриски индустриски потрошувачи во регионот. Со цел да се илустрира дека е можно да се подобри балансираноста и да се опише јадрото на алгоритмот, со користење на овие електрични броила ние вештачки формираме поедноставен мрежа со топологија прикажано на Слика 4.1 и $n = 9$. Јазелот N_0 во овој случај ќе биде локална трафостаница, која ги снабдува избраните индустриски потрошувачи, претставени со електричните броила. Во пракса, постојат многу други потрошувачи (електрични броила) поврзани на трафостаницата, но за да се поедностави анализата разгледуваме само неколку од нив.

На почетокот, се пресметува вкупното количество на енергија по фаза

$$\begin{bmatrix} E_r & E_s & E_t \\ 34.626 & 33.471 & 33.141 \end{bmatrix} MWh \quad (4.9)$$

и нејзиниот дисбаланс $d_{E_total} = 1.4668 \%$.

Ако се пресметаат вредностите на (4.9) за секој јазел одделно и се спојат во една матрица, слична на P_{rst} , би можело да се провери дали има подобро позиционирање (со почетен $s \neq [0 \ 0 \ \dots \ 0]$) на ротирачки склопки, што ќе резултира со помал дисбаланс. Користење на тие информации може да се избере подобра

¹Податоци од електрични броила,
<https://dl.dropboxusercontent.com/u/3705908/DataSet.zip>

почетна состојба s на статички начин, односно може да се оствари подобрување на карактеристиките на системот, без било какво динамичко движење на ротациони склопки. Подобрување на балансираноста е очигледна и не постои нарушено однесување во било кој дел на системот во однос на првичната конфигурација. Значи, ако ротациони склопки на избраните 9 електрични броила биле позиционирани само во согласност со

$$s_{static} = \begin{bmatrix} 0 & -1 & 1 & 0 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}^T \quad (4.10)$$

наместо првичната состојба $s = [0 \ 0 \ \dots \ 0]$, би имале

$$\begin{bmatrix} E_{r_static} & E_{s_static} & E_{t_static} \end{bmatrix} = \begin{bmatrix} 33.671 & 33.846 & 33.721 \end{bmatrix} MWh \quad (4.11)$$

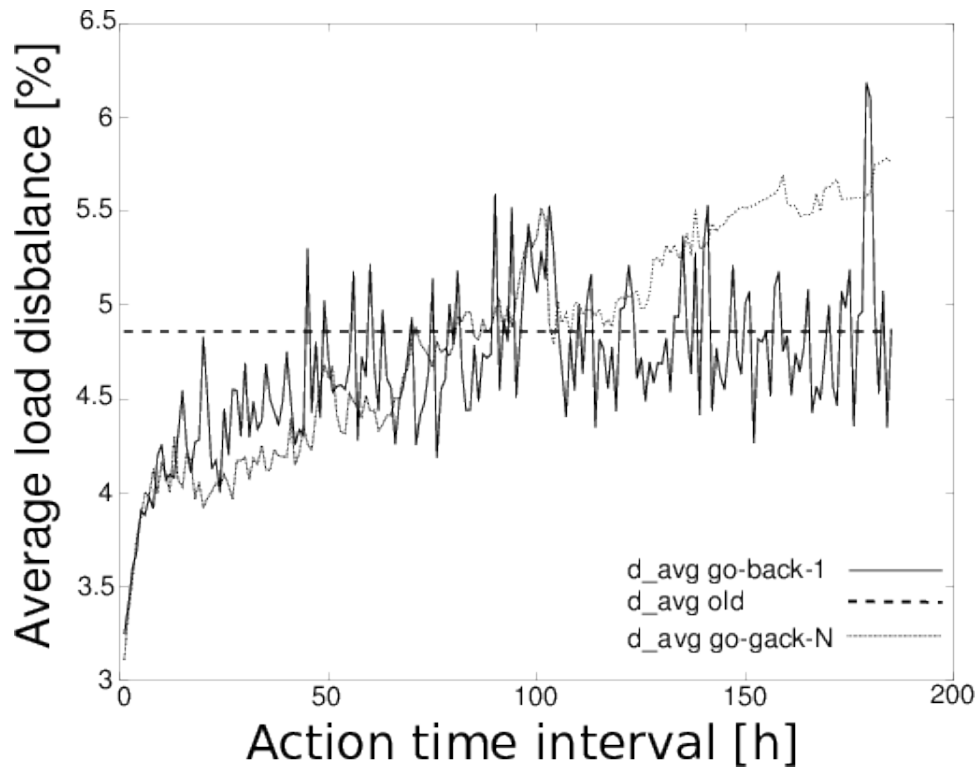
што резултира со $d_{E_total_static} = 0.1729 \%$ и претставува значително подобрување, споредено со (4.9). Компанијата сопственик на дистрибутивната мрежа може да ги анализира мерењата на годишно или месечно ниво и потоа да ги позиционира ротационите склопки во согласност со оптималното s_{static} , добиено од нашиот алгоритам.

Дозволувајќи континуирани динамички движења на ротациони склопки, разгледаваме два брзи и едноставни концепти за пресметка на следната состојба на склопките, s . Првиот (случај 1) е да се зачува само последното мерење (од последните 15 минути) и да се користи за да се пресмета на следниот s . Вториот концепт (случај 2) е да се сумираат N претходни мерења (од последниот момент на реконфигурација) и да се користи тој резултат за да се пресмета на следниот s . Просечните вредности на дисбаланс на товарот добиени со користење на овие два концепти се прикажани на Слика 4.4, каде што $d_{avg_go_back_1}$ е за првиот концепт и $d_{avg_go_back_N}$ за вториот.

Просечниот дисбаланс за секој од вкупно 2976 ($4 * 24 * 31$) записи е

$$d_{old_avg} = \frac{\sum_{i=1}^{2976} d_i}{2976} = 4.9 \% \quad (4.12)$$

каде што „old“ се однесува на почетната конфигурација на системот, без било какви корекции. Оптимизација на балансираност во оптоварувањето според сценариото во случај 1 резултираше со $d_{avg_case1} = 3.1 \%$, а сценариото во случај 2 резултираше со $d_{avg_case2} = 3.3 \%$. Сценариото во случај 1 е побрзо и троши помалку меморија,



Слика 4.4 Средниот дисбаланс на товарот како функција од временскиот интервал на промена на ротационите склопки. Нивото на дисбаланс е пропорционално со нивото на активните загуби на моќност.

но резултатот е малку непредвидлив и полош од резултатот во сценариото во случај 2, кој е побавен и бара повеќе меморија.

Секое движење на ротациони склопки во текот на секојдневното работење на дистрибутивната мрежа може да се смета како негативен фактор поради краткиот прекин на напонот за време на ротација. Значи, не е мудро да се ротираат склопките на една минута или неколку минути. Но, ако ротацијата се појавува еднаш на час или еднаш на ден, имплементацијата на нашиот алгоритам нема да го наруши концептот на стабилна дистрибутивна мрежа, а исто така и дистрибутивната мрежа проширена со DILB ќе биде избалансирана во однос на товари по фаза. Со цел да се одреди оптималниот временски интервал за ротација на склопките ние го симулиравме просечниот дисбаланс како функција од интервалот на ротација и добивме график прикажан на Слика 4.4. Графикот на $d_{avg_go_back_N}$ покажува значително подобрување на балансираноста дури и за интервали подолги од 48 часа (2 дена) со што се потврдува хипотезата за можно подобрување без чести ротации на склопките. Важно е да се напомене дека употреба на покомплексни алгоритми за конфигурација на склопките и предвидување на товарот (со машинско учење

или други методи) се очекува да резултира со попрецизна конфигурација на ротационите склопки и поефективен механизам на балансирање како целина.

4.1.4 Паметна дистрибутивна мрежа со DILB

Оптимален начин за интегрирање на концептот DILB во постоечка дистрибутивна мрежа евозможен со додатоци за електричните броила, кои ќе бидат во улога на ротациони склопки. функционалноста на мерење на електрична енергија од електричното броило ќе биде едноставно наследена. Всушност предлагаме дополнување на постоечките дистрибутивни мрежи со DILB електрични броила (електрични броила со DILB додаток) кои ќе бидат меѓусебно поврзани преку каблите на електричната мрежа со користење на некои од техниките за модулација на сигналот. Сите потрошувачи позади електричното броило (куќа, стан и сл.) ќе бидат еквивалентирани трофазен потрошувач, што може да се ротира со цел да се добие дистрибутивна мрежа балансирана по фази. Покомплициран сценарио би било да се инсталираат ротациони склопки во секој штекер и/или уред, што ќе доведе до подлабока грануларност и подобри резултати на балансирањето.

Можно е да се примени едноставна аналогија на протоколот за динамичка конфигурација на уредите (од англ. „Dynamic Host Configuration Protocol“ - DHCP), добро познат во компјутерските мрежи. Во паметната DILB дистрибутивна мрежа ќе има еден DILB сервер (аналогно на DHCP сервер), но наместо динамичко доделување на адреси, неговата работа ќе биде да му ја додели на вредноста s (-1 , 0 или 1) на секое електрично броило.

Проширување на концептот на балансирање од еден јазел (Слика 4.1) на целото дрво (Слика 4.2), односно дистрибутивна мрежа, се очекува да резултира со постабилна и посигурна мрежа со помали загуби на моќност. Загубата на моќност во една трифазна линија се движи помеѓу $3 * R * I^2$ (за идеално избалансирани товари, со 0 струја во неутралниот проводник) и $18 * R * I^2$ (за целосно неуррамнотежени товари, односно $9 * R * I$ во една од фазите и истите $9 * R * I$ во неутралниот проводник), па подобрување на балансот на товар во секој поединечен јазел резултира со намалување на загубата во линијата која ги поврзува тој јазел со остатокот од мрежата [112]. Разгледувајќи ја дистрибутивната мрежа како множество од линии, интуитивно е дека добро балансирана мрежа ќе има намалени загуби на енергија, споредено со слабо балансирана мрежа.

Функционалната имплементација на алгоритмот, што се извршува во внатрешноста на DILB серверот, мора да биде свесна за топологијата на дистрибутивната мрежа, со цел да генерира оптимален вектор S_{min} . Може да се фокусира на постиг-

нување на минимум (1) балансираност на товарот во корен-јазелот или некој друг јазел (или приоритетна листа на јазли) или (2) вкупната загуба на енергија во мрежата. Сепак, двата случаи резултираат со комплексни (од англ. „NP-hard“) проблеми [104] и дополнително во (2) детална структура на мрежата (линија геометрија, итн.) треба да бидат дефинирани во DILB софтверот, што предизвикува многу сложена инсталација и конфигурација. Во овој момент, важно е да се напомене дека споменатото d_{min} важи само за еден јазел, додека вкупната балансираност на целата дистрибутивна мрежа D_{PDN} е покомплексен израз (различен од d_{min}), што содржи детали за геометријата на линиите.

Со цел да се дизајнира генеричко и лесно решение, ние предлагаме хеuristicчки и рекурзивен пристап. Процесот на оптимизација започнува во корен-јазелот (N_0 на Слика 4.2) и рекурзивно со пристап по длабочина ги балансира јазлите листови, а потоа се враќа назад, до корен-јазелот и ги оркестрира деца-јазлите, земајќи го в предвид балансот во тековниот јазел. Нашето решение предизвикува значително подобрување на балансираноста на товарот и минимизирање на загуба на енергија во слабо балансираните дистрибутивни мрежи, но не гарантира (1) идеално избалансирани фази мерено во коренот на корен-јазелот (или други јазли) или (2) минимална загуба на моќност (што е NP-тежок проблем [104]).

Со цел да се верифицира влијанието на алгоритмот, дефинираме хеuristicчко D_{PDN} како

$$D_{PDN} = \frac{d_{N1} + d_{N2} + \dots + d_{Nn}}{n}, \quad (4.13)$$

каде N_0, N_1, \dots, Nn се сите јазли во дистрибутивната мрежа и d_{Nx} го означува вкупниот дисбаланс (дефиниран во (4.3)) измерен во јазелот Nx .

Предложеното решение беше тестирано на реалните модели на дистрибутивни мрежи IEEE 34-Bus Mod1, IEEE 34-Bus Mod2, 37-Bus и 123-Bus. Резултатите се прикажани во Табела 4.1. Измерените вредности на имплементацијата на IEEE 34-Bus Mod1 [113] се прикажани во продолжение. Иницијалните загуби во мрежата се

$$\begin{aligned} DP_{34-Bus_Mod1} &= 266.875 \text{ kW}, \\ DQ_{34-Bus_Mod1} &= -26.274 \text{ kvar} \end{aligned} \quad (4.14)$$

со вкупен дисбаланс $D_{PDN_{34-Bus_Mod1}} = 21.41 \%$ и вкупен товар измерен во коренот

$$\begin{aligned} \begin{bmatrix} P_r & P_s & P_t \end{bmatrix} &= \\ \begin{bmatrix} 1274 & 1252 & 1247 \end{bmatrix} &\text{ kW}. \end{aligned} \quad (4.15)$$

Табела 4.1 Минимизација на активните и реактивните загуби на моќност.

Модел	DP [kW]	DQ [kvar]	DP_{DILB} [kW]	DQ_{DILB} [kvar]	Минимизација $\frac{DP-DP_{DILB}}{DP}$ [%]
34-Bus Mod1	266.875	-26.274	262.009	-31.455	1.82
34-Bus Mod2	258.984	25.880	229.575	-1.689	11.35
37-Bus	151.959	385.707	145.955	375.288	3.95
123-Bus	91.722	190.373	87.075	177.322	5.07

Табела 4.2 Вкупен товар по фази измерен во корен-јазелот.

Модел	Иницијално [kW]				По DILB [kW]			
	P_r	P_s	P_t	D_{PDN} [%]	P_r	P_s	P_t	D_{PDN} [%]
34-Bus Mod1	1274	1252	1247	21.41	1263	1255	1255	16.92
34-Bus Mod2	1274	1252	1247	17.64	1167	1355	1251	16.89
37-Bus	671	583	825	18.77	907	508	664	15.05
123-Bus	1630	1125	1365	13.14	1395	1490	1235	11.34

Иако е очигледно дека товарот е и првично добро избалансиран помеѓу фазите, DILB алгоритам предизвикува одредено намалување на загубата од 5 kW или 1,82 %. Секоја година, ова мало намалување на загубата во мрежата на IEEE 34-Bus MOD1 ќе резултира со заштеда на 42.626 kWh. Активните и реактивните загуби во мрежата по примената на DILB се

$$\begin{aligned} DP_{34-Bus_Mod1_DILB} &= 262.009 \text{ kW}, \\ DQ_{34-Bus_Mod1_DILB} &= -31.455 \text{ kvar}, \end{aligned} \quad (4.16)$$

со вкупен дисбаланс $D_{PDN_{34-Bus_Mod1_DILB}} = 16.92 \%$, што е 4.49 % better, compared to the initial disbalance.

Во некои од моделите на мрежите резултантниот дисбаланс измерен во корен-јазелот (d_{root}) е подобрена (34-Bus MOD1, 123-Bus), но во други модели таа е полоша од почетната (34-Bus Mod2, 37-Bus), како што е прикажано во Табела 4.2. Но, во секој случај, вредноста на D_{PDN} и намалувањето на загубата на активна моќност е главната цел на DILB.

Во случајот на моделот на 34-Bus Mod2 мрежата, минимизацијата на активната загуба достигнува 11,35 %, што претставува значително подобрување, што ќе резултира со заштеда на 257.623 kWh годишно.

4.1.5 Заклучок

Беше презентирани нов пристап во решавањето на проблемот на балансирање на товар и минимизација на активните загуби. Илустрацијата на концептот беше воведена со користење на едноставна мрежа со еден родител-јазел и неговите деца, а потоа генерализирана за цела дистрибутивна мрежа. Беа прикажани можностите за проширување на архитектурата со DILB и резултатите од симулацијата со вистински податоци од локални електрични броила и моделите IEEE 34-Bus (MOD1 и Mod2), 37-Bus и 123-Bus, со што беше потврдено позитивното влијание на нашето решение за вкупниот баланс на товарот и минимизирање на загубите на моќност во дистрибутивната мрежа.

4.2 Алгоритам за дистрибуирана компензација

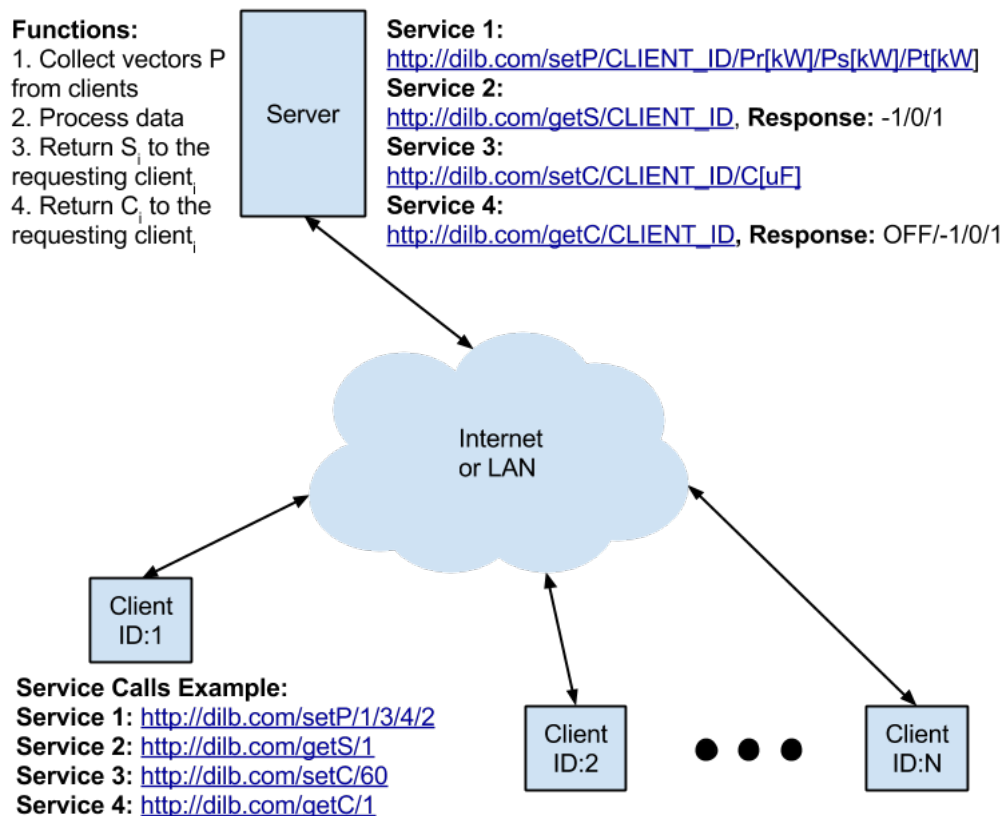
Подобрувањето на стабилноста на дистрибутивната мрежа и минимизација на загубите се многу популарни истражувачки полиња поради големиот број на корисници и како резултат на високите профити. Покрај другите фактори, балансирањето на товарот помеѓу фазите и компензација на реактивна моќност се секогаш актуелни теми, бидејќи тие директно влијаат врз активните загуби на моќност и се важни за стабилноста на електричната мрежа [114].

Во минатото, неколку архитектури на кола и системи за промена на фазата биле дизајнирани за да се реши проблемот со балансирање на товар. Авторите на [115] развиле нов метод на ожичување и полиса за прилагодување, таква што монофазните потрошувачи се поврзани на трите фази. Кога товарот е небалансиран, тогаш контролерот врши промена на фазите на потрошувачите. Во [115] се користи геометриски пристап за решавање на проблемот на балансирање. Се мерат и споредуваат радиусите на круговите, формирани од релациите на текот на моќноста во гранките и се одредува максимална јамка за балансирање на товарот графички. Авторите во [116] експериментално валидираат модел на балансирање за активен и реактивен товар трифазна мрежа, врз основа на асиметрична трансверзална реактивна компензација. Го димензионираат системот со ABC (од англ. „Adaptive Balancing Compensator“) или DLB (од англ. „Dynamic Load Balancer“) и компензаторот е составен од две три-фазни кола, еден во делта конекција, а другиот во у-врска (од англ. „wye-connection“) со неутралниот проводник. Неодамна една нова метода за балансирање беше предложена и беше анализирана нејзиното влијание врз неколку модели на мрежи [117].

Истражувањето на слични теми резултираше со неколку патенти. Измислен е трофазен трансформатор [118] со повеќе секундарни намотки, кои ги балансираат примарните струи во случај на небалансиран товар. Исто така, дизајниран е уред за автоматско маневрирање на балансирањето на товарот (ALBM) [119] поврзан во серија со ниско-напонската страна на дистрибутивен трансформатор во трафостаница. Авторите на [120] моделираат начин на балансирање, кој се состои од откривање на потребата за оптоварување прекинувач од влезната фаза кон излезната, со примена на конверзија на еднонасочна струја на излезната фаза, со цел да се добие еднонасочна репрезентација на излезната фаза. Авторите во [121] имаат развиено мерен уред кој им овозможува на монофазните потрошувачи да бидат селективно поврзани со некоја од комбинациите А-В, А-С или В-С на вертикалата од три фази на собирницата само со движење на конектори конекторот на основната линија. Авторот на [122] измислил уред за балансирање на монофазни потрошувачи поврзани на повеќе-фазна електрична мрежа. Тој содржи конвертор, кој селективно модифицира фазното поместување помеѓу неговите излезни напони и фазите на повеќе-фазната електрична мрежа.

Исто така постои и широко множество на решенија, кои се однесуваат на проблеми на реактивна моќност. Централизиран систем за компензација на реактивна енергија за дистрибутивни мрежи со низок напон е предложен во [123], што резултира со значително намалување на загубите во оптималниот капацитет на централизиран систем за компензација на реактивна енергија. Авторите во [124] дизајнираат дистрибуирана контролна стратегија за компензација на реактивна енергија во паметни микро-мрежи и ги намалуваат загубите при дистрибуција. Во насока на микро-генераторите, како паметни агенти кои може да го измерат својот фазен напон и да ги поделат овие податоци со други агенти да се прилагоди нивото на реактивна енергија внесена во мрежата, подобрување на регулацијата на напон и минимизирање на загубите се постигнуваат во [125]. Три-фазен конвертер со четири-прекинувачи е опишан во [126] и се фокусира кон насока на компензација на реактивната моќност во општ случај на неизбалансиран напонски извор.

Користејќи концепт на хиерархиско, клиент-сервер адресирање, предлагаме алгоритам (метод), кој ги интегрира балансирањето на товар и компензацијата на реактивна енергија. Предложена е хардверска имплементација на прекинувачот и измерено и анализирано е нејзиното однесување со фокус на краткиот временски интервал за промена на фазата и влијанието врз приклучените потрошувачи. Целосната архитектура на систем за управување со податоците и контрола е прикажана во детали.

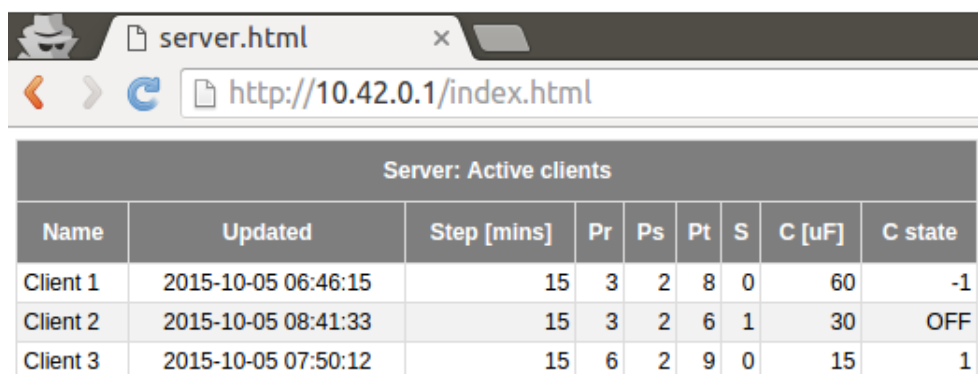


Слика 4.5 Систем Архитектура и комуникациски детали. Серверот е поставен на Интернет или во локална мрежа. Клиентите се поврзани со (или вградени во) електричните броила. Секој клиент ги користи услуги на серверот 1/2/3, испраќа свои мерења и добива оптимална ротациона состојба и состојба на дистрибуираниот кондензатор.

Следува прегледот на поглавјето. Дел 4.2.1 ја прикажува општата архитектура на системот, што е проследено со детални хардверски шеми од имплементацијата на прототипот прототип во делот 4.2.4. Однесувањето на системот е претставено преку мерење на сигналот во делот 4.2.5 и делот 4.2.6 претставува заклучок и ги опишува придобивките од решението применето во дистрибутивните мрежи.

4.2.1 Општа архитектура

Пресметувачката моќ и комуникациските можности на дигиталните електрични броила отвораат нова перспектива на поглед кон добро познати проблеми во дистрибутивните мрежи и подобрување на постоечките решенија. Општата идеја е да се прошири електричното броило со ротациона склупка која е во состојба да ги ротира фазите, како и со релативно мали по големина кондензатори за



Server: Active clients								
Name	Updated	Step [mins]	Pr	Ps	Pt	S	C [uF]	C state
Client 1	2015-10-05 06:46:15	15	3	2	8	0	60	-1
Client 2	2015-10-05 08:41:33	15	3	2	6	1	30	OFF
Client 3	2015-10-05 07:50:12	15	6	2	9	0	15	1

Слика 4.6 Серверот прикажува листа на поврзани уреди и пресметува оптимално s . Достапен е со http/https на пр. <https://dilb.com> (10.42.0.1 во нашиот случај).

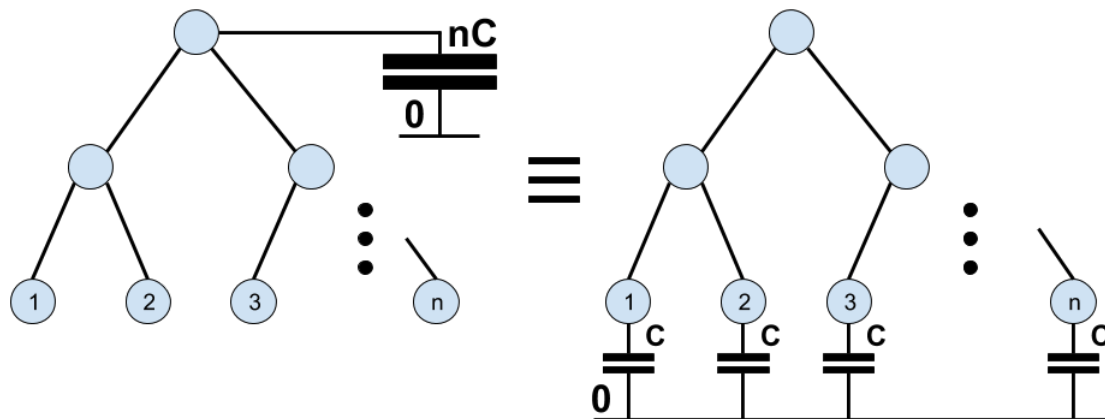
компензација и искористувајќи ги мрежните можности да се воспостави комуникација меѓу сите електрични броила, со цел размена на измерените вредности во реално време. Едноставна обработка на собраните мерења ќе резултира со оптимална конфигурација (во однос на балансирано оптоварување и намалување на реактивна моќност) на ротационите склопки на секое електрично броило во делот од мрежата.

Паметниот додаток на постоечката дистрибутивна мрежа се состои од серверот и клиентите, Слика 4.5. Серверот е веб апликација која е достапна преку Интернет или поставена на локален сервер достапен преку локалната мрежа. Клиентот е терминална станица поврзана со (или вградена во) електричното броило. Веб апликацијата овозможува регистрирање на дистрибутивна мрежа (за објект, населба и сл.) и конфигурација на нејзината топологиски параметри (важно за пресметка на загубите на моќност).

4.2.2 Балансирање на товар

Клиентот постојано испраќа мерења до серверот, добива вектор на ротација s и ги ротира фазите во соодветната состојба. На серверот постојано се пресметува оптималниот вектор s .

Комуникацијата е сервисно ориентирана, односно клиентите комуницираат со серверот преку веб сервиси поставени на одредени адреси. Двата најважни сервиси за клиентот се (1) да испрати ажурирање на мерењата до серверот и (2) да добие состојба на ротација. Двата сервиси може да се интегрираат во единечно HTTP барање, но важно е да се напомене дека интервалот на ажурирање може да се разликува од дозволениот интервал за ротација.



Слика 4.7 Дистрибуирана компензација на реактивна моќност. Вкупната капацитивност (nC) на кондензаторската батерија додадена во корен-јазелот (лево) е еквивалентна на сумата на n дистрибуирани кондензатори (C) поставени во секое електрично броило (десно).

Серверот во нашиот пример е веб апликација во Јава и неговиот главен екран прикажува листа на поврзани клиенти и нивните параметри, Слика 4.6. Оптималниот s се пресметува во позадина на одреден временски интервал. Кога терминалната станица ја бара својата вредност S , ќе ја добие веднаш, наместо да чека вредноста да биде пресметана дополнително. Интервалот на повторната пресметка треба да биде конфигуриран од страна на администраторот на дистрибутивната мрежа.

4.2.3 Компензација на реактивна моќност

Серверот постојано ќе ги чита активните и реактивните вредности на моќност од сумарно електрично броило сместено во трафостаница (или во централната точка на напојувањето на една зграда) и ќе ја пресметува состојбата на сите кондензатори во системот, Слика 4.7. Состојбата ќе определува на која фаза треба да се поврзе кондензаторот (состојба $0/1/-1$ за фази $R/S/T$ соодветно) или дали треба да се исклучи (состојба OFF).

Пресметката на оптималната состојба на кондензаторите ќе резултира со вектор на состојби c_{states} . Еден пример на таков вектор е следниот

$$c_{states} = \left[0 \quad -1 \quad 1 \quad OFF \quad -1 \quad OFF \quad -1 \right]^T. \quad (4.17)$$

Со цел да се подобри прецизноста, дистрибуираните кондензатори може да бидат со различни големини, како C , $\frac{C}{2}$, \dots , $\frac{C}{4}$ итн.

Претпоставувајќи дека најголемиот кондензатор ќе биде

$$Q_c = 1000 \text{ var} = 1 \text{ kvar} \quad (4.18)$$

и користејќи ги релациите

$$Q_c = wCU^2, \quad C = \frac{Q_c}{wU^2} = \frac{Q_c}{2\pi fU^2} \quad (4.19)$$

за дистрибутивна мрежа со следните карактеристики

$$f = 50 \text{ Hz}, \quad U = 230 \text{ V} \quad (4.20)$$

ќе се добие кондензатор со

$$C \approx 60 \mu F. \quad (4.21)$$

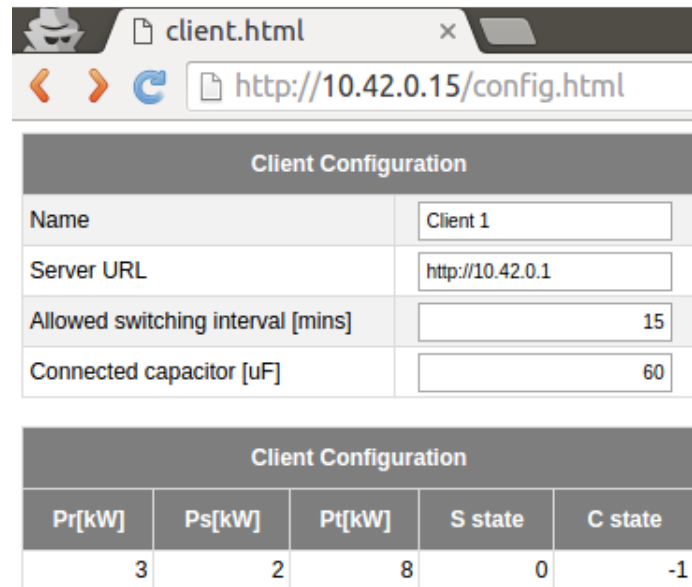
Големината на соодветниот кондензатор би била околу 100 mm висина со 50 mm во дијаметар ² и пропорционално помали димензии за $\frac{C}{2}$, $\frac{C}{4}$ итн., што се доволно мали за да бидат вградени во електричното броило. На пример, зграда со триесет станови, од кои десет имаат вградени кондензатори со големина C ($60 \mu F$), други десет имаат $\frac{C}{2}$ ($30 \mu F$) и останатите десет $\frac{C}{4}$ ($15 \mu F$), може да постигне вкупна капацитивност

$$C_{total} = 10\left(C + \frac{C}{2} + \frac{C}{4}\right) = 1050 \mu F \quad (4.22)$$

и вкупна реактивна моќност

$$Q_{c_total} \approx 17,5 \text{ kvar}. \quad (4.23)$$

Поврзаниот капацитет динамички ќе биде прилагоден според тековната индуктивна моќност во одреден временски период, така што во претходниот пример ќе варира од $15 \mu F$ ($\approx 249 \text{ var}$) до $1050 \mu F$. Димензионирањето на кондензаторите може да варира во зависност од видот на објектот и највисокото очекувано ниво на реактивна моќност.



Слика 4.8 Клиентот овозможува конфигурациска форма и го прикажува товарот по фаза во системот. Достапен е преку неговата локална адреса (на пр. 10.42.0.15). Адресата на серверот треба да се додаде во конфигурацијата на клиентот (<http://10.42.0.1/> означува <https://dilb.com> во нашиот пример).

4.2.4 Прототип клиент

Комуникацискиот и процесирачки модул терминалната станица е имплементиран со Raspberry Pi модел B³. Тој ги задоволува хардверските потреби и обезбедува доволна софтверска платформа (како што се оперативен систем, мрежа, влезно/излезни порти и сл.) за имплементација на клиентот. Оперативниот систем се вчитува од SD-картичка или дури и преку мрежа, што го прави многу лесен за одржување или надоградба. Цената на Raspberry Pi е околу \$40 по парче (тоа сигурно ќе се намали за поголема количина).

Плочата Raspberry Pi е монтирана на врвот на главната плоча, Слика 4.9. Главната плоча се состои од 6 релеи⁴ за да се покријат сите можни состојби на премин помеѓу фазите ($R \rightarrow R, R \rightarrow S, R \rightarrow T, S \rightarrow S, S \rightarrow T$ итн.), два релеи за состојбите на кондензаторот и едно реле за вклучување и исклучување на кондензаторот. Една клетка на прекинувачот за една излезна фаза е прикажана на Слика 4.11. Ротационата склопка спомената погоре беше заменета со релеи за

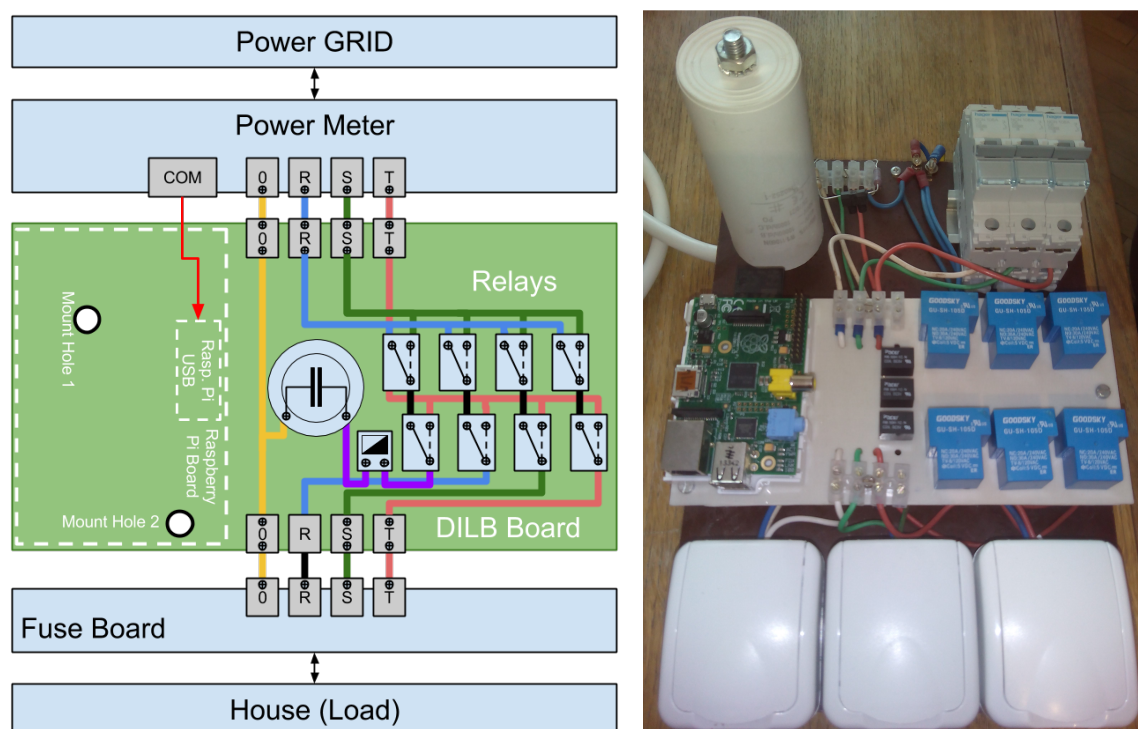
²https://dl.dropboxusercontent.com/u/3705908/w1_11060n.pdf

³Raspberry Pi модел B,

<https://www.raspberrypi.org/products/model-b/>

⁴5V DC на напотката и максимална струја од 30A,

<http://www.goodsky.co.uk/datafiles/GU.pdf>

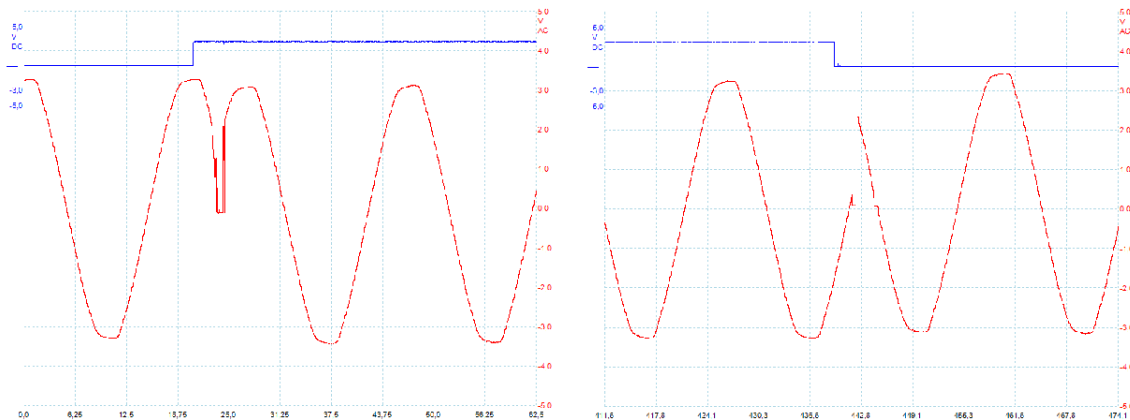


Слика 4.9 Шематски приказ на клиентската плоча, поврзана со електричното броило и панелот со осигурувачи (лево) и фотографија од имплементиранот прототип систем (десно). Кондензаторот е со капацитивност од $60 \mu F$.

да се обезбеди подобра скалабилност на системот (за применливост во мрежи со 2-фази, 3-фази, 4-фази итн.). Продукциската имплементација на клиентот може да се подобри со тиристори или специјално изработени електронски модули, со кои ќе се заменат ротационите склопки. Вкупната цена на клиентот може да се менува со промена на модулот за промена на фазата.

Во нашиот случај, Raspberry Pi користи Linux (дистрибуција RASPBIAN, јадро 3,12), кој ги обезбедува целиот потребен мрежен софтвер. Системот треба да биде поврзан со комуникациски интерфејс (COM) на електрично броило, со цел да ги добие мерењата и е поврзан на локална компјутерска мрежа за да пристапи до серверот. Пристапот до параметрите на електричното броило е најчесто ограничен, но системот е од интерес на дистрибутивната компанија и ако биде имплементиран од неа, тогаш сите потребни привилегии за читање на податоци ќе бидат достапни.

Минимално дозволеният интервал за промена може да се разликува помеѓу различни дистрибутивни мрежи, па имплементиравме едноставен интерфејс за конфигурација, во форма на веб апликација во PHP поставена на клиентскиот Raspberry Pi модул, Слика 4.8. Секој клиент треба да биде конфигуриран со име,



Слика 4.10 Напон на потрошувачот во моментот кога релето се вклучува (лево) и кога се исклучува (десно). Сигналот што предизвикува промена на релето е со сина боја, додека напонскиот сигнал е со црвена боја.

Табела 4.3 Робустност на прототипот - време на одговор.

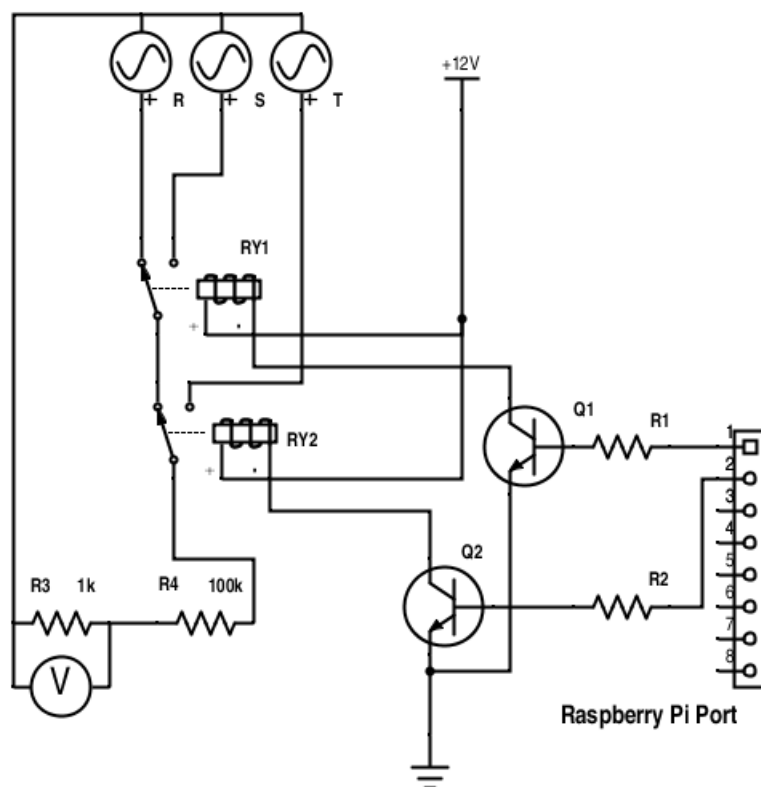
Мерење	Временско доцнење
Доцнење по промена на контролниот сигнал	$\sim 3 \text{ ms}$
Период на осцилација на сигналот по промена	$\sim 5 \text{ ms}$
Детекција на проблем	$3 \times$ (интервал на промена)
Време за ресетирање на клиентот	~ 30 секунди
Време за ресетирање на серверот	~ 1 минута

адреса на серверот, дозволен интервал за промена и вредноста на поврзаниот кондензатор. Товарот по фаза е прикажан на дното на страницата за конфигурација.

Во случај на проблем со комуникацијата со централниот сервер, на клиентскиот јазел ќе се задржи неговата последна состојба. Ако одговорот од серверот недостасува за три прекинувачки интервали (се дефинира преку корисничкиот интерфејс на клиентот), клиентот ќе се рестартира и иницијалната фазна конфигурација ќе биде вратена. Клиентот и серверот се исто така заштитени со внатрешни безбедности тајмери, кои ќе ги рестартираат ако заглават. Апликацијата се стартува веднаш по рестартирањето на системот и на јазелот се враќа во работна состојба. Повеќе детали за стабилноста и времето на одговор се прикажани во Табела 4.3.

4.2.5 Анализа на промената на фаза

Процесот на промена на фаза, сепак, не е добредојден поради краткорочниот прекин на електрична енергија за потрошувачот. Со цел да се анализира сигналот за време на преминот беа направени мерења со осцилоскоп преку напонски делител



Слика 4.11 Единечна ќелија на една излезна фаза од модулот за промена на фазите.

со два отпорници, како што е прикажано на Слика 4.11. Мерењата беа направени во случај кога релето се вклучува, Слика 4.11 (лево), и во случај кога релето се исклучува, Слика 4.11 (Десно).

Промената од една во друга фаза предизвикува деформација на кривата на напонскиот сигнал и осцилација во околу 5 ms , но сепак не постои значаен прекин, од гледна точка на потрошувачот. Јасно е дека промената нема да влијае на уреди кои содржат батерија или се вклучени во UPS (од англ. „Uninterruptible Power Supply“), но направивме серија на испитувања и набљудувања за да се провери влијанието на најчесто користените апарати во домаќинството, што се прикажани во Табела 4.4. Промената на фаза е незабележлива за тестираните уреди, односно не предизвикува никакви промени во однесувањето на уредите, не го рестартира уредот и е невидлива за корисникот на уредот.

Прекинувачот не ја менува насоката на фазите и нема да и наштети на три-фазната опрема (на пример, моторите не се врти во спротивна насока), но тоа може да предизвика промена во брзината и да влијае на прецизноста на опремата

Табела 4.4 Незабележлива промена на фаза - тестирани уреди.

Тип на уреди (тестирани уреди)	незабележлив премин за корисниот на уредот
Сијалици (жичана, неонска)	да
Уреди со мотор (фрижидер, соковник, правосмукалка)	да
Уреди за греење/ладење (бојлер, греач, ладилник)	да
Електронски уреди (ТВ, компјутер, монитор, WiFi-насочувач)	да

или производствените работи/машини, кои користат на монофазен или трифазен мотор, па решението треба да се користи придружено со филтер/стабилизатор или целосно да се избегнува во такви средини.

4.2.6 Заклучок

Беа претставени општата архитектура на проширување за дистрибутивната мрежа за динамичко интелигентно балансирање на товар и дистрибуирана компензација на реактивна моќност, деталните хардверски шеми на прекинувачот (додаток за електрично броило) и имплементацијата на конфигурациско сценарио, што ќе ги намали активните и реактивните загуби на моќност во дистрибутивната мрежа. Очекуваното однесување беше покажано со имплементација на прототип хардвер, мерења со осцилоскоп и експерименти со вистински апарати за домаќинство и електрична опрема која потврди дека интервалот на премин е занемарливо краток и дека концептот е применлив во постоечките дистрибутивни мрежи.

4.3 Онтологија за електроенергетски систем

Паметните електрични мрежи веќе ги користат придобивките на информатичката и комуникациската технологија (ИКТ) и тие веќе се применуваат во електричните системи. Сепак, големата количина на податоци кои се генерирани од паметните мрежи бара соодветно моделирање, обработка и објавување на податоците за да резултира со вистински паметна мрежа и разбирање на податоци од страна на дистрибутивните компании, како и потрошувачите на енергија. Мрежите во иднина ќе имаат тенденција да ги трансформираат на рачните операции, заедно

со електро-механички компоненти, во „паметна мрежа“ (од англ. „smart grid“) и автоматизирани процеси, што ќе бара собирање и обработка на дополнителни податоци, па сеопфатното напредно моделирање на податоците ќе биде од суштинско значење [127]. Паметната мрежа е секогаш проследена со паметно управување со енергијата, која е отворена за предизвиците на податочно-водените принципи за управување со енергијата во информациска инфраструктура, собирање и управување на податоци, интеграција и споделување на податоци, обработка и анализа, безбедност и приватност [128].

Моќта на собраните и обработени податоци е корисна во било кој дел од електричната мрежа. Во [129], податочно-водениот пристап се применува на следење на перформансите на ветерници со користење супервизорска контрола и за извлекување на податоци (од англ. „Supervisory Control and Data Acquisition“ - SCADA) и резултатите покажуваат висока точност во откривање на абнормални профили. Оптимален проток на енергија во комбинација со контрола на оптоварувањето се постигнува со користење на податочно-водениот пристап во [130], што резултира со поресметковна ефикасност на робустен пристап и баланс помеѓу цената и стабилноста на податочно-водените решенија. Моделирање на податоците во електрична мрежа е разгледувано и во [131] и воведена е централна онтологија. Таа користи заеднички слој на поврзување за сите агенти и нуди унифициран пристап до зачуваните податоци, информации и знаење. Модел за теоретското знаење за електричните мрежи се предлага во [132]. Тоа е базирано на онтологиите и семантички веб технологии и фокусирано на решавање на проблемот на нормализирано моделирано знаење за управување и анализа на податочни множества од доменот на електрични мрежи.

Од друга страна, науката за веб технологиите веќе произведе напредно множество на стандарди и алатки за моделирање на податоци, кои, исто така, може да се применат во областа на податоци за управување со електроенергетскиот систем. Schema.org [133], како иницијатива на Google и Microsoft, ги обединува постоечките различни шеми на податоци и станува стандард. Всушност, Schema.org се смета за еден од главните промотори на принципите на Семантичкиот веб ширум светот, покрај поголем број на организации и поединци во нивните бизнис потреби [134].

Иако се појави неодамна, schema.org веќе се применува во различни области. Авторите во [135] имаат развиено претставување, обработка и визуелизација на податоци од доменот на здравството со користење на семантички алатки и се увезени повеќе од 20.000 податоци за поврзани со ХИВ, за смртност и сродни величини. Семантичките алатки биле применети во подобрување на системи за

домашна здравствена заштита и систем размислување и мапирање, изграден со онтологија (OWL), едноставен протокол и RDF јазик за пребарување (од англ. „Simple Protocol and RDF Query Language“ - SPARQL) и SPARQL нотација на заклучување (од англ. „SPARQL Inferencing Notation“ - SPIN) е претставен во [136]. Пристап кој обезбедува интегрирани и информации за различни теми поврзани со туризмот е претставен во [137] и авторите воведуваат концепт на адаптација врз основа на семантички описи на контекстот на корисникот и интегрирани извори на информации и опишуваат прототип имплементација на тој концепт. Во [138] се претставени тековните напори за користење на технологиите на поврзани податоци за подобрување на видливоста на Интернет на туристичките сервиси од Инсбрук и неговата околина. Придонес во областа на транспортот применет на градот Лондон, Велика Британија се истражува во [139] и иновативна рамка за адресирање на бариерите за пристап до информации, преку имплементација на складиште од поврзани податоци за објавување, поврзување и конзумирање на отворени податоци.

Како резултат на досегашната работа во оваа област, веќе се објавени неколку податочни портали кои нудат податоци од различни области, како што се DBPedia ⁵ и SPARQL сервисите ⁶ и WikiData ⁷. DBPedia е иницијатива на заедницата за да се извлечат структурирани информации од Википедија и да станат достапни на Интернет. Со комбинирање на податоците од електричната мрежа со општите гео-политички и социо-економски податоци од ваквите веб и податочни центри, одредени индикатори за град или држава може да се пресметаат, што ќе ја поедностави оценка на квалитетот на електричната мрежа и веројатно ќе овозможи рангирање на градовите и државите според тие индикатори [140].

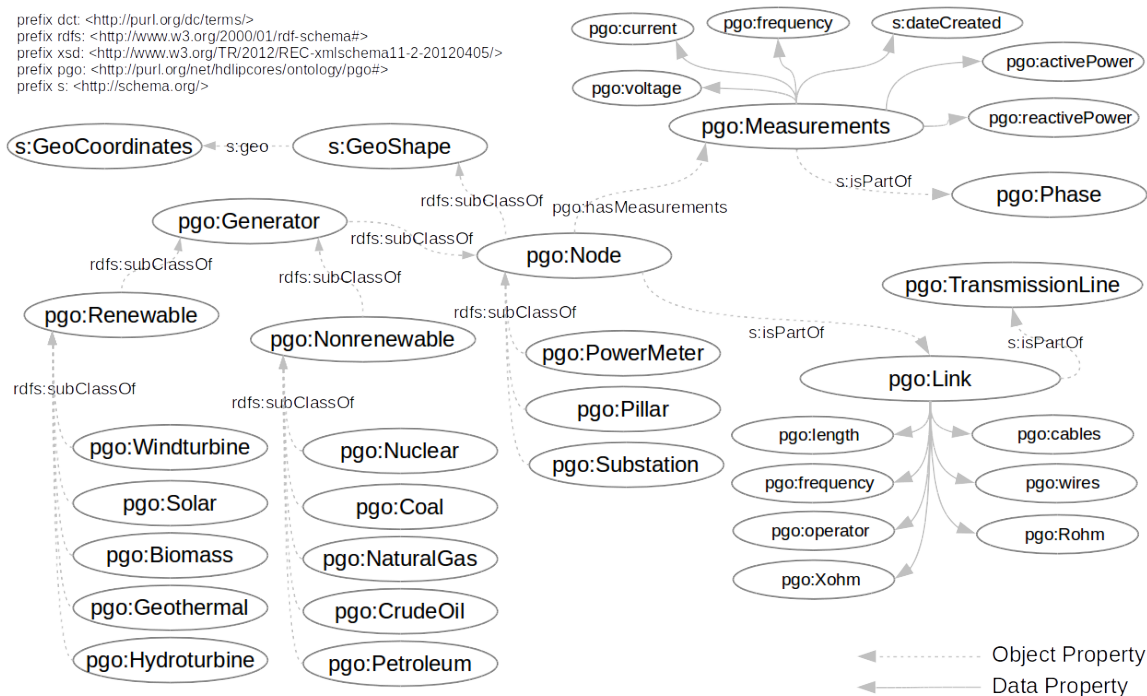
Разгледувајќи ги тековните проблеми во електроенергетските системи и поттикнати од достигнувањата на schema.org во други области, во овој труд ќе се прошири schema.org за да ја опфати областа на електроенергетските системи и ќе се развие онтологија за електрична мрежа (од англ. „Power Grid Ontology“ - PGO), т.е. шема на податоци за анотација на податоци од доменот на електроенергетски системи, врз основа на речникот schema.org.

Прегледот на поглавјето е следниот. Во Дел 4.3.1 се прикажани онтологијата на електричната мрежа и ентитетите и релациите на податоците. Придобивките од користењето, како што се извештаи и филтри, од складиштето на податоци врз

⁵DBPedia, <http://wiki.dbpedia.org/>

⁶SPARQL сервиси, <http://live.dbpedia.org/sparql> и <https://dbpedia.org/sparql>

⁷WikiData, <https://www.wikidata.org/>



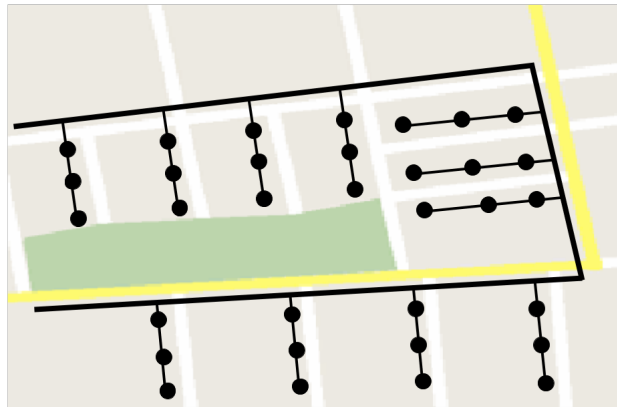
Слика 4.12 Онтологијата за електроенергетски систем (PGO) обезбедува модел на податоци од доменот на електроенергетски систем (како мрежна топологија и континуирани и периодични мерења).

основа на онтологија се претставени во Дел 4.3.2. Дел 4.3.3 претставува заклучок и ги наведува идните можности за работа.

4.3.1 Онтологија за електроенергетски систем

Онтологијата за електроенергетски систем (PGO), прикажана на Слика 4.12, воведува податочен модел за анотација на податоците на електроенергетските системи. PGO е развиена врз база на schema.org и реискористува и наследува повеќе ентитети и релации (означени со „s:“ на Слика 4.12). Сепак, за најпсецифичните потреби на доменот се додадени нови ентитети и релации (означени со „pgo:“ на Слика 4.12). Централниот ентитет е јазел (од англ. *Node*), кој претставува јазел во мрежата, како што се генератор, трафостаница, столб од преносна линија и електрично броило. Генераторите може да бидат обновливи и необновливи.

Секој јазел има своја географска локација (претставена од „s:GeoShape“ и „s:GeoCoordinates“), вклучувајќи географска должина, широчина и евентуално поштенска адреса. Во секој јазел може периодично или континуирано да се мерат вредности (како што се напон, струја, фреквенција, активна и реактивна моќност).



Слика 4.13 Мрежа од паметни јазли.

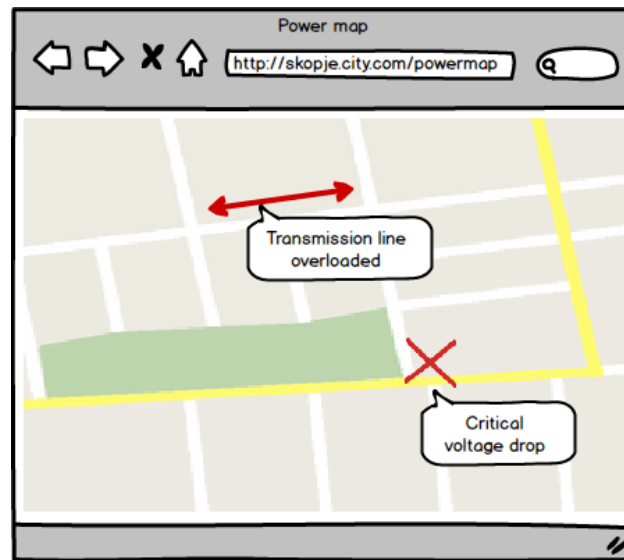
Во повеќе-фазните системи, мерењата треба да се прават по фаза (претставена од со „`rgo:Phase`“ во онтологијата).

Два јазли може да се поврзат со врска (претставена со „`pdg:Link`“) што дополнително е опишани со должина, оператор, импеданса, фреквенција, број на кабли и бројот на жици. Секоја врска може да биде дел од преносна линија, со што се обезбедува посеопфатна голема-слика на целата географска област или град, кон паметен електроенергетски систем и паметен град.

Откако ќе биде анотирано доволно големо множество на податоци со PGO, може да се користи за генерирање на периодични и континуирани извештаи, како што се наоѓање на најоптоварениот јазел или линија или да се најдат јазлите со променлива фреквенција или најчести падови на напон. Неколку други извештаи/прашања би можеле да се дефинираат на следниот начин:

- Листа на преносни линии со напон под одредено ниво;
- Листа на критично-оптоварени преносни линии;
- Листа на критично-оптоварени генератори;
- Јазли со најчесто промени на струјата;
- Просечена оптовареност на јазли на одредена улица во некој град;
- Јазли со струја под одредено ниво;

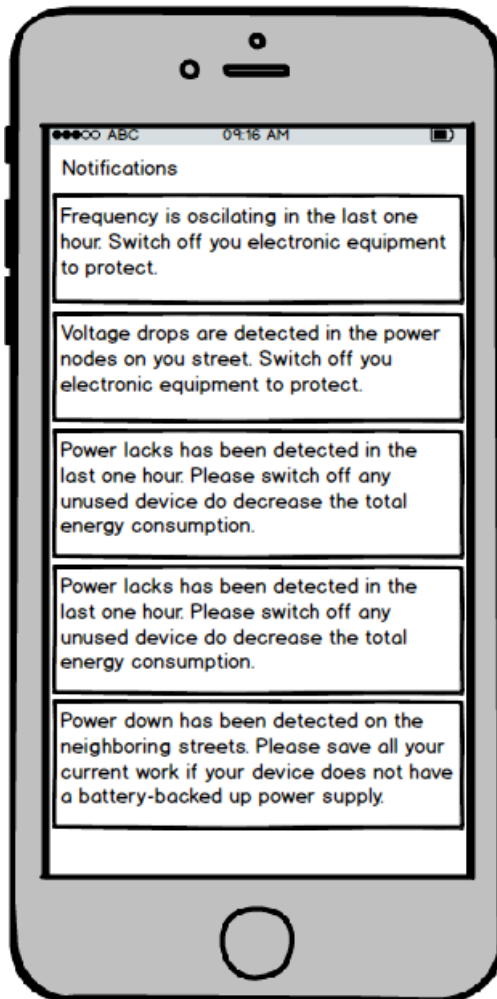
Од друга страна, може да се обезбедат корисни алатки за потрошувачите, како географска мапа на дистрибутивниот систем во градот, по аналогија на мапите во [140], претставувајќи ги сите преносни линии и јазли и можеби алармирајќи



Слика 4.14 Мапа на моќност.

ги жителите за било какви проблеми со стабилноста периодично или во реално време, како што е прикажано на Слика 4.13. Овие алатки ќе бидат корисни и за дистрибутивните компании и за граѓаните, бидејќи тие ќе се обезбедат поглед од високо ниво на електрична мрежа со сите потребни параметри за компаниите, како што е прикажано на Слика 4.14, што ќе им овозможи да донесуваат навремени одлуки. Исто така ќе има навремени информации за можни преоптоварувања за граѓаните, така што тие ќе реагираат со исклучување на некои од уредите, барем за да ги заштитат, а тоа ќе резултира како негативна повратна врска, што ќе го врати електроенергетскиот систем во стабилна состојба. Неколку можни видови на известувања за паметните телефони се претставени на Слика 4.15.

Анотацијата на електроенергетските податоци со помош на PGO ќе биде едноставен процес, бидејќи повеќето од типовите на јазли веќе содржат мерни уреди, па дури и комуникациски модули. На пример, мал дел од улицата веќе содржи десетици паметни електрични броила, како што е прикажано на Слика 4.13, кои може да ги измерат потребните вредности и да ги испратат преку локална мрежа или Интернет. Сите овие јазли имаат нивни координати и припаѓаат на преносна линија, што е концепт и структура, исто така, предвидени во нашата онтологија.



Слика 4.15 Известувања за паметен телефон.

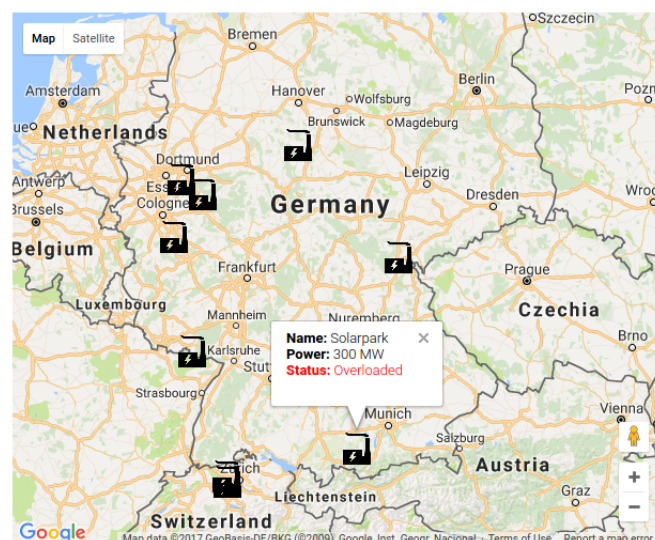
4.3.2 Реални примери

Неколку податочни множества беа објавени неодамна, како што се GridKit ⁸ и SciGRID ⁹, што ги искористивме за евалуација на нашиот концепт и самата онтологија. Базата на GridKit ги содржи јазлите и врските од мрежата на Германија како посебни датотеки. Ние ги аотиравме и ги споивме во еден RDF модел користејќи ги ентитетите и релациите на нашата онтологија. Податоци како што се името, гео-координати, напон и должина врски добиени со податочното множество, се достапни преку нашата SPARQL сервис ¹⁰.

⁸GridKit, <https://github.com/bdw/GridKit>

⁹SciGrid, <http://scigrd.de/>

¹⁰PGO SPARQL сервис, <http://hdlipcores.finki.ukim.mk/sparql>



Слика 4.16 Електрични центри во Германија.

Табела 4.5 Топ 10 оператори според должината на врските.

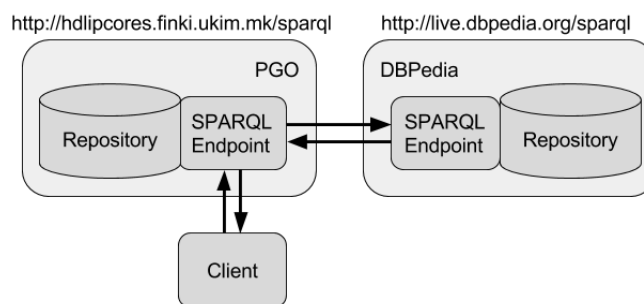
оператор	должина (m)
RWE	2.44718e+06
TenneT	2.12231e+06
TenneT TSO GmbH	1.48374e+06
50Hertz Transmission	682354.0
50Hertz	671870.0
Amprion	479220.0
50Hertz Transmission GmbH	364219.0
EnBW	318911.0
TenneT TSO GmbH;E.ON Netz GmbH	215094.0
EnBW;Amprion	202539.0

Различни видови на прашања може да се поставуваат на складиштето. Една едноставна илустрација, ќе биде да се добијат електричните центри во Германија и да се прикажат на мапа, како што е прикажано на Слика 4.16. Покрај тоа, ако податоците за електраните се достапни во реално време, тогаш соодветните податоци за состојбата може да се прикажуваат и може да се испраќаат соодветни известувања. Деталите за пребарувањето со SPARQL и резултатите се прикажани во Дел 4.3.4 во Листа 4.1 и Табела 4.10.

Топ десет операторите во Германија, подредени според вкупната должина на врските во електроенергетската мрежа, што ги поседуваат се прикажани во Табела 4.5 и SPARQL прашањето е прикажано во Листа 4.2.



Слика 4.17 Критични врски во Германија.



Слика 4.18 SPARQL сервиси. Еден дел од пребарувањето со SPARQL се извршува на PGO складиштето, а другиот дел се доставува до DBPedia. Потоа, податоците добиени од DBPedia се спојуваат со податоците од PGO и дополнително се филтрираат пред да бидат испратени на клиентот.

Посложени пребарувања може да резултираат со подлабоко статистика, како и извештаи на податоци во реално време. Еден таков пример е добивање на критичните врски, како што е прикажано на Слика 4.17. Ако кликнете на врската ќе се прикаже прозорец кој ги прикажува важните податоци за избраната врска. Во моментот, нема податоци за мерењата, па затоа податоците кои се прикажани на Слика 4.17 се само илустративен пример, што треба да го опише однесувањето. Детали за SPARQL прашањето се наведени во Листа 4.3.

Уште повеќе, со комбинирањето на податоците од електричната мрежа со други постоечки податоци од DBPedia, може да резултира со индикатори и броеви како што се должина линкови на метар квадратен или број на генератори и создадена енергија по квадратен метар или по глава на жител. На пример, вкупната должина

Табела 4.6 Градови и жители на радиус од 20 км од електрична централа.

Централа	Растојание (m)	Град	Популација
Gemeinschaftskraftwerk Kiel	3021.53	Kiel	240832
Koepchenwerk	5607.79	Hagen	191241
Koepchenwerk	11579.8	Dortmund	575944
Koepchenwerk	18027.1	Bochum	361876
Kraftwerk Scholven	10144.6	Bottrop	117450
Kraftwerk Scholven	11287.4	Gelsenkirchen	260900
Kraftwerk Scholven	15564.4	Herne, N. R.-W.	166187
Kraftwerk Scholven	15747.0	Oberhausen	214990
Kraftwerk Scholven	16571.7	Essen	589075
Kraftwerk Scholven	19545.8	Bochum	361876
Statkraft Kraftwerk Knapsack II	11083.7	Cologne	1057327
ВКУПНО			4137698

на далноводи во Германија од 10 100 *km* и вкупно население од 82 175 700 ќе резултираат со нов индикатор за должина електрични врски по глава на жител во вредност од 0,12290738 *m*.

Со комбинирање на нашето PGO складиште (преку нашиот SPARQL сервис) и складиштето на DBPedia (преку сервисот на DBPedia ¹¹, како што е прикажано на Слика 4.18) и воведување на погрануларни индикатори, би можеле да се најдат градовите сместени во радиус од 20*km* (или било кое друго растојание) од електрична централа и да се пресмета вкупниот број на население, што живее околу нив, како што е прикажано во Табела 4.6. Целосното SPARQL прашање е презентирano во Листа 4.4.

Користење на расположливите податоци за трафостаници во нашето складиште и податоците за населението на градот од складиштето на DBPedia, може да се воведат нов индикатор изразен како жители по трафостаница. Првите десет градови со популација над 100 000 во Германија, подредени според бројот на жители по трафостаница се наведени во Табела 4.7 и целосното SPARQL прашање е презентирano во Листа 4.5. Од друга страна, последните десет градови рангирани според овој индикатор се наведени во Табела 4.8.

Друг показател може да биде бројот на оператори по град. Во овој случај, податоците за операторите на јазлите од складиштето на PGO се користи и во комбинација со географските податоци (локација, население и област) од DBPedia. За да се подобри прецизноста (во однос на прашањата во претходните примери/индикатори) на овој индикатор направивме апроксимација на географскиот

¹¹DBPedia SPARQL сервис, <http://live.dbpedia.org/sparql>

Табела 4.7 Топ десет градови (со над 100 000 жители) подредени според бројот на жители по трафостаница.

Град	Популација	Трафостаница	Жители/Трафостаница
Hamburg	1774242	3	591414
Munich	1517868	7	216838
Nuremberg	498876	3	166292
Potsdam	161468	1	161468
Oldenburg	160907	1	160907
Münster	300000	2	150000
Chemnitz	243521	2	121760
Jena	105192	1	105192
Hanover	518386	5	103677
Kassel	195530	2	97765

Табела 4.8 Последните десет градови (со над 100 000 жители) подредени според бројот на жители по трафостаница.

Град	Популација	Трафостаница	Жители/Трафостаница
Bottrop	117450	28	4194
Herne, North			
Rhine-Westphalia	166187	31	5360
Mülheim	168956	29	5826
Oberhausen	214990	30	7166
Gelsenkirchen	260900	30	8696
Braunschweig	250556	22	11388
Leverkusen	161279	14	11519
Solingen	161366	13	12412
Bochum	365406	28	13050
Krefeld	235860	15	15724

Табела 4.9 Градови (со над 100 000 жители) подредени според бројот на оператори.

Град	Поп.	Пов. [m^2]	Радиус	Оп.	Поп./Оп.	Пов. m^2 /Оп.
Hamburg	1774242	755.0	15502.4	6	295707	125.833
Cologne	1057327	405.15	11356.2	5	211465	81.03
Dortmund	575944	280.4	9447.44	4	143986	70.1
Duisburg	488005	232.82	8608.66	4	122001	58.205
Bielefeld	327199	257.8	9058.72	4	81799	64.45
Leverkusen	161279	78.85	5009.87	3	53759	26.2833
Bochum	365406	145.4	6803.11	3	121802	48.4667
Salzgitter	106077	223.96	8443.27	3	35359	74.6533
Dresden	536107	328.8	10230.4	3	178702	109.6
Hamm	182022	226.26	8486.51	3	60674	75.42

облик на градот со круг, изведувајќи го радиусот според вкупната површина на градот. Резултатите се прикажани во Табела 4.9 и целосното SPARQL прашање е претставено во Листа 4.6.

4.3.3 Заклучок и идна работа

Беше претставен модел на податоци за електроенергетски систем со онтологијата (PGO), што ги опфаќа потребните ентитети и релации за анотација на топологија и мерењата во електроенергетскиот систем. Со цел да се евалуира предложениот модел, неколку прашања и извештаи на податоци беа презентирани и неколку алатки за крајниот корисник беа концептуално покажани, со цел да се забележи употребливоста за енергетските компании, како и за граѓаните.

Таквиот формален модел на податоци може да поедностави многу претходни решенија за електроенергетските системи поврзани со податоци и може значително да се подобри нашата претходна работа во областа на динамичко интелегентно балансирање на товар во дистрибутивни електрични мрежи [55] и модул за компензација на реактивна моќност вградена во електричните броила [56], со семантичка анотација на податоци, што ќе овозможи користење семантичките алатки.

4.3.4 SPARQL прашања и извештаи

The following SPARQL queries could be directly executed at our PGO SPARQL endpoint ¹² and some of them, in the background, will also use data from DP Pedia ¹³.

Следните SPARQL прашања може да бидат извршени директно на нашиот PDG SPARL сервис ¹⁴, а некои од нив во позадина ќе ги користат податоците од DBPedia ¹⁵.

```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
PREFIX pgo:
<http://purl.org/net/hdlipcores/ontology/pgo#>
PREFIX s:
<http://schema.org/>

SELECT ?name ?lat ?lng
WHERE {
    ?generator rdf:type pgo:Generator .
    ?generator s:name ?name .
    ?generator s:geo ?geoCoordinates .
    ?geoCoordinates s:latitude ?lat .
    ?geoCoordinates s:longitude ?lng
}
```

Листа 4.1 SPARQL кој ги прикажува сите електрични централи.

¹²PGO SPARQL endpoint, <http://hdlipcores.finki.ukim.mk/sparql>

¹³DB Pedia SPARQL endpoint, <http://live.dbpedia.org/sparql>

¹⁴PGO SPARQL сервис, <http://hdlipcores.finki.ukim.mk/sparql>

¹⁵DB Pedia SPARQL сервис, <http://live.dbpedia.org/sparql>

Табела 4.10 Резултати: Листа од сите електрични центри.

name име	lat/lng lat/lng
Gemeinschaftskraftwerk Kiel	10.1789348237874 54.3388198256183
Kernkraftwerk Grohnde	9.40954428358131 52.0343295468096
Koepchenwerk	7.45130176031297 51.4130023629132
Kavernenkraftwerk Säckinggen	7.95921958358891 47.5783600004331
Solarpark	10.6123957294136 48.0313556441031
Kavernenkraftwerk Wehr	7.94281008665733 47.6530214005328
Kraftwerk Scholven	7.00616826157462 51.5997745902162
Gode Wind I	6.98646123119149 54.0178365789963
Grubengas-Heizkraftwerk	7.23233534324937 49.3579040643531
Pumpspeicherkraftwerk Hohenwarte II	11.4747302318416 50.6039905453239
Statkraft Kraftwerk Knapsack II	6.84810730266128 50.8617173194521

```
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
PREFIX pgo:
<http://purl.org/net/hdlipcores/ontology/pgo#>
PREFIX s:
<http://schema.org/>

SELECT
?operator
SUM(xsd:float(?length)) AS ?total_links_length
WHERE {
    ?link rdf:type pgo:Link .
    ?link s:name ?name .
    ?link pgo:length ?length .
    ?link pgo:operator ?operator
}
GROUP BY ?operator
ORDER BY DESC(?total_links_length)
LIMIT 10
```

Листа 4.2 SPARQL кој ги прикажува првите 10 оператори според должината на врските.

```

PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
PREFIX pgo:
<http://purl.org/net/hdlipcores/ontology/pgo#>
PREFIX s:
<http://schema.org/>

SELECT DISTINCT
(?link) ?link_name ?node1_lat ?node1_lng
?node2_lat ?node2_lng ?length
WHERE {
?link rdf:type pgo:Link .
?link s:name ?link_name .
?link pgo:length ?length .
?link pgo:voltage '380000' .
?link pgo:operator 'RWE' .
?link s:hasPart ?node1 .
?link s:hasPart ?node2 .
?node1 s:geo ?gc1 .
?gc1 s:latitude ?node1_lat .
?gc1 s:longitude ?node1_lng .
?node2 s:geo ?gc2 .
?gc2 s:latitude ?node2_lat .
?gc2 s:longitude ?node2_lng .
?node1 pgo:hasMeasurements ?n1Measurements .
?n1Measurements pgo:reactivePower ?reactivePower .
FILTER
(
(
xsd:float(?activePower)
/
math:sqrt(
xsd:float(?activePower)*xsd:float(?activePower) +
xsd:float(?reactivePower)*xsd:float(?reactivePower)
)
) < 0.9
)
}

```

Листа 4.3 SPARQL кој ги прикажува критичните врски со висока реактивна моќност.


```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX pgo: <http://purl.org/net/hdlipcores/ontology/pgo#>
PREFIX s: <http://schema.org/>
#remote prefixes
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX afn: <http://jena.apache.org/ARQ/function#>
SELECT ?genName
MAX (
bif:acos(bif:sin(xsd:float(?lat) * 3.14159 / 180) *
bif:sin(xsd:float(?genLat) * 3.14159 / 180)+
bif:cos(xsd:float(?lat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLong) * 3.14159 / 180 -
xsd:float(?long) * 3.14159 / 180))*6371000
) as ?distance
xsd:string(?cityName) as ?city
MAX(?pop) as ?population
WHERE {
  SERVICE <http://live.dbpedia.org/sparql> {
    # The remote part of the query
    ?city rdf:type dbo:City;
      rdfs:label ?cityName;
      dbo:country ?country;
      dbo:country dbr:Germany;
      geo:lat ?lat;
      geo:long ?long;
      dbo:populationTotal ?pop .
      FILTER (lang(?cityName) = 'en')
    # End of The remote part of the query
  }#End of Service
  SERVICE <http://localhost:8890/sparql> {
    # The remote part of the query
    ?generator rdf:type pgo:Generator .
    ?generator s:name ?genName .
    ?generator s:geo ?geoCoordinates .
    ?geoCoordinates s:latitude ?genLat .
    ?geoCoordinates s:longitude ?genLong .
    # End of The remote part of the query
  }#End of Service
  FILTER(
(bif:acos(bif:sin(xsd:float(?lat) * 3.14159 / 180)*
bif:sin(xsd:float(?genLat) * 3.14159 / 180)+
bif:cos(xsd:float(?lat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLong) * 3.14159 / 180 -
xsd:float(?long) * 3.14159 / 180))*6371000) < 20000.0) .}
group by ?genName ?cityName
order by ?genName ?distance

```

Листа 4.4 SPARQL кој ги прикажува градовите и популацијата на радиус од 20 km од електричните центри.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX pgo: <http://purl.org/net/hdlipcodes/ontology/pgo#>
PREFIX s: <http://schema.org/>
#remote prefixes
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX afn: <http://jena.apache.org/ARQ/function#>

SELECT
xsd:string(?cityName) as ?city
MAX(?pop) as ?population
COUNT(?node) as ?substations
(MAX(?pop)/COUNT(?node)) as ?citizensPerSubstation
WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    # The remote part of the query
    ?city rdf:type dbo:City;
        rdfs:label ?cityName;
        dbo:country ?country;
        dbo:country dbr:Germany;
        geo:lat ?lat;
        geo:long ?long;
        dbo:populationTotal ?pop .
    FILTER (lang(?cityName) = 'en')
    FILTER (?pop > 100000)
    # End of The remote part of the query
  }#End of Service

  SERVICE <http://localhost:8890/sparql> {
    # The remote part of the query
    ?node rdf:type pgo:Substation .
    #?node s:name ?nodeName .
    ?node s:geo ?geoCoordinates .
    ?geoCoordinates s:latitude ?genLat .
    ?geoCoordinates s:longitude ?genLong .
    # End of The remote part of the query
  }#End of Service
  FILTER(
(bif:acos(bif:sin(xsd:float(?lat) * 3.14159 / 180)*
bif:sin(xsd:float(?genLat) * 3.14159 / 180)+
bif:cos(xsd:float(?lat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLong) * 3.14159 / 180 -
xsd:float(?long) * 3.14159 / 180))*6371000
) < 20000.0) .}
group by ?cityName
order by DESC(MAX(?pop)/COUNT(?node))
LIMIT 10

```

Листа 4.5 SPARQL кој ги прикажува топ 10 градовите (со повеќе од 100 000 жители) подредени според популација по трафостаница.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX pgo: <http://purl.org/net/hdlipcores/ontology/pgo#>
PREFIX s: <http://schema.org/>
#remote prefixes
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX afn: <http://jena.apache.org/ARQ/function#>
SELECT xsd:string(?cityName) as ?city MAX(?pop) as ?population
(MAX(?areaTotal) / 1000000) as ?area
MAX(bif:sqrt(?areaTotal / 3.14159 )) as ?radius
COUNT(distinct ?operator) as ?operators
(MAX(?pop)/COUNT(distinct ?operator)) as ?citizensPerOperator
((MAX(?areaTotal) / 1000000)/COUNT(distinct ?operator))
as ?areaPerOperator
WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    # The remote part of the query
    ?city rdf:type dbo:City;
    rdfs:label ?cityName;
    dbo:areaTotal ?areaTotal;
    dbo:country dbr:Germany;
    geo:lat ?lat;
    geo:long ?long;
    dbo:populationTotal ?pop .
    FILTER (lang(?cityName) = 'en')
    FILTER (?pop > 100000)
    # End of The remote part of the query
  }#End of Service
  SERVICE <http://localhost:8890/sparql> {
    # The remote part of the query
    {?node rdf:type pgo:Pillar .
    ?node pgo:operator ?operator .
    ?node s:geo ?geoCoordinates .
    ?geoCoordinates s:latitude ?genLat .
    ?geoCoordinates s:longitude ?genLong .
    } UNION {
    ?node rdf:type pgo:Substation .
    ?node pgo:operator ?operator .
    ?node s:geo ?geoCoordinates .
    ?geoCoordinates s:latitude ?genLat .
    ?geoCoordinates s:longitude ?genLong .
    }# End of The remote part of the query
  }#End of Service
  FILTER((bif:acos(bif:sin(xsd:float(?lat) * 3.14159 / 180)*
bif:sin(xsd:float(?genLat) * 3.14159 / 180)+
bif:cos(xsd:float(?lat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLat) * 3.14159 / 180)*
bif:cos(xsd:float(?genLong) * 3.14159 / 180 -
xsd:float(?long) * 3.14159 / 180))*6371000
) < bif:sqrt(?areaTotal / 3.14159) ) .}
group by ?cityName order by DESC(COUNT(distinct ?operator))
LIMIT 10

```

Листа 4.6 SPARQL кој ги прикажува градовите (со повеќе од 100 000 жители) подредени според бројот на оператори.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:pgo="http://purl.org/net/hdlipcodes/ontology/pgo#"
  xmlns:s="http://schema.org/"
  xmlns:datarepo="http://purl.org/net/hdlipcodes/ontology/pgo/data#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <pgo:Link rdf:about="datarepo:link-175370">
    <pgo:length>458.244020145921</pgo:length>
    <pgo:operator>RWE</pgo:operator>
    <s:name>380 kV Hessen Ost; Trebur Nord+Sud;380kV Trebur Sud;
    380 kV Trebur Nord;380 kV Hessen Ost</s:name>
    <pgo:voltage>380000;380000;380000;380000</pgo:voltage>
    <s:hasPart>
      <pgo:Pillar rdf:about="datarepo:node-48536">
        <s:geo>
          <s:GeoCoordinates rdf:about="datarepo:node-48536-gs-gc">
            <s:latitude>8.47704075248088</s:latitude>
            <s:longitude>50.0935645616639</s:longitude>
          </s:GeoCoordinates>
        </s:geo>
        <s:name>380 kV Hessen Ost; Trebur Nord|Sud;
        110 kV Kriftel Nord; 380 kV Hessen Ost; Trebur Nord+Sud;
        110 kV Kriftel Nord+Sud</s:name>
        <pgo:operator>RWE</pgo:operator>
        <pgo:hasMeasurements>
          <pgo:Measurements
            rdf:about="datarepo:node-48536-measurements">
              <pgo:voltage>380000;110000;380000;110000</pgo:voltage>
              <pgo:frequency>50;50;50</pgo:frequency>
            </pgo:Measurements>
          </pgo:hasMeasurements>
        </pgo:Pillar>
      </s:hasPart>
      <s:hasPart>
        <pgo:Substation rdf:about="datarepo:node-18221">
          <s:geo>
            <s:GeoCoordinates
              rdf:about="datarepo:node-18221-gs-gc">
                <s:latitude>8.47130677635285</s:latitude>
                <s:longitude>50.0972089406338</s:longitude>
              </s:GeoCoordinates>
            </s:geo>
            <s:name>Umspannwerk Kriftel</s:name>
            <pgo:operator>Amprion</pgo:operator>
            <pgo:hasMeasurements>
              <pgo:Measurements
                rdf:about="datarepo:node-18221-measurements">
                  <pgo:voltage>380000;110000</pgo:voltage>
                  <pgo:frequency>50</pgo:frequency>
                </pgo:Measurements>
              </pgo:hasMeasurements>
            </pgo:Substation>
          </s:hasPart>
        </pgo:Link>
      </rdf:RDF>

```

Листа 4.7 Системски генериран RDF опис. Со помош на онтологијата, генерираме соодветен RDF опис за компонентите на електроенергетскиот систем.

Глава 5

Податочно воден пристап во областа на големи хетерогени системи

5.1 ISO-стандардизирана архитектура на платформа за паметен град

Според Обединетите Нации (ОН), 54 % од светската популација живее во урбани области, а тој однос се очекува да се зголеми на 66 % до 2050 [141]. Со тоа, управување со урбаните области станува еден од најважните развојни предизвици на 21-от век. Гартнер [142] предвидува дека 6,4 милијарди поврзани уреди ќе бидат во употреба во светот во 2016 година, што е за 30 % повеќе во однос на 2015 година, од кои 4 милијарди (62,5 %) ќе бидат во сопственост на граѓаните. Овие поврзани уреди (сензорски уреди) може да вклучуваат, на пример, паметни телефони, системи за играње со вграден сензор, вклучувајќи музички системи, како и сензори во возилата. Корисниците што носат мобилни сензорски уреди стануваат извор на различни податоци, додека движењето на корисникот создавање на можности за просторно густо и временско мерење кои, инаку не би било можно да се остварат. Уреди за поврзување, според тоа, може да ги обезбедат потребните податоци, кои по собирањето и обработката ќе достават информации и комуникациски (ИКТ) технолошки решенија, односно платформи за управување со урбани области.

Концептот на паметен град ги интегрира традиционалните (транспорт, енергија) и модерната ИКТ комуникациска инфраструктура со социјалната инфраструктура (интелектуален капитал и социјален капитал) и социјалните мрежи за да се обезбеди одржлив економски развој и висок квалитет на живот. Тоа вклучува паметна урбанизација [143], паметна потрошувачката на енергија [144],

и паметна употреба на големите урбани компјутерски пресметувања [145], [146]. Еден урбан IoT систем [147] е дизајниран за поддршка на визијата на паметен град преку искористување на најнапредни комуникациски технологии за да се обезбеди на додадена вредност на услуги за општинската администрација и за граѓаните. Сепак, една мала група на паметни апликации за наоѓање на информации со користење на општинските податоци достапни на Интернет не е доволна за градот да се нарече паметен. Покрај тоа, дури и кога граѓаните што имаат уреди поврзани на Интернет и објавуваат податоците како извештаи, потоци [148] кои биле анализирани со континуиран семантички механизми [149], повторно е тешко да се тврди дека градот е паметен. Всушност, посеопфатни фактори за градот треба да се разгледуваат и подобруваат, како што се управувањето со градот [150], прашањата за животната средина [151] и ја истакната улога на граѓаните [152]. Мора да има метрика на „памет“ на урбаната средина и заедничка дефиниција за тоа што навистина претставува паметен град [153].

На првиот меѓународен стандард за податоците на градот кој вклучува 100 индикатори за општинските услуги и квалитетот на животот, беше објавен во мај 2014 година од страна на Меѓународната организација за стандардизација (ISO) [154]. Овој нов меѓународен стандард беше развиен со користење рамки на глобални индикатори и е тестиран од страна на повеќе од 250 градови во светот. Ова е стандарден воден од потребите, предводен и создаден од градовите за градовите. ISO 37120 дефинира и утврдува дефиниции и методологии за множество на индикатори за мерење на перформансите на општинските услуги и квалитетот на живот. индикаторите на ISO 37120 се поделени во 17 категории на општинските услуги и квалитетот на живот: економија, образование, енергија, животна средина, финансии, одговор на пожар и итни интервенции, управување, здравство, рекреација, безбедност, домување, цврст отпад, телекомуникации и иновации, транспорт, урбанистичко планирање, отпадни води и вода и санитарни услови. Тие се одраз на квантитативни податоци и мерења на социјалните, економските и еколошките перформанси на градот. Овие показатели, сепак, се статички (или споре се менуваат) мерки кои покажуваат просечна перформанси на градот.

Во овој труд развиваме концепт на платформа која ги надополнува бавно растечките индикатори со ниска резолуција (годишни, квартални или месечни) дефинирани во ISO стандардот со индикатори со податоци/знаење со висока резолуција, извлечено од социјалните мрежи и индикатори фокусирани на личноста, која може да обезбедат модели за податоци во реално време и заклучување или предвидување идните индикатори за паметен град. Затоа, софтверското решение

ќе се обезбеди методи за набљудување и мерење на феномени од заеднички интерес (на пример, условите во сообраќајот, загадувањето на воздухот, бучавата во урбаните средини), на големи географски области, кои го експлоатираат својството на мобилност на уредите со сензори. Со комбинирање на податоците од социјалните мрежи, вести, блогови и други извори на податоци, се создава генерички преглед на градот, кој може да ги насочи решенијата и технологиите кон транзиција кон паметен град.

Со изградба на алатки за анализа на податоци и анализирајќи ги тековните вести, електронски достапните податоци од уредите со сензори, и социјалните мрежи, ќе обезбедиме богат податочен пристап кон прогнозирање на индикаторите на паметниот град. Платформата ќе има за цел собирање и обработка на податоци, заклучување на знаење, и трансформација на податоци во лесно разбирливи информации преку обезбедување на првата платформа за паметен град базирана на IoT, податоци и пресметување во облак, која ги интегрира различни индикатори, вклучувајќи ги и индикатори на ISO 37120. Во повеќе детали, оваа платформа обезбедува систем со контекстот на социјална апликација. Поконкретно, можните показатели на градот, предвидени од страна на стандардот ISO 37120 за паметни градови, се извлекуваат како влез во предложената платформа, а потоа овие индикатори со ниска резолуција и во повеќето случаи, застарени податоци се дополнети со податоци на граѓаните добиени преку повеќе сензорски уреди или собрани од социјалните мрежи, вести и блогови.

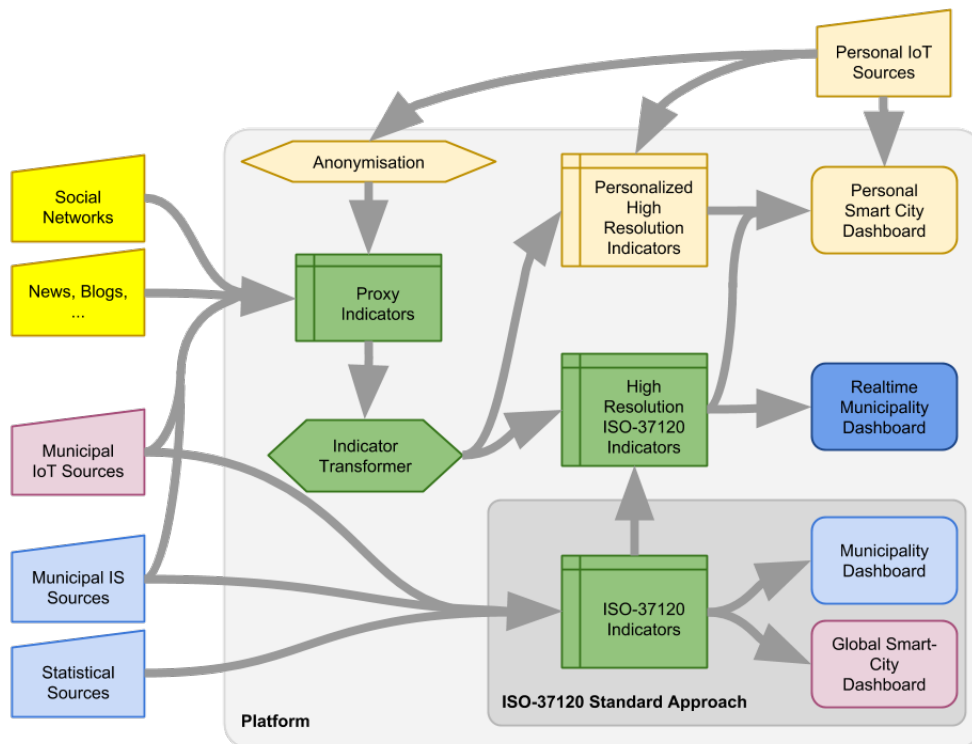
Ова е преглед на поглавјето. Во Дел 5.1.1 е даден преглед на постоечките платформи за паметен град и се адресирани иновациите во нашиот концепт и пристап. Текот на податоците и архитектура на платформата се дадени во Дел 5.1.2. Илустрација на веб-порталот на прототипот е претставена во Дел 5.1.3. Дел 5.1.4 го заклучува поглавјето и ги наведува идните можности за работа.

5.1.1 Достигнувања во досегашниот развој на платформи

Во овој дел ќе биде даден краток преглед на развојот на ИКТ платформи. ThingSpeak е отворена платформа (на GitHub), која овозможува врски помеѓу нештата и луѓето (<https://thingspeak.com>). Таа вклучува функции: собирање на податоци во реално време и складирање, MATLAB анализа и визуелизација, сигнали, распоредување, комуникација со уредите, отворен програмски интерфејс (API) и географски податоци. ThingWorx (<http://www.thingworx.com>) е платформа за развој на софтвер кој е комбинација на клучните функционалности на Веб 2.0, пребарување и социјална соработка за изградба на иновативни апликации

кои ги поврзуваат, зачувуваат и поврзуваат активностите и податоци од луѓето, системите и предметите, вклучувајќи поврзаните машини, уреди, сензори и опрема. NIMBITS (<http://www.nimbits.com/index.jsp>) е отворена платформа за интернетот на предмети за поврзување на луѓе, сензори и уреди во облакот. Тоа е софтвер на дистрибуиран облак за да го зачува и процесира низите од податоци во облак. EVERYTHING (<https://evrythng.com>) управува милијарди интелигентни IoT идентитети за вашите производи во облакот, давајќи му на секој уред трајно и адресибилно присуство на веб. Paraimpu (<https://www.paraimpu.com>) е социјална алатка, со цел да им овозможи на луѓето да се поврзат, создаваат и споделуваат услуги и уреди за да се создадат персонализирани апликации на за Интернетот на предмети. SensorCloud (<http://www.sensorcloud.com>) е платформа која нуди складиште на сензорски податоци, визуелизација и далечинско управување, што користи пресметување во облак за да се обезбеди одлична скалабилност, брзо исцртување и аналитика. Arkessa (<http://www.arkessa.com>) им овозможува на организациите следење, управување и контрола на далечински уреди и објекти. Нуди безбедна и сигурен безжичен систем за мобилните комуникации. Платформата Axeda (<http://www.ptc.com/axeda>) е платформа во облак за изградба и распоредување на напредни апликации, жичено и безжично. Таа им овозможува создавање на бизнис логиката и без програмирање. Таа поддржува мобилна, сателитска и интернет комуникација, оптимизирана за минимални трошоци за пренос. Xively (<https://xively.com>) нуди платформа како сервис кој им овозможува на IoT уредите да се поврзат во облак. Xively е наградувана напредна IoT платформа и апликативно решение, кое ја поедноставува изградбата и водењето на поврзани бизниси.

Постојат многу иницијативи и организации со цел подобрување на споредбата и планирањето на градот. Центарот за напредна просторна анализа (од англ. „Centre for Advanced Spatial Analysis“ - CASA, <http://www.bartlett.ucl.ac.uk/casa>) е една од водечките сили во науката за градови. Тие се фокусирани на генерирање на нови знаења и согледувања за употреба во планирање на градот и идеи во насока на компјутерски-базирана визуелизација и моделирање. Лабораторијата за сензибилен град на MIT (<http://senseable.mit.edu/>) развива и нуди алатки за запознавање на градовите и држи чекор со индустриските партнери, како и локалните власти, граѓаните и маргинализираните заедници. QUT Urban Informatics (<http://www.urbaninformatics.net/>) наметнува пристап со главна улога на луѓето и има тенденција да се спојат физичките и дигиталните слоеви на мрежите на луѓе и урбаната инфраструктура, преку таргетирање на социо-културните, економски,

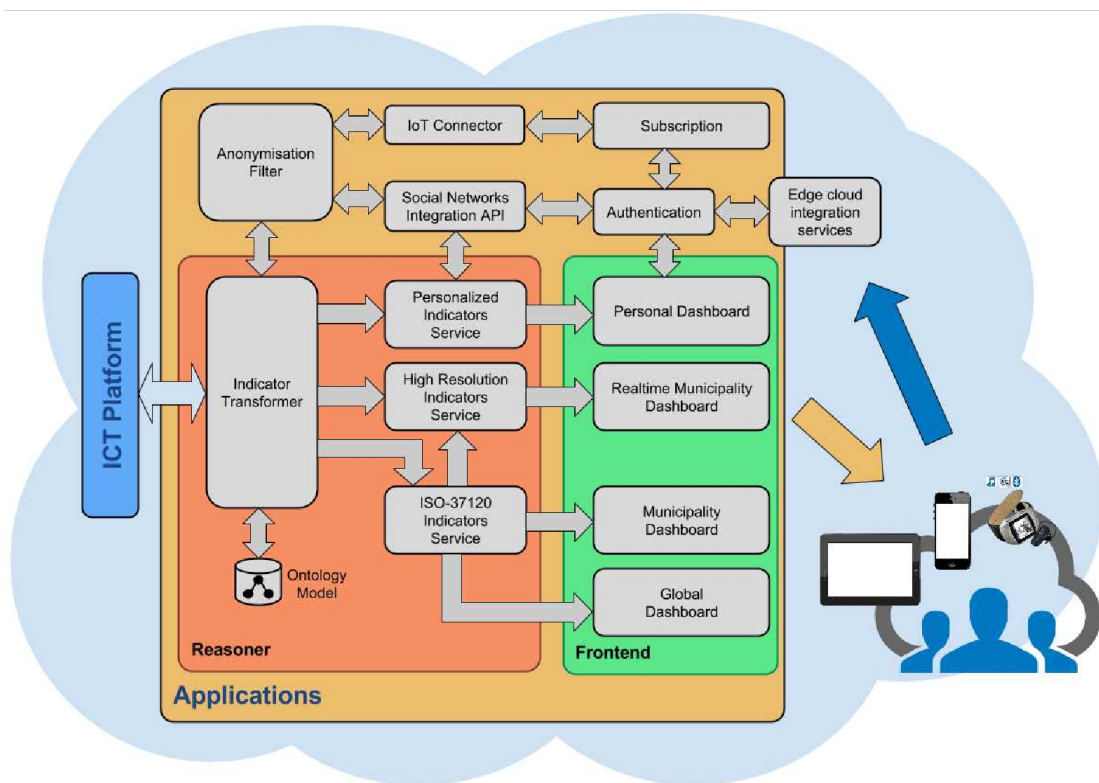


Слика 5.1 Дијаграм на текот на податоците на платформата.

еколошки и технолошки сфери. Националниот центар за гео-пресметување во Ирска (<https://www.maynoothuniversity.ie/national-centre-geocomputation-ncg>) се фокусира на регистрирање, складирање, анализа и визуелизација на просторните податоци и исто така работи со секундарни извори на податоци, на пример шеми за анализа на криминалот, цените на куќите и здравствени податоци.

Постојат многу постоечките платформи за градови во Велика Британија, како Бирмингем, Брајтон, Кардиф, Единбург, Глазгов, Лидс, Лондон и Манчестер (<http://citydashboard.org/choose.php>) и во целиот свет, како Даблин [155] (<http://www.dublindashboard.ie/>), Амстердам (<http://citydashboard.org/amsterdam/>), Чикаго итн. Сеопфатен преглед на компаративни иницијативи кои биле направени од градовите е претставен во [156], а директно влијание на податоците врз градовите се анализира во [157].

Споменатите паметни платформи нудат лесен кориснички интерфејс (UI) и ажурни податоци, но ниту еден од нив не е врз основа на меѓународните стандарди и тие се со недостаток на различни точки на гледање (временски, просторни и сл.) за различни типови на корисници. Решавајќи ги овие недостатоци, ние ја развиваме првата архитектура на платформа врз основа на стандардот ISO 37120.



Слика 5.2 Архитектура на апликацискиот слој.

Главните предности и придонесот на нашата архитектура може да се сумираат како:

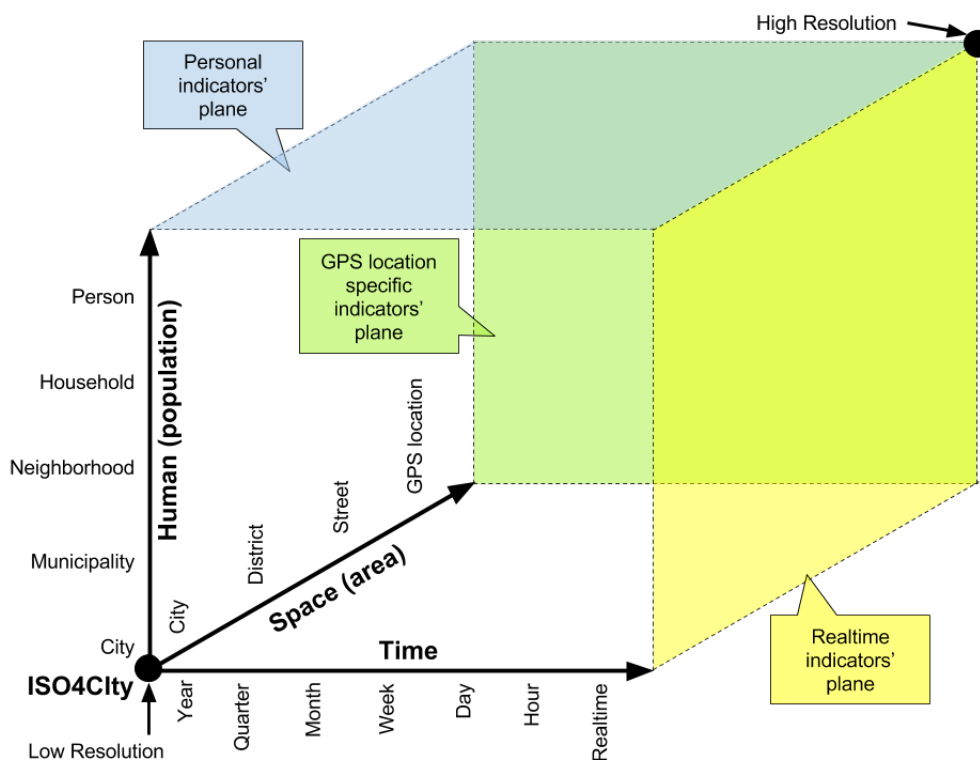
- Архитектура на платформа базирана на стандардот ISO 37120.
- Целосна интеграција на податоците од различни сензорски мрежи, социјални мрежи, вести, блогови, и други извори на податоци што резултираат во контекстно зависна и енергетски ефикасна платформа за IoT, големи податоци и пресметување во облак.
- Концептот на архитектурата на платформата овозможува: (1) координирано, холистичко и подобро планирање на градот и прогнозирање; (2) известувања во реално време за инфраструктурните услови, предвидување и спречување на проблеми, поефикасно искористување на ресурсите; и (3) двонасочна комуникација помеѓу властите и луѓето и ефикасна соработка со граѓаните преку пристап до податоците на градот во реално време.

- Иновативен концептот на 3D простор за индикаторите што резултира со лично-ориентирана, просторно-временска резолуција на претставувањето на градот.

5.1.2 Архитектура на платформата

Јадрото на архитектурата, прикажано на Слика 5.1, е базирано на индикатори на ISO-37120 (или ISO4City, ISO4C) и проширена на погрануларно (временски, просторни) множество на ISO-37120 индикатори со висока резолуција кои обезбедуваат трансформација на основните индикатори кон персонализиран поглед во реално време. За да прилагодат податоците, дополнителна контекстуализација кон гледна точка на граѓанин се врши со помош на филтерот за персоналните индикатори со висока резолуција, кој е директно под влијание на личните IoT извори на податоци. Податоците што доаѓаат од корисниците поминува низ филтер на анонимизација, а потоа целото податочко множество се намалува преку филтерот за посредни индикатори и се трансформираат кон ISO-37120 индикатори со висока резолуција, со користење на трансформаторот за индикатори. Блокот на личните IoT извори е всушност интегрирање на личните IoT сензори (паметни уреди во сопственост на граѓаните). Интеграција на лични IoT покренува прашања за приватност и сигурност на личните податоци [158].

Архитектурата нуди интеграција на податоците достапни од неколку извори подредени во три групи: (1) Статистичките податоци (податоци од постоечките информациона системи во сопственост од градските власти, (2) Податоците што се предвидени од страна на владини и невладини агенции, агенциите на ЕУ, и отворени податоци достапни во периодични извештаи или во континуиран режим), сензорски податоци (податоци од сензори инсталирани од страна на градот, податоци од сензорите во сопственост на различни добавувачи, и (3) податоци од сензорите во сопственост на граѓани, како што се преносни сензори, мобилни телефони, сензори во возилото итн.), и социјални податоци (податоци извадени од социјалните мрежи, блогови, и податоци собрани од весници или портали за собирање на вести). Преносен индикатор претставува индиректна мерка која е приближна или е поврзана со одредена појава во отсуство на директна мерка. Податоците од сензорите, социјалните мрежи, вести, блогови и различни извори (податоци од мобилните оператори) се користат за мерење на индикаторите. Така, индикаторите, кои произлегуваат од процесите во реално време овозможуваат мерење на индикатори за градот во висока резолуција.



Слика 5.3 3D простор на ISO индикатори. Резолуцијата на ISO индикаторите се зголемува со луѓе (население), простор (област) и време, што резултира со прилагодени информации за корисниците.

Нашата идејата за зголемување на резолуцијата на постоечките индикатори и/или воведување на нови (слични) индикатори резултираше со 3D дискретен простор (прикажан на Слика 5.3) со следните оски и нивните дискретни вредности:

- Временска: година, квартал, месец, недела, ден, час, во реално време;
- Просторна (по област): град, област, улица, GPS локација (географска ширина, должина); и
- Човечка (по популација): град, регион/општина (1/10 од населението во градот), соседство (1/10 од населението во општина), домаќинство, лице.

За економскиот индикатор „стапка на невработеност градот“ можеме да го земеме за пример индикаторот „стапка на невработеност во домаќинствата во област“. Или, наместо енергетскиот индикатор „вкупно потрошувачка на електрична енергија по глава на жител“, може да се дефинира нов индикатор „месечно користење на електрична енергија по домаќинство во една улица“. На корисниците

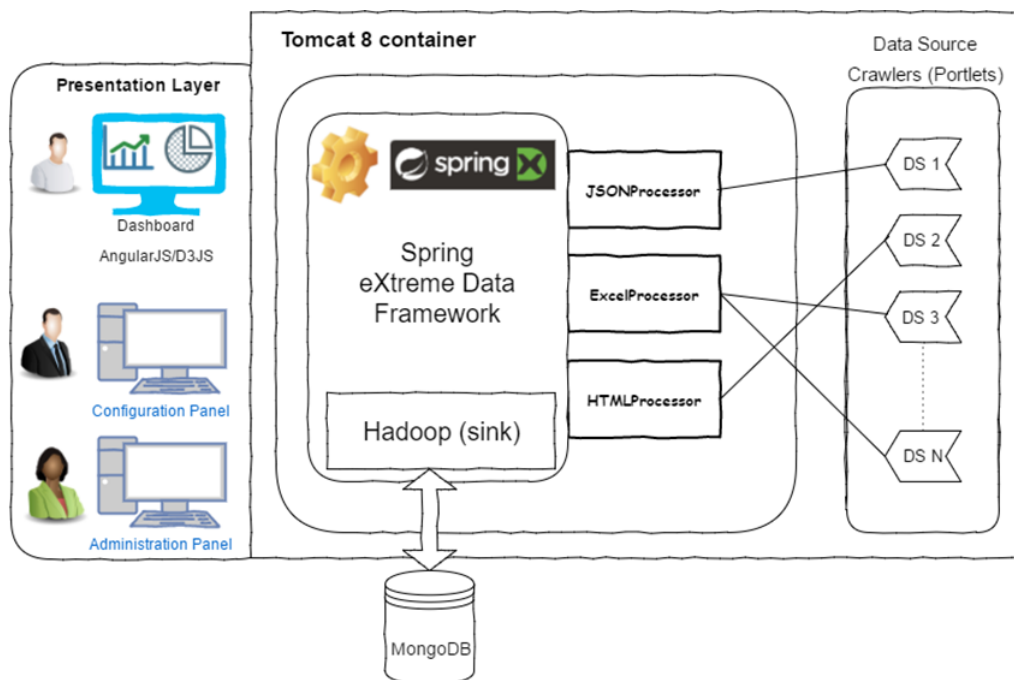
на оваа платформа врз основа на предложените архитектура ќе добијат тековни информации на секоја локација во градот за одреден индикатор (или група на индикатори). Треба да забележиме дека не за сите индикатори постојат сите три димензии. Некои од индикаторите се добро дефинирани само ако се мери на годишно ниво и за целиот град. Групата на индикатори со висока резолуција е основа за следење во реално време на виталните процеси на градот.

Архитектурата на апликацискиот слој е прикажана на Слика 5.2 и се состои од интерфејс за интеграција со модулот за собирање на податоци, основна интелигенција и логика на размислување, помошни блокови за автентикација и безбедност и кориснички интерфејс, прикажан преку веб апликација и мобилна апликација, веб сервиси и безбедни интеграција со IoT. Програмскиот интерфејс (API) за интеграција со социјалните мрежи ги поврзува корисничките сметки на граѓаните. Резултатот е да се рангираат граѓаните според нивната вклученост во подобрување на општинските сервиси и квалитетот на живот, да се споделуваат резултатите во континуитет (дневно, неделно), да се делат совети за подобрување на општинските сервиси и квалитетот на живот и на тој начин да ги мотивира и да ги поттикне граѓаните да учествуваат активно со користење на оваа платформа.

Моделот на онтологијата претставува семантички модел на податоци потребен за трансформација на податоците во индикатори и правила на одлучување, кои се резултат на истражувањата во насока на подобрување резолуцијата на индикаторите и нивната персонализација. Трансформаторот на индикатори првично ги оркестрира податоците од различни интероперабилни сервиси на платформата и поминува преку сервисите на индикатори (како JSON или Јава објекти). ISO 37120 сервисот ги обработува податоците и ги проследува до корисниците преку глобалната и општинскиот поглед. Софтверскиот блок за индикатори со висока резолуција едноставно наследува блок со ISO индикатори треба да ја подигне нивната резолуцијата со користење на дополнителни податоци од граѓаните, како и семантичкото знаење стекнато од страна на системот, односно од пристапот на машинско учење. Сервисот за персонализирани индикатори ги интегрира личните податоци на граѓаните (собрани директно од социјалните мрежи или преку филтерот за анонимизација и платформата за податоци), со цел да ги персонализира податоците на индикаторот кон личните приоритети на граѓаните.

5.1.3 Прототип

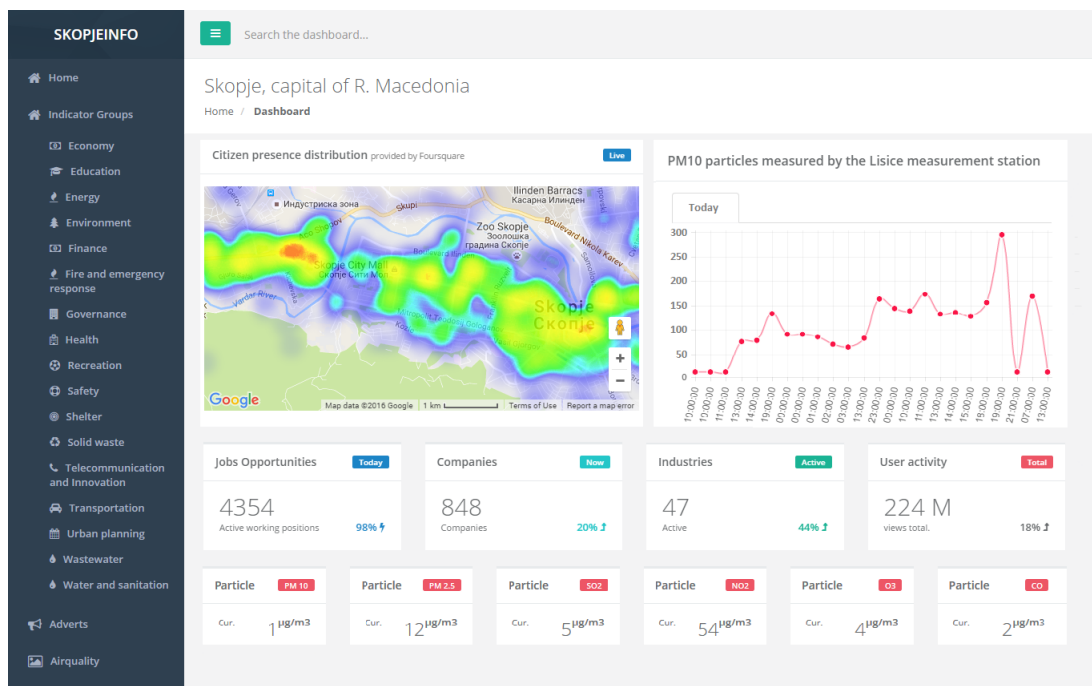
Имплементиравме прототип [159] следејќи отворени стандарди и технологии на софтвер со отворен код, како и корпоративна рамка за развој на софтвер, целосно



Слика 5.4 Имплементација со програмската околина на Јава.

достапен во Spring платформата во Јава (Spring MVC) и неговите модули, како технологија во раст и развој, прикажано на Слика 5.4. Контејнерите и аспект ориентираното програмирање (од англ. „Aspect Oriented Programming“ - AOP) се погодни за модуларен развој на софтвер и се применуваат за имплементација на блоковите со индикатори во нашиот систем [160]. Spring сервисите се основа за сервисите на индикатори, додека Spring контролерите се вклопуваат во интеграцијата на корисничкиот интерфејс. Од друга страна, Spring XD е проект кој овозможува брз развој на податочно-водени апликации и обезбедува интеграција податочни потоци итн.

Платформата е овозможена за користење на постојните јавни сервиси во Скопје, Македонија. Неколку од нив се имплементирани од страна на Факултетот за информатички науки и компјутерско инженерство, како што се апликацијата „Green Route“, која обезбедува информации на патниците за најбрзиот, најевтиниот и најеколошкиот пат до дестинации во главниот град и ги покажува во реално време сообраќајниот метеж и загадувањето на воздухот (<http://skopjegrreenroute.mk>). Податоците за протокот на сообраќај, метеж и загадување на воздухот (емисии на CO₂) се земени од сензори инсталирани од страна на локалните власти на градот на главните крстосници и најфреквентните сообраќајни правци во градот. Податоците за загадувањето на воздухот се собрани од неколку мерни станици за



Слика 5.5 ISO стандардизирана платформа за паметен град, која ги прикажува категориите на индикатори (горе-лево), филтер според оските на 3D просторот со висока резолуција (горе) и податоците за градот Скопје.

квалитет на воздух во Скопје, кои може да мерат 5 индикатори на квалитетот на воздухот: CO, O3, NO2, PM2.5 и PM10. Основните информации за Скопје, интересни за нови посетители, се достапни од SPARQL сервис на Dbpedia, имплементиран според концептот на поврзани отворени податоци и овозможува ефикасен начин за поврзување со соодветните бази на податоци, важни за привлекување на вниманието на посетителот. Информациите за временската прогноза се земени од локалните сервиси за временска прогноза. Таквите информации, во последно време, може да се користи за пресметка на ефикасен планер на патување. Во однос на резолуцијата на ISO-индикатори според 3D координатите, е зголемена до одредена личност, GPS локација и податоци прикажани речиси реално време.

Податоците за вработеноста се земени од постојното решение (<http://prv.mk>), имплементирано во соработка со Агенцијата за вработување на РМ, која агрегира огласи за работа на едно место од сите извори низ Македонија и овозможува лесен интерфејс (мобилен и веб) за преглед на огласите. Во моментот имаме речиси 10k активни корисници кои се обидуваат да го решат нивниот работен статус со користење на нашата апликација. PRVmk овозможува интелигентни сервиси, кој за дадени основни информации и способности на корисникот, ќе

предложи активни огласи каде што може да аплицира за работа. Намалувањето на ризици од катастрофи, исто така, е опфатен во нашиот прототип. Во соработка со Центарот за управување со кризи, имаме неодамна развиена јавен сервис кој обезбедува пристап до детални информации за опасни настани како земјотреси, поплави и пожари, како и потенцијални опасности, како што се силни грмежи и обилни врнежи од снег, кои може да ги прекинат секојдневните активности на корисниците (<http://cuk.finki.ukim.mk>). Со користење на апликација за мобилни уреди, системот го известува корисникот во реално време за потенцијалните и постојните опасности. Во соработка со Министерството за внатрешни работи, развиена е паметна апликација која презема податоци од документите објавени од страна на Министерството за внатрешни работи и ги класифицира во соодветен тип на настан извлекувајќи мета-податоци од суровиот текстуален формат со користење на напредни алгоритми за процесирање на природни јазици. Потоа, ги претставува настани на соодветна мапа, прикажувајќи ги сите настани од последните неколку дена, класифицирани како сообраќајна несреќа, грабеж, насилство, употреба на дрога и тн. Во нашата платформа, се подобрува нивото прикажување на криминални настани во Скопје, обезбедувајќи детална статистика на стапката на појава настани групирани по региони или општини за различни сезони во годината (<http://crimemap.finki.ukim.mk>).

Општинските приоритети може да се калибрираат со влезни податоци на граѓаните преку платформата „мојата општина“ (<http://moja-opstina.mk>) - онлајн алатка овозможена од страна на Програмата за развој на Обединетите нации (УНДП, од англ. „United Nations Development Programme“) и Факултетот за информатички науки и компјутерско инженерство. Таа е наменета за прибирање на мислењата на граѓаните за решавање на проблеми во локалната самоуправа, како и за подобрување на локалните и приоритети за развој на градот. Социјалните мрежи Flickr и Foursquare, исто така, учествуваат во платформата за обезбедување на дополнителни информации: Flickr API овозможува галерија од најпопуларните фотографии на најфреквентните места во Скопје, додека Foursquare API се користи за генерирање на мапа на активни посетители на јавните места во реално време.

5.1.4 Заклучок

Различни не-стандардизирани платформи за паметен град биле предложени во минатото. Применувајќи го стандардот ISO 37120 за општинските сервиси и квалитетот на живот, ја претставивме архитектурата за стандардизирана платформа

за паметен град, која го надградува ISO 37120 стандардот во однос на нови индикатори поврзани со постоечките основни индикатори. Платформата интегрира податоци од хетерогени извори, ги обработува и обезбедува резултати сервиси и нуди визуелизација на индикаторите реално време во различни временски резолуции. Архитектурата беше тестирана на податоците обезбедени од страна на градот Скопје, Македонија, така што расположливите податоци на општините беа соединетисо податоците од социјалните мрежи и прилагодени во форма соодветна на ISO категории и индикатори.

Покрај веб апликацијата, во иднина се планира и развој на мобилна апликација за најпопуларните платформи (Андроид, iPhone и Windows), како и за напредни паметни уреди (како што се паметни часовници, паметни ТВ приемници и др, кои ги претставуваат корисникот облак и личните IoT уреди). Овие апликации, не само што ќе ја донесат платформа поблиску до граѓаните овозможувајќи персонализиран пристап кон ISO 37120 стандардот, но ќе им овозможи на граѓаните да ги споделат своите сензорски податоци, правејќи ја платформата моќна алатка за подобрување на општинските услуги и квалитетот на животот.

Глава 6

Заклучок и идна работа

Резултатите од истражувањето во рамките на докторската дисертација придонесоа кон популаризација на податочно водениот пристап во областа на хардверски кола и системи. Резултатите од истражувањето беа интегрирани во HDL IP Cores системот во форма на напредни алатки за дизајн на електронски кола, DILB прототипот за електрични мрежи и ИСО-стандардизираната платформа за паметен град, имплементирана за градот Скопје.

Резултатите од истражувањето придонесоа за надградување на функционалностите и создавањена нови кориснички сценарија во рамките на HDL IP Cores системот, но и во други научни трудови и системи. Исто така беше остварено подобрување на точноста и прецизноста на алгоритмите за определување на меѓусебна сличност и компатибилност на компонентите, како и подобрување на алгоритмите за класификација и пребарување на компоненти. Во рамките на HDL IP Cores системот беше создадено и независно податочно множество, кое може да се користи и од други научници или развивачи на софтвер за тестирање на нови научни модели и методи, како и за развој на нови алатки за дизајн на електронски кола. Анализата во насока на напредни алатки за дизајн на електронски кола ќе обезбеди теоретска основа за развој и подобрување во насока на поедноставен развој на нови компоненти, нивно споделување, валидација и верификација, како и меѓусебна поврзаност и тековно подобрување на постоечките компоненти. Исто така, податочните модели и податочните множества, што ќе се создадат во рамките на истражувањето се очекува да бидат добра основа за развој и тестирање на нови методи и функционалности.

Резултатите од истражувањето поттикнаа примена на податочно-базирани техники и методи кон оптимизација на електричните мрежи, но и директно влијаеја врз подобрување на карактеристиките на DILB прототипот. Развиените

податочни модели и алгоритми резултира со значителни подобрувања во доменот на динамичко балансирање на товарот и минимизација на реактивната енергија. Истражувањето во насока на оптимизација на електричните мрежи ќе придонесе кон подобрување на механизмите за динамичко балансирање на товарот во повеќе фазните мрежи, како и минимизација на реактивната енергија. Но, се очекува начинот на решавање на споменатите проблеми да поттикне и размислувања за решавање на повеќе други познати проблеми во дистрибутивните мрежи со помош на податочни водениот пристап.

Применувајќи го стандардот ISO 37120 за општинските сервиси и квалитетот на живот, ја претставивме архитектурата за стандардизирана платформа за паметен град, која го надградува ISO 37120 стандардот во однос на нови индикатори поврзани со постоечките основни индикатори. Платформата интегрира податоци од хетерогени извори, ги обработува и обезбедува резултати сервиси и нуди визуелизација на индикаторите реално време во различни временски резолуции. Архитектурата беше тестирана на податоците обезбедени од страна на градот Скопје, Македонија, така што расположливите податоци на општините беа соединетисо податоците од социјалните мрежи и прилагодени во форма соодветна на ISO категории и индикатори.

Покрај веб апликацијата, во иднина се планира и развој на мобилна апликација за најпопуларните платформи (Андроид, iPhone и Windows), како и за напредни паметни уреди (како што се паметни часовници, паметни ТВ приемници и др, кои ги претставуваат корисникот облак и личните IoT уреди).

Резултатите од спроведеното истражување во докторската дисертација се очекува да поттикнат примена на податочни водениот пристап во различни области, а особено за решавање на проблеми во полето на хардверски кола и системи. Податочни водениот пристап ќе отвори можност за подобрување на постоечките и развој на нови кориснички сценарија и методи за подобрување на карактеристиките на постоечките системи, но и ќе наметне нов научен пристап во решавањето на постоечките проблеми.

Спроведеното истражување ќе придонесе за издвојување на најдобри практики за примена на податочни водениот пристап во областа на хардверски кола и системи. Специфичните чекори се очекува да се идентификуваат во сите сегменти од податочни водениот пристап, и тоа почнувајќи од процесот на идентификување на корисни податоци и нивно собирање, преку правилно индексирање и структурирање, па се до анализа на податоците и носење заклучоци и преземање на соодветни акции во зависност од заклучоците.

Сублимацијата на најдобри практики за примена на податочно водениот пристап во рамките на хардверските кола и системи ќе послужат како насока на идните истражувања во овие области, а може да послужат за побрза идентификација на податочно водените елементи и нивна примена за подобрување на постоечки апликации и системи. Добиените научни модели се очекува да бидат применливи не само во потесната област на хардверските кола и системи, туку и во други сродни области. Предложените концепти за конкретни подобрувања во алатките и системите, што се предмет на истражувањето, ќе бидат потврдени со научно-мерливи резултати, споредливи или подобри од постоечките научни модели и пристапи за решавање на истите проблеми. Со тоа ќе се зголеми вредноста и значењето на податоците во овој домен и искористувањето на податоците се очекува да добие поголемо влијание во тековните процеси во рамките на дизајн на кола и оперативната контрола на системите.

Литература

- [1] T. Hey, S. Tansley, K. M. Tolle, et al., *The fourth paradigm: data-intensive scientific discovery*, vol. 1. Microsoft research Redmond, WA, 2009.
- [2] C. Anderson, *Creating a data-driven organization*. "O'Reilly Media, Inc. 2015.
- [3] V. R. Melnikov, V. V. Krzhizhanovskaya, A. V. Boukhanovsky, and P. M. Slood, "Data-driven modeling of transportation systems and traffic data analysis during a major power outage in the netherlands," *Procedia Computer Science*, vol. 66, pp. 336–345, 2015.
- [4] Y. Velde, S. W. Lyon, and G. Destouni, "Data-driven regionalization of river discharges and emergent land cover–evapotranspiration relationships across sweden," *Journal of Geophysical Research: Atmospheres*, vol. 118, no. 6, pp. 2576–2587, 2013.
- [5] H. Nampak, B. Pradhan, and M. A. Manap, "Application of gis based data driven evidential belief function model to predict groundwater potential zonation," *Journal of Hydrology*, vol. 513, pp. 283–300, 2014.
- [6] D. Niemeijer, "Developing indicators for environmental policy: data-driven and theory-driven approaches examined by example," *Environmental Science & Policy*, vol. 5, no. 2, pp. 91–103, 2002.
- [7] S. Lewis, T. Foltynie, A. Blackwell, T. Robbins, A. Owen, and R. Barker, "Heterogeneity of parkinson's disease in the early clinical stages using a data driven approach," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 76, no. 3, pp. 343–348, 2005.
- [8] K. Dujardin, A. F. Leentjens, C. Langlois, A. J. Moonen, A. A. Duits, A.-S. Carette, and A. Duhamel, "The spectrum of cognitive disorders in parkinson's disease: A data-driven approach," *Movement Disorders*, vol. 28, no. 2, pp. 183–189, 2013.
- [9] S. Shiboski, C. Shiboski, L. Criswell, A. Baer, S. Challacombe, H. Lanfranchi, M. Schiødt, H. Umehara, F. Vivino, Y. Zhao, et al., "American college of rheumatology classification criteria for sjögren's syndrome: A data-driven, expert consensus approach in the sjögren's international collaborative clinical alliance cohort," *Arthritis care & research*, vol. 64, no. 4, pp. 475–487, 2012.

- [10] M. Oorschot, T. Lataster, V. Thewissen, M. Wichers, and I. Myin-Germeys, "Mobile assessment in schizophrenia: a data-driven momentary approach," *Schizophrenia bulletin*, p. sbr166, 2011.
- [11] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation—a review on the statistical data driven approaches," *European journal of operational research*, vol. 213, no. 1, pp. 1–14, 2011.
- [12] N. P. Tatonetti, P. Y. Patrick, R. Daneshjou, and R. B. Altman, "Data-driven prediction of drug effects and interactions," *Science translational medicine*, vol. 4, no. 125, pp. 125ra31–125ra31, 2012.
- [13] S. J. Lusher, R. McGuire, R. C. van Schaik, C. D. Nicholson, and J. de Vlieg, "Data-driven medicinal chemistry in the era of big data," *Drug discovery today*, vol. 19, no. 7, pp. 859–868, 2014.
- [14] S. Yin, S. X. Ding, A. H. Abandan Sari, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1366–1376, 2013.
- [15] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.
- [16] G. Wang, S. Yin, and O. Kaynak, "An lwpr-based data-driven fault detection approach for nonlinear process monitoring," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2016–2023, 2014.
- [17] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven soft sensors in the process industry," *Computers & Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009.
- [18] F. Darema, "Dynamic data driven applications systems: A new paradigm for application simulations and measurements," *Computational Science-ICCS 2004*, pp. 662–669, 2004.
- [19] X. Jian-Xin and H. Zhong-Sheng, "Notes on data-driven system approaches," *Acta Automatica Sinica*, vol. 35, no. 6, pp. 668–675, 2009.
- [20] W. Tan, Y. Fan, M. Zhou, and Z. Tian, "Data-driven service composition in enterprise soa solutions: A petri net approach," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 686–694, 2010.
- [21] T. Johns and S. Y. be Persuaded, "Two samples of data-driven learning materials'," *Classroom C0ncOrdancing. ELR Journal University of Birm ingham*, vol. 4, p. 84, 1990.
- [22] T. Johns, "1 9 from printout to handout: Grammar and vocabulary teaching in the context of data-driven learning," *Perspectives on pedagogical grammar*, vol. 293, 1994.

-
- [23] M. Schmachtenberg, C. Bizer, and H. Paulheim, "Adoption of the linked data best practices in different topical domains," in *International Semantic Web Conference*, pp. 245–260, Springer, 2014.
- [24] R. Peinl, "Semantic web: State of the art and adoption in corporations," *KI-Künstliche Intelligenz*, vol. 30, no. 2, pp. 131–138, 2016.
- [25] P. F. Patel-Schneider, "Analyzing schema. org," in *International Semantic Web Conference*, pp. 261–276, Springer, 2014.
- [26] D. Zhang, S. Li, Y. Liu, X. S. Hu, X. He, Y. Zhang, P. Zhang, and H. Yang, "A c2rtl framework supporting partition, parallelization, and fifo sizing for streaming applications," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 21, no. 2, p. 19, 2016.
- [27] Z. Ye and C. Hua, "An innovative method of teaching electronic system design with psoc," *IEEE Transactions on Education*, vol. 55, no. 3, pp. 418–424, 2012.
- [28] S. Vemuru, S. Khorbotly, and F. Hassan, "A spiral learning approach to hardware description languages," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pp. 2759–2762, IEEE, 2013.
- [29] D. Mathaikutty and S. Shukla, "Soc design space exploration through automated ip selection from systemc ip library," in *SOC Conference, 2006 IEEE International*, pp. 109–110, IEEE, 2006.
- [30] F. Doucet, S. Shukla, M. Otsuka, and R. Gupta, "Balboa: A component-based design environment for system models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 12, pp. 1597–1612, 2003.
- [31] H. Lange and A. Koch, "Hw/sw co-design by automatic embedding of complex ip cores," in *International Conference on Field Programmable Logic and Applications*, pp. 679–689, Springer, 2004.
- [32] S. Vakili, J. P. Langlois, and G. Bois, "Customised soft processor design: a compromise between architecture description languages and parameterisable processors," *IET Computers & Digital Techniques*, vol. 7, no. 3, pp. 122–131, 2013.
- [33] A. M. Fernandes, R. Pereira, J. Sousa, A. Batista, A. Combo, B. Carvalho, C. Correia, and C. Varandas, "Hdl based fpga interface library for data acquisition and multipurpose real time algorithms," *IEEE Transactions on Nuclear Science*, vol. 58, no. 4, pp. 1526–1530, 2011.
- [34] E. Carpaneto and G. Chicco, "Distribution system minimum loss reconfiguration in the hyper-cube ant colony optimization framework," *Electric Power Systems Research*, vol. 78, no. 12, pp. 2037–2045, 2008.
- [35] A. Ukil and W. Siti, "Feeder load balancing using fuzzy logic and combinatorial optimization-based implementation," *Electric Power Systems Research*, vol. 78, no. 11, pp. 1922–1932, 2008.

-
- [36] Y.-T. Hsiao and C.-Y. Chien, "Multiobjective optimal feeder reconfiguration," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 148, no. 4, pp. 333–336, 2001.
- [37] K. Wang, S. Skiena, and T. G. Robertazzi, "Phase balancing algorithms," *Electric Power Systems Research*, vol. 96, pp. 218–224, 2013.
- [38] Y. Weng, R. Negi, C. Faloutsos, and M. D. Ilić, "Robust data-driven state estimation for smart grid," *IEEE Transactions on Smart Grid*, 2016.
- [39] J. Zhu, E. Zhuang, C. Ivanov, and Z. Yao, "A data-driven approach to interactive visualization of power systems," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2539–2546, 2011.
- [40] L. Xie, Y. Zhang, and M. D. Ilic, "Multi-scale integration of physics-based and data-driven models in power systems," in *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, pp. 129–137, IEEE Computer Society, 2012.
- [41] X. Shi, H. Damgacioglu, and N. Celik, "A dynamic data-driven approach for operation planning of microgrids," *Procedia Computer Science*, vol. 51, pp. 2543–2552, 2015.
- [42] E. Xydas, C. Marmaras, L. M. Cipcigan, N. Jenkins, S. Carroll, and M. Barker, "A data-driven approach for characterising the charging demand of electric vehicles: A uk case study," *Applied Energy*, vol. 162, pp. 763–771, 2016.
- [43] E. C. Kara, J. S. Macdonald, D. Black, M. Bérges, G. Hug, and S. Kiliccote, "Estimating the benefits of electric vehicle smart charging at non-residential locations: A data-driven approach," *Applied Energy*, vol. 155, pp. 515–525, 2015.
- [44] M. Piramanrique, R. Francisco, and J. Sofrony Esmeral, "Data driven fault detection and isolation: a wind turbine scenario," *Tecnura*, vol. 19, no. 44, pp. 71–82, 2015.
- [45] A. Verma and K. Yang, "A data-driven approach for wind turbine performance bench-marking," in *IIE Annual Conference. Proceedings*, p. 1120, Institute of Industrial Engineers-Publisher, 2014.
- [46] E. T. Renani, M. F. M. Elias, and N. A. Rahim, "Using data-driven approach for wind power prediction: A comparative study," *Energy Conversion and Management*, vol. 118, pp. 193–203, 2016.
- [47] J. Zhang, Y. Zhu, W. Shi, G. Sheng, and Y. Chen, "An improved machine learning scheme for data-driven fault diagnosis of power grid equipment," in *High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on*, pp. 1737–1742, IEEE, 2015.

-
- [48] H. Jiang, J. J. Zhang, W. Gao, and Z. Wu, "Fault detection, identification, and location in smart grid based on data-driven computational methods," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2947–2956, 2014.
- [49] F. Raak, Y. Susuki, and T. Hikihara, "Data-driven partitioning of power networks via koopman mode analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2799–2808, 2016.
- [50] Y. Weng, C. Faloutsos, and M. D. Ilić, "Data-driven topology estimation," in *Smart Grid Communications (SmartGridComm)*, 2014 IEEE International Conference on, pp. 560–565, IEEE, 2014.
- [51] T. Berners-Lee, J. Hendler, O. Lassila, et al., "The semantic web," *Scientific american*, 2001.
- [52] T. Berners-Lee and N. Shadbolt, "There's gold to be mined from all our data," *The Times*, London, 2011.
- [53] T. Berners-Lee, "Long live the web," *Scientific American*, vol. 303, no. 6, pp. 80–85, 2010.
- [54] V. Zdraveski, M. Jovanovik, R. Stojanov, and D. Trajanov, "Hdl ip cores search engine based on semantic web technologies," in *International Conference on ICT Innovations*, pp. 306–315, Springer, 2010.
- [55] V. Zdraveski, M. Todorovski, and L. Kocarev, "Dynamic intelligent load balancing in power distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 157–162, 2015.
- [56] V. Zdraveski, M. Todorovski, D. Trajanov, and L. Kocarev, "Dynamic load balancing and reactive power compensation switch embedded in power meters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 4, pp. 422–426, 2017.
- [57] L. Kocarev, V. ZDRAVESKI, and M. TODOROVSKI, "Method and system for dynamic intelligent load balancing," Sept. 11 2015. WO Patent App. PCT/US2015/018,482.
- [58] C. U. Z. V. S. M. T. M. S. M. Kocarev, Ljupco (La Jolla, "Method and system for dynamic intelligent load balancing," March 2017.
- [59] D. Patterson, "People of acm," 2016.
- [60] H. Dibowski, J. Ploennigs, and K. Kabitzsch, "Automated design of building automation systems," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 11, pp. 3606–3613, 2010.
- [61] J. Teich, "Hardware/software codesign: The past, the present, and predicting the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1411–1430, 2012.

- [62] F. Fummi, M. Loghi, M. Poncino, and G. Pravadelli, "A cosimulation methodology for hw/sw validation and performance estimation," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 14, pp. 23:1–23:32, Apr. 2009.
- [63] W. Silva, E. Bezerra, M. Winterholer, and D. Lettnin, "Automatic property generation for formal verification applied to hdl-based design of an on-board computer for space applications," in *Test Workshop (LATW), 2013 14th Latin American*, pp. 1–6, 2013.
- [64] G. Smith, "Time is money," *Design Test, IEEE*, vol. 30, no. 1, pp. 55–57, 2013.
- [65] P. F. Patel-Schneider, *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19–23, 2014. Proceedings, Part I, ch. Analyzing Schema.org*, pp. 261–276. Cham: Springer International Publishing, 2014.
- [66] T. Berners-Lee and N. Shadbolt, "There's gold to be mined from all our data," *The Times*, 2011.
- [67] T. Berners-Lee, "Long live the web," *Scientific American*, Dec. 2010.
- [68] H. Lange and A. Koch, "Hardware/software-codesign by automatic embedding of complex ip cores," *Field Programmable Logic and Application, 14th International Conference , FPL*, 2004.
- [69] S. Vakili, J. Langlois, and G. Bois, "Customised soft processor design: a compromise between architecture description languages and parameterisable processors," *Computers Digital Techniques, IET*, vol. 7, no. 3, pp. 122–131, 2013.
- [70] A. Fernandes, R. Pereira, J. Sousa, A. J. N. Batista, A. Combo, B. Carvalho, C. M. B. A. Correia, and C. A. F. Varandas, "Hdl based fpga interface library for data acquisition and multipurpose real time algorithms," *Nuclear Science, IEEE Transactions on*, vol. 58, no. 4, pp. 1526–1530, 2011.
- [71] S. K. Shukla, F. Doucet, and R. K. Gupta, "Structured component composition frameworks for embedded system design," *HiPC '02 Proceedings of the 9th International Conference on High Performance Computin*, 2002.
- [72] D. Mathaikutty and S. Shukla, "Soc design space exploration through automated ip selection from systemc ip library," in *In proceedings of IEEE International SOC Conference*, 2006.
- [73] N. Bombieri, F. Fummi, and G. Pravadelli, "Automatic abstraction of rtl ips into equivalent tlm descriptions," *Computers, IEEE Transactions on*, vol. 60, no. 12, pp. 1730–1743, 2011.
- [74] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, "High-level synthesis for fpgas: From prototyping to deployment," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, pp. 473–491, April 2011.

- [75] A. Canis, J. Choi, M. Aldham, V. Zhang, A. Kammoona, T. Czajkowski, S. D. Brown, and J. H. Anderson, "Legup: An open-source high-level synthesis tool for fpga-based processor/accelerator systems," *ACM Trans. Embed. Comput. Syst.*, vol. 13, pp. 24:1–24:27, Sept. 2013.
- [76] M. Schmachtenberg, C. Bizer, and H. Paulheim, *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19–23, 2014. Proceedings, Part I*, ch. Adoption of the Linked Data Best Practices in Different Topical Domains, pp. 245–260. Cham: Springer International Publishing, 2014.
- [77] C. Bizer, A. Jentzsch, and R. Cyganiak, "State of the lod cloud," Version 0.3 (September 2011), vol. 1803, 2011.
- [78] E. Prud'Hommeaux, A. Seaborne, et al., "Sparql query language for rdf," W3C recommendation, vol. 15, 2008.
- [79] S. Tramp, P. Frischmuth, T. Ermilov, and S. Shekarpour, "An architecture of a distributed semantic social network," *Semantic Web*, Mar. 2011.
- [80] M. Jovanovik, P. Ristoski, and D. Trajanov, "A system for suggestion and execution of semantically annotated actions based on service composition," in *ICT Innovations 2013*, pp. 97–109, Springer International Publishing, 2013.
- [81] D. Trajanov, R. Stojanov, M. Jovanovik, V. Zdraveski, P. Ristoski, M. Georgiev, and S. Filiposka, "Semantic sky: A platform for cloud service integration based on semantic web technologies," in *Proceedings of the 8th International Conference on Semantic Systems*, pp. 109–116, ACM, 2012.
- [82] *Semantic Web Programming*. Wiley Publishing, Inc., 2009.
- [83] J. Fischer and D. H. Huson, "New common ancestor problems in trees and directed acyclic graphs," *Information Processing Letters archive*, pp. 331–335, 2010.
- [84] *Digital System Design with VHDL*. Pearson Education Limited, 2004.
- [85] V. Zdraveski, M. Jovanovik, R. Stojanov, and D. Trajanov, "Hdl ip cores search engine based on semantic web technologies," *ICT Innovations 2010, Communications in Computer and Information Science*, vol. 83, no. 2, pp. 306 – 315, 2011.
- [86] V. Zdraveski, A. Dimitrovski, and D. Trajanov, "Hdl ip cores system as an online testbench provider," 2014.
- [87] K. Arabi, "Special session 6c: New topic mixed-signal test impact to soc commercialization," in *2010 28th VLSI Test Symposium (VTS)*, pp. 212–212, April 2010.
- [88] C.-Y. Huang, Y.-F. Yin, C.-J. Hsu, T. B. Huang, and T.-M. Chang, "Soc hw/sw verification and validation," in *16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011)*, pp. 297–300, IEEE, 2011.

-
- [89] H. Zhang, C. Wu, W. Zhang, and J. Wang, "Design on soc module-level functional verification platform," in *Mechanic Automation and Control Engineering (MACE)*, 2011 Second International Conference on, pp. 4012–4015, July 2011.
- [90] Q. Meng, Z. Li, and F. Wang, "Functional verification of external memory interface ip core based on restricted random testbench," in *Computer Engineering and Technology (ICCET)*, 2010 2nd International Conference on, vol. 7, pp. V7–253–V7–257, April 2010.
- [91] L. Feng, Z. Dai, W. Li, and J. Cheng, "Design and application of reusable soc verification platform," in *ASIC (ASICON)*, 2011 IEEE 9th International Conference on, pp. 957–960, Oct 2011.
- [92] X. Cheng, A. W. Ruan, Y. B. Liao, P. Li, and H. C. Huang, "A run-time rtl debugging methodology for fpga-based co-simulation," in *Communications, Circuits and Systems (ICCCAS)*, 2010 International Conference on, pp. 891–895, July 2010.
- [93] F. Corno, E. Sánchez, M. S. Reorda, and G. Squillero, "Automatic test program generation: a case study," *IEEE Design & Test of Computers*, vol. 21, no. 2, pp. 102–109, 2004.
- [94] V. Zdraveski and D. Trajanov, "Vhdl ip cores ontology," in *Conference for Informatics and Information Technology (CIIT)*, 2013.
- [95] A. D. Brown, K. G. Nichols, and M. Zwolinski, "Issues in the design of a logic simulator: an improved caching technique for event-queue management," *IEE Proceedings - Circuits, Devices and Systems*, vol. 142, pp. 293–298, Oct 1995.
- [96] D. M. Maksimovi and V. B. Litovski, "Timing simulation with vhdl simulators," in *Microelectronics, 2002. MIEL 2002. 23rd International Conference on*, vol. 2, pp. 655–658, 2002.
- [97] D. M. Maksimovic and V. B. Litovski, "Tuning logic simulators for timing analysis," *Electronics Letters*, vol. 35, pp. 800–802, May 1999.
- [98] G. Martin and G. Smith, "High-level synthesis: Past, present, and future," *IEEE Design & Test of Computers*, vol. 26, no. 4, pp. 18–25, 2009.
- [99] G. De Micheli, "An outlook on design technologies for future integrated systems," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 6, pp. 777–790, 2009.
- [100] S. Hansen, D. Koch, and J. Torresen, "Simulation framework for cycle-accurate rtl modeling of partial run-time reconfiguration in vhdl," in *Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, 2013 8th International Workshop on, pp. 1–8, 2013.
- [101] E. Carpaneto and G. Chicco, "Distribution system minimum loss reconfiguration in the hyper-cube ant colony optimization framework," *Electric Power Systems Research*, vol. 78, no. 12, pp. 2037 – 2045, 2008. Special Issue Papers Presented at the 6th World Energy System Conference 6th World Energy System Conference.

- [102] A. Ukil and W. Siti, "Feeder load balancing using fuzzy logic and combinatorial optimization-based implementation," *Electric Power Systems Research*, vol. 78, no. 11, pp. 1922 – 1932, 2008.
- [103] Y.-T. Hsiao and C.-Y. Chien, "Multiobjective optimal feeder reconfiguration," *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 148, pp. 333–336, Jul 2001.
- [104] K. Wang, S. Skiena, and T. G. Robertazzi, "Phase balancing algorithms," *Electric Power Systems Research*, vol. 96, no. 0, pp. 218 – 224, 2013.
- [105] Y.-Y. Hsu, J.-H. Yi, S. S. Liu, Y. Chen, H. C. Feng, and Y. M. Lee, "Transformer and feeder load balancing using a heuristic search approach," *Power Systems, IEEE Transactions on*, vol. 8, pp. 184–190, Feb 1993.
- [106] J. Zhu, M.-Y. Chow, and F. Zhang, "Phase balancing using mixed-integer programming [distribution feeders]," *Power Systems, IEEE Transactions on*, vol. 13, pp. 1487–1492, Nov 1998.
- [107] J. Zhu, G. Bilbro, and M.-Y. Chow, "Phase balancing using simulated annealing," *Power Systems, IEEE Transactions on*, vol. 14, pp. 1508–1513, Nov 1999.
- [108] M. Dilek, R. Broadwater, J. Thompson, and R. Seqiun, "Simultaneous phase balancing at substations and switches with time-varying load patterns," *Power Systems, IEEE Transactions on*, vol. 16, pp. 922–928, Nov 2001.
- [109] M.-Y. Huang, C.-S. Chen, C. H. Lin, M. S. Kang, H. J. Chuang, and C.-W. Huang, "Three-phase balancing of distribution feeders using immune algorithm," *Generation, Transmission Distribution, IET*, vol. 2, pp. 383–392, May 2008.
- [110] C.-C. Kuo and Y.-T. Chao, "Energy management based on am/fm/gis for phase balancing application on distribution systems," *Energy Conversion and Management*, vol. 51, no. 3, pp. 485 – 492, 2010.
- [111] R. Hooshmand and S. Soltani, "Simultaneous optimization of phase balancing and reconfiguration in distribution networks using bf–nm algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 41, no. 1, pp. 76 – 86, 2012.
- [112] W. Kersting, *Distribution System Modeling and Analysis*, Third Edition. Taylor & Francis, 2012.
- [113] W. Kersting, "Radial distribution test feeders," in *Power Engineering Society Winter Meeting*, 2001. IEEE, vol. 2, pp. 908–912 vol.2, 2001.
- [114] D. Karimipour and F. Salmasi, "Stability analysis of ac microgrids with constant power loads based on popov's absolute stability criterion," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 62, pp. 696–700, July 2015.
- [115] M. Kashem, V. Ganapathy, and G. Jasmon, "A geometric approach for three-phase load balancing in distribution networks," in *Power System Technology, 2000. Proceedings. PowerCon 2000. International Conference on*, vol. 1, pp. 293–298, IEEE, 2000.

- [116] A. Pană, A. Băloi, and F. Molnar-Matei, “Experimental validation of power mechanism for load balancing using variable susceptances in three phase four wire distribution networks,” in EUROCON, 2007. The International Conference on Computer as a Tool, pp. 1567–1572, IEEE, 2007.
- [117] V. Zdraveski, M. Todorovski, and L. Kocarev, “Dynamic intelligent load balancing in power distribution networks,” *International Journal of Electrical Power & Energy Systems*, vol. 73, pp. 157 – 162, 2015.
- [118] T. Reynal, “Load balancing transformer,” Sept. 17 1996. US Patent 5,557,249.
- [119] B. Tavakoli and M. Mojibian, “Automated load balancing for distribution substation feeders,” Jan. 28 2009. EP Patent App. EP20,070,112,916.
- [120] E. Koren, M. Vysotsky, and A. Gusinsky, “Balancing three phase power systems by smooth pahse shifting and clustering,” Aug. 2 2012. US Patent App. 13/358,807.
- [121] L. Polston, L. Ward, and A. Loya, “Device and method for field balancing individual single phase meter loads in a three phase multiple meter stack,” Aug. 10 1999. US Patent 5,936,834.
- [122] S. Guillemin, “Load-balancing device on a polyphase network,” Jan. 2 2014. US Patent App. 13/985,944.
- [123] S. Chen, Y. Foo Eddy, H. Gooi, M. Wang, and S. Lu, “A centralized reactive power compensation system for lv distribution networks,” *Power Systems, IEEE Transactions on*, vol. 30, pp. 274–284, Jan 2015.
- [124] S. Bolognani and S. Zampieri, “A distributed control strategy for reactive power compensation in smart microgrids,” *Automatic Control, IEEE Transactions on*, vol. 58, pp. 2818–2833, Nov 2013.
- [125] S. Bolognani, R. Carli, G. Cavraro, and S. Zampieri, “Distributed reactive power feedback control for voltage regulation and loss minimization,” *Automatic Control, IEEE Transactions on*, vol. 60, pp. 966–981, April 2015.
- [126] J. Klima, J. Skramlik, and V. Valouch, “An analytical modelling of three-phase four-switch pwm rectifier under unbalanced supply conditions,” *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 54, pp. 1155–1159, Dec 2007.
- [127] A. Ipakchi and F. Albuyeh, “Grid of the future,” *IEEE Power and Energy Magazine*, vol. 7, pp. 52–62, March 2009.
- [128] K. Zhou, C. Fu, and S. Yang, “Big data driven smart energy management: From big data to big insights,” *Renewable and Sustainable Energy Reviews*, vol. 56, pp. 215 – 225, 2016.
- [129] H. Long, L. Wang, Z. Zhang, Z. Song, and J. Xu, “Data-driven wind turbine power generation performance monitoring,” *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 6627–6635, Oct 2015.

- [130] Y. Zhang, S. Shen, and J. L. Mathieu, "Data-driven optimization approaches for optimal power flow with uncertain reserves from load control," in 2015 American Control Conference (ACC), pp. 3013–3018, July 2015.
- [131] I. Budinska, V. Oravec, and B. Frankovic, "Central ontology layer for power grid scheduling," in 2006 IEEE International Conference on Mechatronics, pp. 267–271, July 2006.
- [132] Y. Huang and X. Zhou, "Knowledge model for electric power big data based on ontology and semantic web," CSEE Journal of Power and Energy Systems, vol. 1, no. 1, pp. 19–27, 2015.
- [133] R. V. Guha, D. Brickley, and S. Macbeth, "Schema. org: Evolution of structured data on the web," Communications of the ACM, vol. 59, no. 2, pp. 44–51, 2016.
- [134] M. Hepp, "The web of data for e-commerce: Schema. org and goodrelations for researchers and practitioners," in International Conference on Web Engineering, pp. 723–727, Springer, 2015.
- [135] B. Tilahun, T. Kauppinen, C. Keßler, and F. Fritz, "Design and development of a linked open data-based health information representation and visualization system: potentials and preliminary evaluation.," JMIR medical informatics, vol. 2, no. 2, pp. e31–e31, 2013.
- [136] M. Hennessy, C. Oentojo, and S. Ray, "A framework and ontology for mobile sensor platforms in home health management," in Engineering of Mobile-Enabled Systems (MOBS), 2013 1st International Workshop on the, pp. 31–35, IEEE, 2013.
- [137] C. Keller, R. Pöhland, S. Brunk, and T. Schlegel, "An adaptive semantic mobile application for individual touristic exploration," in International Conference on Human-Computer Interaction, pp. 434–443, Springer, 2014.
- [138] I. Toma, C. Stanciu, A. Fensel, I. Stavrakantonakis, and D. Fensel, "Improving the online visibility of touristic service providers by using semantic annotations," in European Semantic Web Conference, pp. 259–262, Springer, 2014.
- [139] C. Ding, M. Wald, and G. Wills, "Open accessibility data interlinking," in International Conference on Computers for Handicapped Persons, pp. 73–80, Springer, 2014.
- [140] V. Zdraveski, K. Mishev, D. Trajanov, and L. Kocarev, "Iso-standardized smart city platform architecture and dashboard," IEEE Pervasive Computing, vol. 16, no. 2, pp. 35–43, 2017.
- [141] "Urbanization prospects." <https://www.un.org/development/desa/en/news/population/world-urbanization-prospects.html>. [Accessed: June 23, 2016].
- [142] Gartner Inc, "6.4 Billion Connected "Things" Will Be in Use in 2016, Up 30 Percent From 2015." <http://www.gartner.com/newsroom/id/3165317>. [Accessed: December 23, 2015].

- [143] R. Kitchin, “The real-time city? big data and smart urbanism,” *GeoJournal*, vol. 79, no. 1, pp. 1–14, 2014.
- [144] H. Bulkeley, P. M. McGuirk, and R. Dowling, “Making a smart city for the smart grid? the urban material politics of actualising smart electricity networks,” *Environment and Planning A*, p. 0308518X16648152, 2016.
- [145] G. McArdle and R. Kitchin, “Improving the veracity of open and real-time urban data,” Available at SSRN, 2015.
- [146] D. Arribas-Bel, K. Kourtit, P. Nijkamp, and J. Steenbruggen, “Cyber cities: social media as a tool for understanding cities,” *Applied Spatial Analysis and Policy*, vol. 8, no. 3, pp. 231–247, 2015.
- [147] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” *IEEE Internet of Things Journal*, vol. 1, pp. 22–32, Feb 2014.
- [148] A. Vakali, L. Anthopoulos, and S. Krco, “Smart cities data streams integration: experimenting with internet of things and social data flows,” in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, p. 60, ACM, 2014.
- [149] A. Sheth, C. Thomas, and P. Mehra, “Continuous semantics to analyze real-time data,” *IEEE Internet Computing*, vol. 14, no. 6, p. 84, 2010.
- [150] A. Meijer and M. P. R. Bolívar, “Governing the smart city: a review of the literature on smart urban governance,” *International Review of Administrative Sciences*, vol. 82, no. 2, pp. 392–408, 2016.
- [151] K. H. Yim, M. C. Ha, C. J. Jo, K.-M. Han, J.-I. Baek, and Y.-U. Ban, “Strategic planning for the smart-green city through urban governance,” *International Journal of Built Environment and Sustainability*, vol. 2, no. 3, 2015.
- [152] L. Berntzen and M. R. Johannessen, “The role of citizen participation in municipal smart city projects: Lessons learned from norway,” in *Smarter as the New Urban Agenda*, pp. 299–314, Springer, 2016.
- [153] V. Albino, U. Berardi, and R. M. Dangelico, “Smart cities: Definitions, dimensions, performance, and initiatives,” *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, 2015.
- [154] “ISO 37120.” http://www.iso.org/iso/catalogue_detail?csnumber=62436. [Accessed: June 23, 2016].
- [155] G. McArdle and R. Kitchin, “The dublin dashboard: Design and development of a real-time analytical urban dashboard.” 2016.
- [156] R. Kitchin, T. P. Lauriault, and G. McArdle, “Knowing and governing cities through urban indicators, city benchmarking and real-time dashboards,” *Regional Studies, Regional Science*, vol. 2, no. 1, pp. 6–28, 2015.

-
- [157] D. Offenhuber and C. Ratti, *Decoding the city: Urbanism in the age of big data*. Birkhäuser, 2014.
- [158] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, “Big data privacy in the internet of things era,” *IT Professional*, vol. 17, no. 3, pp. 32–39, 2015.
- [159] “Prototype application.” <http://skopjeinfo.b1.finki.ukim.mk/>. [Accessed: June 23, 2016].
- [160] C. Walls, *Spring in action 4-th edition*. Shelter Island, NY11964: Manning, 2014.