Okvir za razvoj obitelji poslovnih web aplikacija primjenom ontološkoga modela i generatora programskoga koda

Strmečki, Daniel

Doctoral thesis / Disertacija

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:211:561354

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN

Repository / Repozitorij:

Faculty of Organization and Informatics - Digital Repository

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

Sveučilište u Zagrebu
Fakultet organizacije i informatike
Poslijediplomski doktorski studij “Informacijske znanosti”

DANIEL STRMEČKI

OKVIR ZA RAZVOJ OBITELJI POSLOVNIH WEB APLIKACIJA PRIMJENOM ONTOLOŠKOGA MODELA I GENERATORA PROGRAMSKOGA KODA

DOKTORSKI RAD

Varaždin, 2018

Sveučilište u Zagrebu
Fakultet organizacije i informatike
Poslijediplomski doktorski studij “Informacijske znanosti”

DANIEL STRMEČKI

OKVIR ZA RAZVOJ OBITELJI POSLOVNIH WEB APLIKACIJA PRIMJENOM ONTOLOŠKOGA MODELA I GENERATORA PROGRAMSKOGA KODA

DOKTORSKI RAD

Mentor:
Izv. prof. dr.sc. Ivan Magdalenić

Varaždin, 2018

University of Zagreb
Faculty of Organization and Informatics
Postgraduate doctoral study "Information Sciences"

DANIEL STRMEČKI

FRAMEWORK FOR BUSINESS WEB APPLICATION FAMILIES DEVELOPMENT USING AN ONTOLOGICAL MODEL AND SOURCE CODE GENERATORS

DOCTORAL DISSERTATION

Supervisor:
dr.sc. Ivan Magdalenić

Varaždin, 2018

PODACI O DOKTORSKOM RADU

I. AUTOR

Ime i prezime	Daniel Strmečki
Datum i mjesto rođenja	11.05.1989.
Naziv fakulteta i datum diplomiranja	Fakultet organizacije i informatike, Sveučilište u Zagrebu
Sadašnje zaposlenje	Senior Java Web Developer u ecx.io, an IBM company

II. DOKTORSKI RAD

Naslov	Okvir za razvoj obitelji poslovnih web aplikacija primjenom ontološkoga modela i generatora programskoga koda
Broj stranica, slika, tabela, priloga, bibliografskih podataka	230 stranica, 70 slika, 20 tablica, 10 priloga, 92 citirana rada
Znanstveno područje i polje iz kojeg je postignut akademski stupanj	Društvene znanosti, znanstveno polje informacijske i komunikacijske znanosti
Mentor i voditelj rada	Izv. prof. dr. sc. Ivan Magdalenić
Fakultet na kojem je rad obranjen	Fakultet organizacije i informatike, Sveučilište u Zagrebu
Oznaka i redni broj rada	150

III. OCJENA I OBRANA

Datum sjednice Fakultetskog vijeća na kojoj je prihvaćena tema	27.02.2018.
Datum predaje rada	08.02.2019.
Datum sjednice Fakultetskog vijeća na kojoj je prihvaćena pozitivna ocjena rada	11.12.2018.
Sastav Povjerenstva koje je rad ocijenilo	Prof. dr. sc. Dragutin Kermek, predsjednik Prof. dr. sc. Boris Vrdoljak, član Prof. dr. sc. Daniel Radošević, član Izv. prof. dr. sc. Markus Schatten, zamjena člana
Datum obrane	06.02.2019.
Sastav Povjerenstva pred kojim je rad obranjen	Prof. dr. sc. Dragutin Kermek, predsjednik Prof. dr. sc. Boris Vrdoljak, član Prof. dr. sc. Daniel Radošević, član Izv. prof. dr. sc. Markus Schatten, zamjena člana
Datum promocije	04.05.2019.

SAŽETAK

Beskonačni ciklus ponovnog programiranja različitih obrazaca i modula poslovnih web aplikacija, na nekoliko različitih projekata, naveo je mnoge programere i istraživače da razmotre mogućnosti daljnje automatizacije njihovog razvoja. Kako bi se malim i srednjim poduzećima omogućila dostupnost poslovnog softvera razvijenog po mjeri, potrebno je smanjiti troškove razvoja, a zadržati ili čak povećati razinu kvalitete. Ponovno iskorištavanje postojećih artefakata i podizanje razine apstrakcije pri razvoju pokazali su se učinkovitim pravcima za automatizaciju proizvodnje obitelji softverskih proizvoda. U ovom istraživanju dizajniran je inovativan okvir za razvoj obitelji poslovnih web aplikacija, temeljen na ontološki potpomognutom, generativnom programiranju. Istraživanje je provedeno prema DSRM (engl. *Design Science Research Methodology*) metodologiji za provedbu istraživanja temeljenih na dizajnu. Rad započinje sistematskim pregledom literature s ciljem formiranja baze znanja na ovome području, a zatim slijedi faza dizajna i razvoja koja je rezultirala trima artefaktima: ontološkim modelom, generatorima programskog koda i razvojnim okvirom. Primjena dizajniranog razvojnog okvira demonstrirana je na dvjema studijama slučaja, a analiza rezultata pokazala je da okvir osigurava razvoj na visokoj razini apstrakcije, uz osiguranje visoke razine ponovne iskoristivosti ranije korištenih artefakata.

Ključne riječi: *Okvir, generator, ontologija, web, aplikacija, obitelj*

ABSTRACT

The endless cycle of repetitive coding of various forms and business application modules, on several different projects, has led many developers and researchers wonder about the possibilities for further development automation. In order to enable small and medium-sized businesses to purchase custom-made business software, it is necessary to reduce the cost of development, but maintain or even increase the level of quality. Software reuse and higher abstraction levels are historically proven means for automating the production of software product families. This research proposes a framework for the development of business Web application families, based on an ontology-driven, generative programming approach. The research was conducted according to design science research methodology named *Design Science Research Methodology*. First, a systematic literature review was conducted to constitute a knowledge base in this field. Then, the design and development faze was carried out and it

resulted in three artifacts: an ontological model, code generators and a development framework. The application of the designed framework was demonstrated in two case studies. The analysis of the results showed that the framework provides a higher level of abstraction then the classic development approach and provides high level of reusability of previously used artifacts.

Keywords: *Framework, generator, ontology, web, application, family*

EXTENDED ABSTRACT

A basic goal of software engineering as a discipline is to successfully manage and control software complexity. In the past twenty years, the increase of complexity in software products and high development and maintenance costs have resulted in a large number of unsuccessful projects. With the lack of experienced software developers in the market nowadays, the price of custom-made software went up in recent years. In order to enable small and medium-sized businesses to purchase custom-made business software, it is necessary to reduce the cost of development, but maintain or even increase the level of quality. Software reuse and higher abstraction levels are historically proven means for automating the production of software product families. Software reuse has the potential of increasing its productivity and quality, but numerous factors like complexity, deadlines, budget, technology, architecture and the level of knowledge need to be taken into account. Raising the abstraction level has been the most commonly used software engineering approach to increase software reuse. Researchers are thus constantly looking for higher abstraction levels to enhance productivity and quality of software products. By encapsulating knowledge about lower level operations, developers can think in terms of higher level concepts, thus saving the effort and time of composing lower-level operations. Ontologies were initially used by software applications only to store data and their semantic meaning, but today they are used to aid software development in every phase of its lifecycle. Ontology-driven software engineering is a software development approach where ontologies are used to perform a majority of operations in software development. This research proposes a framework for the development of business Web application families based on a ontology-driven, generative programming approach. The research was conducted according to design science research methodology named *Design Science Research Methodology*. First, a systematic literature review was conducted to constitute a knowledge base in the field. The

review provided with an overview of different approaches and frameworks applying ontologies in automation of software modeling, development and testing. Papers collected in the literature review were grouped by the main disciplines they belong to, including generative programming, service-oriented architectures, model-driven development, multi-agent systems, test automation, internet of things and software product lines. After that, the design and development faze was carried out and it resulted in creation of three artifacts: an ontological model, code generators and a development framework. The ontologies were developed following *Ontology Development 101*, a mature, iterative methodology with a large number of practical guidelines and rules for making decisions when building ontologies. Given that the main goal was to achieve a high level of re-usability and expandability of the initially created ontological model, we created five initial ontologies concerning: requirements, repository, web forms, web components and user interface. Although the initial ontologies are quite simple, we decided for a modular development approach because of the high possibility for their extension in the future (when they are applied to different projects). The code generators were developed in *Java* using *Apache Jena* framework for reading ontological models from RDF / XML files or URLs. The same framework also allows the execution of SPARQL queries to retrieve data stored in an ontological model. This feature is intensively exploited in code generators, as they generate executable code / complete projects based on the data stored in the ontological model. Several additional frameworks were used to simplify and accelerate the code generation process. *Apache Velocity* is a templating framework that enables developers to use a simple template language that enables access to *Java* object references from within the template files. The next framework used is *Vaadin*, because it provides a large number of free, of-the-shelf Web components, which we would need to develop from scratch otherwise. Although code generators themselves do not use any database, the generated applications will use them. Therefore, *Hibernate* framework was used for object-relational mappings and access / storage of data stored in a relational database. *Apache Maven* was used as a build automation tool and *Eclipse Oxygen Java EE* as a development environment. The designed framework is presented in this dissertation trough a set of activities in his development cycle, enriched by UML diagrams. The development cycle of the framework consists of eight activities. The first four activities deal with ontological specification, specifically: project specification, requirements specification, repository specification and form specification. Ontological specification basically means specification of product specific requirements and software artefacts by creating instances in a prepared ontological model. The fifth activity is model validation, which

is an activity carried out by running reasoners on the ontological model. Except the built-in reasoning tools available in ontology editors like *Protégé*, a custom reasoner is a part of the code generators which checks that all required properties for a given instance are provided. The sixth activity is the development of new and specific functionalities. Since every new project or product variant usually brings some new functionality, we need to extend the ontological model and code generators in order to support their automatic code generation. This is the most complicated activity in the development cycle, but also an activity for which related frameworks are usually lacking support. In seventh activity, we generate executable source code, simply by running the prepared code generators. The last activity in the designed framework is to validate the requirements, by manually testing the generated application. The application of the designed framework was demonstrated in this dissertation in two case studies. The case studies include the implementation of applications for management of UBL orders and quotations. Requirements for the two case studies were drawn from official examples downloaded from the OASIS consortium Web page. The implementation implies creating web forms with appropriate user interface elements, input fields, validation and business logic required for calculation, as well as basic functionality such as user login / logout and user rights definition. The analysis of the results showed that the framework provides a higher level of abstraction then the classic development approach, as well as high level of reusability of previously used artifacts. The results confirmed both hypotheses of this doctoral dissertation.

Anđelima koji me čuvaju.

Vaša vjera u mene i sjećanje na vas moja je vječna snaga i motivacija.

Mojoj majci, ocu i bratu.

ZAHVALA

Zaista je teško pronaći riječi za izraziti zahvalnost svima koji su mi pomogli tijekom svih godina školovanja pa i tijekom proteklih nekoliko godina, koliko radim na ovoj doktorskoj disertaciji.

Prije svega želim zahvaliti mentoru, izv. prof. dr. sc. Ivanu Magdaleniću za svu pomoć, strpljenje, savjete i podršku tijekom cijelog vremena pripreme i izrade doktorskog rada. Osim mentora, čiji su zasluge ipak najveće, moram zahvaliti i prof. dr. sc. Danijelu Radoševiću, prof. dr. sc. Dragutinu Kermeku, prof. dr. sc. Nevenu Vrčeku, prof. dr. sc. Vjeranu Strahonji, prof. dr. sc. Dijani Šimić i doc. dr. sc. Zlatku Stapiću na korisnim preporukama i savjetima koje su mi bile od velike pomoći u pripremi doktorskog rada.

Jednako tako želim izraziti zahvalnost instituciji Fakulteta organizacije i informatike, čiji profesori, zaposlenici i studenti su uvijek bili samo fer i korektni prema meni te spremni pomoći u svakoj fazi mog dugogodišnjeg obrazovanja na ovoj instituciji. Uvijek ću s ponosom isticati grad Varaždin i Fakultet organizacije i informatike koji mi je omogućio vrhunsko obrazovanje i pripremu za uspješnu poslovnu karijeru.

Ovo istraživanje nikad ne bi postalo stvarnost, bez ljubavi, podrške i motivacije koju su mi pružili djevojka, obitelj i prijatelji. Rad sam posvetio članovima obitelji koji su nas prerano napustili. Osobito majci Ljubici, kojoj je moje obrazovanje uvijek bila najvažnija stvar na svijetu, ali i ocu Vladimiru te bratu Nevenu. Hvala vam za svu ljubav, podršku i vjeru u mene. Zahvaljujem i dobrim ljudima grada Varaždinu, koji su mi pomogli prebroditi najteže trenutke, osobito zamjenskim roditeljima Mariji, Zlatku i Jagodi. Na kraju, no nikako ne i najmanje važno, hvala mojoj dugogodišnjoj djevojci Sanji na svoj pruženoj ljubavi i strpljenju. Tvoj osmjeh uljepšao mi je sve naporne dane i uz tvoju podršku za mene ništa nije nemoguće.

SADRŽAJ

1. Uvod ... 1
1.1. Područje istraživanja ... 1
1.1.1. Područje i problematika ... 1
1.1.2. Automatsko i generativno programiranje ... 2
1.1.3. Srodne i povezane discipline ... 3
1.1.4. Ontologije i programsko inženjerstvo ... 4
1.1.5. Kategorizacija primjene ontologija ... 5
1.2. Opis problema ... 8
1.2.1. Motivacija i svrha ... 8
1.2.2. Otvorena pitanja ... 9
1.2.3. Srodni pristupi ... 10
1.2.4. Alternativni pristupi ... 11
1.3. Plan istraživanja ... 13
1.3.1. Obuhvat i ograničenja ... 13
1.3.2. Ciljevi i hipoteze ... 14
1.3.3. Struktura rada ... 15
2. Metodologija ... 17
2.1. Istraživanje temeljeno na dizajnu ... 17
2.1.1. Tip znanstvenog istraživanja ... 17
2.1.2. Metodologija istraživanja temeljenog na dizajnu ... 18
2.1.3. Smjernice za provedbu istraživanja ... 19
2.2. Faze istraživanja ... 20
2.3. Tijek i opseg istraživanja ... 22
2.4. Znanstvene metode ... 24
2.5. Očekivani doprinos ... 25
3. Sustavni pregled literature ... 26
3.1. Metoda ... 26
3.2. Pretraga ... 28
3.3. Rezultati ... 29
3.3.1. Odabrani radovi ... 29
3.3.2. Generativno programiranje ... 30
3.3.3. Uslužno-orijentirane arhitekture ... 31
3.3.4. Razvoj temeljen na modelima ... 32
3.3.5. Više-agentni sustavi ... 33
3.3.6. Automatizacija testiranja ... 34

3.3.7. Linije proizvoda u programiranju ... 35
3.3.8. Internet stvari ... 36
3.4. Diskusija ... 37
4. Dizajn i razvoj ... 40
4.1. Ontološki model ... 40
4.1.1. Motivacija ... 40
4.1.2. Metodologija ... 41
4.1.3. Smjernice ... 42
4.1.4. Priprema za razvoj ... 43
4.1.5. Struktura ontologija i pravila imenovanja ... 46
4.1.6. Definiranje klasa i klasne hijerarhije ... 48
4.1.7. Definiranje atributa i njihovih svojstava ... 53
4.2. Generatori programskog koda ... 60
4.2.1. Korištena tehnologija ... 60
4.2.2. Pristup podacima iz modela ... 62
4.2.3. Rad s predlošcima programskog koda ... 64
4.2.4. Web komponente i korisničko sučelje ... 66
4.2.5. Struktura programskog koda ... 67
4.2.6. Dijagram klasa ... 71
4.3. Razvojni okvir ... 74
4.3.1. Primjena okvira ... 74
4.3.2. Povezivanje modela i generatora ... 75
4.3.3. Razvojni ciklus ... 77
4.3.4. Aktivnosti i dionici ... 82
4.3.5. Ontološka specifikacija ... 84
4.3.6. Razvoj novih i specifičnih funkcionalnosti ... 90
5. Demonstracija i evaluacija ... 92
5.1. Studija slučaja 1 ... 92
5.1.1. Primjer UBL narudžbenice ... 92
5.1.2. Specifikacija projekta i korisničkih zahtjeva ... 93
5.1.3. Specifikacija repozitorija ... 97
5.1.4. Specifikacija obrazaca i komponenata ... 100
5.1.5. Validacija modela i razvoj novih funkcionalnosti ... 104
5.1.6. Generiranje programskog koda ... 108
5.1.7. Validacija korisničkih zahtjeva ... 110
5.2. Studija slučaja 2 ... 111

5.2.1. Primjer UBL ponude ... 111
5.2.2. Specifikacija projekta i korisničkih zahtjeva ... 112
5.2.3. Specifikacija repozitorija ... 114
5.2.4. Specifikacija obrazaca i komponenata ... 116
5.2.5. Validacija modela i razvoj novih funkcionalnosti ... 118
5.2.6. Generiranje programskog koda ... 119
5.2.7. Validacija korisničkih zahtjeva ... 121
5.3. Analiza i zaključak ... 122
6. Diskusija i zaključak ... 126
6.1. Pregled ostvarenih ciljeva ... 126
6.2. Ograničenja istraživanja ... 127
6.3. Mogućnosti za buduća istraživanja ... 128
6.4. Komunikacija ... 129
6.5. Zaključak ... 130
Literatura ... 131
Dodatak A – Popis odabranih radova za SLR ... 139
Dodatak B – Ontologija korisničkih zahtjeva ... 143
Dodatak C – Ontologija repozitorija podataka ... 147
Dodatak D – Ontologija web okvira ... 151
Dodatak E – Ontologija web komponenata ... 153
Dodatak F – Ontologija korisničkog sučelja ... 158
Dodatak G – Bazna ontologija sa instancama provedenih studija slučaja ... 160
Dodatak H – Službeni primjer UBL narudžbenice ... 216
Dodatak I – Službeni primjer UBL ponude ... 222
Dodatak J – CD sa izvornim programskim kodom generatora, ontologijama i provedenim studijama slučaja ... 226
Životopis ... 227

POPIS SLIKA

Slika 1: Tijek znanstvenog istraživanja po fazama 21
Slika 2: Broj odabranih radova po godini objave 29
Slika 3: Komponente bazne ontologije 47
Slika 4: Klasna hijerarhija ontologije korisničkih zahtjeva 49
Slika 5: Klasna hijerarhija ontologije repozitorija podataka 49
Slika 6: Klasna hijerarhija ontologije web okvira 50
Slika 7: Klasna hijerarhija ontologije web komponenata 51
Slika 8: Klasna hijerarhija ontologije korisničkog sučelja 51
Slika 9: Hijerarhijska struktura bazne ontologije 52
Slika 10: Postavke projekta generatora programskog koda 62
Slika 11: Skripta za pokretanje generatora ontološke sheme 62
Slika 12: Generirana shema ontologije korisničkog sučelja 63
Slika 13: Isječak programskog koda sa SPARQL upitom 64
Slika 14: Isječak *Velocity* predloška za klasu entiteta 65
Slika 15: Metoda za generiranje *Java* klase entiteta pomoću *Velocity* predloška 66
Slika 16: Isječak *Velocity* predloška za klasu obrasca 67
Slika 17: Dijagram paketa generatora programskog koda 68
Slika 18: Organizacija programskog koda u *Java* pakete i klase 69
Slika 19: Korišteni resursi - statične datoteke i predlošci koda 70
Slika 20: Dijagram klasa paketa za učitavanje podataka 71
Slika 21: Dijagram klasa paketa za reprezentaciju podataka 72
Slika 22: Dijagram klasa paketa generatora programskog koda 73
Slika 23: Povezivanje ontološkog modela i generatora programskog koda u okvir za razvoj web aplikacija 76
Slika 24: Primjer prijave greške u *Protege*-u pomoću *FaCT++* argumentatora 79
Slika 25: Primjer prijave greške u *Eclipse*-u pomoću argumentatora ugrađenog u generatore 80
Slika 26: Dijagram aktivnosti razvojnog ciklusa primjene okvira 81
Slika 27: Dijagram slučajeva korištenja razvojnog okvira 83
Slika 28: Atributi instance korisničkog zahtjeva 85
Slika 29: Atributi instance korisničke priče 85
Slika 30: Atributi instance kriterija prihvaćanja 85
Slika 31: Atributi instance repozitorija 86
Slika 32: Atributi instance tablice u bazi podataka 86
Slika 33: Atributi instance stupca u bazi podataka 87

Slika 34: Atributi instance grupe obrazaca 88
Slika 35: Atributi instance web obrasca 88
Slika 36: Atributi instance web komponente za unos podataka 88
Slika 37: Atributi instance web komponente gumba 89
Slika 38: Generirani obrazac za upravljanje artiklima 89
Slika 39: Detaljan dijagram aktivnosti specifikacije obrazaca te razvoja novih i specifičnih funkcionalnosti 91
Slika 40: Atributi instance projekta studije slučaja 1 93
Slika 41: Atributi instance korisničkog zahtjeva za studiju slučaja 1 96
Slika 42: Atributi instance korisničke priče za studiju slučaja 1 97
Slika 43: Atributi instance kriterija prihvaćanja za studiju slučaja 1 97
Slika 44: Atributi instance repozitorija studije slučaja 1 98
Slika 45: Atributi instance tablice u bazi podataka za studiju slučaja 1 99
Slika 46: Atributi instance stupca tablice za studiju slučaja 1 100
Slika 47: Atributi instance grupe obrazaca za studiju slučaja 1 103
Slika 48: Atributi instance obrasca za studiju slučaja 1 104
Slika 49: Atributi instance datumskog polja za studiju slučaja 1 104
Slika 50: Instanca tablice sa paginacijom za studiju slučaja 1 106
Slika 51: Isječak programskog koda klase koja reprezentira unosno polje 106
Slika 52: Isječak programskog koda predloška za postavljanje vrijednosti unosnih polja 107
Slika 53: Isječak programskog koda predloška za renderiranje unosnih polja 107
Slika 54: Ispis poruka u *Eclipse* konzoli prilikom generiranja studije slučaja 1 108
Slika 55: Pregled generiranih obrazaca i broja linija koda studije slučaja 1 109
Slika 56: Uređivanje narudžbenice u web pregledniku 110
Slika 57: Atributi instance projekta studije slučaja 2 112
Slika 58: Atributi instance korisničkog zahtjeva za studiju slučaja 2 114
Slika 59: Atributi instance repozitorija studije slučaja 2 115
Slika 60: Atributi instance tablice u bazi podataka za studiju slučaja 2 115
Slika 61: Atributi instance stupca u bazi podataka za studiju slučaja 2 116
Slika 62: Atributi instance grupe obrazaca za studiju slučaja 2 117
Slika 63: Atributi instance obrasca za studiju slučaja 2 118
Slika 64: Atributi instance onemogućenog polja za studiju slučaja 2 118
Slika 65: Ispis poruka u Eclipse konzoli prilikom generiranja studije slučaja 2 119
Slika 66: Pregled generiranih obrazaca i broja linija koda studije slučaja 2 120
Slika 67: Dodavanje nove stavke ponudi u web pregledniku 121
Slika 68: Priprema za analizu ontološke specifikacije 122
Slika 69: Statička analiza generiranog programskog koda za studiju slučaja 1 124

Slika 70: Statička analiza generiranog programskog koda za studiju slučaja 2 124

POPIS TABLICA

Tablica 1: Osnovni pojmovi u ontologiji 45
Tablica 2: Podatkovni atributi ontologije korisničkih zahtjeva 53
Tablica 3: Objektni atributi ontologije korisničkih zahtjeva 54
Tablica 4: Podatkovni atributi ontologije repozitorija podataka 54
Tablica 5: Objektni atributi ontologije repozitorija podataka 55
Tablica 6: Podatkovni atributi ontologije web okvira 56
Tablica 7: Objektni atributi ontologije web okvira 56
Tablica 8: Podatkovni atributi ontologije web komponenata 57
Tablica 9: Objektni atributi ontologije web komponenata 57
Tablica 10: Podatkovni atributi ontologije korisničkog sučelja 58
Tablica 11: Objektni atributi bazne ontologije 58
Tablica 12: Korisnički zahtjevi za studiju slučaja 1 94
Tablica 13: Potrebne tablice u bazi podataka za realizaciju studije slučaja 1 98
Tablica 14: Obrasci i komponente za studiju slučaja 1 100
Tablica 15: Podatkovni atributi dodani u ontološki model 105
Tablica 16: Novi korisnički zahtjevi za studiju slučaja 2 113
Tablica 17: Nove tablice u bazi podataka za realizaciju studije slučaja 2 115
Tablica 18: Novi obrasci i komponente za studiju slučaja 2 117
Tablica 19: Analiza ontološke specifikacije 123
Tablica 20: Statička analiza generiranog programskog koda 123

POPIS KRATICA

API	Aplikacijsko programsko sučelje (engl. *Application Programming Interface*)
BPEL	Jezik izvršavanja poslovnih procesa (engl. *Business Process Execution Language*)
CBD	Razvoj temeljen na komponentama (engl. *Component-Based Development*)
CD	Kompaktna ploča (engl. *Compact Disc*)
CIM	Računalno neovisan model (engl. *Computation Independent Model*)
CSS	Kaskadni stilski jezik (engl. *Cascading Style Sheet*)
DSL	Specifičan jezik domene (engl. *Domain Specific Language*)
DSRM	Metodologija istraživanja temeljenog na dizajnu (engl. *Design Science Research Methodology*)
GPL	Opći programski jezik (engl. *General Purpose Language*)
GUI	Grafičko korisničko sučelje (engl. *Graphical User Interface*)
HTML	Jezik oznaka hiper-teksta (engl. *HyperText Markup Language*)
IDE	Integrirano razvojno okruženje (engl. *Integrated Development Environment*)
IoT	Internet stvari (engl. *Internet of Things*)
JVM	Java virtualni stroj (engl. *Java Virtual Machine*)
MAS	Više-agentni sustavi (engl. *Multi Agent Systems*)
MDD	Razvoj temeljen na modelima (engl. *Model-Driven Development*)
ODD	Razvoj temeljen na ontologijama (engl. *Ontology-Driven Development*)
OWL	Ontološki jezik weba (engl. *Ontology Web Language)*
ORM	Objektno-relacijsko mapiranje (engl. *Object-Relational Mapping*)
PIM	Neovisni model platforme (engl. *Platform Independent Model*)
POJO	Jednostavan *Java* objekt (engl. *Plain Old Java Object*)
PSM	Specifični model platforme (engl. *Platform Specific Model*)
RDF	Okvir za opis resursa (engl. *Resource Description Framework*)
SLR	Sustavni pregled literature (engl. *Systematic Literature Review*)
SOA	Uslužno-orijentirana arhitektura (engl. *Service-Oriented Architecture*)
SOAP	Jednostavan protokol za pristup objektima (engl. *Simple Object Access Protocol*)
SPL	Linije proizvoda u programiranju (engl. *Software Product Lines*)

SPARQL	Standardni jezik upita i protokol za RDF (engl. *SPARQL Protocol and RDF Query Language*)
SQL	Strukturirani jezik upita (engl. *Structured Query Language*)
SWS	Semantičke web usluge (engl. *Semantic Web Services*).
TA	Automatizacija testiranja (engl. *Test Automation*)
TDD	Razvoj temeljen na testiranju (engl. *Test-Driven Development*)
TURTLE	Sažeti jezik za opis RDF trojaka (engl. *Terse RDF Triple Language*)
UBL	Univerzalni poslovni jezik (engl. *Universal Business Language*)
UDDI	Univerzalni opis, otkrivanje i integracija (engl. *Universal Description, Discovery and Integration*)
UI	Korisničko sučelje (engl. *User Interface*)
UML	Unificirani jezik modeliranja (engl. *Unified Modeling Language*)
URL	Jedinstvena lokacija resursa (engl. *Uniform Resource Locator*)
W3C	Konzorcij za svjetsku mrežu weba (engl. *World Wide Web Consortium)*
WSDL	Jezik za opis web usluga (engl. *Web Services Description Language*)
XML	Proširivi jezik oznaka (engl. *eXtensible Markup Language*)

1. UVOD

1.1. Područje istraživanja

1.1.1. Područje i problematika

Disciplina kojom se bavi ovaj doktorski rad je programsko inženjerstvo. Inženjerstvo, općenito, je disciplina koja se bavi metodologijama učinkovitim kod izgradnje kompleksnih i velikih sustava. Programsko inženjerstvo jest primjena inženjerskog pristupa pri razvoju programskih proizvoda. No programsko inženjerstvo ne bavi se samo razvojem programskih proizvoda, već i njegovim radom, performansama i održavanjem. Stoga općeprihvaćena definicija programskog inženjerstva glasi: *Programsko inženjerstvo je primjena sustavnog, disciplniranog, kvantificiranog pristupa razvoju, izvođenju i održavanju programskog proizvoda* (Bourque & Fairley, 2014; Happel & Seedorf, 2006). S obzirom da se inženjerske discipline bave kompleksnim rješenjima, upravo je kontrola kompleksnosti programskih proizvoda jedan od glavnih zadataka programskog inženjerstva. Složenost programskih proizvoda te visoki troškovi razvoja i održavanja rezultirali su značajnim brojem neuspješnih projekata u posljednjih dvadesetak godina. Neki od ključnih problema u razvoju programskih proizvoda su niska razina tehnika dizajniranja i implementacije te izlaganje više detalja nego što je inicijalno bilo planirano (Batory, 2006; Robal, Ojastu, Kalja, & Jaakkola, 2015). Čak i kada upotrijebljena tehnologija razvoja sama po sebi nije komplicirana, inženjering i realizacija projekta obično jesu. Glavni uzrok tome su sposobnost i iskustvo ljudi koji obavljaju razvojne aktivnosti, velik broj uključenih strana te različite interpretacije sustava od strane različitih sudionika (Musen, 2000; van Ruijven, 2013).

Ponovno iskorištavanje artefakata i podizanje razine apstrakcije pri razvoju, sredstva su kojima se disciplina programskog inženjerstva služi kroz cijelu svoju povijest, s ciljem rješavanja navedenih problema. Još od ranih početaka discipline, istraživači i inženjeri nastoje povećati produktivnosti razvoja programskih proizvoda traženjem viših razina apstrakcije. Upravo podizanje razine apstrakcije omogućuje inženjerima i programerima da razmišljaju o konceptima viših razina te jednostavno koriste dostupne funkcionalnosti nižih razina (Visser, 2008). Osim podizanja razine apstrakcije, ponovno iskorištavanje programskih komponenata i

drugih artefakata korištenih u razvoju pokazali su se kao jedan od glavnih doprinosa povećanju kvalitete programskih proizvoda posljednjih godina (García Coria, Castellanos-Garzón, & Corchado, 2014). Unatoč tome, današnje moderno tržište zahtijeva sve brži razvoj programskih proizvoda, ali uz očuvanje visoke razine kvalitete te što je moguće jednostavnije održavanje. Upravo su velika potražnja modernog tržišta i znanstvena znatiželja glavna motivacija istraživačima da i danas nastave konstantnu potragu za metodama i alatima koje će omogućiti razvoj na višim razinama apstrakcije, uz što je moguće višu razinu ponovne iskoristivosti ranije korištenih artefakata (Strmečki, Magdalenić, & Radosević, 2018).

1.1.2. Automatsko i generativno programiranje

Automatsko programiranje je pod-disciplina u sklopu discipline programskog inženjerstva koja se bavi automatizacijom procesa nižih razina, kako bi omogućila programerima rad na višim razinama apstrakcije. Inicijalno je problematika automatskog programiranja bila razmatrana kao problem kompiliranja (specifikacija kompilirana u implementaciju), međutim danas ova disciplina razvija kompleksne modele koji presijecaju poslovnu domenu i domenu programskog inženjerstva. Radi se o zreloj disciplini koja je kroz povijest omogućila razvoj od strojnog koda, preko asemblera, do programskih jezika visokih razina te značajno utjecala na povećanje produktivnosti programera (Barstow, 1984; Rich & Waters, 1988). Iako se ova disciplina primjenjuje već više od 60 godina u programskom inženjerstvu, niti danas ne postoje univerzalna rješenje za automatsko generiranje programskog koda. Međutim, unutar nekih poslovnih domena, danas je moguće ostvariti vrlo visoku razinu automatizacije. Formalni modeli i izvršive specifikacije neki su od modernih pristupa automatskom programiranju koji i dalje nastoje ostvariti osnovni cilj ove discipline, a to jest smanjenje ljudskog napora pri razvoju programskih proizvoda (Insaurralde, 2013).

Generativno programiranje je još jedna pod-disciplina automatskog programiranja koja primjenjuje generatore programskog koda kako bi ubrzala i olakšala proces razvoja programskih proizvoda. Generator možemo definirati kao softver koji prima ulaznu specifikaciju visoke razine te iz nje generira implementaciju programskog proizvoda. Generativno programiranje stavlja fokus na mapiranje između prostora problema i prostora rješenja. Prostor problema odnosi se na skup značajki obitelji programskih proizvoda, a prostor rješenja podrazumijeva implementaciju apstrakcija sadržanih u specifikaciji (Czarnecki & Eisenecker, 2000; Magdalenić, Radošević, & Orehovački, 2013). Generativno programiranje primjenjuje tehnike meta-programiranja, tj. dizajn i razvoj softvera koji čitaju, generaliziraju, analiziraju i transformiraju sami sebe ili drugi softver. Tehnike generativnog programiranja

često se koriste i drugim pod-disciplinama programskog inženjerstva poput linija proizvoda u programiranju i razvoja temeljenog na modelima (Strmečki, Magdalenić, & Kermek, 2016).

1.1.3. Srodne i povezane discipline

Linije proizvoda u programiranju (SPL) ili obitelji programskih proizvoda podrazumijevaju skup programskih proizvoda koji se sastoje od zajedničke arhitekture i skupa ponovno iskoristivih artefakata koji se koriste u sustavnoj proizvodnji (Asikainen, Männistö, & Soininen, 2007). Ova disciplina fokus stavlja na sustavnu, strateški planiranu ponovnu iskoristivost artefakata za proizvodnju nekoliko sličnih programskih proizvoda. Osnovni cilj discipline jest izbjegavanje razvoja programskih proizvoda od nule primjenom konfiguracije i ponovne iskoristivosti artefakata na više projekata. SPL sadrže dva razvojna procesa, inženjering domene i inženjering aplikacije. Cilj domenskog inženjeringa je razvoj zajedničke arhitekture obitelji proizvoda te osmišljavanje plana proizvodnje. Inženjering aplikacije bavi se razvojem pojedinačnog proizvoda čiji se specifični zahtjevi uzimaju u obzir i definiraju se mehanizmi upravljanja varijabilnošću kako bi se proizvod razvio pravovremeno i kvalitetno, ali uz najmanji mogući trošak (Magdalenić et al., 2013; T. Nguyen, Colman, & Han, 2015). SPL zahtijeva značajna ulaganja unaprijed, ali pridonosi smanjenju vremena razvoja i povećanju kvalitetu obitelji programskih proizvoda. Istraživanja su pokazala kako više od 50% industrijskih stručnjaka provodi SPL tek nakon iniciranja nekoliko sličnih varijanti proizvoda koristeći *ad-hoc* tehnike ponovnog iskorištavanja, poput dupliciranja te zatim modificiranja programskog koda (Martinez, Ziadi, Bissyandé, Klein, & Le Traon, 2015).

Razvoj temeljen na komponentama (CBD) pod-disciplina je programskog inženjerstva koja nastoji osigurati primjenu ponovno iskoristivih komponenata u *priključi i koristi* (engl. *plug and play*) stilu razvoja. Radi se o pristupu temeljenom na ponovnoj upotrebi postojećih artefakata, tj. komponenata. Kako bi CBD bio efikasan komponente koje programeri razvijaju moraju biti nezavisne i labavo povezane. Idealni CBD podrazumijeva izgradnju kompleksnih sustava integracijom i povezivanjem već razvijenih i dostupnih komponenata (Ringert, Rumpe, & Wortmann, 2014). Ponovno iskorištavanje komponenata i povezanih artefakata povećava pouzdanosti i produktivnost programskih proizvoda, jer se pri razvoju oslanjamo na već testirane tehnike i komponente. Ponovno iskorištavanje komponenata jedan je od najvažnijih pristupa koji su pridonijeli razvoju kvalitetnijih programskih proizvoda u posljednjih dvadesetak godina (García Coria et al., 2014).

Razvoj temeljen na modelima (MDD) također je pod-disciplina programskog inženjerstva koja naglasak stavlja na apstrakciju temeljenu na modelima i automatsko generiranje programskog koda. U MDD razvoju bitne značajke sustava bilježe se pomoću odgovarajućih modela, a generatori koda se primjenjuju za automatsko generiranje programskog koda iz tih modela (Pahl, 2007b). Modeli predstavljaju djelomičan ili pojednostavljeni prikaz sustava te se u MDD ne koriste samo kao dokumentacija, već kao ponovno iskoristivi artefakti korišteni kroz cijeli životni ciklus programskog proizvoda (Rodrigues da Silva, 2015). MDD je temeljen na tri sloja: sloj modela koji obrađuje strukturne aspekte sustava, sloj modela koji definira sustav neovisno od platforme i sloj modela specifičnog za konkretnu implementaciju. Iako je MDD po definiciji neovisan od platforme, tipični MDD korišten u praksi temelji se na unificiranom jeziku modeliranja (UML) (Katasonov & Palviainen, 2010; Pahl, 2007a). Programeri koji primjenjuju MDD najprije modeliraju kompletan sustav u UML-u, a zatim iterativno poboljšavaju i dograđuju model. Konačni model treba biti dovoljno konkretan da se iz njega može direktno generirati izvršivi programski kod. S obzirom da kod ovog pristupa programeri rade na visokoj razini apstrakcije, postiže se visoka efikasnost, no vrlo je teško primijeniti MDD za programiranje aplikacija opće namjene, jer generatori postaju previše kompleksni (Zimmer & Rauschmayer, 2004).

1.1.4. Ontologije i programsko inženjerstvo

Ontologije su svoj prodor u disciplinu programskog inženjerstva ostvarile preko discipline umjetne inteligencije. Unutar discipline umjetne inteligencije ontologije se primjenjuju za ograničavanje složenosti i organizaciju informacija. Općeprihvaćena definicija ontologija u računalnim znanostima jest da je ontologija formalna i eksplicitna specifikacija zajedničke konceptualizacije. Konceptualizacija podrazumijeva apstraktnu i pojednostavljenu verziju prezentiranog svijeta, odnosno reprezentaciju znanja temeljenu na objektima, konceptima i entitetima. Formalna znači da je računalo može procesuirati, a eksplicitna da postoje jasna ograničenja koja se primjenjuju na reprezentacije. Ontologiju u programskom inženjerstvu možemo definirati kao sredstvo za formalizaciju iscrpne i rigorozne konceptualne sheme unutar određene domene. Pojednostavljeno, ontologija je hijerarhijska struktura podataka koja sadrži relevantne elemente unutar domene te njihove međusobne veze i pravila (C Calero, Ruiz, & Piattini, 2006; P. Wongthongtham, Kasisopha, Chang, & Dillon, 2008; Pornpit Wongthongtham, Chang, & Dillon, 2007). Autori *Dillon et al.* navode kako ontologije ne bi

trebale sadržavati samo hijerarhijske koncepte, već i semantičke odnose koji specificiraju kako su koncepti povezani jedan s drugim. Isti autori također su nagovijestili da će primjena semantike kao jednog od osnovnih mehanizama programskog inženjerstva revolucionarizirati način razvoja programskih proizvoda (Dillon, Chang, & Wongthongthain, 2008).

Ontologije čine skup ponovno iskoristivih, djeljivih resursa znanja te predstavljaju posebnu vrstu artefakta u programskom inženjerstvu. Ontologije uključuju određen pogled na svijet, dizajniran s ciljem eksplicitnog izražavanja značenja skupa postojećih objekata (John, 2010). Posljednjih desetak godina, s pojavom semantičkog weba, predstavljeno je nekoliko ontoloških jezika, no jedan je postao standard Konzorcija za svjetsku mrežu weba (W3C). Radi se o ontološkom jeziku weba (OWL) koji pruža set izraza za definiranje koncepata i odnosa koji se javljaju unutar ontologija. OWL je razvijen s ciljem povećanja razine računalne interpretacije ljudskog znanja. Razvijen je kao proširenje rječnika okvira za opis resursa (RDF), a izveden je iz starije varijante ontološkog jezika naziva DAML+OIL (Bossche, Ross, MacLarty, Van Nuffelen, & Pelov, 2007; Pornpit Wongthongtham, Chang, Dillon, & Sommerville, 2009).

1.1.5. Kategorizacija primjene ontologija

Inicijalno, ontologije su korištene u programskom inženjerstvu samo za pohranu podataka i njihovog semantičkog značenja. Međutim, danas se ontologije koriste kao primarni ili sekundarni artefakti pri razvoju softvera u svakoj fazi njegovog životnog ciklusa. Ontološki upravljano programsko inženjerstvo je disciplina u kojoj se ontologije koriste za obavljanje većine operacija u razvoju softvera, od modeliranja sustava do generiranja programskog koda (Wiebe & Chan, 2012). Autori *Guarino* i *Fensel* klasificirali su ontologije na temelju njihove opće razine:

1) Generičke ontologije – obuhvaćaju opće znanje svijeta i primjenjive su u različitim domenama;
2) Reprezentativne ontologije – pružaju entitete bez izražavanja onoga što predstavljaju i ne pripadaju ni jednoj određenoj domeni;
3) Domenske ontologije – obuhvaćaju znanje koje se primjenjuju u određenoj domeni;
4) Ontologije metoda i zadataka – obuhvaćaju znanje specifično za metode rješavanja određenog problema ili zadatka (C Calero et al., 2006; Strmečki et al., 2016).

Autori *Calero et al* definirali su nešto širu klasifikaciju ontologija na temelju njihovog predmeta konceptualizacije:

1) Ontologije prikaza znanja – koriste se za formaliziranje znanja pod konkretnom paradigmom;
2) Uobičajene ili generičke ontologije – predstavljaju ponovno iskoristivo generičko znanje;
3) Ontologije visoke razine – opisuju opće pojmove;
4) Domenske ontologije – pružaju vokabular za koncepte u određenoj domeni;
5) Ontologije zadatka – opisuju vokabular koji se odnosi na generičku aktivnost;
6) Domenske ontologije zadataka – mogu se ponovno upotrijebiti samo u određenoj domeni;
7) Ontologije metoda – primjenjive su na proces zaključivanja dizajniran za obavljanje određenog zadatka;
8) Ontologije aplikacija - ovise o aplikaciji i često specijaliziraju vokabular domenske ili ontologije zadatka.

Isti autori naglašavaju da, na temelju trenutka kada se koriste, ontologije se mogu podijeliti na one koje se koriste tijekom razvoja ili tijekom izvođenja. Njihova konačna taksonomija ontologija u programskom inženjerstvu sadrži dvije generičke kategorije:

1) Ontologije domene – generičke ili specifične: zahtjevi, dizajn, testiranje, održavanje, konfiguracija, upravljanje, kvaliteta, alati i metode;
2) Ontologije kao programski artefakti – ontologije korištene u razvoju ili ontologije korištene u izvođenju programskog proizvoda (C Calero et al., 2006).

Happel i Seedorf iste su godine kao i *Calero et al.* prepoznali integraciju pristupa programskog inženjerstva i inženjerstva znanja te dokumentirali primjere primjene ontologija kroz cijeli životni ciklus programskog proizvoda: analiza i dizajn (specifikacija zahtjeva i ponovna upotreba komponenata), razvoj (modeliranje proizvoda, modeliranje domene, podrška programiranju i dokumentacija), implementacija i izvođenje (semantički web, web usluge) te održavanje (projektna podrška, testiranje). Navedeni autori kategorizirali su primjenu ontologija u programskom inženjerstvu u četiri osnovna pristupa:

1) Razvoj upravljan ontologijama – ontologije se primjenjuju tijekom razvoja programskog proizvoda za opisivanje problemske domene;

2) Razvoj omogućen ontologijama – ontologije se primjenjuju tijekom razvoja za podršku programerima prilikom obavljanja svakodnevnih zadataka;
3) Arhitekture temeljena na ontologijama – ontologije se primjenjuju kao osnovni artefakti prilikom izvođenja programa;
4) Arhitekture omogućene ontologijama – ontologije se primjenjuju kao pomoćni artefakti prilikom izvođenja programa (Happel & Seedorf, 2006).

Autori *Gašević et al*. nekoliko godina kasnije također su istražili primjenu ontologija u različitim fazama životnog ciklusa programskih proizvoda. Njihovi rezultati pokazali su da se u fazi analize ontologije najčešće primjenjuju za inženjerstvo zahtjeva. U fazi dizajna ontologije se koriste kao modeli programskog proizvoda, poslovnog vokabulara, zaključivanja ili transformacije. U fazi implementacije navedeni autori definiraju tri moguće primjene ontologija:

1) Implementacija programskog proizvoda može se generirati iz ontologije kreirane u fazi analize i dorađene u fazi dizajna;
2) Ontologije se mogu koristiti u izvođenju, za rukovanje OWL ontologija u konkretnom programskom jeziku;
3) Ontologije se mogu koristiti kao dio implementacijske logike u sustavima implementiranim pomoću jezika baziranih na pravilima.

U fazi održavanja programskog proizvoda ontologije se mogu koristiti kao podrška upravljanju znanjem, a moguća je i primjena ontologija u fazi odumiranja ili umirovljivanja programskog proizvoda kao spremišta znanja (Gašević, Kaviani, & Milanović, 2009).

1.2. Opis problema

1.2.1. Motivacija i svrha

Motivacija za provedbu ovog istraživanja proizašla je iz prakse. Današnje moderno tržište programskih proizvoda zahtijeva ubrzani razvoj kvalitetnog softvera po mjeri, najčešće za poslovne potrebe. Osobito se to reflektira na mala i srednja poduzeća koja često imaju poslovne potrebe za razvojem programskih proizvoda po mjeri, ali si isti ne mogu priuštiti zbog visokih troškova razvoja i održavanja. S obzirom da je trenutno potražnja za poslovnim softverom velika, a na tržištu nema dovoljan broj iskusnih programera, informatičke tvrtke često nisu niti zainteresirane za razvoj manjih i jednostavnijih rješenja, prilagođenih malim i srednjim poduzećima. Programeri se pak, u razvoju poslovnog softvera za web, često susreću s velikim brojem ponavljajućih zadataka. Upravo ta činjenica motivira mnoge iskusne programere i istraživače na području programskog inženjerstva da razmotre mogućnosti daljnje automatizacije razvoja softvera, osobito obitelji programskih proizvoda.

Ponovno iskorištavanje postojećih artefakata i podizanje razine apstrakcije pri razvoju pokazali su se učinkovitim pravcima za automatizaciju proizvodnje obitelji programskih proizvoda. Primjena metoda generativnog programiranja za automatizaciju razvoja SPL-ova ima smisla kada postoji značajna sličnost među većim brojem programskih proizvoda. Upravo je to slučaj i sa značajnim brojem poslovnog web softvera za manja i srednja poduzeća. Iako su proizvodi sami po sebi različiti iz perspektive krajnjeg korisnika, tehnologija izrade, korištene komponente i metoda razvoja vrlo su slične ili čak jednake iz perspektive programera. Stoga je glavna svrha ovog istraživanja postići višu razine ponovne iskoristivosti artefakata ranije razvijenih na drugim projektima. Cilj je i općenito podići razinu apstrakcije pri razvoju obitelji poslovnih web aplikacija, kako bi se što je moguće više smanjio broj ponavljajućih programerskih zadataka.

Moderni objektno-orijentirani programski jezici poput *Java-e* i *C#,* kakve danas većinom koristimo za razvoj web softvera za poslovne potrebe, već su visokih razina apstrakcije, ali radi se o jezicima općenite namjene (GPL). Kako bi postigli još više razine apstrakcije pri razvoju softvera u konkretnoj domeni, primjenjujemo razne tehnike modeliranja i meta-programiranja. Meta-programiranje označava sposobnost programskog proizvoda da čita, generira, analizira ili transformira druge programske proizvode ili sam sebe (Strmečki, Radošević, & Magdalenić, 2015). Obično se kao standardna tehnika modeliranja programskih proizvoda danas primjenjuje UML, iz kojeg se direktno generira programski kod u MDD pristupu razvoja. Često korištena alternativa MDD-u je razvoj i primjena domenski specifičnih jezika (DSL), koji se kasnije obično prevode u neki GPL. U ovome doktorskom radu istražiti ćemo drugačiji pristup,

odnosno mogućnost primjene ontologija za modeliranje i automatsko generiranje programskog koda obitelji poslovnih web aplikacija. Ontologije su već duže vrijeme prepoznate kao prikladan način za opisivanje i organiziranje znanja unutar određene domene, no njihova primjena u programskom inženjerstvu i dalje je većinom samo akademska. Nedostaju standardi, okviri, metode i alati koji bi, kroz visoku razinu ponovne iskoristivosti ontološkog znanja, omogućili isplativost ulaganja u ontološki-upravljan razvoj softvera (Strmečki et al., 2016).

1.2.2. Otvorena pitanja

U jednom od najcitiranijih radova na ovome području, autori *Happel i Seedorf* navode kako pristupi integraciji metoda programskog inženjerstva i inženjerstva znanja imaju tendenciju biti akademski te zanemaruju aspekte primjenjivosti, odnosno ne pružaju dovoljno smjernica programerima i inženjerima (Happel & Seedorf, 2006). Posljednjih nekoliko godina, ontologije se najčešće spominju u kontekstu uslužno-orijentiranih arhitektura (SOA) i slabo je istražena njihova primjena za modeliranje i generiranje stolnih, web ili mobilnih aplikacija sa grafičkim korisničkim sučeljem (Strmečki et al., 2016). Uporaba semantičkih informacija danas dobiva sve više na važnosti za razvoj aplikacija i podršku interoperabilnosti između različitih aplikacija. Ontologije se sve češće primjenjuju kao OWL domenski modeli, ali takve modele i danas je teško iskoristiti u mnogim programskim jezicima (Poggi, 2011). Ipak, posljednjih par godina integracija ontologija i objektno-orijentiranih jezika je značajno napredovala, pa se npr. *Apache Jena* biblioteka može koristiti za pristupanje informacijama spremljenim u ontološkim modelima iz *Java* programskog koda. Primjenom takvih biblioteka te raznih modernih web razvojnih okvira i alata moguće je generirati poslovne aplikacije direktno iz ontološke specifikacije. Međutim, primjena ontologija pri razvoju programskih proizvoda, kao i primjena svake druge nove tehnologije ili tehnike, zahtijeva određene investicije. Kako bi se takva investicija isplatila, potrebno je osigurati visoku razinu ponovne iskoristivosti ontološkog znanja na više projekata (Strmečki & Magdalenić, 2016). Razvoj softvera složeni je proces koji zahtijeva sustavne pristupe modeliranju, razvoju i održavanju, kao i adekvatnu podršku u smislu pomoćnih alata. Razvojni procesi u programskom inženjerstvu poprilično su složeni i kao takvi vrlo su dobro istraženi te danas postoje definirani standardi. Primjena ontologija pri razvoju softvera stoga zahtijeva standardizaciju samih ontologija, ali i integraciju ontologija u postojeće razvojne paradigme, kao i razvoj adekvatne programske podrške (Strmečki et al., 2018).

1.2.3. Srodni pristupi

Mogućnost primjene ontologija u programskom inženjerstvu istražuje se već 15-ak godina, stoga je jasno kako je u literaturi moguće pronaći nekoliko srodnih pristupa, odnosno rješenja koja primjenjuju ontologije za generiranje programskog koda poslovnih web aplikacija. U ovome poglavlju spomenuti ćemo tri pristupa najsličnija pristupu predloženom u ovom doktorskom radu te istaknuti njihove prednosti i mogućnosti za poboljšanje.

U pristupu nazvanom *ODASE* (engl. *Ontology Driven Architecture for Software Engineering*) autori *Bossche et al.* primijenili su ontološki potpomognuto generiranje koda na projektu e-osiguranja. Osnovna motivacija za provedbu istraživanja bio je jaz između programera i poslovnih korisnika, uzrokovan neformalnom i nepotpuno dokumentiranom vizijom projekta od strane poslovnog sustava. Fokus istraživanja stavljen je na formalizaciju korisničkih zahtjeva upotrebom ontologija, s ciljem postizanja jasnog dogovora između poslovnih korisnika i razvojnih timova. Upravo je ontologija korištena kao sredstvo osiguranja razmjene znanja i međusobnog razumijevanja. Na prikazanom projektu autori su koristili OWL, *Mercury* logički programski jezik te internu platformu nazvanu *Hedwig*. Interna platforma predstavlja set alata i biblioteka koji omogućuju integraciju ontologija i poslovnog softvera (Bossche et al., 2007). Argumentiran je niz prednosti koje donosi primjena ontologija u automatskom programiranju, no nejasno je koje ontologije su korištene, kako su međusobno povezane te kako je izgledao kompletni razvojni proces temeljen na ovom pristupu. Sam razvoj temeljen na ovom pristupu bilo bi vrlo teško ponoviti s obzirom da je korištena interna platforma te nisu javno objavljene korištene ontologije i programski kod.

Autori *Solis et al.* predstavili su agilnu razvojnu platformu za razvoj web aplikacija pod nazivom *SWB* (engl. *Semantic Web Builder*). Motivacija za izradu ove platforme jest povećanje ponovne iskoristivosti komponenata u specifičnoj domeni (web) te mogućnost izrade kompleksnih sustava u što je mogućem kraćem vremenu. Platforma primjenjuje ontologije za modeliranje korisničkih zahtjeva te zatim iz njih automatski generira infrastrukturu. Radi se o polu-automatskom pristupu razvoja čiji razvojni ciklus započinje reprezentacijom zahtjeva kroz ontologiju. Zatim se primjenjuju transformacijska pravila za generiranje objektno-orijentiranog programskog koda temeljenog na konceptima i atributima definiranim u ontologiji. Kao rezultat dobiva se infrastruktura sustava podržana platformom koja omogućava pohranu semantičkih podataka i korištenje gotovih web komponenata. Dakle, ovaj pristup generira klase u objektno-orijentiranom programskom jeziku iz elementa predefinirane ontologije, dok je specifične funkcionalnosti (poslovna logika, korisničko sučelje) potrebno posebno razvijati te njima

proširiti generirani kod. *SWB* platforma korištena je za razvoj aplikacija za nekoliko vladinih organizacija u Meksiku (Solis, Pacheco, Najera, & Estrada, 2013). Osnovni nedostatak ovog pristupa jest primjena kombinacije generiranog programskog koda, koji se ne smije modificirati i prilagođenog programskog koda koji se mora dodatno ručno razvijati za sve specifične funkcionalnosti.

WISE (engl. *Web Information System auto-construction Environment*) je prototip platforme za razvoj web informacijskih sustava autora *Tang et al.* koji primjenjuje vlastitu ontologiju za opis korisničkih zahtjeva, ali i grafički alat za crtanje korisničkih sučelja. Razvoj ove platforme motiviran je spoznajom da ontologije ne moraju nužno biti samo formalna specifikacija korisničkih zahtjeva, već mogu posložiti i kao modeli za generiranje programskog koda. *WISE* koristi dva tipa ontologija, ontologiju domene za opis statičnih entiteta i ontologiju ponašanja za opis dinamičkih značajki sustava. Sama platforma implementirana je kao set dodataka za *Eclipse* razvojno okruženje. Implementiran je grafički alat koji korisnicima olakšava konstrukciju ontologija domene i ponašanja. Za razliku od *SWB* platforme, *WISE* koristi relacijsku bazu podataka te pruža metode za transformaciju ontologije ponašanja u programski kod koji koristi strukturirani jezik upita na bazu (SQL) (Tang et al., 2006). S obzirom da *WISE* platforma generira programski kod automatski iz predefiniranih ontologija, nije moguće dodati nove elemente ontologije ili korisničkog sučelja bez proširivanja same platforme.

1.2.4. Alternativni pristupi

Alternativni pristup generiranju programskog koda jest jednostavno pisanje svih dijelova programskog koda ručno, pakiranje ponovno iskoristivog programskog koda u biblioteke ili kopiranje dijelova programskog koda s jednog projekta na drugi. Kopiranje programskog koda uzrokuje duplikaciju što nije poželjna osobina kvalitetnog softvera, tako da se u modernom razvoju softverskih proizvoda najčešće koriste kombinacije generativnog programiranja te višestruko iskoristivih modula i biblioteka.

Alternativni pristupi primjeni ontoloških modela za specifikaciju i generiranje programskog koda su UML modeli te DSL-ovi. UML je jezik modeliranja koji pruža standardni način vizualizacije dizajna softvera, ali primjenom profila može se koristiti i za modeliranje specifičnih domena. DSL je pak jezik visoke razine koji podržava koncepte i apstrakcije odrađene domene (Visser, 2008). I UML i DSL jezici danas se primjenjuju za generiranje izvršivog programskog koda. Već smo spomenuli da velika većina MDD pristupa koristi UML

jezik za modeliranje sustava te automatski generira programski kod iz tih modela. Slična je primjena i DSL-ova, osim što je DSL obično tekstualni, a ne vizualni model. DSL-ovima se nastoji poboljšati, odnosno olakšati proces razvoja softvera unutar određene domene, podizanjem razine apstrakcije i najčešće prevođenjem DSL koda u GPL kod.

WebDSL je DSL za kreiranje web aplikacija s bogatim podatkovnim modelom. Osnovna motivacija za kreiranje ovog DSL-a je smanjivanje duplikacije programskog koda te olakšavanje rada programerima na način da se ne moraju brinuti o pisanju i održavanju ponavljajućeg koda. Kreiran je pomoću *Stratego/XT* jezika i seta alata te *Java* okvira za razvoj web aplikacija. *WebDSL* fokusiran je na domenu web aplikacija koje koriste baze podataka i pružaju korisničko sučelje sa nekoliko različitih pogleda na podatke u bazi, ali i mogućnosti izmjene podataka putem web preglednika. *WebDSL* generator generira potpuni programski kod koji nema potrebe ručno nadopunjavati ili podešavati. Autori procjenjuju da donosi smanjenje od 5 do 10 puta, mjereno u broju linija programskog koda u odnosu na ručno programirane aplikacije kroz GPL (Visser, 2008).

WebRatio je pristup baziran na transformaciji DSL-a nazvanog *WebML* (engl. *Web Modeling Language*) u izvršivi programski kod. *WebRatio* generira GPL kod podatkovnih modela i poslovne logike, ali i grafičko sučelje bez ograničavanja kvalitete korisničke interakcije. Vizija kreiranja ovog alata bila je omogućavanje brze i ekonomične isporuke poslovnih web aplikacija po mjeri upotrebom razvojnih alata koji integriraju MDD i CBD. Princip brzog prototipiranja omogućuje ovom pristupu rano uključivanje korisnika u razvojni proces. *WebRatio* generator generira potpuni *Java* programski kod i oslanja se na *Struts* okvir za prezentacijski sloj te *Hibernate* okvir za podatkovni sloj. Ovaj alat primijenjen je u različitim domenama (financije, bankarstvo, moda, javni prijevoz, edukacija, itd.) kroz nekoliko godina i rezultati implementacije ovog MDD pristupa pokazuju da su indeksi produktivnosti do tri puta veći, a troškovi održavanja smanjeni do 80% (Brambilla & Fraternali, 2014).

1.3. Plan istraživanja

1.3.1. Obuhvat i ograničenja

U sklopu ovog doktorskog rada istražiti ćemo mogućnost primjene ontološkog modela mapiranog na generatore programskog koda, s ciljem automatskog generiranja implementacije obitelji poslovnih web aplikacija. Primjena generativnog pristupa pri razvoju poslovnih aplikacija ima smisla kada postoji značajno podudaranje među zahtjevima za razvoj više proizvoda (odnosno obitelji). Stoga je obuhvat ovog istraživanja ograničen na razvoj obitelji poslovnih aplikacija i to isključivo za web platformu.

Razvoj poslovnih web aplikacija vrlo je široko područje i obuhvaća velik broj domena u kojima se takva rješenja primjenjuju. U sklopu ovog doktorskog rada fokus je stavljen na web rješenja po mjeri za poslovne potrebe malih i srednjih poduzeća, koji imaju karakteristike obitelji programskih proizvoda, u smislu značajnog broja sličnih korisničkih zahtjeva, funkcionalnosti i korištene tehnologije razvoja. Obuhvat istraživanja biti će ograničen na klasične poslovne web aplikacije temeljene na relacijskim bazama podataka, objektnim podatkovnim modelima, web komponentama, tablicama za pregled podataka te obrascima za unos, promjenu i pohranu podataka.

U sklopu istraživanja biti će dizajnirano nekoliko inovativnih artefakata, prvenstveno ontološki model i generatori programskog koda. Navedeni artefakti temeljni su dijelovi glavnog artefakta, odnosno razvojnog okvira. Artefakti kreirani u ovom istraživanju omogućiti će automatizirani razvoj poslovnih web aplikacija ograničenih mogućnosti. Osnovni cilj istraživanja je dizajn razvojnog okvira koji će omogućiti proširenje i prilagodbu kreiranih artefakata, s ciljem omogućavanja naknadne implementacije specifične poslovne logike ili novih elementa korisničkog sučelja. Kreirani artefakti u sklopu ovog istraživanja ne mogu zadovoljiti sve korisničke zahtjeve različitih aplikacija ovog tipa, ali primjenom razvojnog okvira moguće je ponovno iskoristiti funkcionalnosti s ranijih projekta te kroz nadogradnju ontološkog modela i generatora programskog koda dodati nove funkcionalnosti.

Demonstracija i evaluacija kreiranih artefakata u ovom istraživanju planira se kroz provedbu studija slučaja. S obzirom da će autor sam razviti i evaluirati okvir, subjektivni utjecaj može predstavljati slabost planiranog istraživanja. Stoga studije slučaja neće biti implementirane prema specifičnim primjerima poslovnih aplikacija, niti korisničkim zahtjevima specifičnih tvrtki, već prema specifikaciji univerzalnog poslovnog jezika (UBL), verzije 2.1. Iako je osnovna svrha UBL-a omogućavanje elektroničke razmjene dokumenata, planirano istraživanje

ne bavi se tim područjem. U ovome istraživanju primijeniti ćemo UBL specifikaciju dokumenata i njihovih atributa, odnosno službene primjere odabranih UBL dokumenata za izvlačenje korisničkih zahtijeva, s ciljem razvoja odgovarajuće programske podrške. Implementacija UBL specifikacije dokumenata podrazumijeva web obrasce s odgovarajućim elementima korisničkog sučelja, unosna polja i odgovarajuće validacije te poslovnu logiku potrebnu za izračun vrijednosti. Osnovne funkcionalnosti većine web aplikacija, poput prijave i odjave korisnika te definicije korisničkih prava korisnika biti će implementirane u sklopu studija slučaja, dok su sve ostale funkcionalnosti izvan opsega planiranog istraživanja. Prilikom evaluacije neće biti korištene metrike podložne subjektivnom utjecaju poput utroška vremena i direktne usporedbe programskog koda, već će se koristiti objektivne metrike koje uspoređuju elemente specifikacije i generirani kod.

1.3.2. Ciljevi i hipoteze

Ovim istraživanjem nastoji se, primjenom ontološki potpomognutog generativnog pristupa programiranju, ubrzati te olakšati razvoj obitelji poslovnih web aplikacija. Radi se o specifičnom tipu aplikacija, kakve često primjenjuju mala i srednja poduzeća, ali i o inovativnom pristupu razvoja koji ima smisla primijeniti kada se razvija više proizvoda (obitelj) temeljnih na značajnom broju zajedničkih korisničkih zahtjeva i artefakata. Osnovno sredstvo kojim se nastoji ubrzati razvoj ovog tipa aplikacija jest podizanje razine apstrakcije, i to sa razine programskog koda na razinu ontološkog modela. Iz te tvrdnje proizlazi prvi cilj istraživanja C1.

C1: Dizajnirati okvir koji će primjenom ontološki potpomognutog, generativnog programiranja podići razinu apstrakcije pri razvoju obitelji poslovnih web aplikacija.

Ontološki model primijeniti ćemo za opis spremišta podataka, web obrazaca, web komponenata, korisničkog sučelja te drugih artefakata korištenih na specifičnim programskim proizvodima. Ontološkim opisom tih artefakata nastoji se omogućiti jednostavna ponovna iskoristivost istih na novim projektima, iz čega proizlazi drugi cilj istraživanja C2.

C2: Postići visoki stupanj ponovne iskoristivosti ranije korištenih artefakata pri razvoju novih proizvoda iz obitelji poslovnih web aplikacija.

Iz definiranih ciljeva proizlaze i istraživačke hipoteze H1 i H2. S obzirom da je cilj C1 fokusiran na podizanje razine apstrakcije pri razvoju, hipotezom H1 nastoji potvrditi da predloženi okvir

zaista podiže razinu apstrakcije i to na temelju usporedbe količine specifikacije i generiranog programskog koda.

H1: Predloženi okvir za razvoj obitelji poslovnih web aplikacija podiže razinu apstrakcije opisa zahtjeva i artefakata mjerenu smanjenjem ontološke specifikacije aplikacije u odnosu na njezin programski kod

Cilj C2 je pak fokusiran na postizanje visokog stupanja ponovne iskoristivosti artefakata pa se hipotezom H2 nastoji potvrditi da primjena ontološkog modela za specifikaciju artefakata poslovnih web aplikacija upravo to omogućuje.

H2: Primjena ontološkog modela, kao specifikacije za generiranje programskog koda, osigurava visoku razinu ponovne iskoristivosti ranije razvijenih i ontološki specificiranih artefakata obitelji poslovnih web aplikacija

Istraživačke hipoteze H1 i H2 biti će potvrđene ili opovrgnute primjenom metode studije slučaja.

1.3.3. Struktura rada

Nakon pregleda područja istraživanja, opisa problema, postojećih rješenja, motivacije te navođenja ciljeva, hipoteza i opsega istraživanja u ovom uvodnom poglavlju, ostatak dokumenta je organiziran u dodatnih pet poglavlja. Struktura rada sukladna je sa obrascem publikacija za istraživanja temeljenih na dizajnu autora *Gregor* i *Hevner* (Gregor & Hevner, 2013).

Drugo poglavlje *2. Metodologija* predstavlja korištenu metodologiju, a radi se o DSRM (engl. *Design Science Research Methodology*) metodologiji za provedbu istraživanja temeljenih na dizajnu. U ovome poglavlju predstavljen je tip istraživanja, korištena metodologija i njezinih šest aktivnosti te dodatne smjernice za provedbu ovog tipa istraživanja. Opisan je tijek i opseg istraživanja, navedene su korištene znanstvene metode u svakoj fazi istraživanja i definiran je očekivani znanstveni doprinos.

Treće poglavlje *3. Sustavni pregled literature* prezentira rezultate provedenog sustavnog pregleda literature (SLR). Pregled literature je proveden prema smjernicama za provedbu SLR-a u programskom inženjerstvu autora Kitchenham i Charters (Kitchenham & Charters, 2007). Ova metoda korištena je kao sredstvo vrednovanja i interpretacije dostupnog istraživanja iz zadanog područja. Baza znanja prikupljena pregledom literature poslužiti će kao početna točka,

ali i mjerilo za dizajnirani okvir. Radovi, odnosno predloženi pristupi, okviri i prototipi, prikupljeni ovim pregledom kategorizirani su prema osnovnim disciplinama kojima se bave.

Četvrto poglavlje *4. Dizajn i razvoj* ujedno je i najvažnije poglavlje ovog istraživanja jer predstavlja rezultate faze dizajna i razvoja, u sklopu koje su dizajnirani i razvijeni glavni doprinosi ovog istraživanja. Radi se o ontološkom modelu, generatorima programskog koda i razvojnom okviru. Ontološki model razvijen je prema odabranoj metodologiji za razvoj ontologija, a generatori programskog koda razvijeni su u *Java* programskom jeziku primjenom nekoliko dodatnih okvira koji su olakšali i ubrzali samu implementaciju. Razvojni okvir prezentiran je uporabom UML dijagramskih tehnika, a razvojni ciklus okvira definiran je kroz osam aktivnosti.

Peto poglavlje *5. Demonstracija i evaluacija* prezentira rezultate demonstracije i evaluacije dizajniranog okvira nad dvjema studijama slučaja. Prva studija slučaja ima za cilj demonstrirati izvedivost i korisnost predloženog pristupa, a druga studija slučaja ima cilj evaluirati razvojni okvir u vidu ponovne iskoristivosti artefakta s prve studije slučaja. Temeljem analize rezultata provedenih studija slučaja potvrditi će se ili opovrgnuti definirane istraživačke hipoteze.

Posljednje, šesto poglavlje *6. Diskusija i zaključak* poslužiti će za raspravu o prevedenom istraživanju, njegovim rezultatima, ostvarenim ciljevima, doprinosima, ali i ograničenjima. U ovome poglavlju naglašena su preostala otvorena pitanja, odnosno mogućnosti za buduća istraživanja na ovome području.

Ovome doktorskom radu priloženo je i nekoliko dodataka koji pružaju detaljniji pogled u rezultate pojedinih faza istraživanja. *Dodatak A* sadrži popis odabiranih radova za SRL, prezentiran u trećem poglavlju. *Dodatci B-G* sadrže dizajnirane i razvijene ontologije u RDF / XML[1] formatu, koje su detaljnije predstavljene u četvrtom poglavlju. *Dodatak J* sadrži CD sa izvornim programskim kodom generatora, ontologijama i provedenim studijama slučaja. Dakle, u ovom dodatku objedinjeni su artefakti generatora programskog koda i ontološkog modela, a moguće je detaljnije proučiti i specifikaciju te generirani programski kod za obje provedene studije slučaja prezentirane u poglavlju 5.

[1] RDF je u prijevodu okvir za opis resursa (na webu), a XML proširivi jezik oznaka. RDF dokumenti su napisani u XML-u, a XML jezik kojeg koristi RDF naziva se RDF / XML.

2. METODOLOGIJA

2.1. Istraživanje temeljeno na dizajnu

2.1.1. Tip znanstvenog istraživanja

Ovo znanstveno istraživanje temeljeno je na dizajnu, stoga najprije moramo definirati što je dizajn. Dizajn se ponajprije bavi sintezom, tj. dizajner mora biti u mogućnosti kombinirati različite dijelove, elemente, koncepte i procese kako bi definirao novu cjelinu. Naglasak je na pronalasku pravog uređenja dijelova, iako sami elemente već mogu postojati. Mi ljudi imamo sposobnost prepoznati da svijet možemo mijenjati stvaranjem, uređivanjem i reorganizacijom elemenata koje nam pruža priroda kako bismo poboljšali kvalitetu našeg života. Uključujemo se u dizajnerske aktivnosti jer vidimo nedostatak cjelovitosti ili put do željenog stanja (Vahidov, 2012).

Istraživanje temeljeno na dizajnu nastoji kreirati artefakte koji služe ljudima, tehnološki je orijentirano i upotrebljava kriterije vrijednosti ili upotrebljivosti artefakata. Razlikujemo četiri osnovna tipa dizajniranih artefakata: konstrukti, modeli, metode i implementacije. Osnovno znanje, odnosno konstrukti formiraju vokabular domene i koriste se za karakterizaciju pojava kao i u prirodnim znanostima. Mogu se kombinirati u konstrukte višeg reda, koje nazivamo modelima, a koriste se za opis zadataka, situacija ili artefakata. Dizajnom se razvijaju i metode, skupovi koraka, odnosno načini obavljanja određenih aktivnosti. Na kraju, sve navedeno može se implementirati s ciljem izvođenja specifičnih zadataka ili primjene artefakta u namijenjenom okruženju. Umjesto da postavljaju teorije, istraživanja temeljena na dizajnu nastoje stvoriti modele, metode i implementacije koje su inovativne i vrijedne (March & Smith, 1995).

Ovaj tip istraživanja kreira nove i inovativne artefakte kojima nastoji riješiti konkretan problem iz prakse. Razlikuje se od rutinskog dizajna, jer inovativnim načinom nastoji riješiti važne, do sada ne riješene probleme. Istraživanje se provodi prema znanstvenoj metodologiji koja osigurava relevantnost, evaluaciju dizajna i doprinos u smislu nove ili poboljšane teorije ili metode. Autori *Hevner et al.* zaslužni su za proboj istraživanja temeljenih na dizajnu u informacijske znanosti. Oni su prepoznali da kreiranje novih informacijsko-komunikacijskih artefakata ne mora nužno biti dobar istraživački rad. Stoga su prezentirali smjernice za provođenje i kriterije za ocjenjivanje istraživanja temeljenog na dizajnu u informacijskim znanostima. Razliku između rutinskog dizajna i istraživanja temeljenog na dizajnu definirali su na sljedeći način. *„Rutinski dizajn je primjena postojećeg znanja u rješavanju organizacijskih*

problema, poput izgradnje financijskih ili marketinških informacija sustav pomoću najbolje prakse iz dostupne baze znanja. Istraživanje temeljeno na dizajnu rješava važne, do sada neriješene probleme na jedinstvene ili inovativne načine.“ Istraživanje temeljeno na dizajnu bavi se važnim problemima za koje je trenutna baza znanja neadekvatna te mora demonstrirati svoj doprinos bazi znanja (Hevner, March, Park, & Ram, 2004).

Doprinos znanstvenih istraživanja temeljenih na dizajnu leži u kombinaciji inovativnosti i korisnosti izgrađenih artefakata. Kako bi se osigurao znanstveni doprinos istraživanja temeljenog na dizajnu potrebno je identificirati i jasno opisati problem, demonstrirati da trenutno ne postoji adekvatno rješenje, razviti i prezentirati inovativni artefakt, evaluirati ga te obrazložiti njegovu korisnost i dodanu vrijednosti (March & Storey, 2008). Cilj istraživanja temeljenog na dizajnu je kreirati bolja rješenja od postojećih, u obliku učinkovitijeg proizvoda, procesa, usluge, tehnologije ili ideje. Istraživanje temeljeno na dizajnu ocjenjuje se najprije po svojoj sposobnosti da jasno predstavi i komunicira novi dizajn / artefakt. Nakon toga procjenjuju se dokazi o poboljšanju u odnosu na trenutna rješenja. Poboljšanje može biti u obliku pozitivne promjene u učinkovitosti, produktivnosti, kvaliteti, konkurentnosti, ili druge kvalitetne mjere, ovisno o ciljevima istraživanja (Gregor & Hevner, 2013).

2.1.2. Metodologija istraživanja temeljenog na dizajnu

Autori *Peffers et al.* definirali su metodologiju za provedbu istraživanje temeljenog na dizajnu u informacijskim znanostima. Metodologija je nazvana je DSRM, a sastoji se od šest faza ili aktivnosti:

1) Identifikacija problema i motivacija – faza koja podrazumijeva definiciju specifičnog problema kojim se istraživanje bavi i opravdanje vrijednosti budućeg rješenja. Buduća vrijednost motivira istraživača u potrazi za rješenjem i pomaže razumijevanju zaključaka istraživanja. Resursi potrebni za ovu aktivnost uključuju poznavanje stanja problema i važnosti njegovog rješavanja.

2) Definicija ciljeva istraživanja – faza u kojoj se definiraju ciljevi istraživanja iz definicije problema, ali i poznavanja onoga što je moguće i izvedivo. Ciljevi mogu, no ne moraju biti kvantitativni, ali trebaju se moći racionalno zaključiti iz specifikacije problema. Potrebni resursi za ovu fazu istraživanja su poznavanje stanja problema i aktualnih rješenja te njihovu učinkovitost.

3) Dizajn i razvoj – faza u kojoj se kreira novi artefakt: konstrukt, model, metoda, ili implementacija (mogu se dodavati i nova svojstva tehničkih, društvenih ili informacijskih resursa). Artefakt istraživanja može biti bilo koji dizajnirani objekt u kojem je doprinos istraživanja ugrađen u sam dizajn. Ova aktivnost uključuje određivanje željene funkcionalnosti i arhitekture, a zatim i kreiranje stvarnog artefakta.

Potrebni su resursi za projektiranje i razvoj, ali i teorijsko znanje koje se može transformirati u rješenje.

4) Demonstracija – faza koja podrazumijeva dokazivanje upotrebljivosti artefakta za rješavanje jednog ili više slučajeva problema. To bi moglo uključivati njegovu upotrebu u eksperimentiranju, simulaciji, studiji slučaja ili drugoj prikladnoj aktivnosti. Potrebni resursi za demonstraciju uključuje učinkovito znanje o korištenju artefakta za rješavanje problema.
5) Evaluacija – faza u kojoj promatramo i mjerimo koliko dobro artefakt podržava rješenje problema. Ova aktivnost uključuje usporedbu ciljeva i stvarnih promatranih rezultata uporabe artefakta u demonstraciji. Samim time ova aktivnost zahtijeva poznavanje relevantnih metrika i tehnika analize. Ovisno o prirodi problema i artefakta, evaluacija može poprimiti različite oblike: od usporedbe funkcionalnosti artefakta s postavljanim ciljevima do mjerenja performansi (vrijeme odgovora, dostupnost i sl.). Na kraju ove aktivnost istraživači moraju odlučiti hoće li se ponovo vratiti na treći korak i pokušati poboljšati učinkovitost artefakta ili ostaviti daljnje poboljšanja za buduća istraživanja ili projekte.
6) Komunikacija – faza komuniciranja problema i njegove važnosti te artefakta i njegove korisnosti, inovativnosti, dizajna i djelotvornosti prema drugim istraživačima i ostaloj zainteresiranoj publici. U znanstvenim istraživačkim publikacijama istraživači mogu koristiti strukturu ovog procesa za strukturiranje članaka (Peffers et al., 2006; Peffers, Tuunanen, Rothenberger, & Samir, 2008).

2.1.3. Smjernice za provedbu istraživanja

Autori *Hevner et al*., koji su među najzaslužnijima za proboj istraživanja temeljenog na dizajnu u informacijske znanosti, definirali su šest osnovnih smjernica za provedbu ovog tipa istraživanja:

1) Problem koji se rješava mora biti relevantan. Kriteriji za procjenu relevantnosti trebaju biti fokusirani na vjernu reprezentaciju i mogućnost implementacije kreiranog artefakta.
2) Znanstveno istraživanje zahtijeva primjenu rigoroznih metoda prilikom izgradnje i evaluacije artefakata. Odabir metrika i procjena njihove prikladnosti važan je dio znanstveno-istraživačkog procesa.
3) Potraga za optimalnim dizajnom često je nedokaziva za realistične informacijske sustave. Dizajn je kreativan i iterativnom proces rješavanja problema kojim se izgrađuju

i evaluiraju svrsishodni artefakti. Dizajn je u suštini proces pretraživanja i konstrukcije artefakta koji dobro obavlja svoju zadaću, primijenjen na određenu klasu problema.

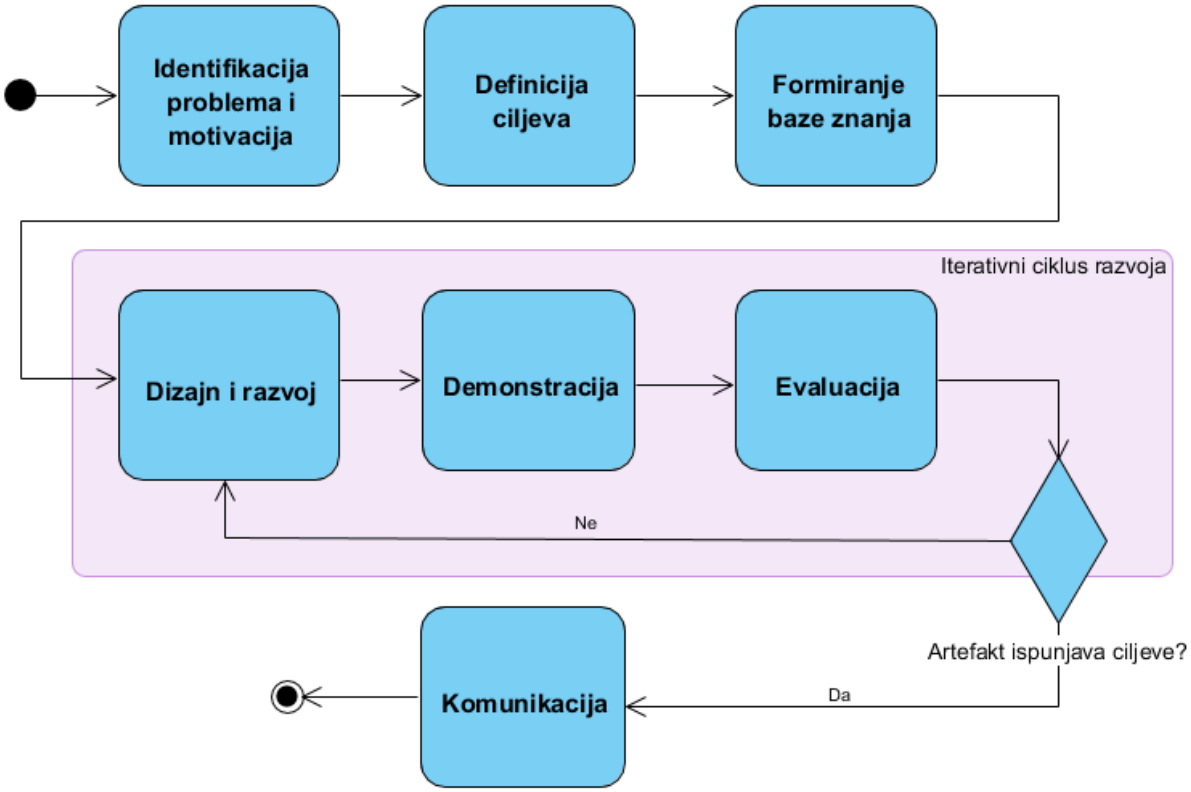
4) Artefakt treba prezentiran na način koji omogućava njegovu evaluaciju i usporedbu s postojećim artefaktima koji se primjenjuju za istu svrhu. Kreirani artefakt može biti konstrukt, model, metoda ili primjer implementacije.
5) Novi artefakt je kompletan i učinkovit kada zadovolji sve zahtjeve i ograničenja problema koji treba riješiti. Evaluacija zahtijeva definiciju odgovarajućih metrika te prikupljanje i analizu odgovarajućih podataka. Informacijsko-komunikacijski artefakti mogu se ocijeniti u smislu funkcionalnosti, cjelovitosti, dosljednosti, točnosti, performansi, pouzdanosti, upotrebljivosti, itd.
6) Učinkovita znanstvena istraživanja temeljena na dizajnu moraju pružiti jasne doprinose na području kojim se bave. Znanstveni doprinos ovog tipa istraživanja obično je sam artefakt, ali doprinos može se očitovati i u novim teorijama, evaluacijskim metodologijama i metrikama (Hevner et al., 2004).

2.2. Faze istraživanja

Istraživanje u sklopu ovog doktorskog rada biti će provedeno prema predstavljenoj DSRM metodologiji autora *Peffers et al.* (Peffers et al., 2008), ali i smjernicama za provedbu istraživanja temeljenog na dizajnu autora *Hevner et al.* (Hevner et al., 2004). DSRM metodologija sastoji se od šest aktivnosti, a temeljem navedenih smjernica u sklopu istraživanja provesti će se i jedna dodatna aktivnost. Radi se o aktivnosti formiranja baze znanja, čiji je cilj obaviti detaljan pregled područja i pokazati kako trenutno ne postoji adekvatno rješenje za problem kojim se istraživanje bavi. Konačni skup od sedam aktivnosti koje se planiraju provesti u sklopu ovog istraživanja su:

(1) Identifikacija problema i motivacija,
(2) Definicija ciljeva,
(3) Formiranje baze znanja,
(4) Dizajn i razvoj
(5) Demonstracija,
(6) Evaluacija i
(7) Komunikacija.

Slika 1 prikazuje tijek istraživanja po definiranim aktivnostima. Sam tijek istraživanja temelji se na DSRM metodologiji i sukladan je s procesnim modelom metodologije. S obzirom da je dizajn iterativan proces rješavanja problema, odnosno kreiranja artefakta koji ispunjava postavljane ciljeve ili zahtjeve, faze (4) *Dizajn i razvoj, (5) Demonstracija* i *(6) Evaluacija* provoditi će se iterativno, a artefakt će se razvijati i poboljšavati do razine definirane ciljevima istraživanja te razine potrebne za provedbu demonstracije kroz planirane studije slučaja.



Slika 1: Tijek znanstvenog istraživanja po fazama

2.3. Tijek i opseg istraživanja

Prva faza, odnosno *(1) Identifikacija problema i motivacija* za provedbu istraživanja, zapravo proizlazi iz prakse. Današnje moderno tržište zahtijeva ubrzani razvoj kvalitetnog web softvera po mjeri za poslovne potrebe. Osobito se to odnosi na mala i srednja poduzeća koja često imaju poslovne potrebe za razvojem softvera po mjeri, ali si isti ne mogu priuštiti zbog visokih troškova razvoja i održavanja. S obzirom da je trenutno potražnja za poslovnim softverom velika, informatičke tvrtke često nisu niti zainteresirane za razvoj manjih i jednostavnijih rješenja, prilagođenih malim i srednjim poduzećima. Motivacija i svrha istraživanja već je detaljnije predstavljena u poglavlju *1.2 Opis problema*.

Druga faza istraživanja jest *(2) Definicija ciljeva* istraživanja. U sklopu poglavlja *1.3.2 Ciljevi i hipoteze* već su definirana dva glavna cilja istraživanja C1 i C2, koji se odnose na povećanje razine apstrakcije pri razvoju obitelji poslovnih web aplikacija i povećanje stupnja ponovne iskoristivosti artefakata korištenih na ranijim projektima.

Treća faza, odnosno *(3) Formiranje baze znanja* planira se provesti primjenom metode sustavnog pregleda literature. Ova metoda koristi se kao sredstvo vrednovanja i interpretacije dostupnog istraživanja iz zadanog područja. Baza znanja prikupljena pregledom literature poslužiti će kao početna točka, ali i mjerilo za dizajnirani okvir. Metoda i rezultati sustavnog pregleda literature prezentirati su u poglavlju *3. Sustavni pregled literature*.

Faza četiri, *(4) Dizajn i razvoj* započinje izgradnjom ontološkog modela koji će se koristiti za modeliranje poslovnih aplikacija, a kasnije i kao specifikacija za generiranje programskog koda. Artefakti iz domene programskog inženjerstva postati će elementi ontologija, a OWL je jedini jezik koji će se koristiti za modeliranje kompletne aplikacije u ovom istraživanju. Nakon kreiranja početnog ontološkog modela, koji će se iterativno proširivati novim elementima, faza dizajna nastavlja se dizajnom i razvojem generatora programskog koda te mapiranjem ontološkog modela na generatore. S obzirom da je jedan od glavnih ciljeva ovog pristupa omogućavanje proširenja i prilagodbu kreiranih artefakata, fokus nije stavljen na implementaciju svih mogućih varijanti poslovne logike, ograničenja i komponenata, već na dizajn razvojnog okvira koji će upravo to omogućiti. Okvir će biti prezentiran upotrebom UML dijagramskih tehnika te nizom procesnih aktivnosti koje je potrebno provesti pri ontološki potpomognutom, generativnom razvoju obitelji poslovnih web aplikacija. Ovo je ključna faza istraživanja kojom se nastoje ispuniti ciljevi C1 i C2, a način provedbe i rezultati biti će prezentirati u poglavlju *4. Dizajn i razvoj*.

Faze *(5) Demonstracija* i *(6) Evaluacija* kreiranih artefakata u ovom istraživanju (ontološki model, generatori programskog koda i razvojni okvir) planiraju se kroz provedbu dvije studije slučaja, odnosno implementacijom dvaju programskih proizvoda iz iste obitelji. S obzirom da će istraživač sam razviti i evaluirati okvir, subjektivni utjecaj istraživača može predstavljati slabost planiranog istraživanja. Stoga studije slučaja neće biti implementirane prema specifičnim primjerima poslovnih aplikacija, niti korisničkim zahtjevima specifičnih tvrtki, već prema UBL 2.1 specifikaciji. U ovoj fazi primijeniti će se UBL specifikacija dokumenata i njihovih atributa za razvoj odgovarajuće programske podrške. Za prvu studiju slučaja planira se implementacija web aplikacije za upravljanje narudžbenicama (prema primjeru iz *Dodatka H*), dok se za drugu studiju slučaja planira implementacija web aplikacije za upravljanje ponudama (prema primjeru iz *Dodatka I*). Kao specifikaciju zahtjeva za navedene studije slučaja biti će korišteni službeni primjeri UBL 2.1 ponude i narudžbenice preuzeti sa stranice OASIS konzorcija[2]. Implementacija UBL specifikacije navedenih dokumenata podrazumijeva web obrasce s odgovarajućim elementima korisničkog sučelja, unosna polja i odgovarajuće validacije te poslovnu logiku potrebnu za izračun vrijednosti. Osnovne funkcionalnosti, poput prijave / odjave korisnika i definicije korisničkih prava korisnika biti će implementirane u sklopu studija slučaja, dok su ostale funkcionalnosti izvan opsega planiranog istraživanja. Prilikom evaluacije neće biti korištene metrike podložne subjektivnom utjecaju poput utroška vremena i direktne usporedbe programskog koda. Ovom fazom se potvrđuju ili odbacuju hipoteze istraživanja H1 i H2. U poglavlju *5. Demonstracija i evaluacija* biti će predstavljeni i diskutirani rezultati ove dvije faze istraživanja.

Posljednja faza istraživanja, *(7) Komunikacija* planira se na način da se određeni dijelovi istraživanja, odrađeni u sklopu ove doktorske disertacije, prezentiraju na znanstvenim konferencijama ili objave u znanstvenim časopisima i to na engleskom jeziku. Osnovni cilj jest komunicirati rezultate istraživanja široj znanstvenoj i stručnoj publici.

[2] Dostupno 16.06.2018. na http://docs.oasis-open.org/ubl/UBL-2.1.html

2.4. Znanstvene metode

U sklopu istraživanja biti će primijenjeno nekoliko znanstvenih metoda, a predstaviti ćemo ih po fazama istraživanja u kojima se primjenjuju. U trećoj fazi istraživanja, odnosno *(3) Formiranje baze znanja*, biti će primijenjena metoda sustavnog pregleda literature. Pregled literature biti će proveden prema smjernicama za provedbu SLR u programskom inženjerstvu autora Kitchenham i Charters (Kitchenham & Charters, 2007). Ovakav oblika preglednog rada već je postao *de facto* standard u istraživanjima na ovom području i prihvaćaju ga gotovi svi znanstveni časopisi. Znanje prikupljeno pregledom literature poslužiti će kao smjernice za dizajn razvojnog okvira.

U četvrtoj fazi, *(4) Dizajn i razvoj*, biti će primijenjena metoda izgradnje ontologija prema specifičnoj metodologiji. Dizajnirani ontološki model sastojati će se od više međusobno povezanih ontologija, a odabrana metodologija za njihovu izgradnju jest *Ontology Development 101* autora Noy i McGuinness (Noy & McGuinness, 2001). Iako postoje i brojne druge metodologije za izgradnju ontologija, odluka za *101* metodologiju temeljena je na njezinoj zrelosti, iterativnosti te velikom broju praktičnih smjernica i pravila za donošenje odluka prilikom izgradnje ontologija koje metodologija sadrži. Odabrana metodologija izgradnje ontologija pokriva sve faze razvoja ontologija, uključujući i složena pitanja koja se odnose na definiranje klasne hijerarhije te svojstava klasa i instanci. Druga znanstvena metoda korištena u ovoj fazi istraživanja jest dizajn, konkretnije dizajn razvojnog okvira. Dizajn je inovativna metoda otkrivanja efikasnog rješenja za zadani problem koja zahtijeva znanje iz domene primjene i domene rješenja. Napredak se u dizajnu ostvaruje iterativno, a artefakt se smatra dovršenim i efikasnim kada zadovolji sve zahtjeve problema koji nastoji riješiti. Zahtjevi za dizajn razvojnog okvira su:

1) Definirati proces ontološki potpomognutog, generativnog razvoja obitelji poslovnih web aplikacija,
2) Omogućiti specifikaciju korisničkih zahtjeva i softverskih artefakata pomoću ontološkog modela,
3) Omogućiti automatsko generiranje programskog koda iz ontološke specifikacije,
4) Osigurati visoku razinu ponovne iskoristivost modela i specifikacije te
5) Pružiti smjernice za proširenje ontologija i generatora.

Primjena ovih dvaju metoda omogućiti će ispunjavanje ciljeva istraživanja C1 i C2.

U finalnim fazama istraživanja *(5) Demonstracija* i *(6) Evaluacija* biti će primijenjena znanstvena metoda studije slučaja. S ciljem demonstracije i evaluacije razvojnog okvira, u istraživanju će biti provedene dvije studije slučaja. Prva studija slučaja ima za cilj demonstrirati izvedivost predloženog pristupa, a druga studija slučaja ima cilj evaluirati razvojni okvir u vidu ponovne iskoristivosti artefakta s prve studije slučaja. Obje studije slučaja bit će korištene za potvrdu ili odbacivanje hipoteze H1, dok će samo druga studija slučaja biti korištena za potvrdu ili odbacivanje H2. Primjena kontroliranih eksperimenata, simulacija i studija slučaja neke su od preporučenih metoda evaluacije dizajniranih artefakata prema DSRM metodologiji i smjernica za provedbu istraživanja temeljenih na dizajnu (Hevner et al., 2004; Peffers et al., 2008).

2.5. Očekivani doprinos

Očekivani znanstveni doprinos istraživanja proizlazi iz ranije definiranih ciljeva i hipoteza istraživanja. Znanstveni doprinos podizanja razine apstrakcije pri razvoju, očitovati će se uvođenjem ontološkog modela kao više razine apstrakcije pri razvoju. Ovaj doprinos očitovati će se kvantitativno i kvalitativno. U kvantitativnom pogledu očekivani doprinos je smanjenje ontološke specifikacije aplikacije u odnosu na njezin generirani programski kod. U kvalitativnom pogledu doprinos će biti primjena ontologija za formalizirani zapis znanja u domeni razvoja obitelji poslovnih web aplikacija, kao i u mogućnosti specifikacije ovoga tipa aplikacija pomoću ontološkog modela. Primjena predloženog okvira za razvoj obitelji poslovnih web aplikacija omogućiti će izbjegavanje ciklusa ponovnog programiranja istih obrazaca te visoku razinu ponovne iskoristivost ranije implementiranih artefakata.

Kao i kod većine istraživanja temeljenih na dizajnu, osnovni znanstveni doprinos istraživanja su sami artefakti. U ovome znanstvenom istraživanju biti će dizajniran i razvijen ontološki model te generatori programskog koda koji čini znanstveni doprinos istraživanja D1.

D1: Ontološki model za specifikaciju zahtjeva i artefakata korištenih u razvoju obitelji poslovnih web aplikacija te generatori programskog koda za automatizaciju razvoja temeljem ulazne ontološke specifikacije.

Na temelju tih artefakata biti će dizajniran razvojni okvir kojemu je cilj definirati procesne aktivnosti te omogućiti proširenje i primjenu artefakata na druge poslovne slučajeve, različite korisničke i funkcionalne zahtjeve. Dizajnirani razvojni okvir čini doprinos istraživanja D2.

D2: Okvir za razvoj obitelji poslovnih web aplikacija koji primjenom ontologija formalizira znanje u području domene i specificira aplikacije na visokoj razini apstrakcije.

3. SUSTAVNI PREGLED LITERATURE

3.1. Metoda

Na području primjene ontologija u programskom inženjerstvu već postoji nekoliko vrlo kvalitetnih pregleda literature autora Calero et al., Gašević et al, Happel i Seedorf (C Calero et al., 2006; Gašević et al., 2009; Happel & Seedorf, 2006). Međutim, navedeni pregledni radovi nisu fokusirani na automatsko programiranje i objavljeni su prije desetak i više godina. Stoga je u sklopu ovog istraživanja, a s ciljem identifikacije trenutno dostupnih rješenja i tehnologija, obavljen SLR fokusiran na primjenu ontologija u automatskom i polu-automatskom programiranju. Rezultati ovog pregleda literature, obavljenog prema smjernicama za provedbu SLR u programskom inženjerstvu autora Kitchenham i Charters (Kitchenham & Charters, 2007), poslužiti će kao početna točka za dizajn razvojnog okvira. SLR jest formaliziran te ponovljiv postupak za dokumentiranje relevantnog znanja na određenom području. Provedeni pregled literature i njegovi rezultati su, u skladu sa metodologijom provedbe ovog doktorskog rada, objavljeni i u znanstvenom časopisu na engleskom jeziku (Strmečki et al., 2018).

Na temeljenu već spomenutih smjernica, SLR proces započinje definiranjem istraživačkih pitanja. Ovaj se pregled usredotočuje na pristupe koji koriste ontologije kako bi postigli djelomično ili potpuno generiranje programskog koda. Stoga su definirana slijedeća tri istraživanja pitanja:

1) Mogu li se ontologije primijeniti u automatskim / generativnim tehnikama programiranja, s ciljem povećanja ponovne iskoristivosti softvera (ili njegovih artefakta)?
2) Mogu li se ontologije primijeniti kao artefakti u razvoju softvera koji omogućuju rad (razvoj programskih proizvoda) na višim razinama apstrakcije?
3) Kako integrirati ontologije u razvojne procese automatskog programiranja s ciljem postizanja visoke razine modeliranja i specifikacije programskih proizvoda?

Uz istraživačka pitanja, a s ciljem pronalaska svih relevantnih studija, postavljeni su dodatni kriteriji uključivanja i isključivanja radova:

1) Biti će uključeni samo radovi napisani na engleskom jeziku i objavljeni u znanstvenim časopisima ili prezentirani na znanstvenim ili stručnim konferencijama.
2) Radovi se moraju odnositi na barem jedno od istraživačkih pitanja.
3) Radovi se neće filtrirati temeljem vrste / tipa istraživanja.
4) Radovi obavljeni 2005. godine i ranije biti će isključeni iz pregleda.
5) Radovi koji se izričito ne odnose na modeliranje, razvoj ili testiranje softvera bit će isključeni.
6) Duplikati i radovi u kojima tvrdnje nisu opravdane biti će isključeni iz pregleda.
7) Fokus se stavlja na primarne istraživačke radove, pa će drugi pregledni radovi biti isključeni.

Kako bi rad bio odabran za ovaj SLR mora zadovoljiti sve kriterije uključivanja i ne zadovoljiti niti jedan kriterij isključenja. S ciljem osiguranja vjerodostojnosti i relevantnosti odabranih radova, definirana su i dva dodatna kriterija kvalitete koji svi odabrani radovi moraju zadovoljiti:

1) Rad se mora temeljiti na analizi podataka, studiji slučaja, praktičnoj primjeni ili teorijskom rasuđivanju umjesto neopravdanih izjava.
2) Rad mora sadržavati opis konteksta u kojem je istraživanje provedeno ili u kojem se primjenjuje predloženo rješenje.

3.2. Pretraga

Pretraga radova je obavljena isključivo u znanstvenim bazama podataka, što znači da nisu pretraživane specifične knjige, niti časopise. Četiri elektronske znanstvene baze odabrane kao izvor radova za ovaj pregled:

1) Scopus[3]
    - Ograničeno na područje istraživanja informatike
2) Web of Science[4]
    - Ograničeno na područje istraživanja informatike
3) IEEE Xplore[5]
4) Google znalac[6]
    - Ograničeno na prvih 50 stranica rezultata (10 rezultata po stranici)

Ove su baze podataka odabrane iz razloga što su bile dostupne autoru u vrijeme provedbe SRL-a i jer osiguravaju pristup radovima iz važnih te visoko-utjecajnih časopisa i konferencija. Početni skup radova dobiven je provedbom automatizirane pretrage baza podataka korištenjem upita temeljenog na identificiranim ključnim riječima na engleskom jeziku:

("Ontologies" OR "Ontology") AND ("Software") AND ("Development" OR "Engineering" OR "Programming") AND ("Generative" OR "Generator" OR "Automatic" OR "Metaprogramming")

Ključne riječi korištene za pretragu u prijevodu su: ontologija / ontologije, softver, razvoj, inženjerstvo / programiranje, generativno / automatsko / generator / meta-programiranje. U prvoj iteraciji dobiveni su radovi filtrirani primjenom kriterija za uključivanje samo na naslov rada. U drugoj iteraciji preostali su radovi filtrirani primjenom kriterija uključivanja i isključivanja na sažetak rada. U ovoj iteraciji, uvod i zaključak bili bi također uzeti u obzir, ako sažetak nije bio dovoljno jasan. Svaka je iteracija kontrolirana od strane mentora kako bi se izbjegla pristranost jednog autora. Radovi koji su ostali uključeni nakon dvije iteracije detaljno su pregledani čitanjem cijelog teksta. Nakon čitanja cjelovitih radova, kriteriji za isključivanje

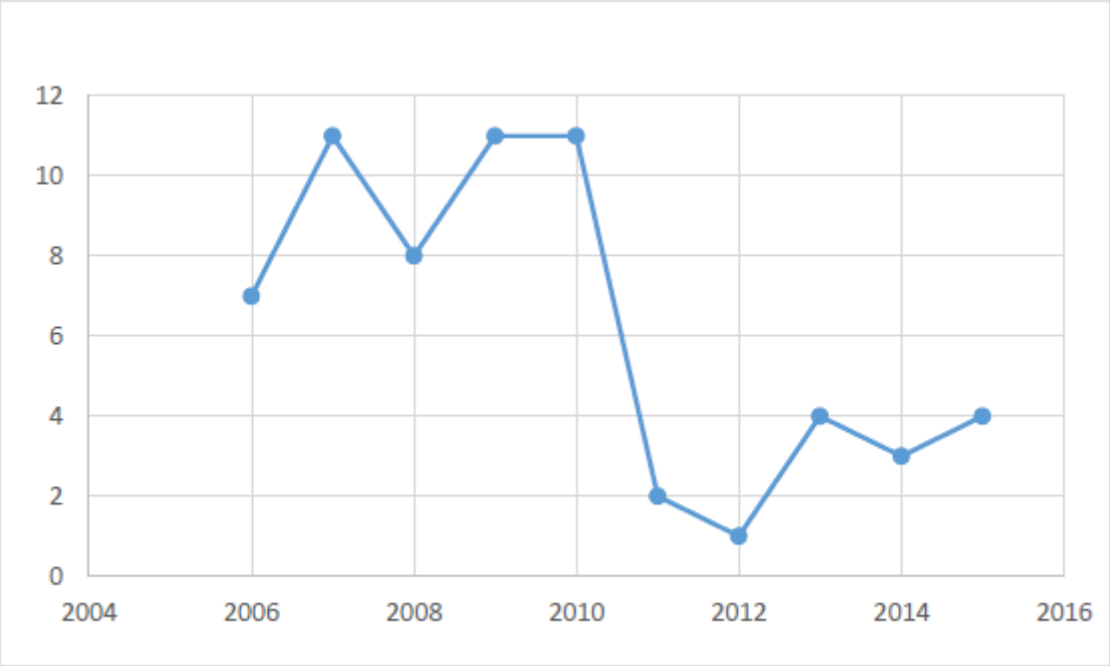
[3] Dostupno 10.04.2016. na https://www.scopus.com/
[4] Dostupno 10.04.2016. na https://webofknowledge.com/
[5] Dostupno 10.04.2016. na https://ieeexplore.ieee.org/
[6] Dostupno 10.04.2016. na https://scholar.google.hr/

bili su primijenjeni na neke radove za koje su autor i mentor zaključili da nisu relevantni za istraživanje.

3.3. Rezultati

3.3.1. Odabrani radovi

U prvoj iteraciji, nakon primjene kriterija uključivanja na naslov radova, pronađeno je ukupno 128 radova. U drugoj iteraciji, nakon primjene kriterija uključivanja i isključivanja na sažetak rada (i uvod ili zaključak po potrebi), preostalo je 67 radova. Nakon čitanja cijelog teksta tih radova, isključena su dodatna 4 rada. Cjeloviti popis 63 odabranih radova nalazi se u *Dodatku A* ovog doktorskog rada. Na *Slici 2* može se primijetiti da je većina odabranih radova objavljena u razdoblju između 2007. i 2010. godine. Iako nakon toga broj radova opada odabrani uzorak od 60-ak radova je značajan, a rezultati će pokazati kako još uvijek postoje mnoga otvorena pitanja i praznine koje treba zatvoriti kako bi učinkovito primijenili ontologije u automatskom razvoju programskih proizvoda.



Slika 2: Broj odabranih radova po godini objave

Nakon čitanja svih odabranih radova odlučili smo ih svrstati u sedam grupa temeljenih na paradigmi razvoja softvera kojoj predloženi pristupu primarno pripada. Najveći broj odabranih

studija predlaže rješenja za automatizaciju opće namjene i stoga su klasificirana u grupu Generativno programiranje. Uslužno-orijentirane arhitekture (SOA) najpopularnija su paradigma među odabranim radovima, što se moglo i očekivati, jer je SOA danas vrlo popularan arhitektonski uzorak i mnogi veliki sustavi bili su u procesu prebacivanja svoje arhitekture na SOA u proteklih 10 godina. Razvoj temeljen na modelima (MDD) sljedeća je najpopularnija paradigma, a zatim slijede više-agentni sustavi (MAS), automatizacija testiranja (TA), linije proizvoda u programiranju (SPL) i internet stvari (IoT).

3.3.2. Generativno programiranje

U nekoliko odabranih radova prepoznat je potencijal ontologija kao specifikacija na visokoj razini apstrakcije, iz kojih se može direktno generirati izvršivi programski kod. Autori *Athanasiadis et al.* primijenili su domensku ontologiju kako bi generirali *Java* programski kod i *Hibernate* objektno-relacijsko mapiranje (ORM) (Athanasiadis, Villa, & Rizzoli, 2007). U uvodu smo već spomenuli srodne pristupe ODASE i WISE koji također primjenjuju ontologije za opis korisničkih zahtjeva iz kojih automatski generiraju programski kod (Bossche et al., 2007; Tang et al., 2006). Autori *Toti i Rinelli* prezentirali su sustav koji je u mogućnosti generirati aplikacijsko programsko sučelje (API) na različitim razinama (pristup podatcima, usluge) iz bilo kojeg RDF repozitorija (Toti & Rinelli, 2015).

Još jedna popularna tema među odabranim radovima jest integracija ontologija sa modernim razvojnim platformama i objektno-orijentiranim jezicima. *Poggi* predlaže pristup reprezentaciji domenskog modela u *Java* programskom kodu primjenom sučelja i anotacija (Poggi, 2011). Autori *Rahmani et al.* realizirali su transformaciju između OWL-a i *Ecore*[7] u obliku dodatka za *Eclipse* integrirano razvojno okruženje (IDE) (Rahmani, Oberle, & Dahms, 2010). Autori *Djuric i Devedžić* te *Alias i Robin* prezentirali su vrlo slične pristupe integraciji ontološke paradigme u GPL jezike. Glavni cilj ova dva pristupa jest omogućiti programerima pisanje koda temeljenog na domeni s bogatom semantikom (Alias & Robin, 2014; Djuric & Devedzic, 2010, 2012). Ontologije su prikladne i za opisivanje softverskih komponenti pa ih u nekim radovima autori primjenjuju za CBD. Primjer je okvir autora *Andreou i Papatheocharous* koji primjenjuje ontologije za profiliranje komponenata te sadrži mehanizam za automatsko pronalažene komponenata (Andreou & Papatheocharous, 2015).

[7] *Ecore* je temeljni meta-model koji koristi *Eclipse Modeling Framework* za izražavanje drugih modela.

Generiranje i automatsko pozicioniranje komponenata korisničkog sučelja još je jedna domena gdje nekoliko autora primjenjuju ontologije. Autori *Luo et al.* predstavili su pristup koji primjenjuje ontologije za automatsko pozicioniranje elemenata korisničkog sučelja (UI) temeljem smjernica za dizajn korisničkog sučelja (Shengyang Luo, Yinglin Wang, & Jianmei Guo, 2009). Autori *Hildisch et al.* primijenili su automatsko generiranje korisničkog sučelja za dinamičke usluge u automobilskoj domeni (Hildisch, Steurer, & Stolle, 2007). Za kraj spomenimo još i pristupe koji istražuju primjenu ontologija za generiranje softverskih artefakata iz tekstualnih specifikacija pa čak i govornog jezika. Ovakav pristup istražili su autori *Tichy i Koerner* te *Talanov et al.*, međutim njihova rješenja su još u fazi prototipa i ne mogu se primijeniti na generalne probleme i domene (Talanov, Krekhov, & Makhmutov, 2010; Tichy & Koerner, 2010).

3.3.3. Uslužno-orijentirane arhitekture

Većina radova koja primjenjuju ontologije u SOA bave se problematikom automatskog otkrivanja i kompozicije Web usluga. SOA koristi standardne protokole i jezike od kojih vrijedi spomenuti jednostavan protokol za pristup objektima (SOAP) i jezik za opis web usluga (WSDL). SOAP je komunikacijski protokol neovisan o platformi koji definira format XML poruka koje se razmjenjuju između dvije aplikacije preko internetskih protokola. WSDL se obično koristi za definiranje programskih sučelja web usluga. Čak je i primjena ontologija u SOA djelomično standardizirana kroz WSDL-S i OWL-S. WSDL-S dodaje semantičko proširenje WSDL-u, dok je OWL-S ontologija koja definira osnovni skup oznaka za opisivanje web usluga (Cui-Mei, 2009; Korthaus, Schwind, & Seedorf, 2007; F. Li, 2009; Xiao, Zou, Tang, Ng, & Nigul, 2009).

Značajan broj alata, okvira, prototipova i obrazaca za postizanje automatskih ili polu-automatskih kompozicija web usluga predloženo je u literaturi. Autori *Bevilacqua et al.* predložili su alat koji može generirati konkretne, izvršive usluge pomoću repozitorija usluga, domenske ontologije i ograničenja (Bevilacqua, Furno, di Carlo, & Zimeo, 2011). *Li et al.* predlažu okvir za polu-automatsku kompoziciju web usluga koji koristi komponentu radnog tijeka (engl. *workflow*) i ontologiju procesa za dizajn te kompoziciju usluga (G. Li, Hai, & Hu, 2008). Sličan uzorak dizajna predlažu i autori *Dourandish et al.* koji primjenjuju više ontologija za modeliranje domena i komponenata radnog tijeka (Dourandish, Zumel, & Manno, 2007). *Hai-tao i Qing-rui* također su predstavili pristup dinamičke kompozicije usluga i to na temelju OWL-S i registra za univerzalni opis, otkrivanje i integraciju (UDDI). Oni koriste modul za normalizaciju i standardizaciju zahtjeva korisnika, normalizirani zahtjev za uslugom šalje se

OWL agentu zaduženom za semantičko označavanje (Hai-tao & Qing-rui, 2010). *Dao et al.* pak predlažu modeliranje i pohranu web usluga u repozitorij uslužnih modela, dok je ontološki repozitorij izgrađen kako bi se definirali semantički odnosi. Kompozicija usluga u ovom pristupu ostvaruje se preko modificiranog algoritma za lanac unatrag (engl. *backward chain*) koji kao ulaz prima modele usluga (Dao, Xu, & Chunlai, 2008). *Han et al.* u svom pristupu primjenjuju nekoliko ontologija: ontologija općeg znanja, domenska ontologija i ontologija usluga. Ovaj pristup zaprima zahtjeve za uslugama, prevodi ih i pretvara u interni format, provodi obradu podudaranja usluga i šalje opis usluge natrag tražitelju (H. Li et al., 2009). Autori *Zuniga et al.* predlažu okvir koji primjenjuje tehnike planiranja umjetne inteligencije te omogućava korisniku da u zahtjevu za uslugom odredi domenu, željeni cilj i ograničenja (Zuniga, Pérez-Alcázar, & Digiampietri, 2010).

U literaturi je predloženo i nekoliko različitih algoritama za rješavanje problema pronalaska adekvatne web usluge. IRIS (engl. *Interoperability and Reusability of Internet Services*) je naziv komponente autora *Radetzki i Cremers* koja sadrži nekoliko algoritama za pronalaženje sličnih operacija, kompoziciju usluga i post-procesiranje (Radetzki & Cremers, 2006). Autori *Liu et al.* koristili su algoritam za podudaranje poruka i algoritam za podudaranje sučelja. Uz potporu ontološke semantike, na prototipu su prikazali izvedivost i učinkovitost predloženog pristupa (Liu, Wang, Chen, & Li, 2008).

Kako bi se krajnjim korisnicima (koji nisu nužno eksperti) omogućilo sastavljanje i korištenje web-usluga, neki radovi fokusirali su se na sakrivanje složenost standarda i alata web usluga. Autori *Xiao et al.* predložili su jednostavnu shemu za opis web usluga pomoću opisnih oznaka (ključnih riječi). Ovaj pristup primjenjuje ontologije za prikupljanje relevantnih informacija vezanih uz određene ciljeve koristeći ekspertna znanja (Xiao et al., 2009). *Kim i Kim* također tvrde kako trenutne metode kompozicije web usluga predstavljaju problem uobičajenim korisnicima. Oni predlažu pristup automatske kompozicije otvorenih API-ja (Kim & Kim, 2016).

3.3.4. Razvoj temeljen na modelima

Ontologije se u MDD načinu razvoju najčešće koriste kao baze znanja i modeli iz kojih se programski kod može automatski generirati. Različiti pristupi u odabranim radovima primjenjuju ontologije na različitim slojevima: model u model transformacije i model u kod transformacije. *Bachmann et al.* predlažu primjenu rječnika kao prikladnog instrumenta za rad s ontologijama u ranim fazama razvoja softvera te poluautomatsko mapiranje rječnika u UML modele (Bachmann, Hesse, Russ, & Kop, 2007). *Pahl* je predložio prilagodbu MDD-a na SOA arhitekturu koristeći ontologije kao alat za integraciju. U predloženom okviru MDD je razvojni

pristup primijenjen na SOA, a ontologije se koriste za semantičko modeliranje (Pahl, 2007b). *Soylu i De Causmaecker* predložili su metodologiju koja spaja MDD i razvoj temeljen na ontologijama (ODD). Oni predlažu korištenje OWL-a umjesto UML-a kao obećavajuće alternative koja omogućava kompleksno modeliranje s podrškom za zaključivanje te mogućnost korištenja ontologija u svakoj fazi razvojnog ciklusa softvera (Soylu & De Causmaecker, 2009). *Bauer i Roser* predstavili su arhitekturu razvojnog okruženja obogaćenog semantikom. Kombinirali su MDD pristup s ontološkim konceptima kroz uporabu ontologija u razvoju, modeliranju i transformaciji modela. Tzv. mostovi se koriste za sinkronizaciju sintakse modela i ontologija aplikacije. Ontologije aplikacije i mostovi mogu se generirati automatski iz modela i modeli se mogu generirati iz aplikacijskih ontologija (Bauer & Roser, 2006; Roser & Bauer, 2008). U već ranije predstavljenom pristupu SWB autora *Solis et al.*, zahtjevi sustava modelirani su pomoću ontologija, a infrastruktura sustava (u objektno-orijentiranom izvornom kodu) automatski se generira pomoću tog znanja. Ovaj pristup koristi unaprijed definiranu ontologiju koja sadrži osnovnu strukturu za transformaciju iz ontologije na objektno-orijentirani kod. Izvor se generira u dva sloja. Osnovni sloj automatski se generira i ne smije se mijenjati, dok prošireni sloj omogućuje programerima implementaciju određenih dodatnih funkcionalnosti (Solis et al., 2013). Autori *Parreiras et al.* razvili su DSL koji pruža razvojno okruženje za mapiranje ontoloških informacija i njegovih objektno-orijentiranih reprezentacija. Usmjeren je na koncepte domene i omogućuje generiranje OWL API-ja na zahtjev (Parreiras, Saathoff, Walter, Franz, & Staab, 2009).

3.3.5. Više-agentni sustavi

Nekoliko pristupa u odabranim radovima koristi MAS kako bi omogućili otkrivanje, sastavljanje i korištenje web usluga od strane agenata. MAS koji se koriste za otkrivanje semantičkih web usluga (SWS) obično koristi ontologije za opisivanje web usluga, ali i za olakšavanje komunikacije između agenata. Takav pristup predlaže *Cui-Mei* koji koristi OWL-S za semantički opis usluga, a WSDL opis usluge za sintaktičku opisivanje funkcionalnosti. Svaka funkcionalnost usluge pronađena u WSDL-u povezana je s odgovarajućim elementom u semantičkom modelu koji omogućavaju agentu interakciju s SWS-om (Cui-Mei, 2009). *Prabhu* je predstavio okvir koji dijeli posao kompozicije usluga između središnjeg registra i klijenta koji je postavio zahtjev. Agenti koriste algoritam za kompoziciju web usluga s pristupom odozdo prema gore, ali svi ostali poslovi ostavljeni su klijentu. Cilj ovog pristupa je minimizirati potrebne izmjene na postojećoj SOA arhitekturi (Prabhu, 2007). Autori *Garcia Coria et al.* pak predlažu sustav sa više agenata koji se temelji na virtualnim organizacijama. Više ontologija koristi se za modeliranje semantičkog znanja uključenih u web usluge, a

datoteka u formatu jezika izvršavanja poslovnih procesa (BPEL) koristi se za pohranu kompozicije web usluge koja zadovoljava zahtjev korisnika (García Coria et al., 2014).

Neki od odabranih pristupa primjenjuju ontologije u procesima projektiranja i implementacije sustava temeljenih na više agenata. *Jansen i Girardi* predstavili su metodologiju za izgradnju domenskih modela temeljenih na ontologiji, DSL-u i generatoru aplikacija kako bi se omogućilo generativno, ponovno korištenje MAS obitelji (Jansen & Girardi, 2006). *Nyulas et al.* demonstrirali su rješenje koje pruža ontologiju i softver za određivanje MAS funkcionalnosti pomoću tehnike razvrstavanja zadatak-metoda. Radi se o rješenju za projektiranje i implementaciju MAS, a podržana je izgradnja detaljnih modela i automatsko postavljanje agenata na temelju tih modela (Nyulas et al., 2008).

Višestruki agenti podržani ontologijama prepoznati su i kao potencijal za rješavanje različitih problema vezanih uz određene domene. Tako su *Zhao et al.* predložili model interakcije testiranja performansi softvera temeljen na pet rola agenata: agent za upravljanje, agent za snimanje, agent za ponavljanje, agent za praćenje i agent za analizu (Chongchong Zhao, Guoxin Ai, Xiao Yu, & Xiaofeng Wang, 2010). *Insaurralde* je čak predložio i upotrebu ontologija i agenata koji igraju ulogu programera, samostalnim izvođenjem aktivnosti razvoja softvera. Ovaj pristup je eksperimentalan i zalaže se za razvoj softvera na autonoman način, umjesto automatiziranog (Insaurralde, 2013).

3.3.6. Automatizacija testiranja

Kako razvoj temeljen na testiranju (TDD) stječe sve veći zamah u praksi, poneka istraživanja bave se načinima automatizacije generiranja testnih slučajeva. Budući da je testiranje softvera proces koji intenzivno koristi znanje, ontologije su primijenjene i u ovoj disciplini za generiranje testnih slučajeva. *Wang et al.* predložili su pristup koji uzima OWL-S kao ulaz u generator za testiranje i pretvara ga u model Petrijeve mreže za formalno predstavljanje strukture i ponašanja usluga. Testni slučajevi generiraju se na temelju analize ponašanja Petrijeve mreže i ontologijama ulaza, izlaza, preduvjeta i učinaka (Wang, Bai, Li, & Huang, 2007). *Li et al.* iskoristili su tehnike reversnog inženjeringa i analize koda kako bi kreirali ontologiju za testiranje grafičkog sučelja. Pravila za generiranje testnih slučajeva izvlače se iz iskustva testera, što znači da se snimaju i pohranjuju različiti unosi testera nad komponentama grafičkog korisničkog sučelja (GUI) (H. Li et al., 2009). Sličan je pristup i autora *Moser et al.* koji su ekspertna znanja pohranili u ontologiju i zatim dinamički prilagođavali generičku skriptu generatora, prema postavkama u modelu podataka. Ovaj pristup omogućava generiranje testnih slučajeva na sustavan način. Testni slučajevi generiraju se iz ontologije i izvoze u XML formatu (Moser, Moser, Dürr, & Biffl, 2010). *Nasser et al.* predlažu okvir koji olakšava određivanje kriterija pokrivenosti i pruža sredstva za generiranje testnih slučajeva za različite

artefakte. Ontologije određuju što treba testirati, a algoritmi za zaključivanje koriste se za odabir testnih ciljeva. Okvir koristi nekoliko ontologija, od kojih je jedna ontologija modela ponašanja koja predstavlja korišteni softverski artefakt i time omogućava generiranje testnih slučajeva na temelju različitih artefakata poput koda, dizajna ili zahtjeva (Nasser, Du, & MacIsaac, 2010).

Osim generiranja testnih slučajeva, ontologije su se pokazala korisnima i za samu provedbu automatskog testiranja te kontrolu nad procesom izvođenja. Autori *Sinha et al.* predložili su okvir u kojem se funkcionalni zahtjevi visoke razine koriste za automatsko generiranje testnih slučajeva. Zahtjevi se u početku specificiraju na prirodnom jeziku, a zatim se analiziraju i specificiraju pomoću formalne ontologije. U sklopu ovoga okvira autori su implementirali i alat za automatsko provođenje testnih slučajeva (Sinha, Pang, Martinez, Kuronen, & Vyatkin, 2015). Pristup autora *Nguyen et al.* koristi domenske ontologije i ontologije za interakciju između agenata, a testni slučajevi se generiraju i izvršavaju neprekidno koristeći više agenata (C. D. Nguyen, Perini, & Tonella, 2008).

3.3.7. Linije proizvoda u programiranju

Kreiranje i upravljanje skupom sličnih softverskih sustava iz zajedničkog skupa softverskih artefakata zahtijeva visoku razinu tehničkog znanja, ali i poznavanje domene primjene. Nekoliko odabranih radova je primijenilo ontologije u upravljanju značajkama, varijabilnosti i konfiguraciji SPL-ova. *Damaševičius et al.* predložili su prikaz varijabilnosti pomoću modela značajki obogaćenim domenskom ontologijom. Ti se modeli, pomoću tehnika meta-programiranja, transformiraju u specifikacije generativnih komponenata. Transformacija se u predloženom pristupu provodi na dvije razine. Najprije se obogaćeni model značajki transformira u meta-programski model, a potom se meta-programski model transformira u sam meta-program, odnosno programski kod (Damaševičius, Štuikys, & Toldinas, 2008). Autori *Duran-Limon et al.* predložili su okvir za automatsko generiranje proizvoda iz SPL arhitekture primjenom modela neovisnog o jeziku implementacije te MDD pristupa za generiranje proizvoda. Ontologija se u ovom pristupu primjenjuje za zaključivanje o generiranju transformacijskih pravila, temeljenih na odabiru značajki proizvoda. Model značajki i SPL arhitektura transformiraju se u ontologiju kako bi se omogućilo zaključivanje temeljem elemenata arhitekture i značajki proizvoda (Duran-Limon, Garcia-Rios, Castillo-Barrera, & Capilla, 2015). *Asikainen et al.* razvili su domensku ontologiju za modeliranje varijabilnosti SPL-ova. Ovaj pristup ujedinjuje značajke pristupa temeljenih na značajkama i arhitekturi te omogućuje modeliranje međusobnih veza. Razvijena ontologija može poslužiti kao osnova za razvoj jezika modeliranja i kao podrška alatima za upravljanje varijabilnostima (Asikainen et al., 2007).

3.3.8. Internet stvari

Budući da su pametni uređaji povezani na internet i pružaju usluge ljudima pa i drugim uređajima, primjena ontologija u IoT disciplini ima mnogo sličnosti s primjenom ontologija u SOA. Ontologije su uspješno primijenjene u IoT na način da omogućuju otkrivanje, korištenje i suradnju između različitih pametnih uređaja. Autori *Xu et al.* prezentirali su pristup modeliranja pametnih kućanstava pomoću četiri domenske ontologije za semantičko otkrivanje uređaja. Ontologija uređaja opisuje koncepte povezane s uređajem, a ontologija okoline opisuje prirodne i osobne profile okoline povezane s uređajem. Ontologija funkcija sadrži osnovno znanje koje uključuje funkcijski povezane koncepte i predloške za svaku kompozitnu funkciju. Ontologija preferencija opisuje ograničenja sustava i pravila o upotrebi i funkcijama uređaja. Generičko rješenje olakšava sastavljanje aplikacija na temelju korisničkih profila i raspoloživih uređaja, međutim generiranje koda za izvršnu platformu još nije implementirano (Xu et al., 2009). Sličan je pristup i autora *Lee et al.* koji su koristili tri ontologije: ontologija uređaja, ontologija funkcija i ontologija preferencija. Njihov sustav sastavlja plan funkcija za usluge koje koriste registrirane uređaje u kućanstvu. Korisnici mogu koristiti registrirane usluge i instalirati generirani kod u stroj za izvršavanje koji pruža okolinu za učitavanje i pokretanje usluga (Lee, Li, Tsai, Son, & Moon, 2009).

U par odabranih radova proučena je i primjena ontologija za modeliranje i razvoj IoT pametnih okuženja. *Katasonov i Palviainen* prezentirali su dizajnerski alat koji omogućuje kreiranje grafičkih modela pametnog prostora i okvir koji omogućava ODD. Ovaj alat temelji se na vizualnom kompozicijskom pristupu koji podiže razinu apstrakcije i podržava ponovnu uporabu postojećih softverskih komponenata i parcijalnih modela. Izvorni kod se generira iz modela i time se djelomično automatizira razvojni proces (Katasonov & Palviainen, 2010). U drugom radu autori *Palviainen et al.* predlažu korištenje dvije vrste API-ja u pametnim okruženjima: semantičko sučelje za programiranje aplikacija i izvršne API-je za krajnje korisnike. Cilj ovog pristupa je uključiti krajnjeg korisnika u razvoj na način da pomaže profesionalnim programerima i omogućavanje krajnjim korisnicima korištenje izvršnog API-ja (Palviainen, Kuusijarvi, & Ovaska, 2014).

3.4. Diskusija

U literaturi smo pronašli pristupe koji koriste ontologije za opisivanje komponenata i omogućavanje njihovog automatskog pronalaženja. Radi se o obećavajućem pristupu, jer mnogi web razvojni okviri danas obično imaju na stotine komponenata koje se mogu obogatiti semantičkim opisima, što pak može omogućiti njihovu lakšu kompoziciju. Komponente u razvoju web i mobilnih aplikacija obično sadrže i korisničko sučelje za interakciju s korisnikom. Nekoliko rješenja predlaže i automatsko generiranje UI komponenata te postavljanja istih na odgovarajuću poziciju na zaslonu. Iako su neki autori koristili dodatni sloj između ontologije i izvornog koda, većina rješenja automatski generira programski kod direktno iz ontologija. Izazov kod ovakvih pristupa jest omogućavanje učinkovite integracije prilagođenih i složenih operacija ili poslovnih slučajeva. U pristupima ovog tipa ontologija se može ponovno koristiti više puta čak i prije samog procesa generiranja koda. Nekoliko odabranih radova bavi se inkorporacijom ontologija u moderne GPL-ove pa čak i primjenom ontologija za generiranje softverskih artefakata korištenjem prirodnih jezičnih tekstova kao što su specifikacija zahtjeva, zahtjevi za promjenama ili prijave grešaka. Integracija ontologija u razvojne procese softvera danas je većinom na akademskoj razini i potrebno ju je testirati u praksi na velikim projektima. Kako bismo u potpunosti iskoristili prednosti koje donosi primjena ontologija u razvoju softvera, potrebne su nam standardizirane razvojne metode i odgovarajuća programska podrška. Jednako tako, da bismo bili učinkoviti u primjenu ontologija za razvoj softvera, moramo postići povrat inicijalnih ulaganja. To možemo postići na način da višestruko ponovno iskoristimo ontološko znanje u svim fazama razvojnog ciklusa softvera.

U literaturi smo pronašli značajan broj okvira i prototipnih alata za postizanje automatskog ili poluautomatskog otkrivanja i sastavljanja web usluga u SOA. Većina predloženih rješenja koristi WSDL-om opisane usluge obogaćene OWL-S semantikom ili implementiraju neke prilagođene ontologije koje koriste za semantičko opisivanje usluga. Prikazani prototipovi pokazali su da je djelotvorna kompozicija web usluga ostvariva, no realizacija složenih kompozicija web usluga još uvijek ostaje otvoreno pitanje. Osim kompleksnih usluga, neka od otvorenih pitanja su još i cjelovitost usluge, pouzdanost, kvaliteta te sigurnost. Semantičko otkrivanje kućanskih aparata i usluga koje pružaju mogu se u određenoj mjeri poistovjetiti sa pristupima za otkrivanje SWS.

U MAS pristupima pronađenim u literaturi ontologije su najčešće primijenjene kao sredstvo komunikacije i zajednički jezik koji omogućuje razumijevanje između agenata. Neki su pristupi

primijenili više agenata kako bi omogućili kompoziciju SWS. Iako još u ranoj eksperimentalnoj fazi, predloženi su i pristupi koji koriste ontološki potpomognute agente koji igraju ulogu razvojnih programera. Za razliku od SOA, u IoT predloženi pristupi koriste prilagođene ontologije za semantički opis znanja potrebnog za generiranje koda i komunikaciju među uređajima. Iako ontologije korištene u IoT nisu standardizirane, one u većini pristupa uključuju pojmove povezane s uređajem, funkcijama uređaja, kompozicijom te pravilima i ograničenjima. Neki pristupi pokazali su da primjena ontologija u razvoju IoT sustava može pružiti djelomičnu automatizaciju procesa razvoja pomoću API-ja visokih razina, ali i grafičkih modela.

Većina MDD pristupa koristi ontologije u svrhu transformacije modela. Ontologije su uglavnom korištene u računalno neovisnom sloju / modelu (CIM) kako bi se automatizirala transformacija od CIM-a do neovisnog modela platforme (PIM). Međutim, postoje i pristupi koji primjenjuju ontologije u sva tri MDD sloja: CIM, PIM i specifični model platforme (PSM). Neki pristupi koriste ontologije u kombinaciji s UML-om kako bi pružile podršku za zaključivanje, dok drugi koriste ontologije kao alternativu UML-u čija je ekspresivnost ograničena. Među MDD rješenjima pronašli smo razvojne okvire koji specificiraju korisničke zahtjeve kroz ontologije i zatim koriste to znanje kako bi automatski generirali izvorni kod. Osnovni nedostatak ovih pristupa, u trenutnom stanju, jest nemogućnost implementacije prilagođenih softverskih rješenja te proširivanje okvira novim značajkama. Ontologije su se pokazale korisnima i u TA, jer mogu obuhvatiti domensko znanje i znanje testiranja softvera. Postoje pristupi koji ekstrahiraju ekspertna znanja u ontologiju i zatim ga koriste za generiranje testnih slučajeva. Drugi pristupi pak kombiniraju ontologije domene i korisničkih zahtjeva sa MAS, kako bi generirali i kontinuirano pokretali testne slučajeve. Pristup koji puno obećava jest primjena ontologija za formalizirano zapisivanje značajki sustava i korisničkih zahtjeva. Razlog tome jest taj što se ontološko znanje može kasnije ponovno iskoristiti za generiranje različitih artefakata, od izvršivog programskog koda pa do testnih slučajeva. Čak i prije faze generiranja programskog koda, ontologije se mogu koristiti za olakšavanje komunikacije između različitih zainteresiranih strana te provjeravanje potpunosti i dosljednosti korisničkih zahtjeva. Ontologije su prodrle i u SPL disciplinu, gdje su predloženi pristupi koji primjenjuju ontologije kako bi obogatili dijagrame značajki, rasuđivali o arhitektonskim elementima i značajkama sustava te modelirali varijabilnosti među proizvodima. Modeliranje varijabilnosti obitelji proizvoda pomoću ontologija zanimljiv je pristup, jer značajke proizvoda obično se mogu mapirati na softverske komponente, što bi moglo omogućiti više razine automatizacije pri razvoju obitelji proizvoda.

U provedenom pregledu literature pronađeno je mnoštvo pristupa koji primjenjuju ontologije za automatsko ili polu-automatsko generiranje programskom koda različitih tipova aplikacija. Jedan od identificiranih nedostataka postojećih pristupa orijentiranih na web domenu su netransparentnosti istraživanja, jer korištene ontologije i izvorni programski kod nekih istraživanja nisu javno objavljeni. Osim toga korištene ontologije su često vrlo kompleksne, a nedovoljno dokumentirane i standardizirane. Većina pristupa pronađenih u literaturi definira samo korištene tehnologije i alate te ostaje nejasno kako se te ontologije uklapaju u procese razvoja softvera. Još jedan bitan uočeni nedostatak postojećih pristupa jest kombiniranje generiranog i prilagođenog programskog koda, gdje se generirani programski kod ne smije ručno modificirati, dok se specifične funkcionalnosti razvijaju ručno i nisu povezane sa procesom generiranja. Korištene ontologije predefinirane su za određene platforme i alate, a s obzirom da prototipske platforme obično nisu dobro dokumentirane, vrlo je teško proširiti takve ontologije novim elementima, kako bi se podržale nove značajke i specifične funkcionalnosti. Ovim istraživanjem planira se riješiti neke od identificiranih nedostataka na način da se dizajnira okvir za ontološki potpomognuti generativni razvoj u domeni razvoja obitelji poslovnih web aplikacija. Razvijene ontologije i izvorni kod generatora programskog koda biti će javno dostupni u sklopu rada. Osim detaljnog opisa ontologija i korištene tehnologije za razvoj generatora, biti će jasno definiran razvojni okvir, odnosno faze razvoja softvera temeljenog na predloženom pristupu. Umjesto potrebe za pisanjem prilagođenog programskog koda, okvir će biti dizajniran na način na bude proširiv, tj. da se nove funkcionalnosti mogu ugraditi u ontološki model i generatore programskog koda.

4. DIZAJN I RAZVOJ

4.1. Ontološki model

4.1.1. Motivacija

Fazu dizajna započinjemo razvojem ontološkog modela, najprije inicijalnih ontologija koje će zatim kroz demonstraciju biti proširene dodatnim elementima. Osnovna ideja dizajna kompletnog razvojnog okvira jest da bude proširiv. Samim time će dizajnirane inicijalne ontologije pružiti kostur za ponovnu upotrebu ontološki specificiranih artefakata, dok će nove elemente biti moguće dodavati prema potrebi specifičnih implementacija / proizvoda. Ontologija uključuje definicije pojmova u domeni i odnosima među njima koji se mogu računalno interpretirati. Glavna motivacija za razvoj ontologija u ovom doktorskom radu jest omogućavanje ponovne uporabe domenskog znanja. Umjesto ugrađivanja domenskih pretpostavki u programski kod, primijeniti ćemo ontološki model kako bismo te pretpostavke lakše pronašli, po potrebi promijenili i učinili razumljivima osobama koje nisu toliko stručne u razvoju softvera. Osobito se to odnosi na dionike iz poslovnog sustava. Jedan od ciljeva je svakako smanjiti jaz između programera i poslovnih korisnika, uzrokovan neformalnom i nepotpunom dokumentacijom projekta. Razvijene ontologije u ovom istraživanju nisu same po sebi svrha, već je osnovna svrha iskoristiti dizajnirane ontologije za specifikaciju korisničkih zahtjeva i značajki sustava te, iz te formalne ontološke specifikacije, automatski generirati izvršivi programski kod određenog proizvoda iz obitelji poslovnih web aplikacija.

Temeljem proučenih kategorizacija ontologija u poglavlju *1.1.5. Kategorizacija primjene ontologija*, zaključujemo kako će u ovom istraživanju biti kreirane domenske ontologije, jer će obuhvaćati znanje koje se primjenjuju u određenoj domeni (programsko inženjerstvo, odnosno razvoj obitelji poslovnih web aplikacija). Konkretnije, raditi će se o domenskim ontologijama zadataka, jer će se koristiti za obavljanje aktivnosti generiranja programskog koda, a mogu se ponovno upotrijebiti samo u ovoj specifičnoj domeni. Ontologije će biti korištene kao programski artefakti u razvoju pa stoga možemo zaključiti da se ovdje radi o pristupu razvoja upravljanog ontologijama.

4.1.2. Metodologija

Ontologije su razvijene prema metodologiji *Ontology Development 101* autora Noy i McGuinness. Ova metodologija odabrana je zbog svoje zrelosti, iterativnosti te velikog broja praktičnih smjernica i pravila za donošenje odluka prilikom izgradnje ontologija. Metodologija koristi iterativni pristup razvoju ontologija: počinje sa razvojem grube verzije ontologije, a zatim se preispituje i pročišćuje ontologija te dodaju pojedinosti. Autori metodologije ističu kako nema jedinstvenog pravilnog načina modeliranja domene, već najbolje rješenje gotovo uvijek ovisi o primjeni koju autor ima na umu i proširenjima koja predviđa. Koncepti u ontologiji trebaju biti bliski objektima (fizički ili logički) i odnosima u domeni od interesa. To su najvjerojatnije imenice (objekti) ili glagoli (odnosi) u rečenicama koje opisuju domenu. Nakon što definiramo inicijalnu verziju ontologija, možemo ju evaluirati pomoću softvera ili raspravom sa stručnjacima, ili oboje. Kao rezultat toga, gotovo sigurno ćemo morati preispitati početnu ontologiju. Ovaj proces iterativnog dizajna vjerojatno će se nastaviti kroz cijeli životni ciklus ontologije. Upravo takav životni ciklus predviđen je za ontologije koje će se koristiti u predloženom razvojnom okviru. Jedan od glavnih ciljeva dizajna okvira jest omogućavanje proširivanja ontološkog modela, kako bi se zadovoljili specifični korisnički zahtjeve ili dodale nove značajke obiteljima programskih proizvoda.

U praktičnom smislu, razvoj ontologije uključuje:

- Definiranje klasa u ontologiji,
- Uređenje klasa u taksonomskoj hijerarhiji,
- Određivanje atributa i opisivanje dopuštenih vrijednosti te
- Ispunjavanje vrijednosti atributa za instance klasa.

Odabrana metodologija definira sedam koraka razvoja ontologije:

1) Određivanje domene i opsega ontologije,
2) Razmatranje ponovne upotrebe postojećih ontologija,
3) Identifikacija važnih pojmova u ontologiji,
4) Definiranje klasa i klasne hijerarhije,
5) Određivanje svojstva klasa, odnosno definiranje atributa,
6) Definiranje svojstava i tipova atributa te

7) Kreiranje instanci (Noy & McGuinness, 2001).

4.1.3. Smjernice

Autori odabrane metodologije za razvoj ontologija definirali su i smjernice za provođenje pojedinih zadataka prilikom dizajniranja ontologije. Kod definiranja klasne hijerarhije autori opisuju odnos pod-klasa na sljedeći način: klasa A je pod-klasa od B ako je svaka instanca od A također instanca od B. Hijerarhija klasa je tranzitivna, što znači da ako je B pod-klasa od A i C je pod-klasa od B, onda je C i pod-klasa od A. Održavanje dosljedne hijerarhije klase može postati izazovno kada se domene i ontologije razvijaju. Klase u ontologiji predstavljaju pojmove u domeni, a ne riječi koje označavaju te pojmove. Stoga sinonimi za isti koncept ne predstavljaju različite klase. Sestrinske klase u hijerarhijskoj strukturi su klase koje su direktne pod-klase iste nad-klase. Sve sestrinske klase u hijerarhiji (osim onih u korijenu) moraju biti na istoj razini općenitosti. Ne postoje stroga pravila za broj izravnih pod-klasa koje određena klasa može imati. Međutim, mnoge dobro strukturirane ontologije imaju između dvije i desetak izravnih pod-klasa. Ako klasa ima samo jednu izravnu pod-klasu, možda postoji problem prilikom modeliranja ili ontologija nije potpuna. Ako pak postoji više od desetak pod-klasa za određenu klasu, možda ih je moguće dodatno kategorizirati.

Jedna od najtežih odluka koje treba donijeti tijekom modeliranja ontologija jest kada uvesti novu klasu, odnosno kada treba predstaviti razliku kroz različite vrijednosti atributa. Hijerarhija klasa u ontologiji ne bi smjela biti niti izrazito ugniježđena, niti vrlo ravna. Pronalaženje odgovarajuće ravnoteže nije lako, ali postoji nekoliko smjernica koje autori metodologije sugeriraju. Pod-klase obično imaju dodatne atribute ili ograničenja koje nad-klasa nema ili sudjeluju u različitim odnosima od nad-klase. Drugim riječima, novu klasu u hijerarhiju uvodimo obično samo kada postoji nešto što možemo reći o ovoj klasi, a da ne možemo reći o nad-klasama. Iznimka su klase u terminološkim hijerarhijama koje ne moraju uvoditi nova svojstva. Ako je razlika u domeni važna te ako smatramo da su objekti različite vrste, onda bi za njih svakako trebali kreirati novu klasu, a ne razlikovati ih kroz vrijednosti određenog atributa.

Odluka gdje završavaju klase u ontologiji, a započinju individualne instance, počinje s odlučivanjem što je najniža razina granuliranosti u reprezentaciji. Razina granuliranosti određena je potencijalnom primjenom ontologije. Pojedinačne instance su najspecifičniji pojmovi predstavljeni u bazi znanja. Jedna smjernica koju možemo slijediti jest da ako koncepti

tvore prirodnu hijerarhiju, onda bismo ih trebali predstavljati kao klase. Također, autori navode kako ontologija ne bi trebala sadržavati sve moguće informacije o domeni, tj. ne treba specijalizirati (ili generalizirati) više nego što je potrebno za primjenu ontologije.

Određivanje konvencija za imenovanje koncepata u ontologiji, a zatim strogo pridržavanje tih konvencija, ne samo da čini ontologiju jednostavnijim za razumijevanje, nego također pomaže izbjeći neke uobičajene pogreške u modeliranju. Možemo uvelike poboljšati čitljivost ontologije ako dosljedno koristimo mala i velika početna slova za nazive koncepta. Na primjer, uobičajeno je koristiti veliko početno slovo za nazive klasa, a malo početno slovo za nazive atributa. Neke metodologije temeljene na znanju sugeriraju upotrebu prefiksa i sufiksa u imenovanja klasa i atributa kako bi ih lakše razlikovali (Noy & McGuinness, 2001).

4.1.4. Priprema za razvoj

4.1.4.1. Određivanje domene i opsega ontologije

U ovoj, prvoj fazi razvoja ontologije prema odabranoj metodologiji, potrebno je odgovoriti na nekoliko osnovnih pitanja o ontologiji koju ćemo kreirati. Odgovori na ova pitanja pomažu ograničiti opseg modela:

- Koja je domena koju će obuhvatiti ontologija?
- Za što ćemo koristiti ontologiju?
- Za koju vrstu pitanja bi nam ontologija trebala pružiti odgovore?
- Tko će koristiti i održavati ontologiju?

Domena koju će obuhvatiti ontologija jest razvoj obitelji poslovnih web aplikacija, što znači da će ontologija sadržavati znanje o korisničkim zahtjevima i softverskim artefaktima korištenim u razvoju poslovnih aplikacija baziranih na web obrascima. Ontologije će se koristiti za specifikaciju pojedinih programskih proizvoda iz obitelji. Osim specifikacije, ontologije će biti korištene i za automatsko generiranje programskog koda proizvoda. Kreirana ontologija treba pružiti odgovore na slijedeća pitanja:

- Za kojeg klijenta razvijamo određeni projekt / proizvod?
- Koji su korisnički zahtjevi te kriteriji prihvaćanja za pojedini zahtjev?
- Kakav tip repozitorija i koja struktura podataka će se koristiti za pohranu podataka?
- Koji web obrasci će biti dostupni korisnicima aplikacije kroz izbornik?

- Koji tipovi komponenata i drugi softverski artefakti će biti korišteni na pojedinim obrascima?
- Na kojoj lokaciji na ekranu će pojedina komponenta korisničkog sučelja biti prikazana?

Dodavati nove elemente te održavati ontologiju mogu isključivo programeri, jer će isti biti zaduženi i za njihovo mapiranje na generatore programskog koda. Međutim, gotovu ontologu primarno će koristiti poslovni analitičari, a po potrebi i educirani poslovni korisnici. Ontološki model u kombinaciji sa generatorima programskog koda moći će se koristiti za brzo prototipiranje, ali i generiranje kompletnih web aplikacija.

4.1.4.2. Razmatranje ponovne upotrebe postojećih ontologija

Korištena metodologija sugerira da je uvijek vrijedno razmotriti što je netko drugi već razvio i provjeriti možemo li precizirati i proširiti postojeće ontologije za našu domenu i specifični zadatak. Mnoge ontologije dostupne su u elektroničkom obliku i mogu se uvesti u razvojno okruženje za razvoj ontologija. Konkretno, u ovom doktorskom radu korišteno je razvojno okruženje *Protege*[8], verzije 4.3.0.

U skladu sa metodologijom u ovoj fazi pretraženo je nekoliko baza ontologija uključujući DAML[9], Swoogle[10] i Watson[11]. Međutim, nije pronađena gotova ontologiju koja bi se mogla iskoristiti za ovu specifičnu primjenu. Osim toga, s obzirom da je osnovni cilj omogućiti kasnije proširenje ontologije, u ovoj inicijalnoj fazi dizajnirati ćemo što je moguće jednostavnije ontologije, koje će se kasnije iterativno proširivati novim atributima, a po potrebi i klasama.

4.1.4.3. Identifikacija važnih pojmova u ontologiji

Sljedeći korak prema odabranoj metodologiji jest kreirati popis svih pojmova koje želimo opisati ontologijom. U ovoj fazi dizajna ontologije trebamo odgovoriti na pitanja:

- Koji su pojmovi o kojima želimo pričati?
- Koja svojstva imaju ti pojmovi?
- Što bismo željeli reći o njima?

[8] Dostupno 29.06.2018. na https://protege.stanford.edu/
[9] Dostupno 29.06.2018. na http://www.daml.org/ontologies/keyword.html
[10] Dostupno 29.06.2018. na http://swoogle.umbc.edu/2006/
[11] Dostupno 29.06.2018 na http://watson.kmi.open.ac.uk/WatsonWUI/index.html

U *Tablici 1* definirani su osnovni pojmovi koji će biti korišteni u ontološkom modelu. Radi se o početnom popisu pojmova i atributa koji će s iterativno biti proširivani, a pojedini pojmovi biti će dodatno kategorizirani u hijerarhiju klasa.

Tablica 1: Osnovni pojmovi u ontologiji

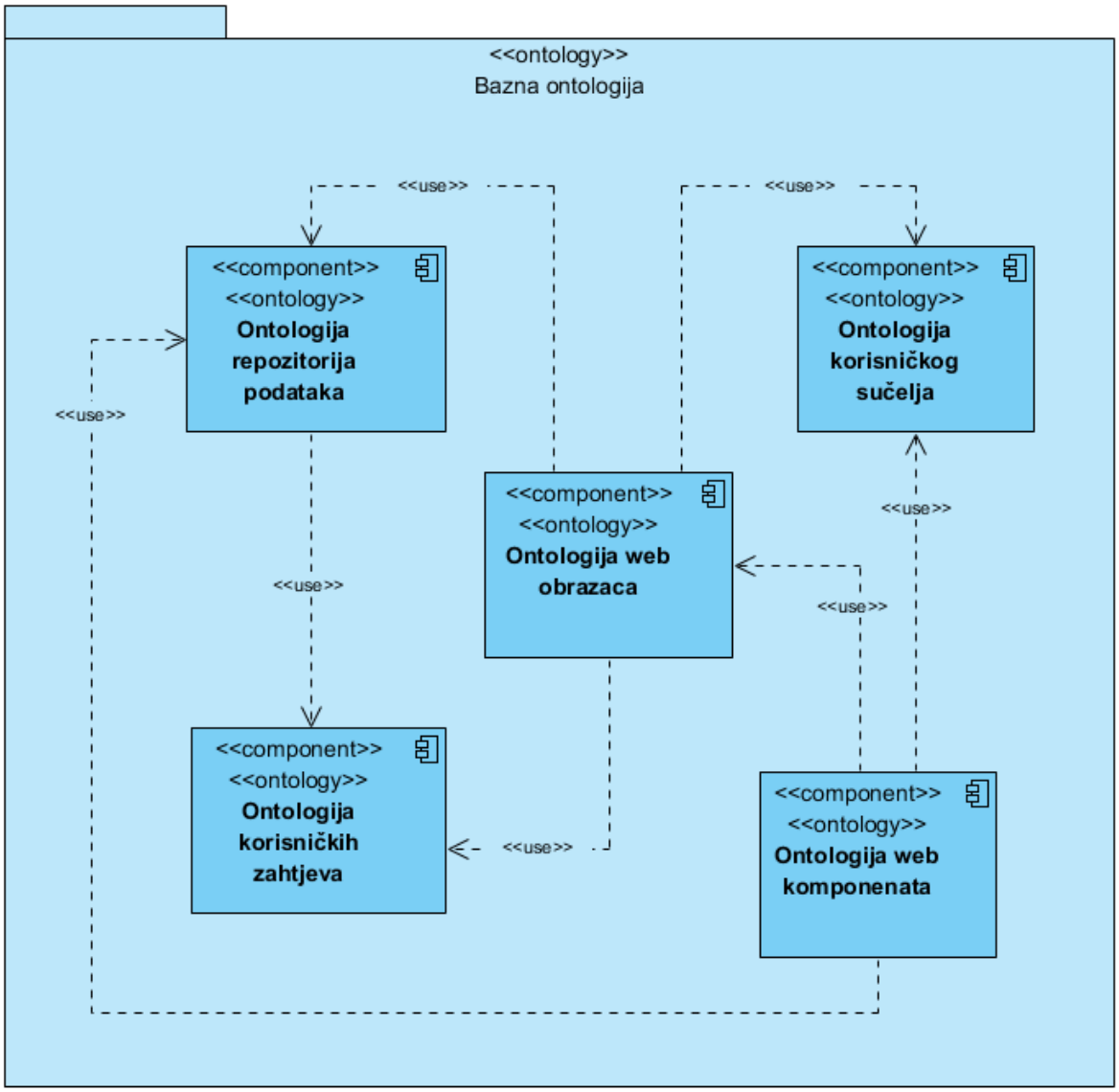
Pojam	**Svojstva**	**Opis**
Klijent	Naziv, adresa, email, tvrtka…	Korisnik za kojeg razvijamo projekt / aplikaciju
Projekt	Identifikator, naziv, opis…	Aplikacija koju razvijamo, odnosno specificiramo pomoću ontologije
Korisnički zahtjev	Opis…	Zahtjevi korisnika koje treba implementirati. Sadrži korisničke priče i kriterije prihvaćanja.
Repozitorij	Naziv, URL, upravljački program, korisničko ime i lozinka…	Repozitorij za trajnu pohranu podatka. Ako se radi o klasičnoj relacijskoj bazi podataka tada sadržava tablice i atribute.
Izbornik	Identifikator, naslov…	Grupa web obrazaca koja se prikazuje u glavnom izborniku aplikacije.
Obrazac	Identifikator, naslov…	Web obrazac za prikaz i/ili unos podataka kojem korisnik pristupa kroz izbornik.
Komponenta	Identifikator, naslov, tip…	Web komponenta za prikaz ili unos podataka. Tipično tablica za prikaz podataka, unosno polje za unos različitih tipova podataka (tekst, broj, datum, itd.) ili gumb za okidanje određene akcije (spremanje, brisanje, itd.)
Korisničko sučelje	Boja pozadine, boja teksta, visina, dužina, pozicija…	Definicija pozicije i izgleda web komponenata na ekranu

4.1.5. Struktura ontologija i pravila imenovanja

S obzirom da želimo postići visoku razinu ponovne iskoristivosti, ali i proširivosti inicijalno kreirane ontologije, odlučili smo se za modularni pristup razvoju. To zapravo znači da ćemo zasebno definirati pet inicijalnih ontologija fokusiranih na jednu pod-domenu:

- Ontologija korisničkih zahtjeva,
- Ontologija repozitorija podataka,
- Ontologija web okvira,
- Ontologija web komponenata i
- Ontologija korisničkog sučelja.

Iako će inicijalne ontologije biti prilično jednostavne, na ovakav modularni razvoj ontologija odlučili smo se zbog mogućnosti njihovog proširivanja u budućnosti, kada će se primjenjivati za različite projekte, odnosno proizvode. Moderni alati za razvoj ontologija, poput korištenog *Protege*-a, omogućuju uvoz ontologija tako da na jednostavan način možemo uvesti svih pet ontologija u jednu baznu ontologiju, u kojoj će biti definirane veze između njih i u kojoj će se kasnije kreirati instance. Osim toga, ovakva podjela na pod-domene omogućiti će budućim novim korisnicima lakše razumijevanje ontološkog modela. U ovoj fazi istraživanja neće biti kreirane instance, već samo klase i atributi. Instance će kasnije biti definirane za svaku od studija slučaja u fazi demonstracije, poglavlje *5. Demonstracija i evaluacija*. Također, ontologija predstavljena u ovom poglavlju nije finalna, već će novi elementi (uglavnom atributi) biti dodani po potrebi prilikom provedbi studija slučaja. Na *Slici 3* prikazane su komponente bazne ontologije te veze među njezinim komponentama, odnosno ontologijama repozitorija podataka, korisničkih zahtjeva, web obrazaca, web komponenata i korisničkog sučelja.



Slika 3: Komponente bazne ontologije

Temeljem ranije spomenutih smjernica autora metodologije, definirali smo slijedeća pravila imenovanja za elemente ontologija:

- Nazivi klasa, atributa i instanci biti će na engleskom jeziku, s obzirom da će na standardnom engleskom jeziku biti pisan i programski kod generatora.
- Nazivi klasa biti će bez razmaka i interpunkcijskih znakova, s velikim početnim slovom svake riječi (engl. *upper camel case*).
    - Npr. *Client, Component, Form*, itd.
- Nazivi atributa biti će bez razmaka i interpunkcijskih znakova, s velikim početnim slovom svake osim prve riječi (engl. *lower camel case*).

    - Npr. *backgroundColor, repositoryName,* itd.

- Nazivi instanci biti će bez razmaka i interpunkcijskih znakova, s velikim početnim slovom prve riječi koje su međusobno razdvojene podvlakama (engl. *snake case* ili *underscores*).
    - Npr. *c1_tab_Tax_schemas, c1_tab_Persons, c1_for_Edit_articles,* itd.

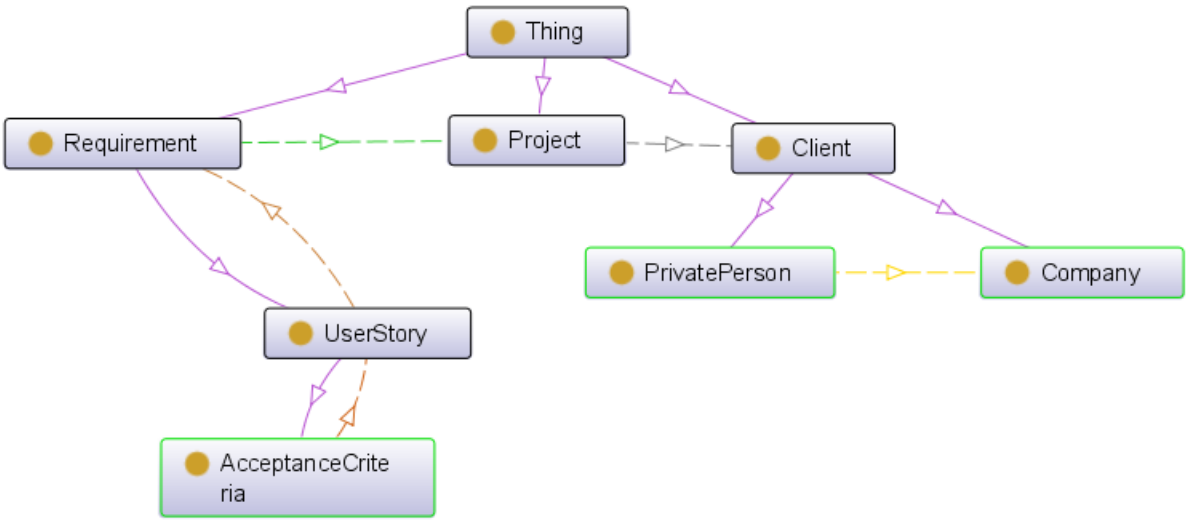
Osim toga, za lakše snalaženje u alatu *Protege,* kada postoji velik broj atributa i instanci, definirali smo i slijedeća dodatna pravila imenovanja:

- Prva riječ atributa treba označavati na koju domenu (klasu) se odnosi, osim ako se ne radi o atributu primjenjivom na više domena.
    - Npr. *clientAddress, clientEmail, columnType, columnTitle,* itd.
- Nazivi atributa koji sadržavaju Booleove vrijednosti imati će dodan fiksni prefiks „*is*“.
    - Npr. *isColumnValueNotNull, isInputFieldDisabled,* itd.
- Sve instance imati će prefiks od pet znakova od kojih prva 2 znaka označavaju projekt, a slijedeća tri znaka klasu kojoj instance pripadaju.
    - Npr. u nazivu instance *c1_tab_Tax_schemas, c1* označava prvu studiju slučaja (engl. *Case 1*), a *tab* označava klasu tablica (engl. *Table*). Nakon toga slijedi sam naziv instance, odnosno *Tax_schemas*.

4.1.6. Definiranje klasa i klasne hijerarhije

4.1.6.1. Ontologija korisničkih zahtjeva

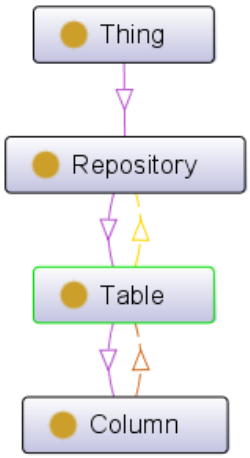
Kako i priliči informatičkim projektima, definiranje klasa započinjemo ontologijom korisničkih zahtjeva. No prije samih korisničkih zahtjeva potrebno je definirati projekt, odnosno aplikaciju i klijenta za kojeg se projekt / proizvod izrađuje. U ovoj jednostavnoj klasnoj hijerarhiji prikazanoj na *Slici 4* s desne strane vidi se da klasa koja predstavlja klijenta ima pod-klase za privatne osobe i tvrtke, s time da privatna osoba može biti član jedne ili više tvrtke. Svaki projekt pripada određenom klijentu, dok se za pojedini projekt definira više korisničkih zahtjeva. Korisnički zahtjevi najprije se definiraju kao grube priče (engl. *epic story*), a kasnije se oplemenjuju i dijele na više manjih i detaljnijih korisničkih priča. Za svaku korisničku priču definiraju se kriteriji prihvaćanja. Kriteriji prihvaćanja iznimno su važni, jer su njihovim zadovoljavanjem smatra da je korisnička priča implementirana, tj. da je specifičan korisnički zahtjev zadovoljan. Ovdje se radi o danas prilično standardnom agilnom pristupu definiranja korisničkih zahtjeva. Osnovni cilj je da korisnički zahtjevi budu što je moguće kompletni te da je razvojnom timu jasno kako zadovoljiti navedene zahtjeve.



Slika 4: Klasna hijerarhija ontologije korisničkih zahtjeva

4.1.6.2. Ontologija repozitorija podataka

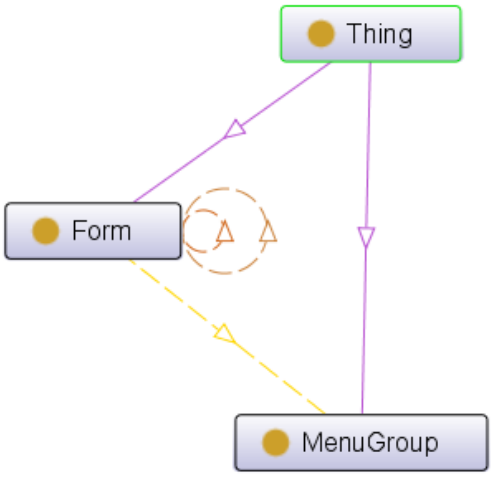
Ontologija repozitorija podataka ima jednostavnu strukturu klasa, jer je u inicijalnoj verziji podržan rad samo sa relacijskim bazama podataka. Stoga klasna hijerarhija prikazana na *Slici 5*, osim reprezentacije repozitorija podataka, sadrži još samo klase za opis tablica i stupaca u relacijskim bazama podataka. Relacijska baza podataka sadrži više tablica, dok svaka tablica sadrži jedan ili više stupaca. Po potrebi se ova ontologija može proširiti na način da se definira drugačija struktura za pohranu podataka.



Slika 5: Klasna hijerarhija ontologije repozitorija podataka

4.1.6.3. Ontologija web okvira

Ontologija web okvira također ima jednostavnu strukturu klasa, prikazanu na *Slici 6*. Unutar ove ontologije definiraju se sami okviri koji korisnik koristi za unos ili prikaz podataka te grupe okvira koje se koriste za prikaz opcija u glavnom izborniku aplikacije. Na slici je također vidljivo da obrazac može imati referencu na drugi obrazac. Ova veza dodana je kako bismo mogli implementirati slučajeve kada obrazac za pregled podataka treba preusmjeriti korisnika na obrazac za uređivanje podataka, i obrnuto.

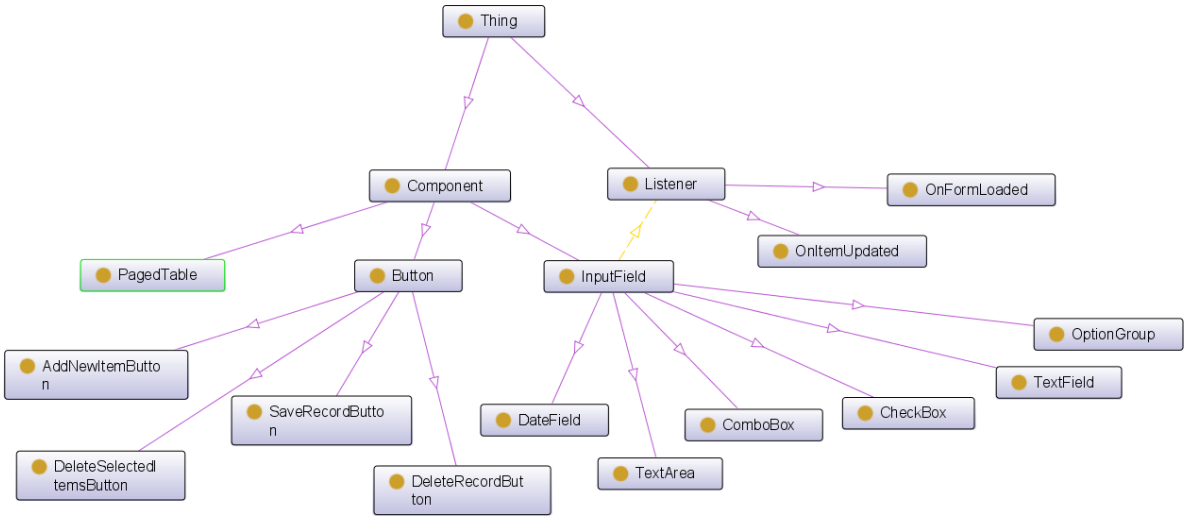


Slika 6: Klasna hijerarhija ontologije web okvira

4.1.6.4. Ontologija web komponenata

Ontologija web komponenata najkompleksnija je od pet kreiranih ontologija, a njezina hijerarhijska struktura prikazana je na *Slici 7*. Sastoji se od komponenata i slušača, dok se komponente dalje dijele na unosna polja, akcijske gumbe i tablice sa paginacijom[12]. Unosna polja nadalje smo podijelili na standardne tipove unosnih polja kakve koristimo na web obrascima, poput tekstualnih unosnih polja, numeričkih unosnih polja, kontrolnih polja, datumskih unosnih polja, itd. Akcijske gumbe podijelili smo na pod-klase prema akcijama koje okidaju. U trenutnoj verziji ontologije to su akcije za spremanje i brisanje zapisa u repozitoriju, odnosno spremanje i brisanje njihovih stavaka (povezanih zapisa). Također, u počenoj verziji ontologije, podržana su dva tipa slušača događaja i to za događaje obrazac učitan i stavka ažurirana. Nove tipove unosnih polja, akcijske gumbe i slušače moguće je vrlo jednostavno dodati u postojeću strukturu ontologije.

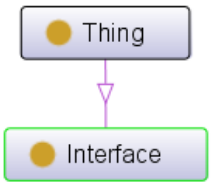
[12] Naziv *„tablica sa paginacijom“* koristi se kako bismo razlikovali klasu komponente tablice za prikaz podataka na obrascu od klase tablice u bazi podataka.



Slika 7: Klasna hijerarhija ontologije web komponenata

4.1.6.5. Ontologija korisničkog sučelja

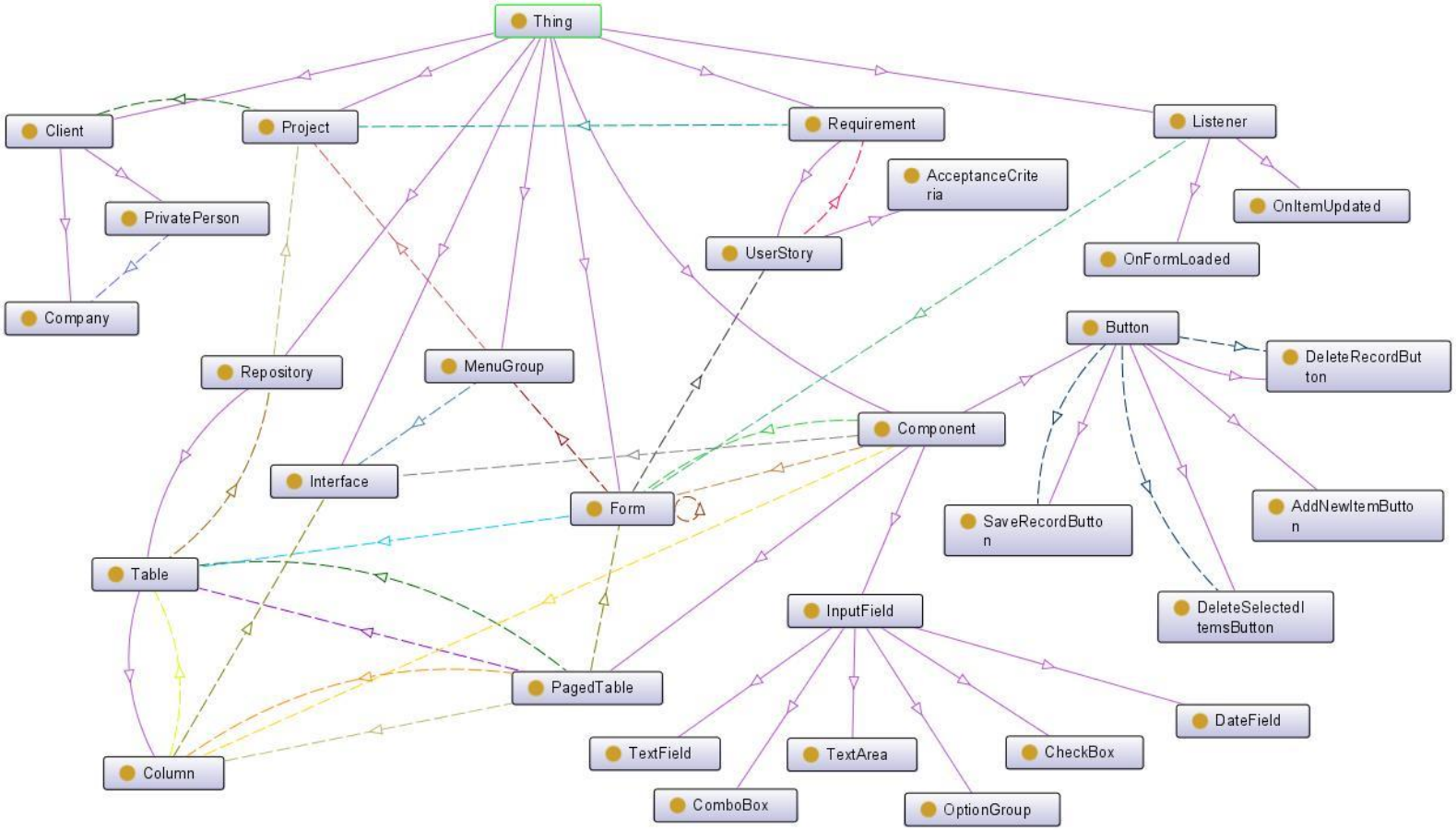
Ontologija korisničkog sučelja najjednostavnija je od svih kreiranih ontologija i sastoji se samo od jedne klase koja definira atribute izgleda i pozicioniranja komponenata na ekranu. Hijerarhijska struktura ove ontologije prikazana je na *Slici 8*. Ontologija korisničkog sučelja mogla bi biti puno kompleksnija, no fokus ovog doktorskog rada nije korisničko iskustvo, niti najbolje moguće pozicioniranje elemenata na ekranu. Ipak, izdvojili smo ontologiju korisničkog sučelja kao zasebni model kako bi se u budućnosti mogla lakše proširiti dodatnim elementima.



Slika 8: Klasna hijerarhija ontologije korisničkog sučelja

4.1.6.6. Bazna ontologija (temeljni model)

Sve navedene ontologije uvezli smo u jednu baznu ontologiju, čija je hijerarhijska struktura prikazana na *Slici 9*. U baznoj ontologiji definirani su preostali objektni atributi koji povezuju klase iz uvezenih ontologija,



Slika 9: Hijerarhijska struktura bazne ontologije

4.1.7. Definiranje atributa i njihovih svojstava

4.1.7.1. Ontologija korisničkih zahtjeva

Tablica 2 prikazuje podatkovne atribute ontologije korisničkih zahtjeva. Radi se o atributima za opis klijenata, privatnih osoba i tvrtki te atributima za opis projekta, odnosno aplikacije koja se izrađuje. Korisnički zahtjevi, uključujući korisničke priče i kriterije prihvaćanja opisuju se tekstualno, korištenjem istog opisnog atributa. *Tablica 3* prikazuje objektne atribute ontologije korisničkih zahtjeva. Ovi atributi korišteni su za definiranje veza među klasama i vidljivi su kao strelice na dijagramu klasne hijerarhije, odnosno *Slici 4*. Kompletna ontologija korisničkih zahtjeva izvezena u RDF / XML formatu dostupna je u *Dodatku B* ovog doktorskog rada.

Tablica 2: Podatkovni atributi ontologije korisničkih zahtjeva

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Ime	*personFirstName*	*PrivatePerson*	*String*
Prezime	*personFirstName*	*PrivatePerson*	*String*
Naziv tvrtke	*companyName*	*Company*	*String*
Adresa	*clientAdress*	*Client*	*String*
Web stranica	*clientWebsite*	*Client*	*String*
Telefonski broj	*clientTelephone*	*Client*	*String*
Adresa e-pošte	*cllientEmail*	*Client*	*String*
Opis	*description*	*Project, Requirement*	*String*
Identifikator projekta	*projectID*	*Project*	*String*
Naziv projekta	*projectTitle*	*Project*	*String*
Osnovni paket	*projectBasePackage*	*Project*	*String*

Tablica 3: Objektni atributi ontologije korisničkih zahtjeva

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Projekt za klijenta	*projectForClient*	*Project*	*Client*
Zaposlenik tvrtke	*employeeOfCompany*	*PrivatePerson*	*Comany*
Korisnički zahtjev za projekt	*requirementForProject*	*Requirement*	*Project*
Korisnička priča za korisnički zahtjev	*storyIsPartOfRequirement*	*UserStory*	*Requirement*
Kriterij prihvaćanja korisničke priče	*acceptanceCriteriaForStory*	*AcceptanceCriteria*	*UserStory*

4.1.7.2. Ontologija repozitorija podataka

Tablica 4 prikazuje podatkovne atribute ontologije repozitorija podatka. Radi se o atributima za opis strukture relacijske baze podataka, odnosno repozitorija, tablica i stupaca. Osobama upoznatima s domenom relacijskih baza podataka, svi atributi ove ontologije trebali bi biti poznati te stoga nećemo ulaziti u detaljnije opise atributa. *Tablica 5* prikazuje objektne atribute ontologije repozitorija podatka. Ovi atributi korišteni su za definiranje veza među klasama i vidljivi su kao strelice na dijagramu klasne hijerarhije, odnosno *Slici 5*. Kompletna ontologija repozitorija podataka izvezena u RDF / XML formatu dostupna je u *Dodatku C* ovog doktorskog rada.

Tablica 4: Podatkovni atributi ontologije repozitorija podataka

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Naziv repozitorija	*repositoryName*	*Repository*	*String*
Adresa repozitorija	*repositoryURL*	*Repository*	*String*
Upravljački program	*respositoryDriver*	*Repository*	*String*
Korisničko ime	*respositoryUsername*	*Repository*	*String*
Lozinka	*respositoryPassword*	*Repository*	*String*
Naziv tablica	*tableName*	*Table*	*String*
Identifikator stupca	*columnID*	*Column*	*String*

Naslov stupca	*columnTitle*	*Column*	*String*
Tip stupca	*columnType*	*Column*	*String*
Pridruživanje stupca po šifri	*columnJoinByID*	*Column*	*String*
Oznaka identifikatora	*isColumnValueAnIdentifier*	*Column*	*Boolean*
Oznaka generirane vrijednosti	*isColumnValueGenerated*	*Column*	*Boolean*
Oznaka zabrane nedefiniranih vrijednosti	*isColumnValueNotNull*	*Column*	*Boolean*
Oznaka jedinstvene vrijednosti	*isColumnValueUnique*	*Column*	*Boolean*
Oznaka pridruživanja više-na-jedan	*isColumnJoinedManyToOne*	*Column*	*Boolean*
Oznaka pridruživanja jedan-na-više	*isColumnJoinedOneToMany*	*Column*	*Boolean*

Tablica 5: Objektni atributi ontologije repozitorija podataka

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Stupac je sadržan u tablici	*columnIsPartOfTable*	*Column*	*Table*
Tablica pripada repozitoriju	*tableIsPartOfRepository*	*Table*	*Repozitorij*

4.1.7.3. Ontologija web okvira

Tablica 6 prikazuje podatkovne atribute ontologije web okvira. Radi se o atributima za opis obrazaca i načina njihovog grupiranja u glavni izbornik aplikacije. Samo jedan obrazac unutar određenog projekta može biti označen kao naslovni obrazac, a moguće je i definirati grupe obrazaca koje će biti skrivene, tj. neće se prikazivati u glavnom izborniku. *Tablica 7* prikazuje objektne atribute ontologije web okvira. Ovi atributi korišteni su za definiranje veza među klasama i vidljivi su kao strelice na dijagramu klasne hijerarhije, odnosno *Slici 6*. Kompletna ontologija web okvira izvezena u RDF / XML formatu dostupna je u *Dodatku D* ovog doktorskog rada.

Tablica 6: Podatkovni atributi ontologije web okvira

Atribut	**Naziv (engl.)**	**Domena (engl.)**	**Opseg (engl.)**
Identifikator obrasca	*formID*	*Form*	*String*
Naslov obrasca	*formTitle*	*Form*	*String*
Oznaka naslovne forme	*isFormHome*	*Form*	*Boolean*
Naslov grupe	*menuGroupTitle*	*MenuGroup*	*String*
Oznaka skrivene grupe	*isMenuGroupHidden*	*MenuGroup*	*Boolean*

Tablica 7: Objektni atributi ontologije web okvira

Atribut	**Naziv (engl.)**	**Domena (engl.)**	**Opseg (engl.)**
Obrazac pripada grupi	*formIsPartOfMenuGroup*	*Form*	*MenuGroup*
Zasebni obrazac za ažuriranje podataka	*formHasSeparateEdit*	*Form*	*Form*
Zasebni obrazac za pregled podataka	*formHasSeparateView*	*Form*	*Form*

4.1.7.4. Ontologija web komponenata

Tablica 8 prikazuje podatkovne atribute ontologije web komponenata. Radi se o atributima za opis komponenata generalno te atributima za dodatni opis unosnih polja kako bismo omogućili implementaciju različitih tipova validacija (poput obaveznih polja). *Tablica 9* prikazuje objektne atribute ontologije web komponenata. Kod ove ontologije radi se samo o jednom objektom atributu za povezivanje unosnih polja sa slušačima događaja, vidljivom na dijagramu klasne hijerarhije, odnosno *Slici 7*. Kompletna ontologija web komponenata izvezena u RDF / XML formatu dostupna je u *Dodatku E* ovog doktorskog rada.

Tablica 8: Podatkovni atributi ontologije web komponenata

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Identifikator komponente	*componentID*	*Component*	*String*
Naslov komponente	*componentLabel*	*Component*	*String*
Oznaka obaveznog unosa polja	*inputFieldRequiredLabel*	*InputField*	*String*
Poruka neuspješne validacije polja	*inputFieldValidatorMessage*	*InputField*	*String*
Regularni izraz za validaciju polja	*inputFieldValidatorRegeEx*	*InputField*	*String*
Dužina polja	*inputFieldWidth*	*InputField*	*Integer*
Oznaka onemogućenog polja za unos	*isInputFieldDisabled*	*InputField*	*Boolean*

Tablica 9: Objektni atributi ontologije web komponenata

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Ažuriranje polja temeljem slušača	*inputFieldUpdateListener*	*InputField*	*Listener*

4.1.7.5. Ontologija korisničkog sučelja

Tablica 10 prikazuje podatkovne atribute ontologije korisničkog sučelja. Radi se o atributima za definiranje dimenzija, pozicije i boje UI komponenata na ekranu korisnika. Ova ontologija u početnoj verziji sadrži samo jednu klasu te stoga ne sadrži niti jedan objektni atribut. Kompletna ontologija korisničkog sučelja izvezena u RDF / XML formatu dostupna je u *Dodatku F* ovog doktorskog rada.

Tablica 10: Podatkovni atributi ontologije korisničkog sučelja

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Boja pozadine	*backgroundColor*	*Interface*	*String*
Boja teksta	*textColor*	*Interface*	*String*
Visina	*uiHeightInPixels*	*Interface*	*Integer*
Širina	*uiWidthInPixels*	*Interface*	*Integer*
Indeks pozicije	*uiPositionIndex*	*Interface*	*Integer*

4.1.7.6. Bazna ontologija

Tablica 11 prikazuje podatkovne atribute bazne ontologije koji predstavljaju poveznicu između klasa iz ontologija korisničkih zahtjeva, repozitorija podataka, web okvira, web komponenata i korisničkog sučelja. Kompletna bazna ontologija, uključujući sve atribute te instance dodane za potrebe demonstracije (studije slučaja), izvezena u RDF / XML formatu dostupna je u *Dodatku G* ovog doktorskog rada.

Tablica 11: Objektni atributi bazne ontologije

Atribut	Naziv (engl.)	Domena (engl.)	Opseg (engl.)
Komponenta koristi stupac podataka	*componentForDataColumn*	*Component*	*Column*
Komponenata na obrascu	*componentForForm*	*Component*	*Form*
Obrazac za korisničku priču	*formForUserStory*	*Form*	*UserStory*
Slušač za obrazac	*listenerForForm*	*Listener*	*Form*

Grupa obrazaca na projektu	*menuGroupForProject*	*MenuGroup*	*Project*
Tablica sa paginacijom koristi tablicu u repozitoriju	*pagedTableMainTable*	*PagedTable*	*Table*
Tablica sa paginacijom prikazuje stupac	*pagedTableShowColumn*	*PagedTable*	*Column*
Repozitorij podataka pripada projektu	*repositoryForProject*	*Repository*	*Project*
Korisnički zahtjev pripada projektu	*requirementForProject*	*Requirement*	*Project*
Sučelje za stupac	*uiForColumn*	*Intereface*	*Column*
Sučelje za komponentu	*uiForComponent*	*Intereface*	*Component*
Sučelje za grupu obrazaca	*uiForMenuGroup*	*Intereface*	*MenuGroup*

4.2. Generatori programskog koda

4.2.1. Korištena tehnologija

Prije nego započnemo s dizajnom te kasnije i samim razvojem, odnosno programiranjem generatora programskog koda, moramo obrazložiti odabir tehnologije. Programski jezik u kojem ćemo razvijati generatore biti će *Java* (korištena verzije 8, nadogradnja 151)[13] i to iz nekoliko razloga. Prvi razlog jest višegodišnje iskustvo autora u radu s ovim programskim jezikom te njegova globalna popularnost i prihvaćenost. Drugi razlog je da se radi o objektno-orijentiranom programskom jeziku s vrlo dobrom podrškom za rad s predlošcima programskog koda te rad s ontologijama.

Nakon što smo odlučili koji programski jezik ćemo koristiti za razvoj generatora, prvi problem s kojim se susrećemo jest kako pristupiti ontološkom modelu i podacima iz generatora koda. Srećom, već postoji *Java* okvir za semantički web koji je otvorenog koda i koji možemo iskoristiti upravo u tu svrhu. Radi se o *Apache Jena* (korištena verzija 3.2.0)[14] okviru koji se, između ostalog, može koristiti za čitanje ontoloških modela iz RDF / XML datoteka ili URL-ova. Isti okvir omogućava i korištenje SPARQL upita za dohvaćanje podataka spremljenih u ontološkom modelu. Upravo će ta značajka biti intenzivno korištena u generatorima programskog koda koji će stvarati izvršivi kod temeljem podataka iz ontološkog modela, odnosno ontološke specifikacije proizvoda.

Sada kada imamo tehnologiju kojom možemo pročitati podatke iz ontološkog modela, sljedeći problem koji se nameće jest kako transformirati model u izvršivi programski kod. Generatori programskog koda iz ontološkog modela mogu se izraditi u čistom *Java* programskom jeziku pomoću *Jena* okvira, bez potrebe za bilo kakvim dodatnim okvirima i bibliotekama. Unatoč tome, kako bismo ubrzali i olakšali razvoj generatora, upotrijebiti ćemo još nekoliko dodatnih alata i okvira. Prvi od njih je *Apache Velocity* (korištena verzija 1.7)[15], a radi se o okviru za primjenu predložaka u *Java* programskom jeziku. Ovaj okvir omogućava programerima da koriste jednostavni jezik predloška koji unutar datoteka predložaka omogućuje pristup referentnim objektima definiranim u *Java* kodu. Sljedeći korišteni okvir je *Vaadin* (korištena

[13] Dostupno 07.07.2018. na http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
[14] Dostupno 07.07.2018. na https://jena.apache.org/download/
[15] Dostupno 07.07.2018. na https://velocity.apache.org/download.cgi

verzija 7.7.2)[16] koji omogućava korištenje velikog broja besplatnih web komponenata, koje bi u suprotnom morali razvijati iz početka. Ovaj okvir koristi komponentni pristup razvoja korisničkog sučelja za brzu izgradnju web aplikacija u *Java* programskom jeziku. Iako sami generatori programskog koda ne koriste bazu podatka, generirane aplikacije će ih koristiti. Stoga smo iskoristili *Hibernate* (korištena verzija 5.2.2)[17] okvir za ORM te pristup i pohranu podataka u bazi podataka. Za potrebe demonstracije u poglavlju *5. Demonstracija i evaluacija* koristili smo *MySQL* bazu podatka (korištena verzija 5.7)[18]. *Apache Maven* (korištena verzija 3.3.9)[19] korišten je kao alat za automatizaciju izgradnje generatora, a *Eclipse Oxygen Java EE* (korištena verzija 4.7.2)[20] kao IDE. *Maven* rješava dva aspekta izgradnje softvera: prvo, opisuje kako se gradi softver, i drugo, opisuje njegove ovisnosti. Na *Slici 10* vidljivo je da u projektu za razvoj generatora programskog koda ne učitavamo sve navedene ovisnosti (biblioteke okvira), već samo *Apache Jena i Apache Velocity*. Razlog tome jest taj što će svaka generirana aplikacija biti zasebni, odnosno novi *Maven* projekt u kojim će biti učitane preostale ovisnosti, odnosno okviri *Vaadin* i *Hibernate* te upravljački program baze podataka.

[16] Dostupno 07.07.2018. na https://vaadin.com/framework/releases/
[17] Dostupno 07.07.2018. na https://hibernate.org/orm/releases/
[18] Dostupno 07.07.2018. na https://dev.mysql.com/downloads/
[19] Dostupno 07.07.2018. na https://maven.apache.org/download.cgi
[20] Dostupno 07.07.2018. na http://www.eclipse.org/downloads/eclipse-packages/

```
onto2form/pom.xml
1  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3    <modelVersion>4.0.0</modelVersion>
4
5    <groupId>hr.foi.dstrmecki</groupId>
6    <artifactId>onto2form</artifactId>
7    <version>0.0.1-SNAPSHOT</version>
8    <packaging>jar</packaging>
9
10   <name>onto2form</name>
11   <url>http://maven.apache.org</url>
12
13   <properties>
14     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15   </properties>
16
17   <dependencies>
18     <dependency>
19         <groupId>org.apache.jena</groupId>
20         <artifactId>jena-text</artifactId>
21         <version>3.2.0</version>
22     </dependency>
23     <dependency>
24         <groupId>org.apache.velocity</groupId>
25         <artifactId>velocity</artifactId>
26         <version>1.7</version>
27     </dependency>
28     <dependency>
29         <groupId>org.apache.httpcomponents</groupId>
30         <artifactId>httpclient</artifactId>
31         <version>4.5.3</version>
32     </dependency>
33   </dependencies>
34 </project>
35
```

Slika 10: Postavke projekta generatora programskog koda

4.2.2. Pristup podacima iz modela

Već smo spomenuli da ćemo koristiti *Apache Jena* okvir za pristupanje strukturi i podacima iz ontološkog modela, a sada ćemo malo pobliže opisati navedenu implementaciju. *Jena* dolazi s alatom za generiranje ontološke sheme (*schemagen.bat*) koji se koristi za generiranje *Java* klasa koje sadržavaju strukturu ontologija (nazive ontoloških klase i atributa). Navedeni alat pokreće se preko naredbenog retka operativnog sustava, a potrebno mu je proslijediti određeni set parametara, kao što je prikazana na *Slici 11*.

```
onto_script.bat
1 set JENA_HOME=C:\Program Files\Apache\apache-jena-3.6.0
2 set ONTO_HOME=D:\Fax\_disertacija\Git-Workspace\onto2form\ontologies
3 set PROJECT_HOME=D:\Fax\_disertacija\Git-Workspace\onto2form\src\main\java
4 set PARAMS=--ontology --uppercase --classnamesuffix Ontology --package "hr.foi.dstrmecki.onto" -o %PROJECT_HOME%\
5
6 "%JENA_HOME%\bat\schemagen.bat" -i "%ONTO_HOME%\Actions.owl"  %PARAMS% & "%JENA_HOME%\bat\schemagen.bat" -i "%ONTO_HOME%\Base.owl"
```

Slika 11: Skripta za pokretanje generatora ontološke sheme

Skripta sa *Slike 11* generira *Java* klase za svaku ontologiju, a primjer jedne od njih prikazan je na *Slici 12*. Radi se o vrlo jednostavnoj ontologiji korisničkog sučelja, koja sadrži samo jednu klasu, i nekoliko podatkovnih atributa.

```
InterfaceOntology.java
1  /* CVS $Id: $ */
2  package hr.foi.dstrmecki.onto;
3  import org.apache.jena.rdf.model.*;
5
6  /**
7   * Vocabulary definitions from D:\Fax\_disertacija\Git-Workspace\onto2form\ontologies\Interface.owl
8   * @author Auto-generated by schemagen on 01 srp 2018 13:37
9   */
10 public class InterfaceOntology {
11     /** <p>The ontology model that holds the vocabulary terms</p> */
12     private static final OntModel M_MODEL = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM, null );
13
14     /** <p>The namespace of the vocabulary as a string</p> */
15     public static final String NS = "http://www.w3.org/2002/07/owl#";
16
17     /** <p>The namespace of the vocabulary as a string</p>
18      * @return namespace as String
19      * @see #NS */
20     public static String getURI() {return NS;}
21
22     /** <p>The namespace of the vocabulary as a resource</p> */
23     public static final Resource NAMESPACE = M_MODEL.createResource( NS );
24
25     public static final DatatypeProperty BACKGROUND_COLOR = M_MODEL.createDatatypeProperty( "http://www.foi.hr/dan
26
27     public static final DatatypeProperty TEXT_COLOR = M_MODEL.createDatatypeProperty( "http://www.foi.hr/danielstr
28
29     public static final DatatypeProperty UI_HEIGHT_IN_PIXELS = M_MODEL.createDatatypeProperty( "http://www.foi.hr/
30
31     public static final DatatypeProperty UI_POSITION_INDEX = M_MODEL.createDatatypeProperty( "http://www.foi.hr/da
32
33     public static final DatatypeProperty UI_WIDTH_IN_PIXELS = M_MODEL.createDatatypeProperty( "http://www.foi.hr/d
34
35     public static final OntClass INTERFACE = M_MODEL.createClass( "http://www.foi.hr/danielstrmecki/ontologies/201
36
37 }
38
```

Slika 12: Generirana shema ontologije korisničkog sučelja

Generirane klase, koje služe kao rječnik za rad s ontologija u *Java* programskom kodu, koriste za jednostavnije pisanje SPARQL upita pomoću kojih se dohvaćaju podaci (odnosno instance) iz bazne ontologije. Na *Slici 13* prikazan je isječak koda generatora koji koristi SPARQL upit za dohvaćanje svih akcijskih gumba na određenom web obrascu. Izvršavanje upita kroz *Jena* okvir vraća rezultate u obliku ontoloških resursa u *Java* programskom kodu, koje je zatim potrebno transformirati u odgovarajuće *Java* klase (u ovom slučaju je to jednostavna klasa za reprezentaciju gumba, *Button.java*).

```java
FormDataLoader.java

163
164     StringBuilder buttonsStringBuilder = OntologyUtil.getQueryStringBuilder();
165     buttonsStringBuilder.append("SELECT ?resource ?type WHERE {  ");
166     buttonsStringBuilder.append(    "?resource rdf:type ?type . ");
167     buttonsStringBuilder.append(    "?resource base:componentForForm <" + formResource.getURI() + "> . ");
168     buttonsStringBuilder.append(    "?type rdfs:subClassOf* <" + ComponentsOntology.BUTTON + ">");
169     buttonsStringBuilder.append("}");
170     try (QueryExecution buttonsQueryExecution = QueryExecutionFactory.create(buttonsStringBuilder.toString(), model)) {
171         ResultSet buttonsResultSet = buttonsQueryExecution.execSelect();
172         while (buttonsResultSet.hasNext()) {
173             QuerySolution buttonsQuerySolution = buttonsResultSet.next();
174             Resource componentResource = buttonsQuerySolution.getResource("resource");
175             Resource typeResource = buttonsQuerySolution.getResource("type");
176             Button button = new Button(componentResource, typeResource);
177             components.add(button);
178         }
179     }
180
```

Slika 13: Isječak programskog koda sa SPARQL upitom

4.2.3. Rad s predlošcima programskog koda

Rad s predlošcima programskog koda olakšali smo primjenom okvira *Apache Velocity*. Na *Slici 14* prikazan je primjer jednog *Velocity* predloška. Radi se o predlošku za generiranje klase entiteta u *Javi*. Svaka klasa generirana koristeći ovaj predložak predstavlja jednu tablicu u bazi podataka i sadrži anotacije za opis njezinih stupaca. *Hibernate* okvir koristi ove anotacije za ORM te omogućuje automatsko kreiranje tablica u bazi podatka. *Slika 15* prikazuje primjer metode za generiranje novih instanci, odnosno *Java* datoteka klasa, pomoću *Velocity* predloška. Radi se o istom primjeru, tj. metodi za generiranje klase entiteta u *Javi*, koja koristi ranije spomenuti predložak. *Velocity* koristi predložak, putanju odredišne datoteke i *Java* objekt nazvan kontekst koji sadrži podatke ili objekte kojima se omogućava pristup unutar predloška. Na ovome primjeru vidljivo je kako se u objektu konteksta definiraju varijable koje označavaju naziv klase i naziv tablice u bazi podataka. Na predlošku, tj. na *Slici 14*, vidljivo je kako se navedene varijable koriste u *Velocity* predlošku (linije 23 i 24). *Velocity* sintaksa koristi znak dolar ($) za pristup varijablama proslijeđenim kroz objekt konteksta.

Entity.java.vm

```
1 package $package;
2
3 import java.util.Date;
4 import java.util.Set;
5 import java.io.Serializable;
6
7 import javax.persistence.CascadeType;
8 import javax.persistence.Column;
9 import javax.persistence.Entity;
10 import javax.persistence.GeneratedValue;
11 import javax.persistence.Id;
12 import javax.persistence.Table;
13 import javax.persistence.EnumType;
14 import javax.persistence.Enumerated;
15 import javax.validation.constraints.NotNull;
16 import javax.persistence.JoinColumn;
17 import javax.persistence.ManyToOne;
18 import javax.persistence.OneToMany;
19
20 import org.apache.commons.lang3.builder.ToStringBuilder;
21
22 @Entity
23 @Table(name="$tableName")
24 public class $className implements Serializable {
25
26     private static final long serialVersionUID = 1L;
27
28     @Id
29     @GeneratedValue
30     @Column
31     private Long id;
32
33     #foreach($column in $columns)
34         #if ($column.columnID != "id")
35             #if ($column.isColumnAnIdentifier)
36     @Id
37             #end
38             #if ($column.isColumnValueGenerated)
39     @GeneratedValue
40             #end
41             #if ($column.isColumnValueNotNull)
```

Slika 14: Isječak *Velocity* predloška za klasu entiteta

```java
DataGenerator.java
1  package hr.foi.dstrmecki.gen;
2
3⊕ import java.io.File;
16
17 public class DataGenerator implements Generator {
18
19⊖     @Override
20△     public void doGenerate(OntModel model, File projectRootDir, String projectID) {
21         RequirementDataLoader requirementsData = RequirementDataLoader.getInstance(projectID, model);
22         RepositoryDataLoader repositoryData = RepositoryDataLoader.getInstance(projectID, model);
23
24         String dataPackage = requirementsData.getBasePackage() + ".data";
25         File dataSourceDir = generateDataPackageDir(projectRootDir, dataPackage);
26
27         List<Table> tables = repositoryData.getTables();
28         tables.forEach(table -> {
29             VelocityContext entityFileContext = new VelocityContext();
30             entityFileContext.put("package", dataPackage);
31             entityFileContext.put("tableName", table.getTableName());
32             entityFileContext.put("className", GenerativeUtil.getJavaClassName(table.getTableName()));
33             entityFileContext.put("columns", table.getColumns());
34             entityFileContext.put("StringUtils", StringUtils.class);
35             generateEntityFile(entityFileContext, dataSourceDir, table);
36             LogUtil.log("Generated table " + table.getTableName());
37         });
38
39         VelocityContext commonFilesContext = new VelocityContext();
40         commonFilesContext.put("package", dataPackage);
41         generateCommonEntityFiles(commonFilesContext, dataSourceDir);
42     }
43
```

Slika 15: Metoda za generiranje *Java* klase entiteta pomoću *Velocity* predloška

4.2.4. Web komponente i korisničko sučelje

Primjenom *Vaadin* okvira olakšali smo generiranje web obrazaca, jer se radi o okviru koji omogućava korištenje standardnih web komponenata po principu *uključi i igraj*. Na *Slici 16* prikazan je *Velocity* predložak za generiranje klasa web obrazaca. Na predlošku je vidljivo kako se ovisno o kontekstu na okviru učitavaju različite web komponente iz *Vaadin* okvira. Neke od standardnih komponenata vidljivih na slici su skočni prozor, tablica sa paginacijom, unosna polja i akcijski gumbi. *Vaadin* okvir omogućuje dodavanje web komponenata na okvire isključivo kroz *Java* programski kod, bez potrebe za definiranjem prilagođenog HTML i *JavaScript* koda. Naravno, pod uvjetom da govorimo o dostupnim komponenata unutar okvira i kada nema potrebe za prilagođavanjem istih. Okvir omogućuje i kreiranje vlastitih komponenata po mjeri te korištenje dodataka i komponenta koje su razvili i podijelili drugi autori. Te dodatke moguće je pretražiti na *Vaadin* direktoriju[21] i vrlo jednostavnu uključiti ih u svoj projekt koristeći *Maven* alat za upravljanje ovisnostima.

[21] Dostupno 08.07.2018. na https://vaadin.com/directory

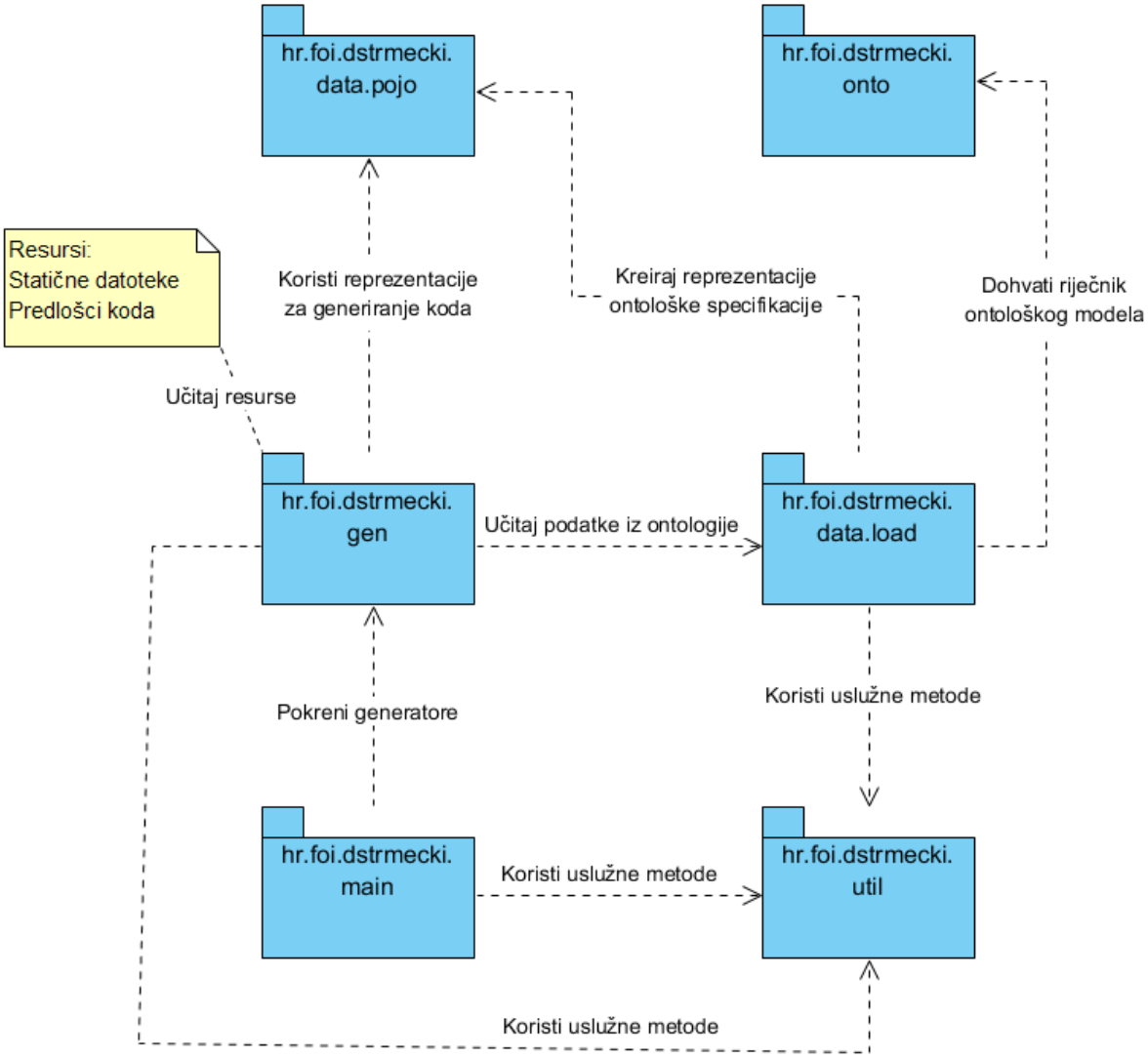
```
ViewEdit.java.vm
33 public class ${form.getFormID()} extends BaseView {
34
35     public static final String VIEW_ID = #if(${form.getIsHome()})""#else"${form.getFormID()}"#end;
36     public static final String VIEW_TITLE = "${form.getFormTitle()}";
37
38     #if ($hasAddNewItemButton)
39     private Window editWindow;
40     #end
41
42     #foreach($pagedTable in $pagedTables)
43     private PagedTable ${pagedTable.componentID};
44     private VerticalLayout ${pagedTable.componentID}HolderLayout;
45     private VerticalLayout ${pagedTable.componentID}Layout;
46     private HorizontalLayout ${pagedTable.componentID}Pagination;
47         #if (${pagedTable.isNotMain})
48     private HorizontalLayout ${pagedTable.componentID}ActionLayout;
49     private ComboBox ${pagedTable.componentID}InputCombo;
50     private Button ${pagedTable.componentID}AddButton;
51     private Button ${pagedTable.componentID}DeleteButton;
52         #end
53     #end
54
55     #if ($hasSaveRecordButton)
56     private Button ${saveRecordButton.getComponentID()};
57     #end
58     #if ($hasDeleteRecordButton)
59     private Button ${deleteRecordButton.getComponentID()};
60     #end
61
62     #if ($hasDeleteSelectedItemsButton)
63     private Button ${deleteSelectedItemsButton.getComponentID()};
64     #end
65     #if ($hasAddNewItemButton)
66     private Button ${addNewItemButton.getComponentID()};
67     #end
68
69     #if (${form.hasMainInputFields()})
70     private TextField mainIdInput;
71     #end
72     #foreach($inputField in $inputFields)
73     private $inputField.inputClass ${inputField.componentID};
```

Slika 16: Isječak *Velocity* predloška za klasu obrasca

4.2.5. Struktura programskog koda

Programski kod dizajniranih i razvijenih generatora organiziran je u šest *Java* paketa, kao što je prikazano na *Slici 17*. Osnovni paket (engl. *main*) sadrži klase koje služe za pokretanje aplikacije i to u dva moda rada, generiranje ontološke sheme i generiranje programskog koda. Paket generatora (engl. *gen* kratica od riječi *generator*) sadrži, kako mi i ime govori, generatore različitih dijelova programskog koda finalne aplikacije, poput generatora projekta, generatora klasa entiteta, generatora okvira, itd. Uslužni paket (engl. *util* kratica od riječi *utility*) sadrži pomoćne klase sa statičnim metodama koje koriste klase iz gotovo svih drugih paketa. Paket ontološkog modela (engl. *onto* kratica od riječi *ontology*) sadrži strukturu i vokabular ontološkog modela generiranu pomoću *Apache Jena* okvira. Njegove klase koristi paket za učitavanje podatka (engl. *data load*) koji sadrži klase sa metodama koje izvode SPARQL upite nad ontologijama i tako učitavaju podatke u jednostavne *Java* klase unutar paketa za

reprezentaciju podataka (engl. *data POJO*[22]). Klase unutar ovog paketa služe kao reprezentacija ontoloških podataka u *Java* programskom kodu. Generatori koriste upravo te klase, zajedno sa statičnim datotekama i predlošcima programskog koda kako bi automatski generirali programski kod iz ontološkog modela i specifikacije.

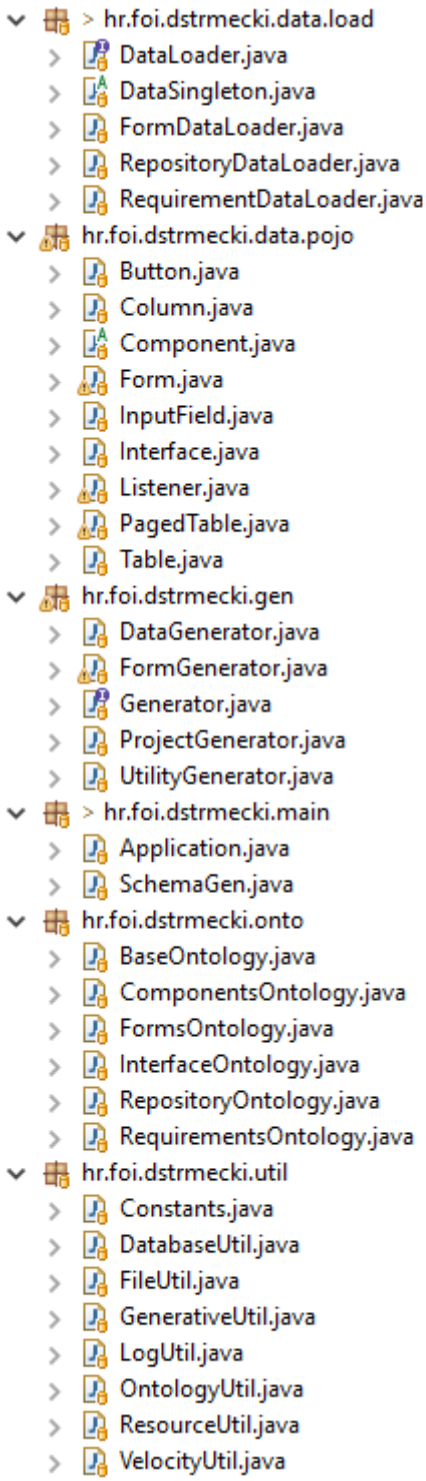


Slika 17: Dijagram paketa generatora programskog koda

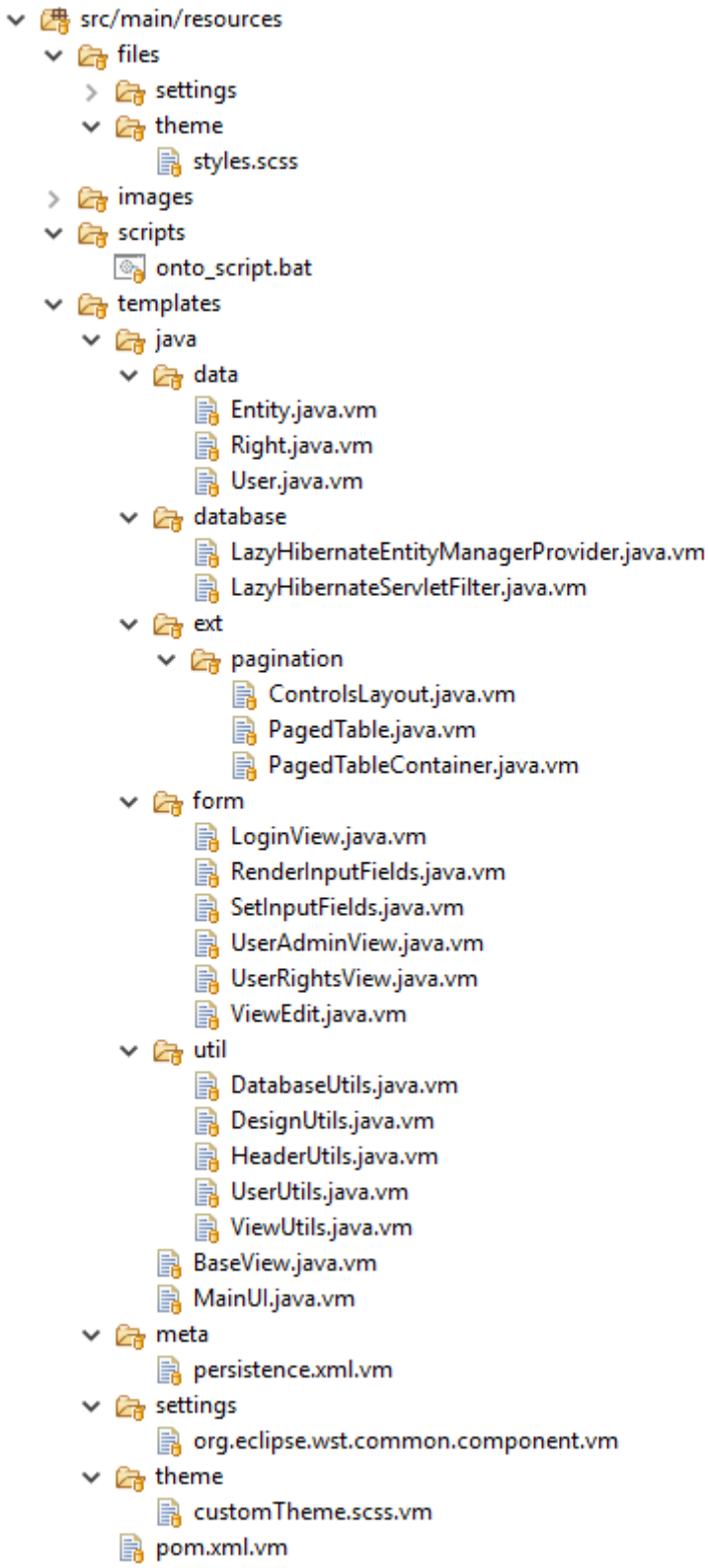
Na *Slici 18* prikazana je kompletna struktura *Java* programskog koda generatora, odnosno raspored svih klasa po paketima. *Slika 19* pak prikazuje korištene resurse, koji se sastoje od

[22] POJO je kratica običan "stari" *Java* objekt, tj. objekt koji nije vezan nikakvim posebnim ograničenjima niti okvirima.

predložaka *Java* programskog koda, predložaka konfiguracije aplikacije te statičnih datoteka poput slika, skripti i dizajna u formatu kaskadnog stilskog jezika (CSS).



Slika 18: Organizacija programskog koda u *Java* pakete i klase

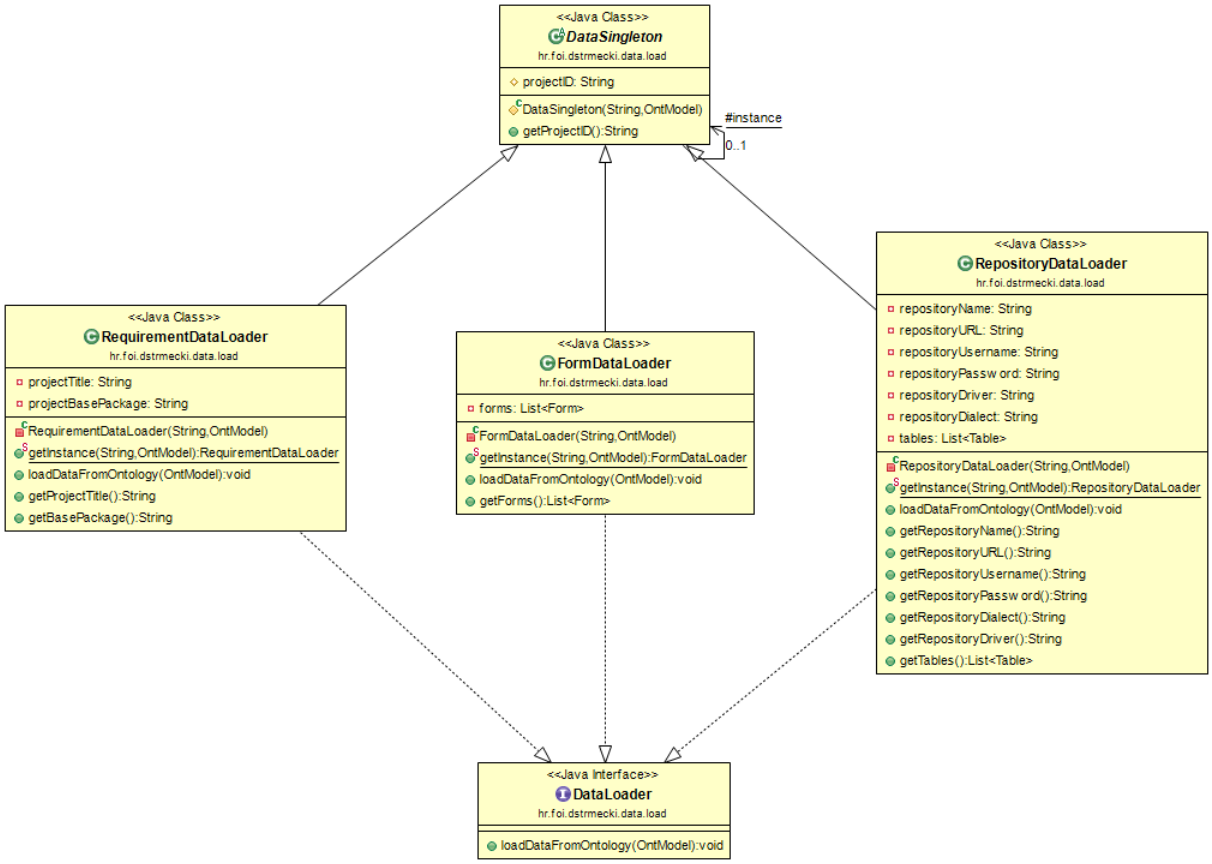


Slika 19: Korišteni resursi - statične datoteke i predlošci koda

4.2.6. Dijagram klasa

4.2.6.1. Učitavanje podataka

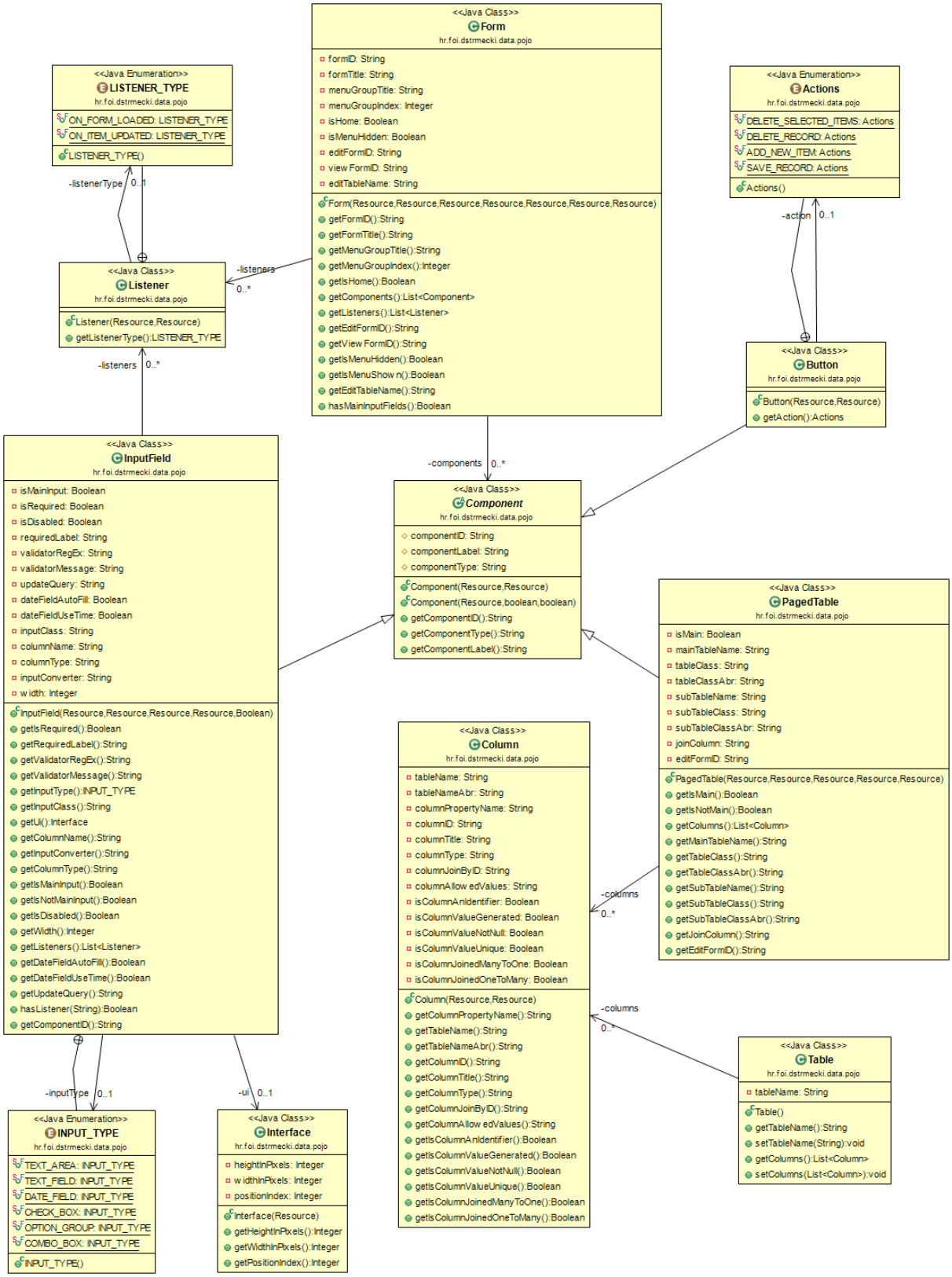
Paket za učitavanje podatka, čiji je dijagram klasa prikazan na *Slici 20*, sastoji se od jednog sučelja te jedne apstraktne klase (napisane prema *singleton* uzorku) koja povezuje specifični projekt / proizvod sa odgovarajućim podacima iz ontološkog modela. S obzirom se istim podacima pristupa iz više različitih klasa generatora, primjenom *singleton* uzorka osiguravamo da se podaci iz ontologije učitaju samo jednom te potom čuvaju u radnoj memoriji do kraja procesa generiranja programskog koda.



Slika 20: Dijagram klasa paketa za učitavanje podataka

4.2.6.2. Reprezentacija podataka

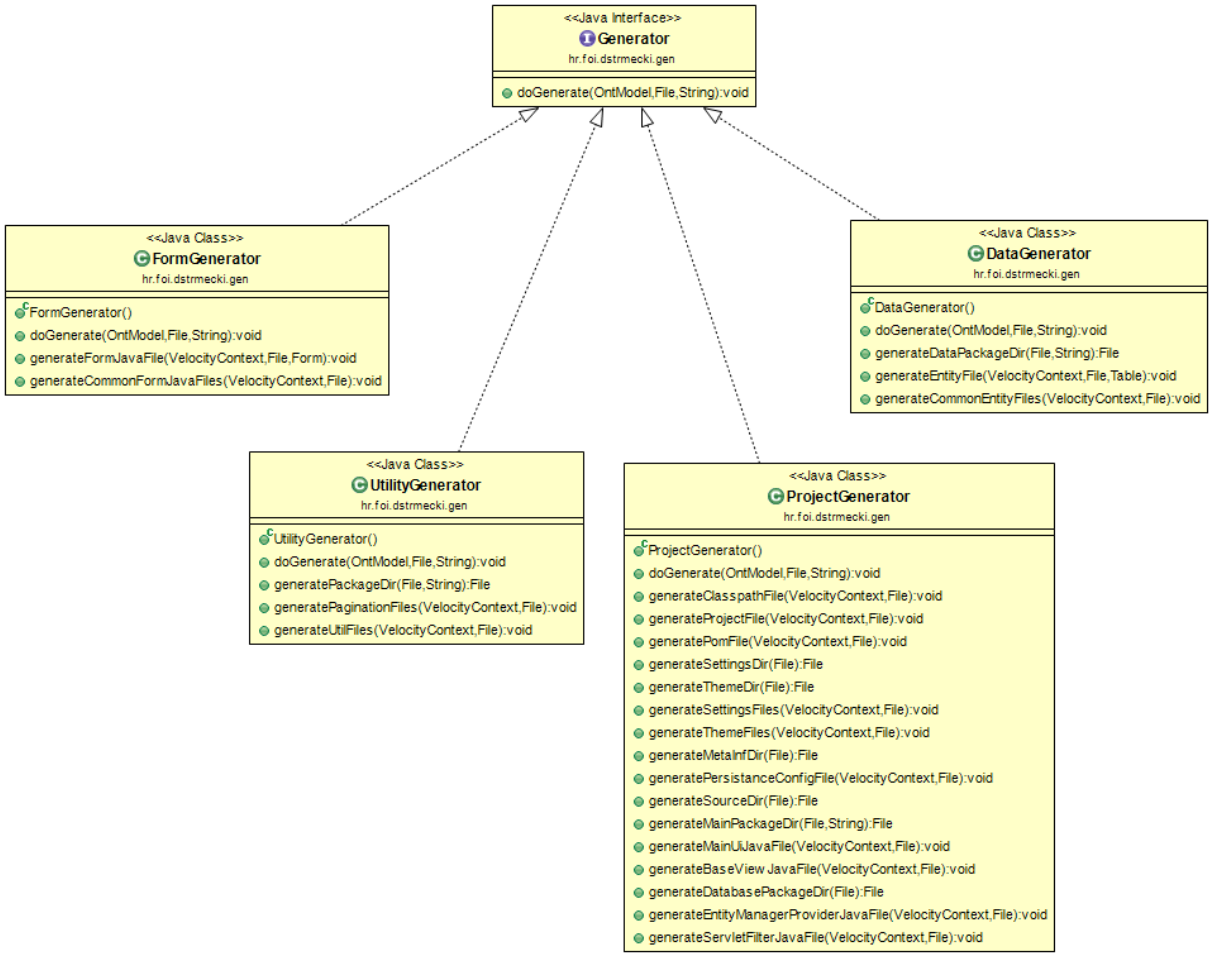
Učitane podatke iz ontologije pretvaramo u objekte jednostavne, tzv. POJO klase kako bi s podacima lakše baratali u *Java* kodu i *Velocity* jeziku predložaka. Dijagram klasa paketa za reprezentaciju podataka prikazan je na *Slici 21*, a sastoji se okvira, komponenata i slušača. Na dijagramu je vidljivo kako je podržano dosta različitih web komponenata te njihovih različitih pod-tipova. Proširivanjem apstraktne klase komponente moguće je, na jednostavan način, povećati broj podržanih komponenata u generatorima programskog koda.



Slika 21: Dijagram klasa paketa za reprezentaciju podataka

4.2.6.3. Generatori programskog koda

Paket za generiranje programskog koda, čiji je dijagram klasa prikazan na *Slici 22*, sastoji se od nekoliko tipova generatora. Sučelje te više tipova generatora kreirano je kako bi se u budućnosti moglo lakše proširiti kod novim implementacijama generatora. Trenutno su implementirani generator projekta, generator uslužih klasa, generator klasa entiteta i generator obrazaca. Generator projekta zadužen je za generiranje novog *Java*, odnosno *Maven* projekta te odgovarajućih konfiguracijskih datoteka. Generator uslužnih klasa generira klase sa statičnim metodama koje se koriste na više različitih mjesta u projektu. Generator klasa entiteta generira klase koje predstavljaju *Hibernate* entitete te služe za kreiranje tablica u bazi podataka, kao i za dohvaćanje te spremanje podataka u bazu. Na kraju, generator obrazaca služi za generiranje kompletnih i funkcionalnih web obrazaca, koji uključuju *Vaadin* web komponente, slušače događaja te metode za prikaz i operacije nad podacima iz baze podataka.



Slika 22: Dijagram klasa paketa generatora programskog koda

4.3. Razvojni okvir

4.3.1. Primjena okvira

Nakon što smo prezentirali dizajnirani ontološki model i razvijene generatora programskog koda, posvetiti ćemo se primjeni navedenih artefakata u okviru za razvoj obitelji poslovnih web rješenja. U ranijim poglavljima bavili smo se modelima, alatima i bibliotekama koje možemo koristiti prilikom generativnog razvoja softvera temeljenog na ontološkoj specifikaciji, a sada ćemo se posvetiti definiranju razvojnog okvira temeljenog upravo na tim artefaktima i tehnologijama. Vrijedi napomenuti kako, ovdje definirani okvir, ima smisla upotrijebiti pri razvoju obitelji poslovnih web aplikacija temeljenim na web obrascima, ne nužno i na drugim tipovima aplikacija. Ukoliko razvijate poslovnu web aplikaciju za vrlo specifičnu primjenu, sa specifičnim komponenata i funkcionalnostima, koje vjerojatno nećete moći ponovno iskoristi na drugim projektima, tada je preporuka koristi neke druge razvojne okvire ili jednostavno klasični način razvoja softvera. Dizajnirani okvir primjenjuje inovativni način razvoja, točnije ontološki potpomognuti pristup generativnom programiranju koji zahtijeva veća inicijalna ulaganja (prvenstveno vrijeme razvojnih programera), a kasnije se isplaćuje kroz ponovnu uporabu razvijenih modela i artefakata na više projekta, odnosno proizvoda iz iste obitelji. Prilikom odluke za primjenu ovog okvira potrebno je razmotriti tip aplikacije koja se razvija, broj projekata koji će se razvijati u budućnosti i postotak artefakata koji će se moći ponovno iskoristiti. Okvir je primjenjiv za razvoj obitelji poslovnih web aplikacija temeljenih na web obrascima koji pohranjuju podatke u relacijsku bazu podataka. Cilj primjene okvira jest dugoročno smanjiti troškove razvoja obitelji poslovnih web aplikacija. Iz tog razloga se odluka za primjenu ovog okvira treba temeljiti na dugoročnoj analizi isplativosti, koju bi svaka tvrtka trebala provesti za sebe.

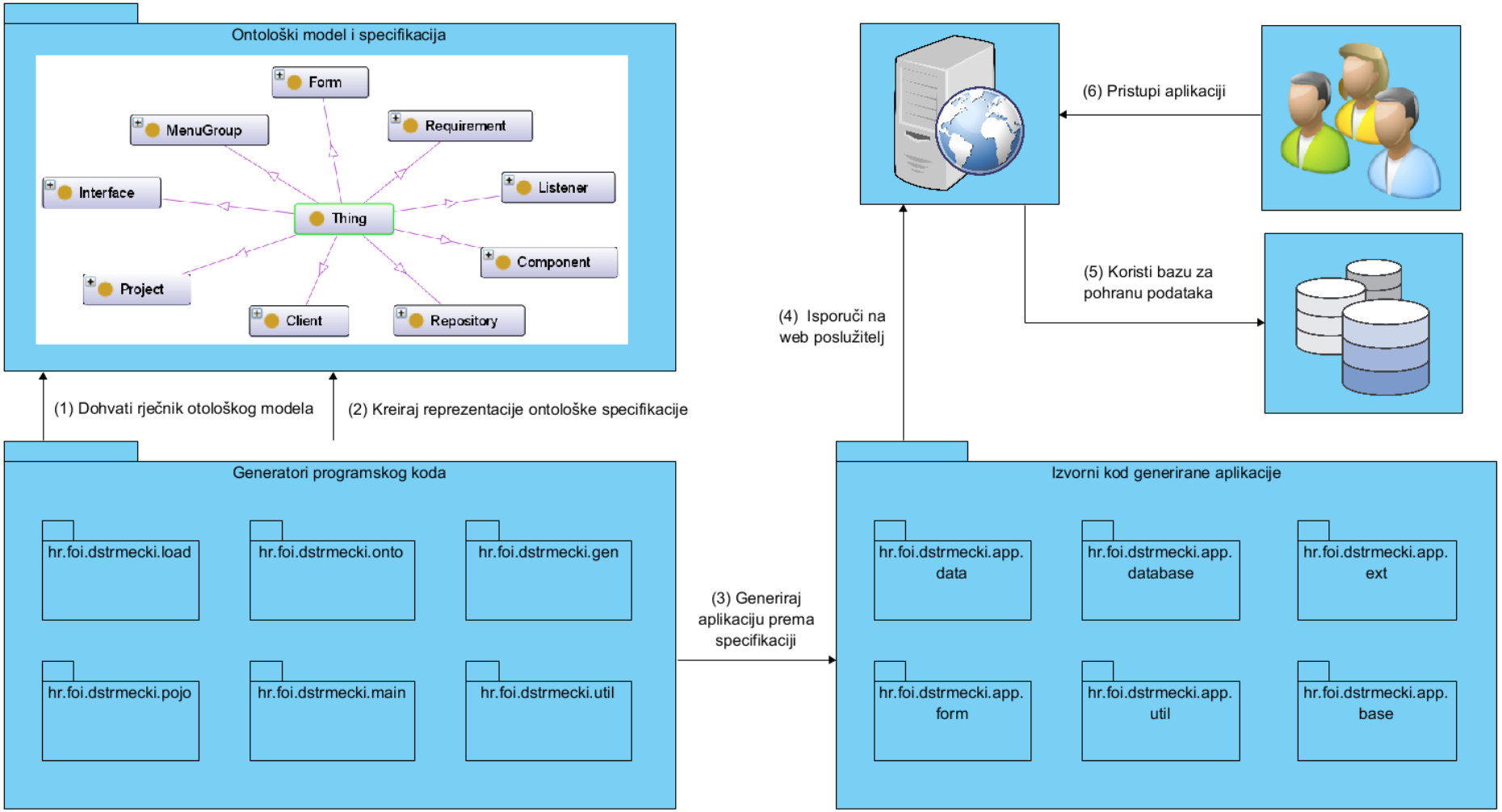
Osim dugoročnih financijskih ušteda, primjena ovog razvojnog okvira omogućuje poslovnim korisnicima lakše uključivanje u razvojni proces. Ontološki modeli, uz alate poput *Protege*-a, koji omogućuju vizualizaciju modela, puno su lakši za razumijevanje poslovnim korisnicima od programskog koda. Stoga je jednostavnije i aktivnije uključiti korisnike u izgradnju same ontologije, kako bi i oni bolje razumjeli kompleksnost izgradnje programskog proizvoda. Na taj način postiže se bolja povezanost i viša razina razumijevanja između poslovnih analitičara, programera i krajnjih korisnika. Osim uključivanja korisnika u proces modeliranja, odnosno specifikacije zahtijeva, okvir omogućuje i brzo prototipiranje primjenom tehnika generativnog programiranja. Brzo prototipiranje omogućuje poslovnih korisnicima rano stjecanje dojma o

izgledu i načinu radu programskog proizvoda te validaciju zahtjeva i dizajna u ranim fazama razvoja, kada promjene još nisu toliko skupe.

4.3.2. Povezivanje modela i generatora

Osnovni koncept predloženog razvojnog okvira podrazumijeva mapiranje ontološkog modela na generatore programskog koda. Cilj mapiranja ovih artefakata jest generirati izvršivi programski kod ontološki specificirane aplikacije. Na *Slici 23* prikazan je koncept razvojnog okvira u kojem se web aplikacija iz obitelji najprije ontološki specificira, a zatim automatski generira na temelju te specifikacije. Sam proces specifikacije aplikacije podrazumijeva kreiranje različitih tipova instanci u ontološkom modelu. Specifikacija aplikacije mora biti potpuna kako bi se mogao generirao programski kod bez grešaka. Nakon što je aplikacija specificirana pokreće se proces generiranja programskog koda. Generatori koriste rječnik ontološkog modela, koji je u programskom kodu reprezentiran klasama koje predstavljaju ontologije, a sadrže konstante koje definiraju objektne i podatkovne atribute (primjer je prikazan na *Slici 12*). Na ovaj način osigurano je da ontološki model sadrži sve klase i atribute korištene u generatorima programskog koda. Kada to ne bi bio slučaj, javila bi se greška prilikom kompiliranja programskog koda generatora. Prema tome, generatori programskog koda sadrže internu reprezentaciju strukture ontološkog modela, koju koriste za dohvaćanje podataka iz ontologije. Podaci o specifikaciji aplikacije dohvaćaju se iz ontologije izvođenjem SPARQL upita. Na temelju rezultata provedenih upita kreiraju se *Java* objekti koji reprezentiraju ontološku specifikaciju prilikom izvođenja programskog koda generatora i koriste se u *Velocity* predlošcima. Generatori programskog koda mapiraju specifikaciju obrazaca i entiteta sa odgovarajućim predlošcima programskog koda. Rezultat ovog procesa su generirane klase izvornog koda finalnog proizvoda, odnosno aplikacije. Proces generiranja programskog koda završava nakon što se generiraju sve potrebne klase aplikacije, što uključuje *Vaadin* web obrasce, *Hibernate* entitete i uslužne klase. Kompletni generirani programski kod aplikacije u tom trenutku može se provjeriti, kompilirati i instalirati na odgovarajući *Java* web poslužitelj. Osim web poslužitelja potrebno je osigurati i pristup bazi podataka[23]. Korisnici pristupaju web obrascima generirane aplikacije preko web preglednika.

[23] U demonstraciji je korišten *Apache Tomcat* poslužitelj te *MySQL* baza podataka. Dostupno 01.09.2018. na https://tomcat.apache.org/download-90.cgi odnosno https://dev.mysql.com/downloads/



Slika 23: Povezivanje ontološkog modela i generatora programskog koda u okvir za razvoj web aplikacija

4.3.3. Razvojni ciklus

Dizajnirani razvojni okvir temeljeni se na modeliranju, odnosno specifikaciji poslovne web aplikacije kroz ontološki model te generiranju programskog koda iz te specifikacije. Navedeni proces provodi se kroz sljedećih osam razvojnih faza ili aktivnosti:

1) Specifikacija projekta
2) Specifikacija korisničkih zahtjeva
    a. Specifikacija korisničkih priča
    b. Specifikacija kriterija prihvaćanja
3) Specifikacija repozitorija
    a. Specifikacija tablica
    b. Specifikacija stupaca i ograničenja
4) Specifikacija web obrazaca
    a. Specifikacija izbornika
    b. Specifikacija web komponenata
    c. Specifikacija događaja i slušača
    d. Specifikacija korisničkog sučelja
5) Validacija modela
6) Razvoj novih i specifičnih funkcionalnosti
7) Generiranje programskog koda
8) Validacija korisničkih zahtjeva

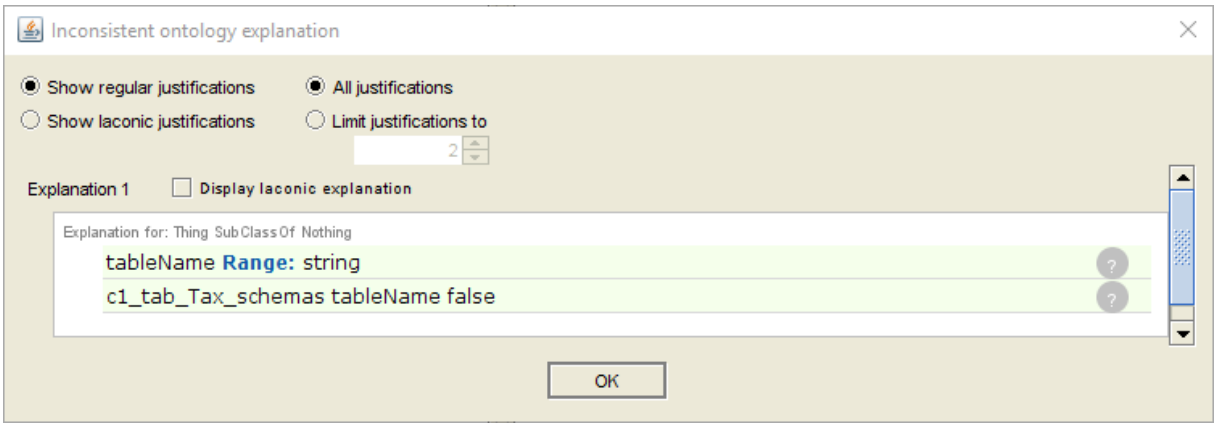
Razvoj novog projekta započinje s aktivnošću *(1) Specifikacija projekta*, gdje se definira sam projekt i klijent za kojeg se projekt razvija. Radi se početnoj fazi gdje se u ontološkom modelu kreira nova instanca projekta i bilježe se osnovni podaci o projektu poput naziva projekta, kodnog imena, naziva paketa, opisa projekta, klijenta (privatna osoba ili tvrtka) itd. Druga aktivnost jest *(2) Specifikacija korisničkih zahtjeva* u kojoj se najprije u grubo specificiraju korisnički zahtjevi, najčešće za svaki obrazac posebno. Nakon toga se za svaki korisnički zahtjev definira jedna ili više korisničkih priča. Za svaku korisničku priču se definira jedan ili više kriterija prihvaćanja. Radi se o iznimno bitnom koraku, jer upravo zadovoljavanje svih

kriterija prihvaćanja određuje uspješan završetak razvoja aplikacije. Korisnički zahtjevi, korisniče priče i kriteriji prihvaćanja spremaju se kao instance ontološkog modela povezane sa konkretnim projektom, a biti će ponovno korištene u posljednjoj aktivnosti *(8) Validacija korisničkih zahtjeva*.

Slijedi aktivnost broj tri, odnosno *(3) Specifikacija repozitorija* gdje se kreira instanca repozitorija, odnosno baze podataka za konkretan projekt. Definira se naziv i tip baze podataka te upravljački program koji će se koristiti. Zatim se za kreirani repozitorij kreiraju instance tablica, a svaka tablica povezuje se sa jednom ili više instanci stupaca. Slijedeća aktivnost je *(4) Specifikacija obrazaca* koja započinje kreiranjem instanci grupa web obrazaca koje se koriste za formiranje glavnog izbornika aplikacije. Nakon toga se kreiraju instance za svaki web obrazac posebno te instance web komponenata korištene na pojedinim obrascima. Specifikacija jedne web komponente može se povezati, odnosno koristiti na jednom ili više obrazaca. Web komponente unosnih polja povezuju se i sa odgovarajućom tablicom te stupcem u bazi podataka iz koje čitaju i u koje spremaju podatke. Web komponentu tablice sa paginacijom, koja se koristi za prikaz podataka iz baze podataka, također je potrebno povezati sa odgovarajućim stupcima konkretnih tablica u bazi koje želimo prikazati korisniku na ekranu. Određene komponente moguće je povezati sa instancama slušača kako bi ažurirale svoje vrijednosti ili stanja nakon izvršenja određenog događaja, npr. izračunati novu sumu iznosa nakon dodavanje nove stavke. Komponente također povezujemo sa instancama korisničkog sučelja. Ove instance služe za određivanje pozicije i izgleda web komponente na ekranu korisnika. Time završavamo četiri aktivnosti koje se bave specifikacijom aplikacije, odnosno kreiranjem instanci ontološkog modela. Na *Slici 25* vidljivo je kako prve četiri aktivnosti pune ontološki model sa podacima (instancama), dok preostale četiri aktivnosti razvojnog ciklusa koriste ontološki model i njegove instance (specifikaciju) kako bi ih validirali i iz njih automatski generirali izvršivi programski kod. Vrijedi još napomenuti kako se aktivnosti *(2) Specifikacija korisničkih zahtjeva, (3) Specifikacija repozitorija* i *(4) Specifikacija obrazaca* obično obavljaju iterativno, za svaki obrazac posebno, iz jednostavnog razloga radi lakšeg fokusiranja ljudi koji obavljaju ove aktivnosti na jedan dio problema.

Nakon što smo dovršili specifikaciju aplikacije kroz predefinirani ontološki model, slijedeća faza *(5) Validacija modela* bavi se validacijom naše specifikacije. U tu svrhu možemo iskoristiti

neki od postojećih semantičkih argumentatora (engl. *reasoner*)[24]. *FaCT++* jedan od zadanih semantičkih argumentatora u alatu *Protege*. Na *Slici 24* prikazan je primjer greške koju *FaCT++* prijavljuje u *Protege*-u u slučaju da pronađe instancu sa pogrešno definiranim tipom atributa. Osim toga, u korištene generatore programskog koda ugrađen je i prilagođeni argumentator za dizajnirani ontološki model, koji provjerava da je za svaki tip instance definiran skup obaveznih podatkovnih atributa. Na *Slici 25* prikazan je primjer prijave greške od strane generatora u *Eclipse*-u u slučaju kada jedna od instanci tablica u bazi podataka nema definiran jedan obavezni atribut (naziv tablice). Prema tome aktivnost validacije modela u sklopu dizajniranog razvojnog okvira obavljamo pokretanjem ugrađenih semantičkih argumentatora u alatu *Protege* i testnim pokretanjem generatora programskog koda u *Eclipse*-u.



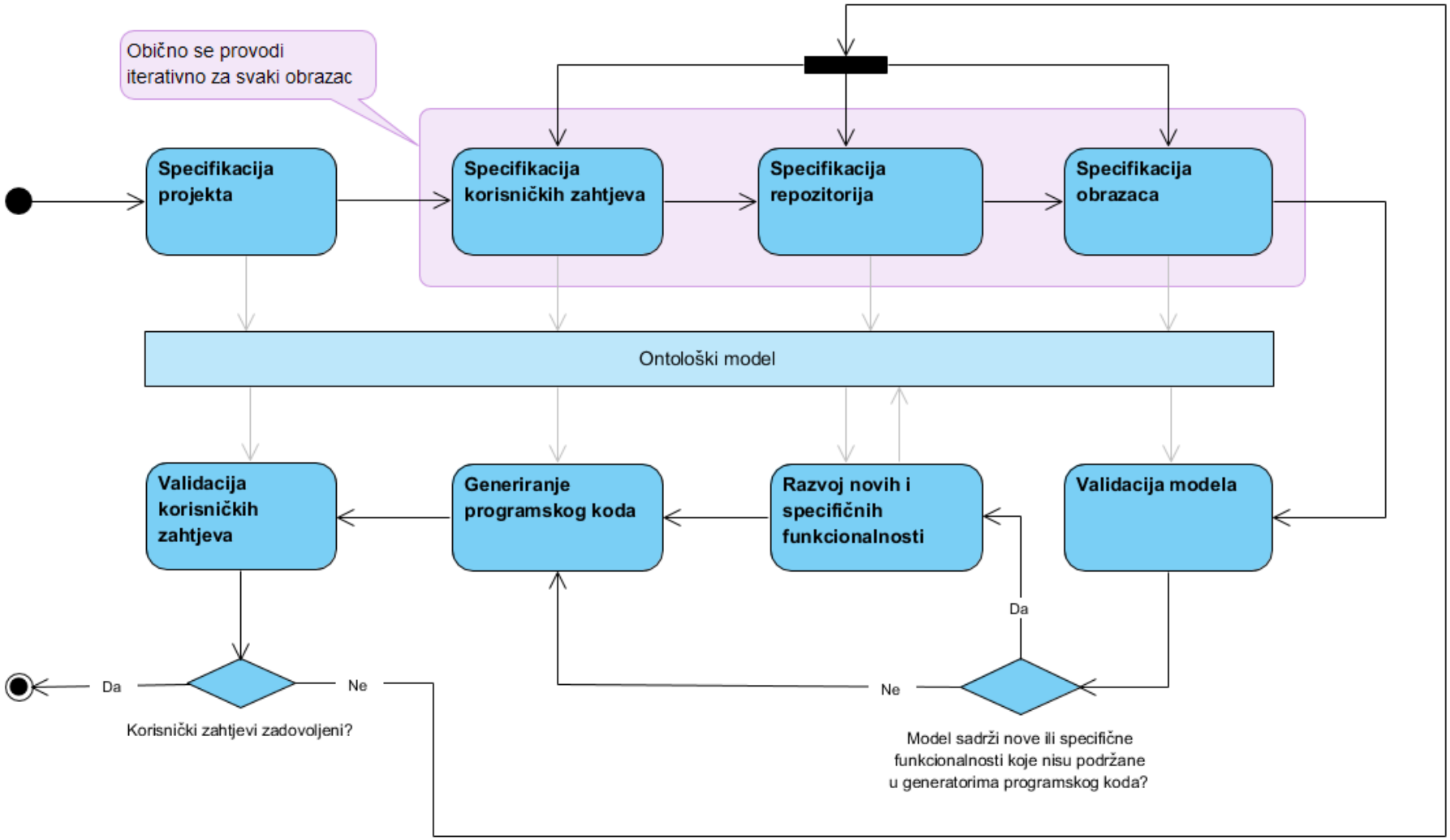
Slika 24: Primjer prijave greške u *Protege*-u pomoću *FaCT++* argumentatora

[24] Semantički argumentator je softver koji može dokazati logičke posljedice iz skupa utvrđenih činjenica ili aksioma.

```
Markers  Properties  Servers  Data Source Explorer  Snippets  Problems  Console  Debug
<terminated> Application [Java Application] C:\Program Files\Java\jdk1.8.0_151\bin\javaw.exe (21. srp 2018. 11:48:49)
DEBUG: Loading ontology model...
log4j:WARN No appenders could be found for logger (Jena).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
DEBUG: Ontology model loaded.
DEBUG: Creating root folder...
DEBUG: Root folder created.
DEBUG: Generating project structure...
Exception in thread "main" java.lang.RuntimeException: Required property tableName not found for resorce c1_tab_Tax_schemas
        at hr.foi.dstrmecki.util.OntologyUtil.getPropertyStringRequired(OntologyUtil.java:51)
        at hr.foi.dstrmecki.data.load.RepositoryDataLoader.loadDataFromOntology(RepositoryDataLoader.java:80)
        at hr.foi.dstrmecki.data.load.RepositoryDataLoader.<init>(RepositoryDataLoader.java:30)
        at hr.foi.dstrmecki.data.load.RepositoryDataLoader.getInstance(RepositoryDataLoader.java:35)
        at hr.foi.dstrmecki.gen.ProjectGenerator.doGenerate(ProjectGenerator.java:18)
        at hr.foi.dstrmecki.main.Application.runGenerators(Application.java:50)
        at hr.foi.dstrmecki.main.Application.main(Application.java:27)
```

Slika 25: Primjer prijave greške u *Eclipse*-u pomoću argumentatora ugrađenog u generatore

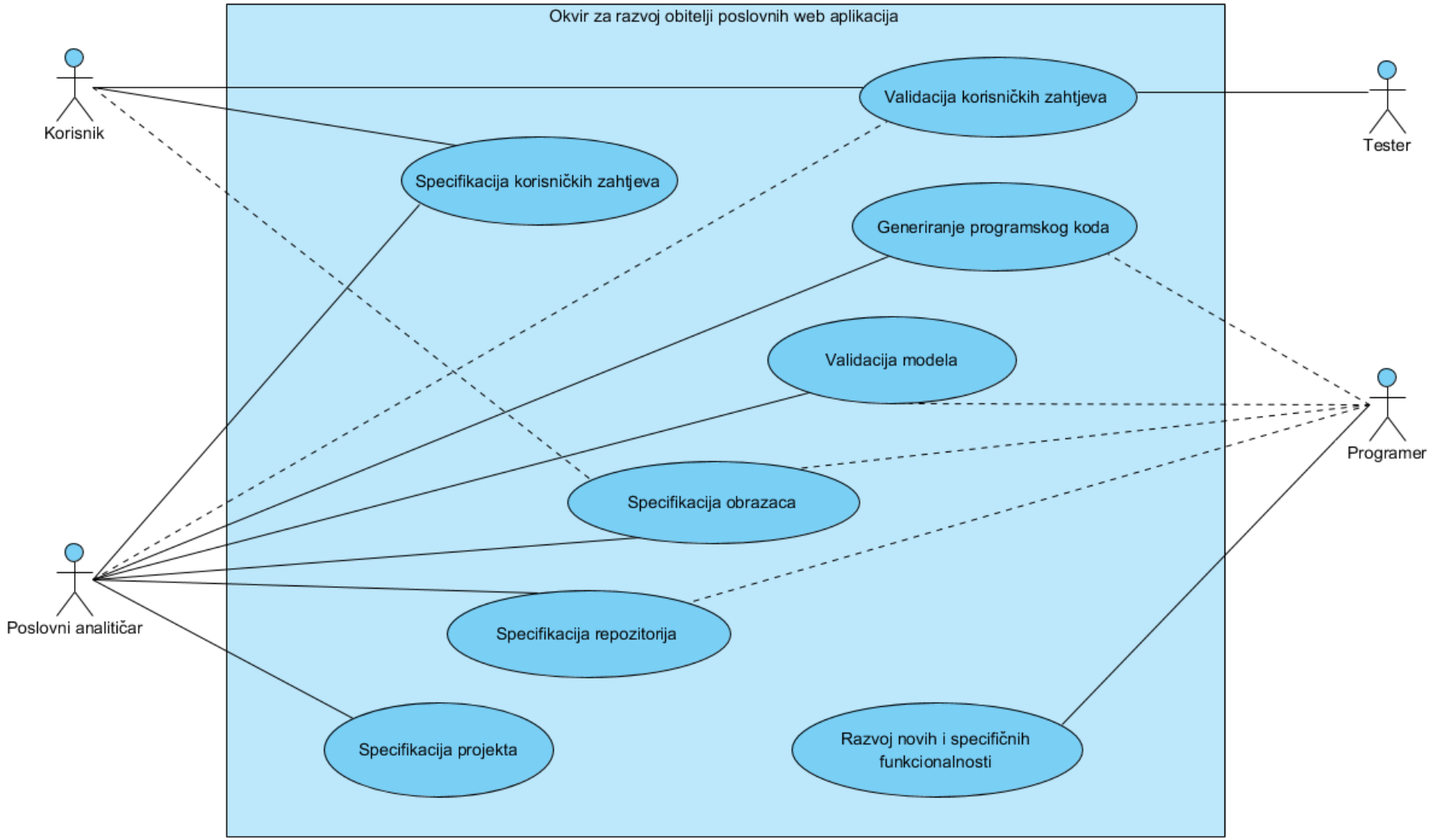
Na *Slici 26* vidljivo je da se slijedeća aktivnost *(6) Razvoj novih i specifičnih funkcionalnosti* provodi samo u slučaju da model sadrži nove ili specifične funkcionalnosti koje trenutno nisu podržane u generatorima programskog koda. U tom slučaju programer mora najprije razviti specifičnu funkcionalnost na reprezentativnom primjeru generirane aplikacije te zatim ugraditi tu funkcionalnost u generatore programskom koda. U tu svrhu programer može dodati ili izmijeniti programski kod nekog od postojećih generatora ili definirati novi generator i datoteke predložaka. Po završetku ove aktivnosti potrebno je uskladiti model sa izmjenama u generatoru, što podrazumijeva dodavanje novih atributa ili čak klasa u ontološki model. Ovisno o programskom proizvodu koji razvijamo i broju ranije generiranih proizvoda iz iste obitelji, ova aktivnost može biti vrlo kompleksna, a može se i u potpunosti preskočiti. Slijedeća aktivnost u razvojnom ciklusu jest *(7) Generiranje programskog koda.* Ova aktivnost obavlja se automatizirano, jednostavnim pokretanjem generatora programskog koda klikom na gumb. Treba ipak napomenuti kako postoji mogućnost intervencije programera u ovoj fazi, i to u slučaju da je došlo do pogreške prilikom nadogradnje generatora ili specifikacije proizvoda. Posljednja aktivnost razvojnog ciklusa je *(8) Validacija korisničkih zahtjeva*, unutar koje se provjeravaju korisnički zahtjevi, a osobito kriteriji prihvaća definirani u fazi *(2) Specifikacija korisničkih zahtjeva*. U slučaju da su svi korisnički zahtjevi zadovoljeni te da su performanse aplikacije zadovoljavajuće ovo je aktivnost u kojoj projekt završava. Međutim ova posljednja aktivnost ima i povratnu vezu na aktivnosti specifikacije aplikacije, u slučaju da neki od korisničkih zahtjeva nisu u potpunosti zadovoljeni. U tom slučaju se cijeli proces vraća na neku od prethodnih faza specifikacija te kasnije eventualno razvoja novih ili specifičnih funkcionalnosti.



Slika 26: Dijagram aktivnosti razvojnog ciklusa primjene okvira

4.3.4. Aktivnosti i dionici

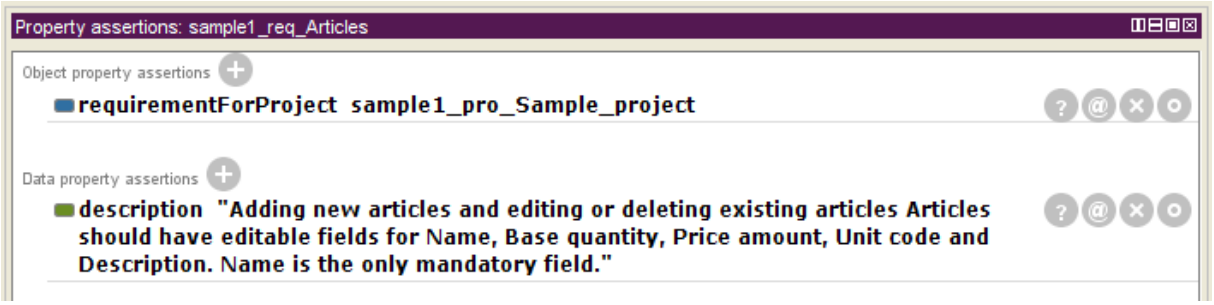
Svaku aktivnost unutar predstavljenog razvojnog ciklusa obavlja jedan ili više dionika, kao što prikazuje *Slika 27*. Na dijagramu slučajeva korištenja punom crtom povezani su dionici koji su direktno zaduženi za obavljanje određene aktivnosti, dok su isprekidanom crtom povezani dionici koji po potrebi mogu biti uključeni (ili konzultirani) prilikom obavljanja aktivnosti. S obzirom da se ovaj razvojni okvir može koristiti u vodopadnom, ali i u inkrementalnom ili agilnom načinu razvoja softvera, u ovaj prikaz uključeni su samo osnovni, tj. nužni dionici (korisnik, poslovni analitičar, programer i tester). Dionici specifični za pojedine metodologije ili okvire, poput vlasnika proizvoda ili *Scrum* majstora (engl. *master*) neće biti razmatrani u ovome radu. Jedna od glavnih ideja ovog razvojnog okvira jest rasteretiti programere ponavljajućih zadataka i omogućiti im da se bave gotovo isključivo razvojem novih funkcionalnosti. Samim time njihova uloga prilikom specifikacije i generiranja pojedinog proizvoda iz obitelji postaje konzultativna. Za obavljanje aktivnosti specifikacija, konkretno aktivnosti *(1) Specifikacija projekta, (2) Specifikacija korisničkih zahtjeva, (3) Specifikacija repozitorija* i *(4) Specifikacija obrazaca* stoga je zadužen poslovni analitičar. Pretpostavka za obavljanje ovih aktivnosti jest da poslovni analitičar dobro poznaje razvojni okvir i korišteni ontološki model te da ima višu razinu formalnog obrazovanja u području informatike ili odgovarajuću razinu radnog iskustva u modeliranju, specifikaciji, razvoju ili testiranju softvera. Osim same specifikacije proizvoda, poslovni analitičar zadužen je i za pokretanje generatora, tj. aktivnosti *(7) Generiranje programskog koda*. Poslovni korisnik pak obavlja, zajedno sa poslovnim analitičarem, aktivnosti vezane uz specifikaciju korisničkih zahtijeva, korisničkih priča i kriterija prihvaćanja, odnosno aktivnost *(2) Specifikacija korisničkih zahtjeva*. Korisnik također može biti konzultiran i u aktivnosti *(4) Specifikacija obrazaca*, jer se upravo u toj aktivnosti definiraju izbornici, obrasci i raspored komponenata na ekranu, s kojima će korisnik kasnije biti u direktnoj interakciji. Programer je direktno zadužen samo za aktivnost *(6) Razvoj novih i specifičnih funkcionalnosti*, a pruža konzultantske usluge, odnosno podršku poslovnom analitičaru u fazama specifikacije i generiranja programskog koda. Posljednju aktivnost *(8) Validacija korisničkih zahtjeva* obavljaju tester i korisnik nezavisno. Tester obavlja funkcionalno i performansno testiranje aplikacije prema korisničkih zahtjevima definiranim u ontološkoj specifikaciji, dok krajnji korisnik provodi tzv. test prihvaćanja.



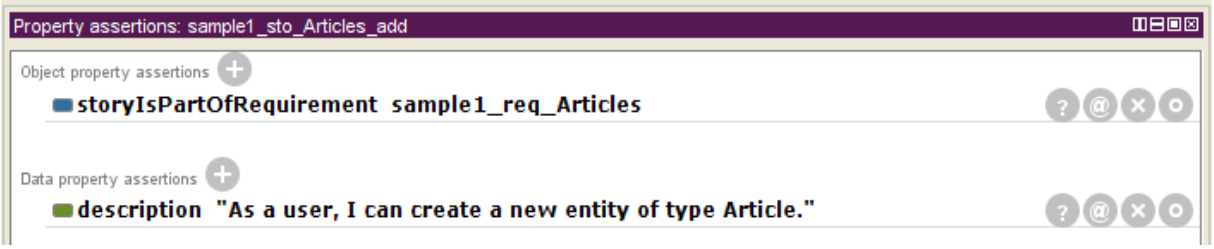
Slika 27: Dijagram slučajeva korištenja razvojnog okvira

4.3.5. Ontološka specifikacija

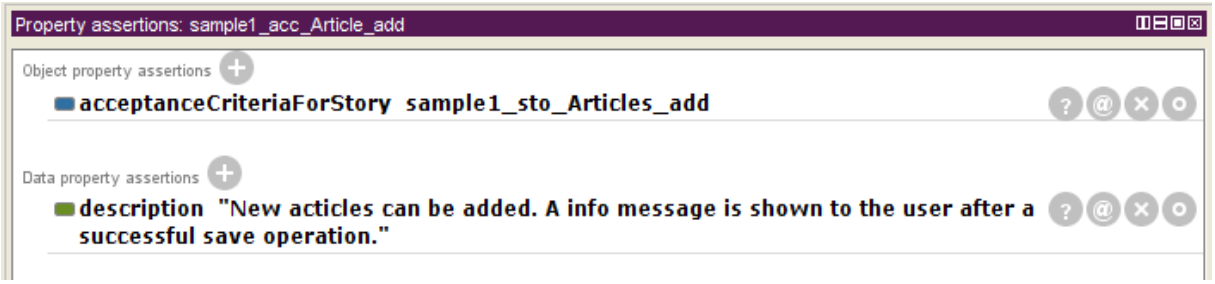
Specifikacija proizvoda provodi se kroz kreiranje instanci unutar dizajniranog ontološkog modela. Za provedbu aktivnosti specifikacije preporučuje se uporaba alata *Protege,* koji je korišten i za sam razvoj ontološkog modela. Ovaj alat omogućuje jednostavno dodavanje novih instanci kroz korisničko sučelje. Prilikom kreiranja instanci iznimno je bitno pridržavati se pravila imenovanja definirana u poglavlju *4.1.5. Struktura ontologija i pravila imenovanja,* jer broj instanci unutar ontološkog modela brzo raste prilikom specifikacije novih proizvoda. Stoga je iznimno bitno na jednostavan način moći razlikovati instance koje pripadaju određenom proizvodu i određenoj klasi. Već smo napomenuli kako se aktivnosti *(2) Specifikacija korisničkih zahtjeva, (3) Specifikacija repozitorija* i *(4) Specifikacija obrazaca* obično obavljaju iterativno, što ćemo sada pobliže prikazati na primjeru definiranja jednog korisničkog zahtjeva i obrasca koji zadovoljava taj korisnički zahtjev. Proces specifikacije počinjemo sa grubim opisom korisničkog zahtjeva, odnosno kreiranjem instance korisničkog zahtjeva sa atributom opisa, kao što je prikazano na *Slici 28*. U ovom primjeru radi se o zahtjevu za omogućavanjem vođenja šifarnika artikala unutar aplikacije. To konkretno podrazumijeva implementaciju mogućnosti dodavanja, uređivanja i brisanja artikala. Osim toga, ovim korisničkim zahtjevom definirana su polja koja pojedini artikl treba sadržavati, a definirano je i jedno obavezno polje bez kojeg nije moguće spremiti artikl. Za ovaj primjer korisničkog zahtjeva potrebno je definirati više korisničkih priča, no radi pojednostavljenja, u ovom primjeru prikazati ćemo samo jednu od njih. Radi se o korisničkoj priči kreiranja novog artikla, čiji atributi su prikazani na *Slici 29*. Ova konkretna korisnička priča odnosi se na mogućnost korisnika da kreira nove zapise artikala. Za navedenu korisničku priču potrebno je kreirati i kriterij prihvaćanja, čiji atributi su prikazani na *Slici 30*. Kriterij prihvaćanja opisuje uvjete iz perspektive korisnika koji moraju biti zadovoljeni kako bi se završila korisnička priča. U ovome slučaju korisnik očekuje obrazac sa poljima za unos artikala te informativnu poruka kada su podaci o artiklu uspješno spremljeni. Kako bismo u potpunosti zadovoljili ovaj korisnički zahtjev, bilo bi potrebno na isti način definirati još i korisničke priče te kriterije prihvaća za uređivanje i brisanje artikala. Radi se radi o gotovo pa identičnoj proceduri kreiranja novih instanci korisničkih zahtjeva, korisničkih priča i kriterija prihvaćanja, razlika je jedino u vrijednostima atributa.



Slika 28: Atributi instance korisničkog zahtjeva



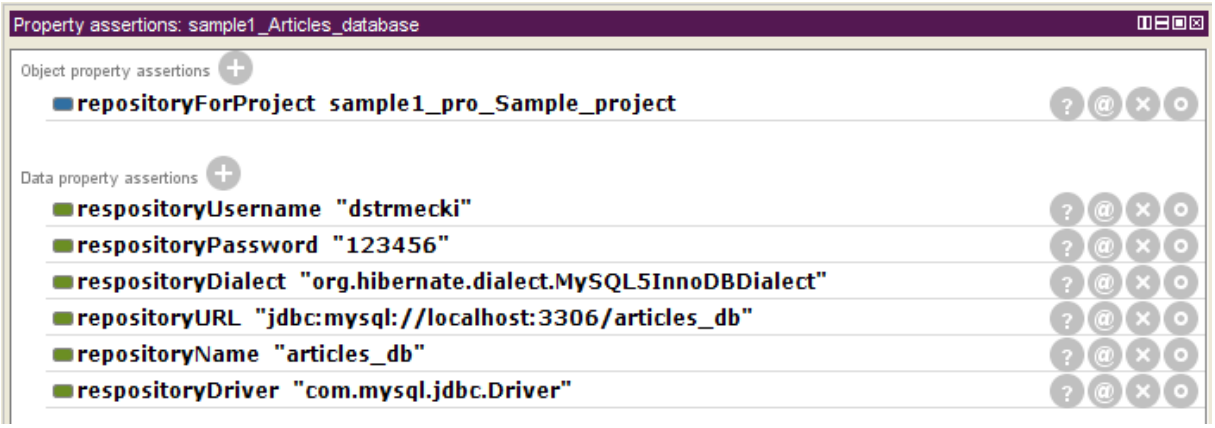
Slika 29: Atributi instance korisničke priče



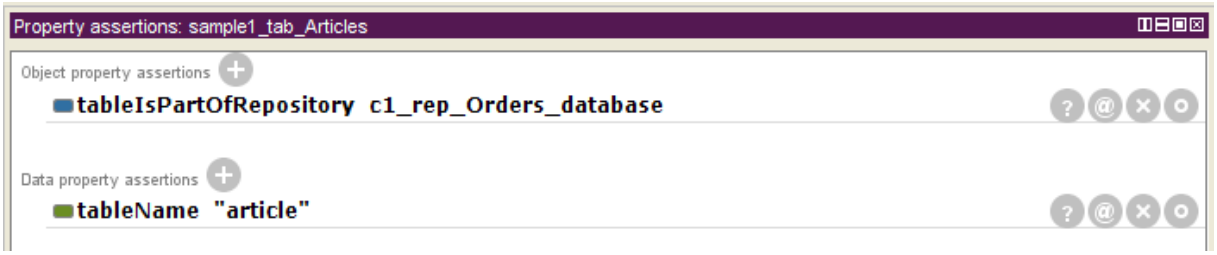
Slika 30: Atributi instance kriterija prihvaćanja

Nakon što smo završili sa specifikacijom korisničkog zahtjeva možemo započeti sa specifikacijom repozitorija, tj. tablica i stupaca u bazi podataka potrebnih za zadovoljenje ovog konkretnog korisničkog zahtjeva. Najprije kreiramo novu instancu repozitorija koja predstavlja bazu podataka, a povezana je sa instancom projekta pomoću objektnog atributa. Atributi instance repozitorija prikazani su na *Slici 31*. Na slici je vidljivo da je za repozitorij definiran *MySQL* dijalekt, odgovarajući upravljački program, URL adresa, naziv baze te korisničko ime i lozinka potrebni za pristup bazi. Nakon kreiranja instance repozitorija, potrebno je kreirati po jednu instancu za svaku tablicu u bazi podataka. U ovom primjeru, kako bi zadovoljili konkretni korisnički zahtjev, potrebna je samo jedna tablica čiji atributi su prikazani na *Slici 32*. Instanca tablice artikala definira samo naziv tablice u bazi te vezu na instancu repozitorija. Kako bismo

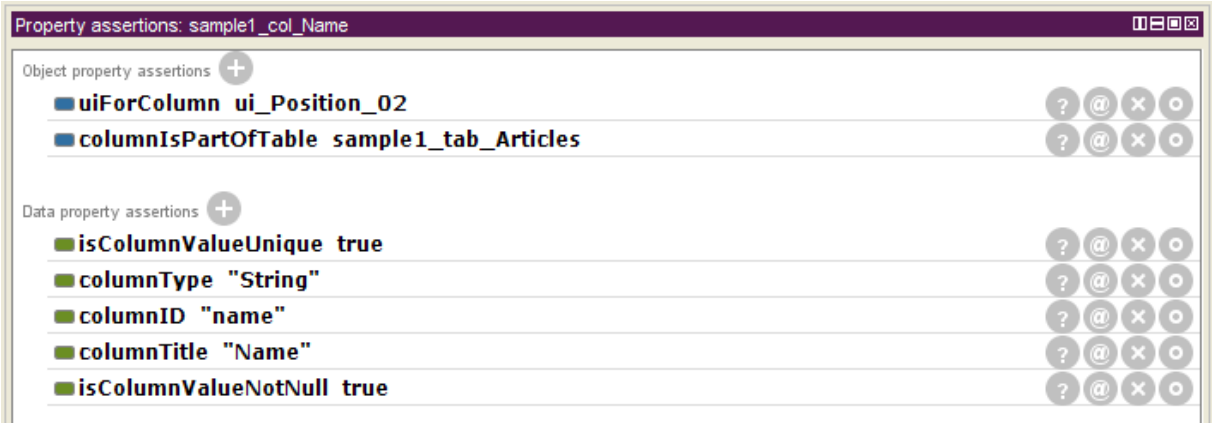
zadovoljili korisničku priču potrebno je definirati nekoliko stupaca povezanih s ovom instancom tablice. Radi pojednostavljenja, prikazati ćemo instancu samo jednog primjera stupca, jer se radi o vrlo sličnim instancama koje se razlikuju samo u vrijednostima atributa. Primjer instance stupca koja predstavlja naziv artikla prikazan je na *Slici 33*. U ovom primjeru stupac je povezan sa tablicom artikla pomoću objektnog atributa te sadrži podatkovne atribute koji definiraju identifikator, naziv i tip stupca te oznake kako stupac u bazi podataka može sadržavati samo uvijek definirane, unikatne vrijednosti. Kako bismo u potpunosti specificirali tablicu artikala, bilo bi potrebno na isti način definirati još instance stupaca za količinu, cijenu, mjernu jedinicu i opis artikla.



Slika 31: Atributi instance repozitorija



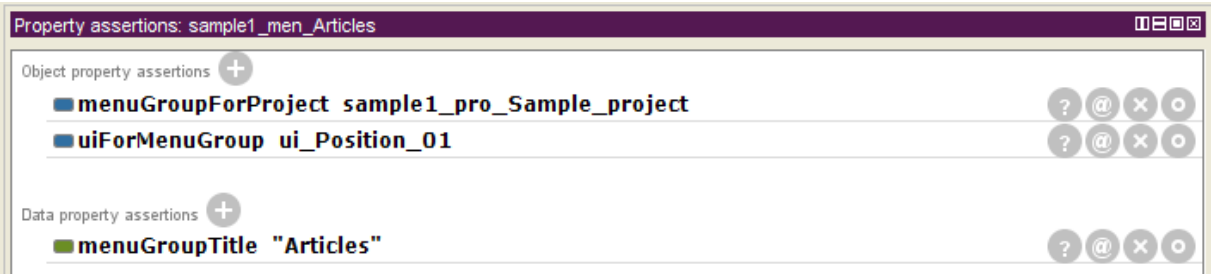
Slika 32: Atributi instance tablice u bazi podataka



Slika 33: Atributi instance stupca u bazi podataka

Nakon specifikacije repozitorija, preostaje još specificirati web obrazac za kreiranje artikala i korištene komponente na obrascu. Prije nego započnemo sa specifikacijom samog obrasca, potrebno je kreirati instancu grupe obrazaca kojoj će budući obrazac pripadati. Atributi instance grupe obrazaca prikazani su na *Slici 34*, a sadrže naziv grupe, vezu na odgovarajući projekt i vezu na instancu korisničkog sučelja koja definira poziciju grupe u glavnom izborniku aplikacije. Sada možemo kreirati instancu web obrasca i povezati je sa kreiranom instancom grupe. Atributi instance web obrasca za pregled i uređivanje artikala prikazani su na *Slici 35*. Osim veze na grupu, na ovoj instanci definiran je još samo identifikator i naslov obrasca koji će biti prikazan korisnicima u glavnom izborniku. U ovome trenutku preostaje još samo definirati komponente koje će biti korištene na obrascu. Za implementaciju odabrane korisničke priče biti će potrebno definirati instance komponenata za unos podataka te komponente gumba koji će koristiti za pokretanje akcije poput pohrane i brisanja zapisa. Na *Slici 36* prikazani su atributi jedne komponente za unos i uređivanje naziva artikla korištene na ovom obrascu. Na slici je vidljivo kako je komponenta povezana sa odgovarajućim obrascem pomoću objektnog atributa. Ova instanca unosne komponente sadrži i vezu na stupac u bazi podataka iz kojeg će čitati i u koji će zapisivati podatke, kao i vezu na instancu korisničkog sučelja koja definira poziciju unosnog polja na ekranu korisnika. Kako bismo u potpunosti implementirali odabranu korisničku priču, bilo bi potrebno na isti način definirati još komponente za unos količine, cijene, mjerne jedinice i opisa artikla. Osim komponenata za unos podataka, potrebno je definirati još i komponente gumba. Na *Slici 37* prikazani su atributi instance gumba za dodavanje novog artikla. Instanca komponente gumba također mora biti povezana sa odgovarajućim obrascem, a sadržava samo identifikator i oznaku komponente. Kompletnu specifikaciju ovog web obrasca možemo iskoristiti za automatsko generiranje programskog

koda. Nakon instalacije generiranog programskog koda na web poslužitelj, obrazac iz perspektive korisnika, u web pregledniku prikazan je na *Slici 38*.

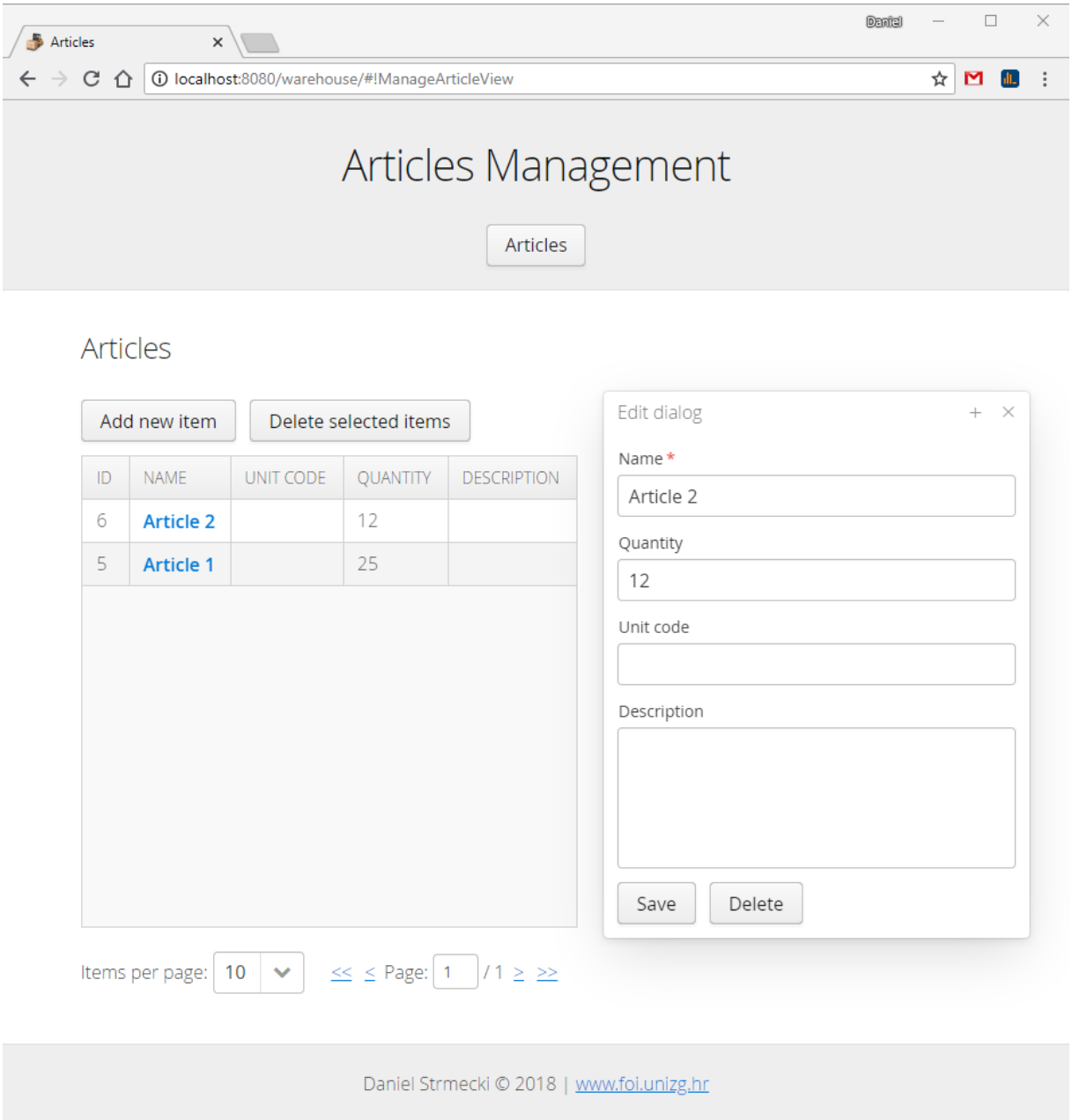


Slika 34: Atributi instance grupe obrazaca

Slika 35: Atributi instance web obrasca

Slika 36: Atributi instance web komponente za unos podataka

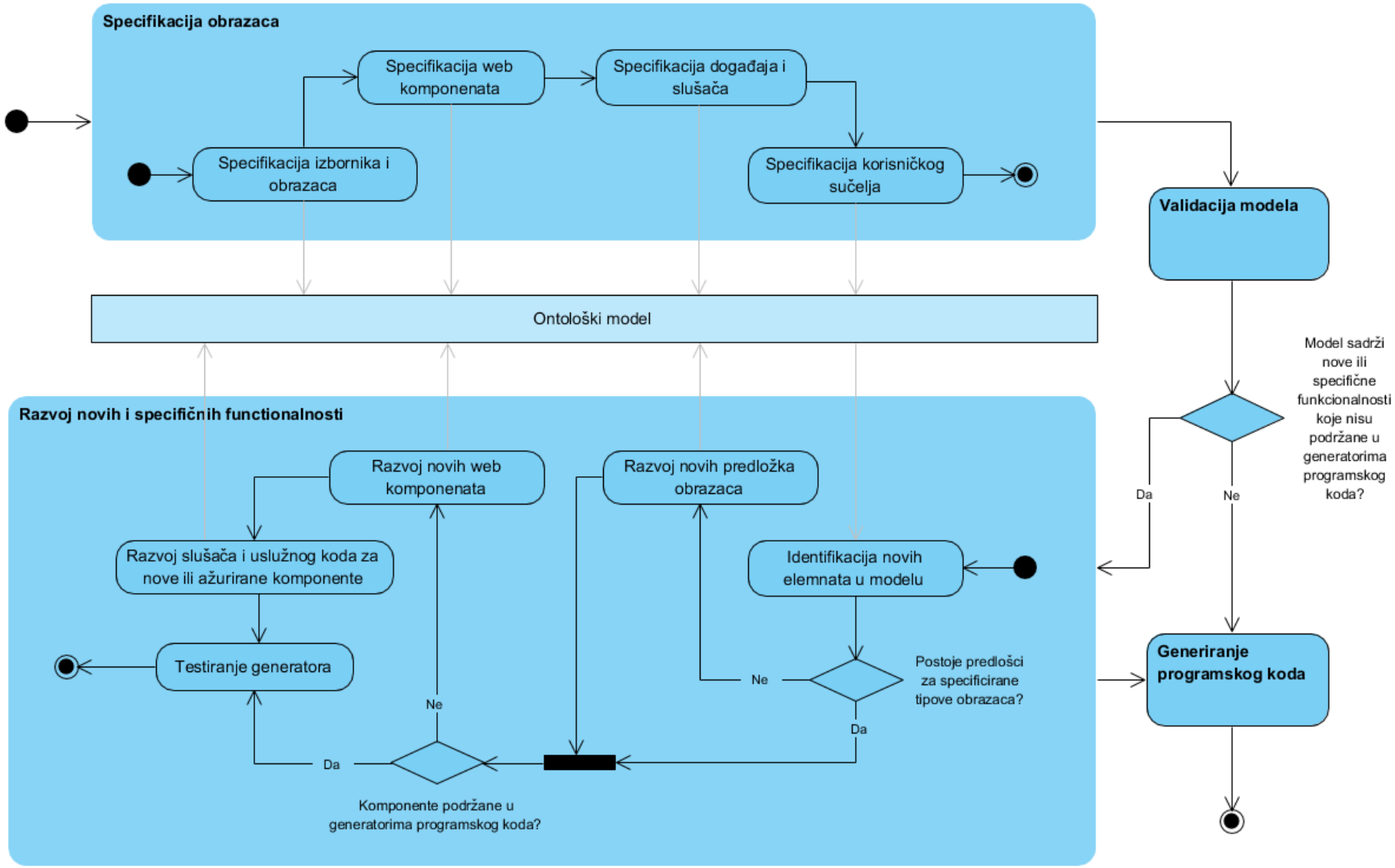
Slika 37: Atributi instance web komponente gumba



Slika 38: Generirani obrazac za upravljanje artiklima

4.3.6. Razvoj novih i specifičnih funkcionalnosti

Mogućnost razvoja novih i specifičnih funkcionalisti jedna je od glavnim prednosti ovog razvojnog okvira, koja ga razlikuje od srodnih pristupa, predstavljanih u poglavlju *1.2.3. Srodni pristupi*. Ontološki model i generatori programskog koda dizajnirani su i građeni modularno kako bi ih se mogli prilagoditi specifičnim zahtjevima, odnosno funkcionalnostima budućih proizvoda. *Slika 39* prikazuje detaljni dijagram aktivnosti specifikacije obrazaca te razvoja novih i specifičnih funkcionalnosti, odnosno kako specifikacija aplikacije utječe na razvoj novih funkcionalnosti. Prilikom provedbe aktivnosti *(4) Specifikacija obrazaca* poslovni analitičar konzultira programera o načinu modeliranja novih ili specifičnih funkcionalnosti. Takve funkcionalnost mogu se modelirati unaprijed, iako još nisu podržane u generatorima programskog koda. Nakon aktivnosti *(5) Validacija modela*, slijedi aktivnosti *(6) Razvoj novih i specifičnih funkcionalnosti*, ali samo u slučaju da je razvoj, odnosno proširenje generatora programskog koda zaista potrebno. Ako jest, tada ova aktivnosti započinje identifikacijom novih elemenata u ontološkom modelu. Nakon toga slijedi provjera postoje li za sve definirane obrasce dostupni odgovarajući predlošci, tj. da li je obrasce uopće moguće automatski generirati pomoću trenutno dostupnih predložaka. U slučaju da u ontološkoj specifikaciji postoji novi tip obrasca, koji nije podržan u trenutnoj verziji generatora programskog koda, tada je za njega potrebno definirati nove predloške, ažurirati programski kod generatora te po potrebi doraditi ontološki model. Slična je procedura i sa dodavanjem podrške za generiranje novih tipova komponenata. Ukoliko ontološka specifikacija sadrži novi tip komponente ili novu funkcionalnost na postojećoj komponenti, koju u trenutnoj verzijom generatora nije moguće automatski generirati, tada je također potrebno ažurirati programski kod generatora te po potrebi doraditi ontološki model. S obzirom da web komponente mogu koristiti slušače događaja i razne oblike popratnog uslužnog programskog koda, programer će prema potrebi dodati i te dijelove koda u generator te ažurirati model kako bi bio u skladu sa podržanim funkcionalnostima generatora. Prilikom provedbe ovih aktivnosti programer će više puta pokretati generatore programskog koda u cilju njihovog testiranja, a po završetku razvoja potrebno je pokrenuti test nad svim finalnim specifikacijama proizvoda kako bi se uvjerili da nove funkcionalnosti nisu utjecale na prestanak rada neke od postojećih funkcionalnosti.



Slika 39: Detaljan dijagram aktivnosti specifikacije obrazaca te razvoja novih i specifičnih funkcionalnosti

5. DEMONSTRACIJA I EVALUACIJA

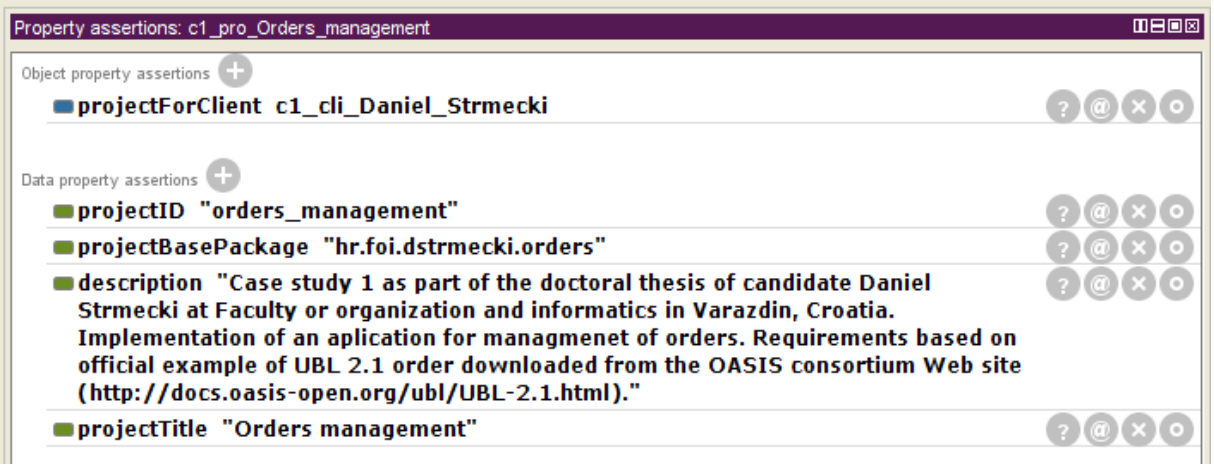
5.1. Studija slučaja 1

5.1.1. Primjer UBL narudžbenice

U studijama slučaja primijeniti ćemo UBL specifikaciju dokumenata i njihovih atributa za razvoj odgovarajuće programske podrške. Prva studija slučaja implementirati će aplikaciju za upravljanje narudžbenicama, prema službenom primjeru preuzetom sa stranice OASIS konzorcija (dostupno u *Dodatku H* ovog rada). Iz navedene XML specifikacije izvučeni su korisnički zahtjevi, odnosno preuzeta su sva polja, tipovi podataka i veze među njima koju UBL narudžbenica treba sadržavati prema službenom primjeru. Valja napomenuti kako ova studija slučaja neće biti referentna implementacija kompletne UBL specifikacije, već samo dijelova pokrivenih službenim primjerom. Implementacija aplikacije za upravljanje narudžbenicama podrazumijeva kreiranje web obrazaca s odgovarajućim elementima korisničkog sučelja, unosna polja i odgovarajuće validacije te poslovnu logiku potrebnu za izračun vrijednosti. Osnovne funkcionalnosti, poput prijave / odjave korisnika i definicije korisničkih prava korisnika biti će također uključene u implementaciju. Navedene funkcionalnosti implementirati ćemo primjenom dizajniranog okvira te kreiranih artefakata (ontološki model i generatori programskog koda) u sklopu ovog doktorskog rada. S obzirom da trenutna verzija generatora podržava generiranje samo jednostavnih web obrazaca, u sklopu ovih studija slučaja, u fazi *(6) Razvoj novih i specifičnih funkcionalnosti* implementirati ćemo i neke nove funkcionalnosti.

5.1.2. Specifikacija projekta i korisničkih zahtjeva

Prema razvojnom ciklusu dizajniranog okvira, razvoj započinjemo aktivnošću *(1) Specifikacija projekta*. Na *Slici 40* prikazani su atributi kreirane instance projekta koji sadrže generalne podatke o projektu, odnosno prvoj studiji slučaja. Instanca projekta povezana je sa klijentom za kojeg se projekt izrađuje, dok će sama instanca projekta kasnije poslužiti za povezivanje svih ostalih instanci s odgovarajućim projektom.



Slika 40: Atributi instance projekta studije slučaja 1

Nakon toga prelazimo na aktivnost *(2) Specifikacija korisničkih zahtjeva*. Analizom primjera narudžbenice (XML dokumenta iz *Dodatka H* ovog rada) identificirali smo dvanaest web obrazaca koje je potrebno razviti kako bi se omogućilo upravljanje narudžbenicama ovog tipa putem web aplikacije. Za svaki web obrazac raspisani su korisnički zahtjevi, koji su prikazani u *Tablici 13*. Na temeljnu tih korisničkih zahtjeva kreirane su instance u ontološkom modelu. Osim korisničkih zahtijeva za svaki obrazac, dodan je i jedan generički korisnički zahtjev koji se odnosi na upravljanje korisnicima i korisničkim pravima unutar same aplikacije. Prikaz svih instanci korisničkih zahtjeva zauzeo bi previše prostora u ovom dokumentu pa je u ovom poglavlju prikazan samo reprezentativni primjer jednog korisničkog zahtjeva na *Slici 41*. Radi se o korisničkom zahtjevu koji treba omogućiti kreiranje novih korisnika unutar aplikacije te dodjelu korisničkih prava istima. Primjer jedne povezane korisničke priče prikazan je na *Slici 42*. Ova korisnička priča definira da korisnik sa administrativnim privilegijama treba biti u mogućnosti dodijeliti prava za čitanje, dodavanje, ažuriranje i brisanje određenih tipova entiteta. Za navedenu korisničku priču definirano je više instanci kriterija prihvaćanja, a jedna od njih prikazan je na *Slici 43*. Radi se o kriteriju prihvaća koji definira da samo korisnici s

pravom kreiranja specifičnih entiteta zaista mogu kreirati nove entitete kroz korisničko sučelje aplikacije. Sve preostale instance specifikacije moguće je pregledati tekstualno u *Dodatku G* ili digitalno u *Dodatku J* ovog doktorskog rada.

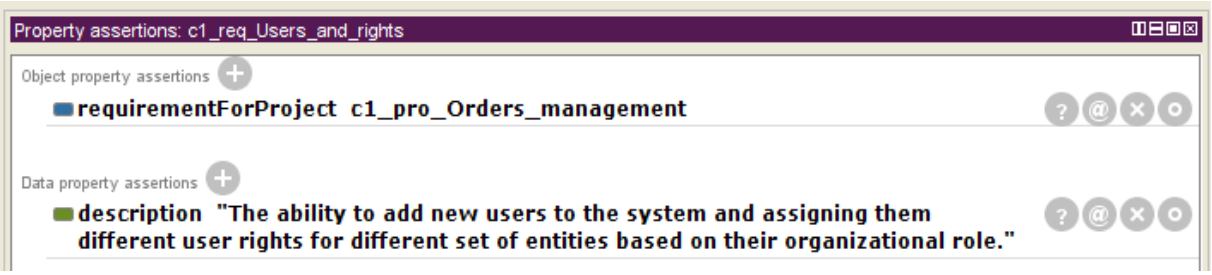
Tablica 12: Korisnički zahtjevi za studiju slučaja 1

Naziv obrasca	Korisnički zahtjev
Adrese	Mogućnost dodavanja novih adresa i uređivanje ili brisanje postojećih adresa. Jedna adresa može biti povezana sa jednom ili više stranaka ili narudžbenica. Adresa treba sadržavati slijedeća polja: identifikator adrese, naziv, naziv ulice, prošireni naziv ulice, broj zgrade, odjel, grad, poštanski kod, poštanska zona, država, državni subjekt. Samo naziv adrese je obavezno polje, a država treba biti polje za odabir.
Artikli	Mogućnost dodavanja novih artikala i uređivanje ili brisanje postojećih artikala. Jedan artikl može biti povezan sa jednom ili više stavaka narudžbenice. Artikl treba sadržavati slijedeća polja: naziv, količina, cijena, mjerna jedinica, identifikator prodavatelja, identifikator artikla i opis. Samo naziv artikla je obavezno polje. Jedan artikl može sadržavati nekoliko povezanih parova atributa i vrijednosti (npr, boja, visina, širina, itd.).
Kontakti	Mogućnost dodavanja novih kontakata i uređivanje ili brisanje postojećih kontakata. Jedan kontakt može se dodati na više stranaka. Kontakt treba sadržavati slijedeća polja: naziv, telefon, faks i e-pošta. Samo naziv kontakta je obavezno polje.
Države	Mogućnost dodavanja novih država i uređivanje ili brisanje postojećih država. Jedna država može se dodati na više adresa. Država treba sadržavati slijedeća polja: naziv i kod. Samo naziv države je obavezno polje.
Reference na dokumente	Mogućnost dodavanja novih referenci na dokumente i uređivanje ili brisanje postojećih. Jedna referenca može biti povezana sa jednom ili više narudžbenica. Referenca na dokument treba sadržavati slijedeća polja: naziv, tip reference, URL i napomena. Naziv i tip reference su obavezna polja. Tip reference može biti ponuda, narudžbenica, inicijator, ugovor ili dodatno.

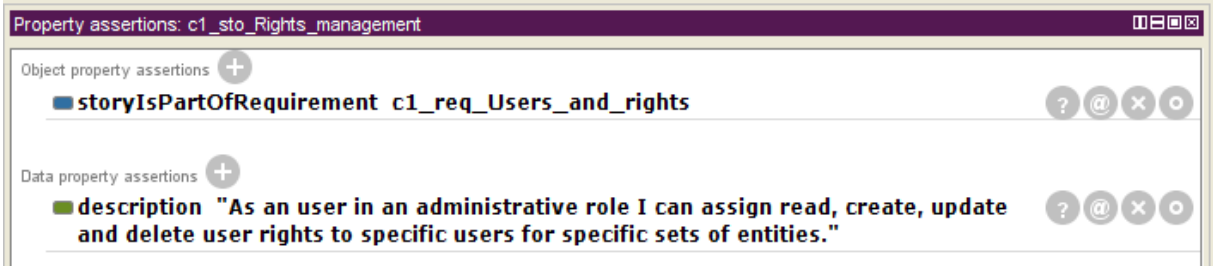
Incoterms[25]	Mogućnost dodavanja novih *Incoterms* i uređivanje ili brisanje postojećih *Incoterms*. Jedan *Incoterm* može biti povezan sa jednom ili više narudžbenica. *Incoterm* treba sadržavati slijedeća polja: naziv i opis. Samo naziv *Incoterm*-a je obavezno polje.
Pravne osobe	Mogućnost dodavanja novih pravnih osoba i uređivanje ili brisanje postojećih pravnih osoba. Jedna pravna osoba može biti povezana sa jednom ili više stranaka. Pravna osoba treba sadržavati slijedeća polja: naziv, identifikator tvrtke i adresa. Samo naziv pravna osobe je obavezno polje, a adresa treba biti polje za odabir.
Narudžbenice	Mogućnost dodavanja novih narudžbenica i uređivanje ili brisanje postojećih narudžbenica. Narudžbenica treba sadržavati slijedeća polja: naziv, datum i vrijeme izdavanja, napomena, vrijedi od / do, šifra valute, šifra računovodstvenih troškova, stranka kupca, stranka prodavatelja, stranka inicijatora, adresa dostave, stranka dostave, uvjeti dostave, posebni uvjeti, iznos doplatka, iznos naknade, iznos linije, iznos poreza i naplativi iznos. Naziv, stranka kupca i stranka prodavatelja su obavezna polja. Stranka kupca, stranka prodavatelja, stranka inicijatora, mjesto isporuke, stranka isporuke, uvjeti isporuke i posebni uvjeti trebaju biti polja za odabir. Iznos linije, iznos poreza i naplativi iznos su polja koja se automatski izračunavaju iz vrijednosti dodanih stavaka. Iznos naplate izračunava se prema formuli: ukupni iznos linije + ukupni iznos poreza + iznos doplatka – iznos naknade. Narudžbenice mogu sadržavati stavke koje se sastoje od slijedeća polja: artikl, količina, mjerna jedinica, iznos poreza, iznos linije, šifra računovodstvenih troškova, oznaka djelomične isporuke i zatražena isporuka od / od. Artikl i količina su obavezna polja za stavke. Oznaka djelomične isporuke je potvrdno polje. Narudžbenice osim stavaka, mogu sadržavati i reference na jedan ili više dokumenta.
Stranke	Mogućnost dodavanja novih stranaka i uređivanje ili brisanje postojećih stranaka. Jedna stranka može biti povezana sa jednom ili više narudžbenica. Stranka treba sadržavati slijedeća polja: naziv, identifikator stranke, adresa, porezna shema i pravna osoba. Samo

[25] Naziv *Incoterms* zapravo predstavlja skraćenicu za međunarodne trgovačke termine (engl. *International commercial terms*). To su službena pravila Međunarodne trgovinske komore za tumačenje trgovinskih termina.

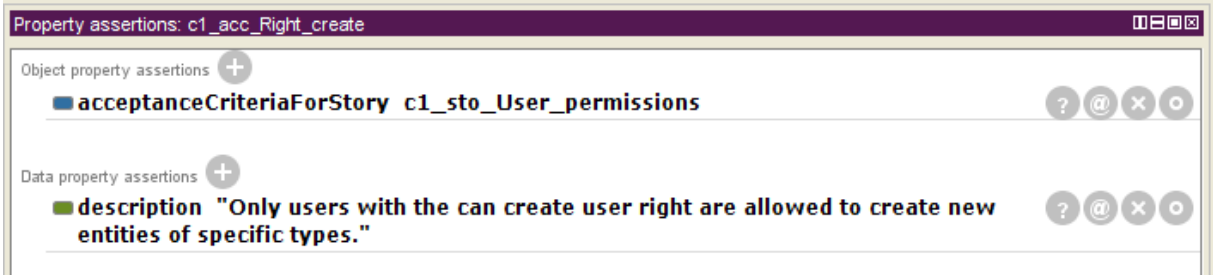
	naziv stranke je obavezno polje. Stranke mogu biti povezane sa više kontakata, kontakata za dostavu i privatnih osoba.
Porezne sheme	Mogućnost dodavanja novih poreznih shema i uređivanje ili brisanje postojećih poreznih shema. Jedna porezna shema može biti povezana sa jednom ili više narudžbenica. Porezna shema treba sadržavati slijedeća polja: naziv i opis. Samo naziv porezne sheme je obavezno polje.
Porezne sheme stranaka	Mogućnost dodavanja novih poreznih shema stranaka i uređivanje ili brisanje postojećih. Jedna porezna shema stranke može biti povezana sa jednom ili više stranaka. Porezna shema stranke treba sadržavati slijedeća polja: naziv, identifikator tvrtke, adresa i porezna shema. Samo naziv porezne sheme je obavezno polje, a adresa i porezna shema trebaju biti polja za odabir.
Privatne osobe	Mogućnost dodavanja novih privatnih osoba i uređivanje ili brisanje postojećih privatnih osoba. Jedna privatna osoba može biti povezana sa jednom ili više stranaka. Privatna osoba treba sadržavati slijedeća polja: naziv, ime, prezime, srednje ime i titulu. Samo naziv privatne osobe je obavezno polje.
Korisnici i prava	Mogućnost dodavanja novih korisnika u sustav i dodjeljivanje korisničkih prava korisnicima na temelju njihove uloge u organizaciji.



Slika 41: Atributi instance korisničkog zahtjeva za studiju slučaja 1



Slika 42: Atributi instance korisničke priče za studiju slučaja 1

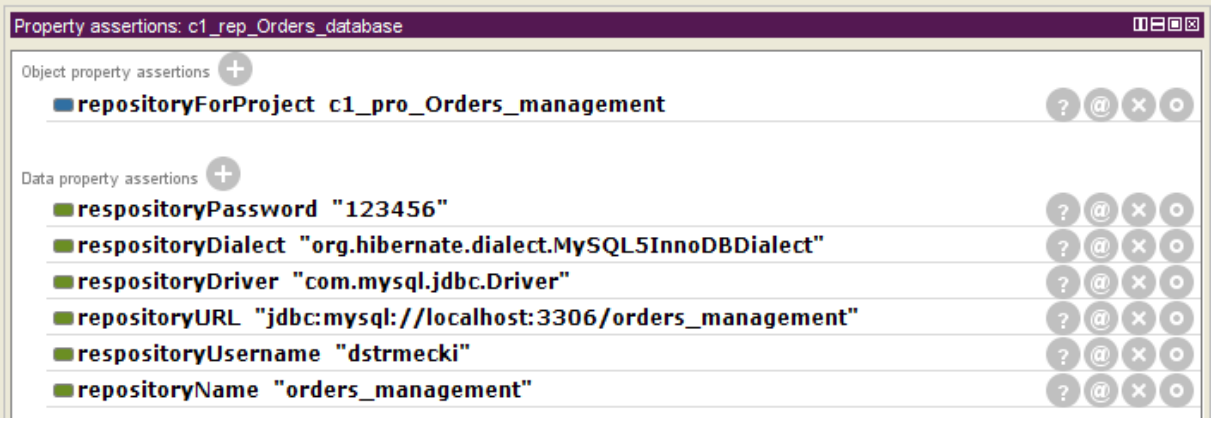


Slika 43: Atributi instance kriterija prihvaćanja za studiju slučaja 1

5.1.3. Specifikacija repozitorija

Slijedeću aktivnost, odnosno *(3) Specifikacija repozitorija*, započinjemo kreiranjem instance koja definira bazu podataka, a čiji atributi su prikazani na *Slici 44*. Instanca repozitorija povezana je sa instancom projekta prve studije slučaja i sadrži podatkovne atribute potrebne za spajanje aplikacije na bazu podataka. Nakon kreiranja instance repozitorija sastavljamo popis potrebnih tablica i stupaca potrebnih za realizaciju ranije definiranih korisničkih zahtjeva. Ovaj popis prikazan je u *Tablici 13*. Prikaz svih instanci tablica i stupaca zauzeo bi previše prostora u ovom dokumentu pa su u ovom poglavlju prikazani samo reprezentativni primjeri. Primjer instance tablice prikazan je na *Slici 45*. Radi se o jednostavnoj instanci tablice koja se povezuje objektnim atributom sa odgovarajućom instancom repozitorija. Primjer instance stupca prikazan je na *Slici 46*. Primjer prikazuje atribute instance stupca za pohranu adrese, koji se koristi za povezivanje jedan naprema više sa tablicom koja sadrži detaljnije informacije o adresama. Na ovom primjeru instance stupca možemo uočiti jednu značajnu prednost primjena ontološke specifikacije (u usporedbi sa klasičnom tekstualnom specifikacijom), a to je da jednom definiranu instancu stupca možemo povezati sa više različitih tablica. Naravno, pod uvjetom da korisnički zahtjevi to dopuštaju. Npr. ako sve tablice u bazi podataka koriste identičan stupac (npr. naziv ili opis), dovoljno ga je specificirati samo jednom i povezati sa

tablicama pomoću objektnog atributa. Preostale instance tablica i stupaca u bazi podataka moguće je pregledati tekstualno u *Dodatku G* ili digitalno u *Dodatku J* ovog doktorskog rada.

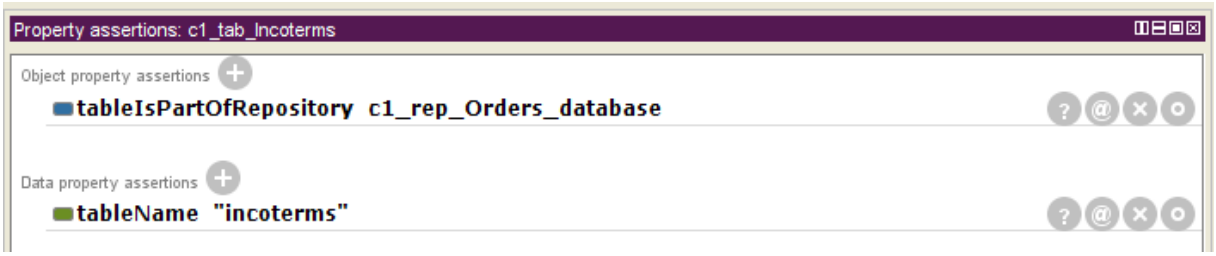


Slika 44: Atributi instance repozitorija studije slučaja 1

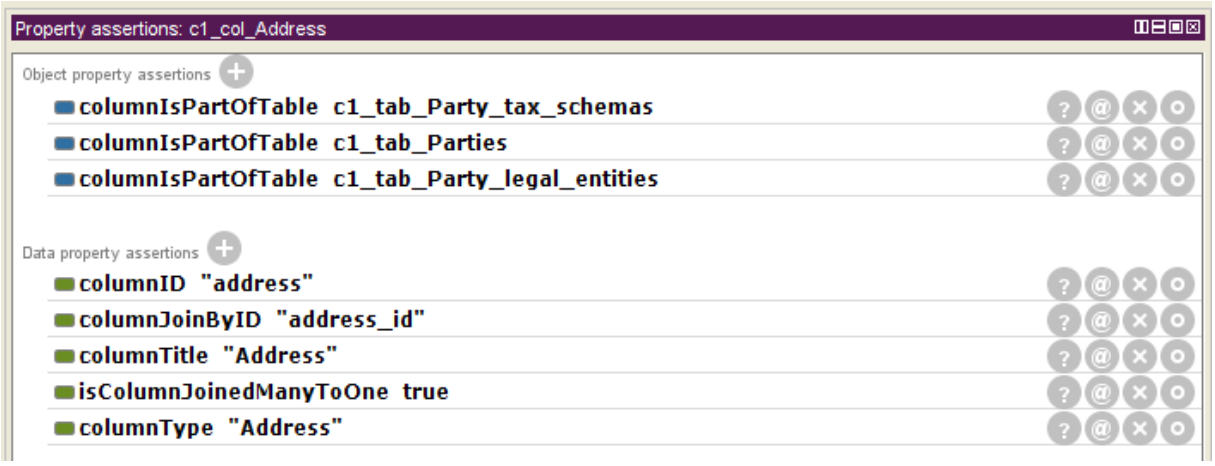
Tablica 13: Potrebne tablice u bazi podataka za realizaciju studije slučaja 1

Naziv tablice	Stupci
Adrese	ID, naziv, identifikator adrese, naziv ulice, prošireni naziv ulice, broj zgrade, odjel, grad, poštanski kod, poštanska zona, država i državni subjekt.
Artikli	ID, naziv, bazna količina, cijena, mjerna jedinica, identifikator prodavatelja, identifikator artikla i opis.
Kontakti	ID, naziv, telefon, fax i e-pošta.
Kontakti - Stranke	Kontakt i stranka.
Kontakti - Dostava	Kontakt i stranka.
Države	ID, naziv i kod.
Reference na dokumente	ID, naziv, URL i napomena.
Reference - Narudžbenice	Referenca na dokument i narudžbenica.
Incoterms	ID, naziv i opis.
Narudžbenice	ID, naziv, datum i vrijeme izdavanja, napomena, vrijedi od datuma, vrijedi do datuma, šifra valute, šifra računovodstvenih troškova, stranka kupca, stranka prodavatelja, stranka inicijatora, adresa dostave, stranka

	dostave, uvjeti dostave, posebni uvjeti, iznos doplatka, iznos naknade, iznos linije, iznos poreza, naplativi iznos i napomena.
Narudžbenice - Stavke	ID, artikl, količina, mjerna jedinica, iznos poreza, iznos linije, šifra računovodstvenih troškova, oznaka djelomične isporuke, zatražena isporuka od datuma, zatražena isporuka do datuma i narudžbenica.
Stranke	ID, naziv, identifikator stranke, adresa, porezna shema stranke i pravni entitet stranke.
Stranke - Pravne osobe	ID, naziv, identifikator tvrtke, adresa, registracija i pravna osoba.
Stranke - Porezne sheme	ID, naziv, identifikator tvrtke, adresa, registracija i porezna shema.
Privatne osobe	ID, naziv, ime, prezime, srednje ime i titula.
Privatne osobe - Stranke	Privatna osoba i stranka.
Porezne sheme	ID, naziv i opis.
Atributi artikala	ID, atribut, vrijednost i artikl.



Slika 45: Atributi instance tablice u bazi podataka za studiju slučaja 1



Slika 46: Atributi instance stupca tablice za studiju slučaja 1

5.1.4. Specifikacija obrazaca i komponenata

Slijedeća aktivnost jest *(3) Specifikacija obrazaca*, koja podrazumijeva i specifikaciju svih komponenata korištenih na pojedinom obrascu. U *Tablici 14* navedeni su svi potrebni obrasci za realizaciju ove aplikacije i njihove komponente koje je potrebno specificirati kroz ontološki model. Obrasce najprije grupiramo kako bismo ih elegantnije prikazali u glavnom izborniku aplikacije. *Slika 47* prikazuje primjer jedne instance grupe obrazaca, koja je povezana sa odgovarajućom instancom projekta i instancom korisničkog sučelja koja definira poziciju grupe obrazaca u glavnom izborniku. *Slika 48* prikazuje primjer jedne instance web obrasca poveznog sa odgovarajućom grupom. Na primjeru ovog obrasca vidljiva je veza prema drugom obrascu, tj. obrascu za uređivanje zapisa. To zapravo znači da će entiteti stranaka biti prikazani na jednom obrascu, ali će se uređivati na drugom (umjesto u skočnom prozoru). *Slika 49* prikazuje primjer instance jednog (datumskog) unosnog polja koje se koristi na obrascu. Ova instanca komponente povezana je sa instancom obrasca na kojem će se koristiti, instancom korisničkog sučelja koja definira njezinu poziciju na ekranu korisnika te stupcem u bazi podataka koji će koristiti za dohvaćanje i pohranu podataka. Preostale instance obrazaca i komponenata moguće je pregledati tekstualno u *Dodatku G* ili digitalno u *Dodatku J* ovog doktorskog rada.

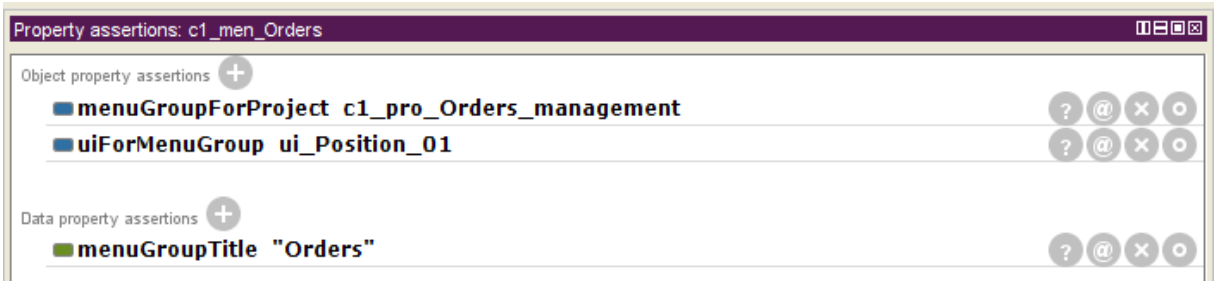
Tablica 14: Obrasci i komponente za studiju slučaja 1

Naziv obrasca	Potrebne komponente
Pregled narudžbenica	Tablica s paginacijom, gumb za dodavanje novog zapisa i gumb za brisanje označenih zapisa.

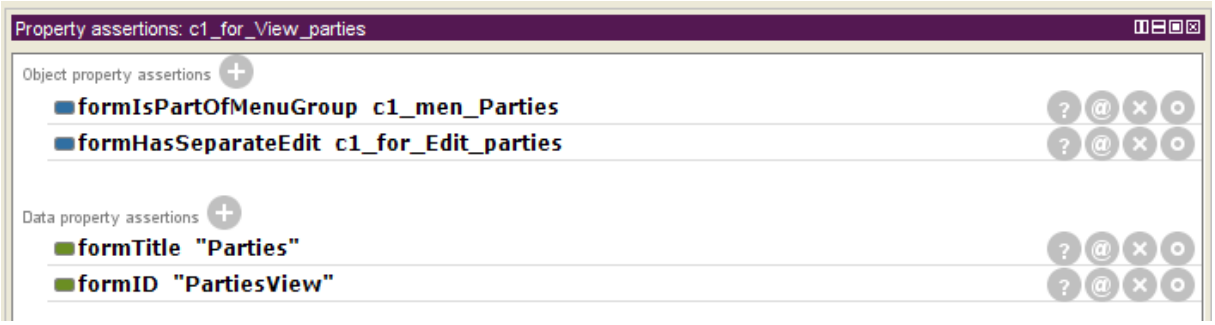
Uređivanje narudžbenica	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, datumsko polje za unos datuma izdavanja, tekstualno polje za unos napomene, datumsko polje za unos datuma važenja od, datumsko polje za unos datuma važenja do, tekstualno polje za unos valute dokumenta, tekstualno polje za unos šifre računovodstvenog troška, polje za odabir stranke kupca, polje za odabir stranke prodavatelja, polje za odabir stranke inicijatora, polje za odabir adrese dostave, polje za odabir stranke dostave, polje za odabir uvjeta dostave, polje za odabir posebnih uvjeta, numeričko polje (dvije decimale) za unos iznosa doplatka, numeričko polje (dvije decimale) za unos iznosa naknade, numeričko polje (dvije decimale) za prikaz ukupnog iznosa linije, numeričko polje (dvije decimale) za prikaz ukupnog iznosa poreza, numeričko polje (dvije decimale) za unos naplativog iznosa, numeričko polje (cjelobrojno) za unos količine, tekstualno polje za unos mjerne jedinice, numeričko polje (dvije decimale) za unos iznosa poreza, numeričko polje (dvije decimale) za unos iznosa linije, potvrdno polje za označavanje djelomične dostave, datumsko polje za unos željenog datuma dostave od, datumsko polje za unos željenog datuma dostave do i polje za odabir artikla.
Pregled artikla	Tablica s paginacijom, gumb za dodavanje novog zapisa i gumb za brisanje označenih zapisa.
Uređivanje artikala	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, numeričko polje (cjelobrojno) polje za unos količine, numeričko polje (dvije decimale) polje za unos cijene, tekstualno polje za unos mjerne jedinice, tekstualno polje za unos oznake prodavatelja, tekstualno polje za unos oznake artikla, tekstualno polje za unos opisa, tekstualno polje za atributa i tekstualno polje za unos vrijednosti.

Pregled i uređivanje kontakata	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, tekstualno polje za unos telefona (validacija formata), tekstualno polje za unos faxa i tekstualno polje za unos e-pošte (validacija formata).
Pregled i uređivanje pravnih osoba	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziv, tekstualno polje za unos identifikatora tvrtke i polje za odabir adrese.
Pregled i uređivanje poreznih shema	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva i tekstualno polje za unos opisa.
Pregled i uređivanje poreznih shema stranaka	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, tekstualno polje za unos identifikatora tvrtke, polje za odabir adrese i polje za odabir porezne sheme.
Pregled i uređivanje privatnih osoba	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, tekstualno polje za unos imena, tekstualno polje za unos prezimena, tekstualno polje za unos srednjeg imena i tekstualno polje za unos titule.
Pregled stranka	Tablica s paginacijom, gumb za dodavanje novog zapisa i gumb za brisanje označenih zapisa.
Uređivanje stranaka	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, tekstualno polje za unos identifikatora stranke, polje za odabir porezne sheme stranke, polje za odabir pravne osobe i polje za odabir adrese.

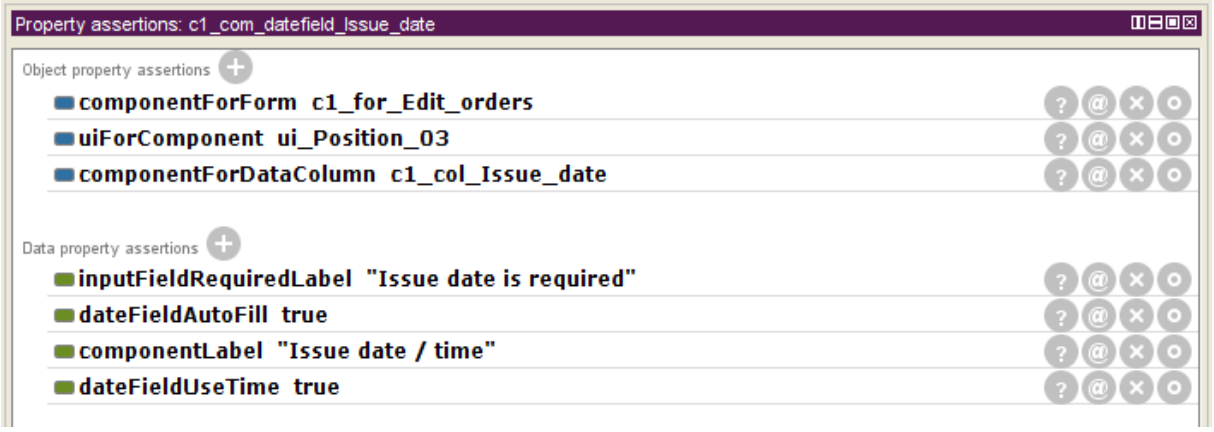
Pregled i uređivanje država	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva i tekstualno polje za unos koda.
Pregled adresa	Tablica s paginacijom, gumb za dodavanje novog zapisa i gumb za brisanje označenih zapisa.
Uređivanje adresa	Tablica s paginacijom, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, tekstualno polje za unos identifikatora adrese, tekstualno polje za unos ulice, tekstualno polje za unos dodatnog naziva ulice, numeričko polje (cjelobrojno) za unos broja zgrade, tekstualno polje za unos odjela, tekstualno polje za unos grada, numeričko polje (cjelobrojno) za unos poštanskog broja, tekstualno polje unos za poštanske zone, tekstualno polje za unos državnog subjekta i polje za odabir države.
Pregled i uređivanje Incoterms	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za naziv i tekstualno polje za opis.
Pregled i uređivanje referenci na dokumente	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, tekstualno polje za unos opisa, tekstualno polje za unos URL-a, tekstualno polje za unos napomene i polje za odabir tipa reference.



Slika 47: Atributi instance grupe obrazaca za studiju slučaja 1



Slika 48: Atributi instance obrasca za studiju slučaja 1



Slika 49: Atributi instance datumskog polja za studiju slučaja 1

5.1.5. Validacija modela i razvoj novih funkcionalnosti

Aktivnost *(5) Validacija modela* provedena je pokretanjem zadanog semantičkog argumentatora *FaCT++* i testnim pokretanjem generatora programskog koda. Primjenom argumentatora pronađene su jednostavne pogreške u specifikaciji, poput pogrešno definiranih tipova vrijednosti ili nedostatka obaveznih atributa. Testnim pokretanjem generatora pronađeno je još nekoliko grešaka povezanih sa objektnim atributima, gdje su nedostajale određene veze među elementima specifikacije ili su objektnim atributima bile povezane pogrešne instance. Nakon ispravljanja pronađenih grešaka slijedi aktivnost *(6) Razvoj novih i specifičnih funkcionalnosti*. S obzirom da se radi o razvoju prvog proizvoda iz ove obitelji, identificirani su određeni elementi u specifikaciji koji nisu bili podržani od strane generatora programskog koda. Stoga smo iste morali doraditi, odnosno razviti te uskladiti generatore programskog koda i ontološki model. Inicijalna verzija generatora podržavala je samo jednostavne obrasce kojima se uređuje jedan zapis u bazi podatka. Međutim, kako je za ovu studiju slučaja bilo potrebno povezivati različite kombinacije zapisa u bazi podataka, kao i dodavati stavke pojedinim

zapisima, navedenu funkcionalnost morali smo implementirati u generatorima. U ovom slučaju nije bilo potrebe za kreiranjem novog predloška obrasca, već samo za nadogradnjom postojećeg predloška. Najprije je dodana mogućnost dodavanja i uređivanja stavaka na obrascu za uređivanje, kroz definiranje objektnih atributa za povezivanje instanca gumba sa instancama obrazaca. Zatim je dodana mogućnost pregleda i dodavanja veza na druge zapise na obrascu za uređivanje, kroz definiranje dodatnih instanci tablica sa paginacijom. Primjer ovakve instance je prikazan na *Slici 50*, gdje je vidljivo da instanca komponente, tipa tablica sa paginacijom, koristi nekoliko dodatnih objektnih atributa kako bi prikazala rezultate iz više spojenih tablica u bazi podataka. Osim toga, na određenim klasama dodani su novi podatkovni atributi kako bi se omogućila implementacija specifičnih korisničkih zahtjeva. Ovi novi atributi prikazani su u *Tablici 15*, a za svaki od njih bilo je potrebno nadograditi generatore programskog koda i predloške. Na primjer, kako bismo podržali novi atribut unosnih polja koji služi za pretvaranje unesenog niza znakova u cjelobrojni ili decimalni broj (atribut *inputFieldConverter*) bilo je potrebno:

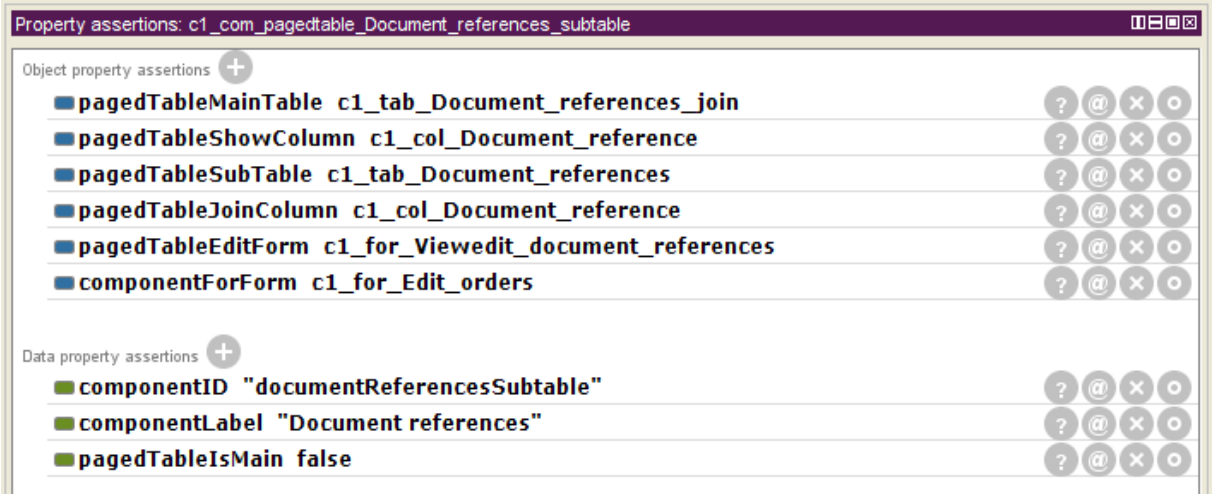
- Dodati novu varijablu u klasu koja reprezentira specifikaciju unosnog polja, kao što je prikazano na *Slici 51*,
- Dodati novu uvjetnu naredbu za pretvaranje vrijednosti u predložak za postavljanje vrijednosti unosnih polja, kao što je prikazano na *Slici 52* i
- Dodati novu uvjetnu naredbu za pretvaranje vrijednosti u predložak za renderiranje unosnih polja, kao što je prikazano na *Slici 53*.

Detaljni način implementacije ostalih funkcionalnosti u generatorima programskog koda moguće je pregledati u izvornom kodu generatora koji je dostupan u *Dodatku J* ovog doktorskog rada.

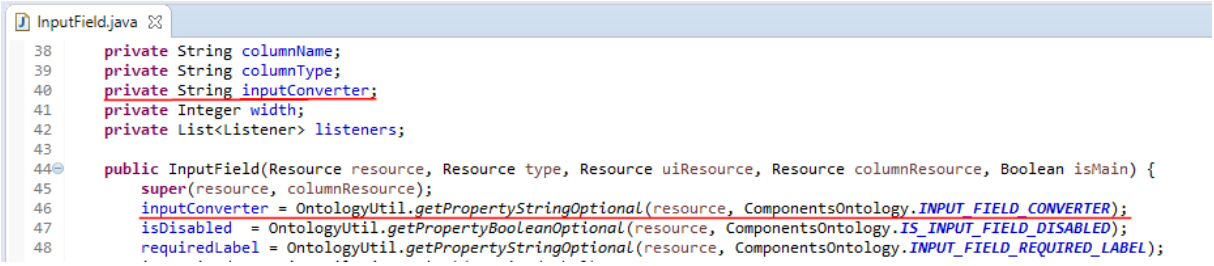
Tablica 15: Podatkovni atributi dodani u ontološki model

Atribut	**Naziv (engl.)**	**Domena (engl.)**	**Opseg (engl.)**
Pretvarač unosnog polja	*inputFieldConverter*	*InputField*	*{"Double" , "Integer"}*
Dozvoljene vrijednosti u stupcu	*columnAllowedValues*	*Column*	*String*

Automatsko popunjavanje datumskog polja	*dateFieldAutoFill*	*DateField*	*Boolean*
Oznaka primarne tablice s paginacijom	*pagedTableIsMain*	*PagedTable*	*Boolean*
Upit za ažuriranje vrijednosti polja	*inputFieldUpdateQuery*	*InputField*	*String*



Slika 50: Instanca tablice sa paginacijom za studiju slučaja 1



Slika 51: Isječak programskog koda klase koja reprezentira unosno polje

SetInputFields.java.vm

```
1          #if ($inputField.inputConverter)
2      try {
3          record.set${StringUtils.capitalize($inputField.columnName)}(($inputField.inputConverter) ${inputFie
4      }
5      catch (ConversionException ex) {
6          ViewUtils.showNotification("${inputField.componentLabel} number conversion error", Type.WARNING_MES
7          ${inputField.componentID}.setValue("0");
8          em.getTransaction().rollback();
9          return;
10     }
11         #elseif ($inputField.inputType == "OPTION_GROUP")
12     record.set${StringUtils.capitalize($inputField.columnName)}(${mainPagedTable.tableClass}.${inputField.cc
13         #elseif ($inputField.inputType == "COMBO_BOX")
14     if (${inputField.componentID}.getValue() != null)
15         record.set${StringUtils.capitalize($inputField.columnName)}(em.find(${inputField.columnType}.class,
16     else
17         record.set${StringUtils.capitalize($inputField.columnName)}(null);
18         #else
19     record.set${StringUtils.capitalize($inputField.columnName)}(${inputField.componentID}.getValue());
20         #end
```

Slika 52: Isječak programskog koda predloška za postavljanje vrijednosti unosnih polja

RenderInputFields.java.vm

```
29     ${inputField.componentID}.setRequired(true);
30     ${inputField.componentID}.setRequiredError("${inputField.requiredLabel}");
31         #end
32         #if ($inputField.inputConverter)
33     ${inputField.componentID}.setConverter(${inputField.inputConverter}.class);
34         #end
35         #if ($inputField.validatorRegEx)
36     ${inputField.componentID}.addValidator(new RegexpValidator("${inputField.validatorRegEx}",
37         #end
```

Slika 53: Isječak programskog koda predloška za renderiranje unosnih polja

5.1.6. Generiranje programskog koda

Aktivnost *(7) Generiranje programskog koda* prilično je trivijalna, jer se svodi na pokretanje generatora programskog koda klikom na gumb te praćenje ispisa poruka u konzoli, kao što je prikazano na *Slici 54*. U ovome primjeru uspješno je generirano šesnaest web obrazaca. Generirani programski kod, odnosno generirane klase ovih web obrazaca prikazane su na *Slici 55*. Samo generirane klase obrazaca (bez klasa entiteta, uslužnih klasa i slično) sadrže više od 6900 linija generiranog programskog koda.

```
Markers  Properties  Servers  Data Source Explorer  Snippets  Problems  Console  Debug  History
<terminated> Application [Java Application] C:\Program Files\Java\jdk1.8.0_151\bin\javaw.exe (2. ruj 2018. 11:31:28)
DEBUG: Ontology model loaded.
DEBUG: Creating root folder...
DEBUG: Root folder created.
DEBUG: Generating project structure...
DEBUG: Project structure generated.
DEBUG: Generating utilities and exstentions...
DEBUG: Utilities and exstentions generated.
DEBUG: Generating data entities...
DEBUG: Generated table document_reference
DEBUG: Generated table contacts_join
DEBUG: Generated table incoterms
DEBUG: Generated table delivery_contacts_join
DEBUG: Generated table country
DEBUG: Generated table item_property
DEBUG: Generated table party_tax_scheme
DEBUG: Generated table order_items
DEBUG: Generated table party
DEBUG: Generated table order1
DEBUG: Generated table person
DEBUG: Generated table persons_join
DEBUG: Generated table tax_scheme
DEBUG: Generated table contact
DEBUG: Generated table document_reference_join
DEBUG: Generated table article
DEBUG: Generated table address
DEBUG: Generated table party_legal_entity
DEBUG: Data entities generated...
DEBUG: Generating forms...
DEBUG: Generated form OrdersView
DEBUG: Generated form ArticlesView
DEBUG: Generated form ContactsViewEdit
DEBUG: Generated form LegalEntitiesViewEdit
DEBUG: Generated form PartyTaxSchemeViewEdit
DEBUG: Generated form PersonsViewEdit
DEBUG: Generated form PartiesView
DEBUG: Generated form CountriesViewEdit
DEBUG: Generated form AddressView
DEBUG: Generated form IncotermsViewEdit
DEBUG: Generated form TaxSchemasViewEdit
DEBUG: Generated form DocumentReferencesViewEdit
DEBUG: Generated form AddressEdit
DEBUG: Generated form ArticlesEdit
DEBUG: Generated form OrdersEdit
DEBUG: Generated form PartiesEdit
DEBUG: Forms generated...
DEBUG: Done.
```

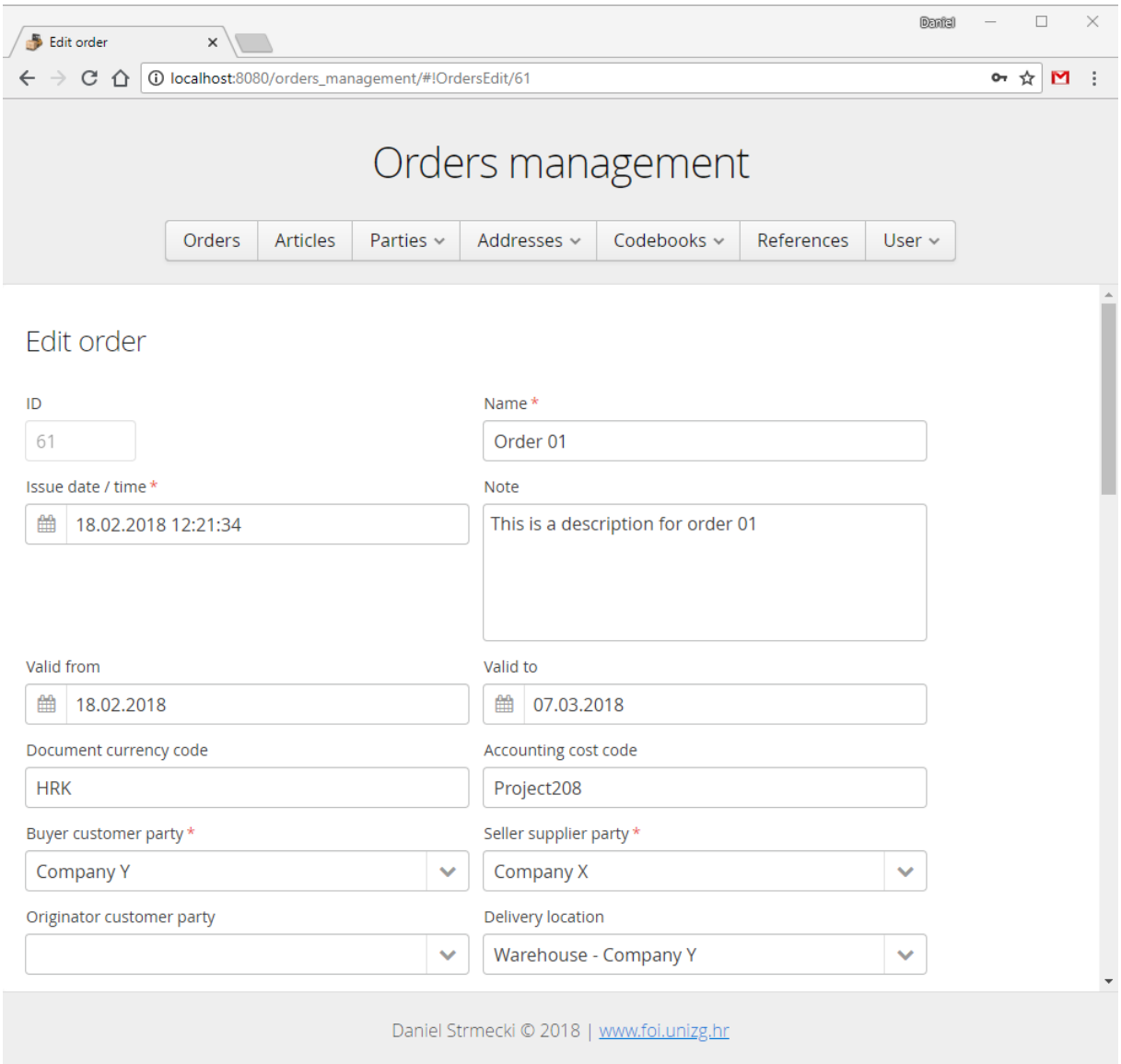
Slika 54: Ispis poruka u *Eclipse* konzoli prilikom generiranja studije slučaja 1

	Lines of Code
src/main/java/hr/foi/dstrmecki/orders/form	6.9k
AddressEdit.java	259
AddressView.java	291
ArticlesEdit.java	481
ArticlesView.java	290
ContactsViewEdit.java	301
CountriesViewEdit.java	282
DocumentReferencesViewEdit.java	312
IncotermsViewEdit.java	282
LegalEntitiesViewEdit.java	303
LoginView.java	86
OrdersEdit.java	1.1k
OrdersView.java	293
PartiesEdit.java	714
PartiesView.java	271
PartyTaxSchemeViewEdit.java	329
PersonsViewEdit.java	321
TaxSchemasViewEdit.java	282
UserAdminView.java	317
UserRightsView.java	383

Slika 55: Pregled generiranih obrazaca i broja linija koda studije slučaja 1

5.1.7. Validacija korisničkih zahtjeva

Aktivnost *(7) Validacija korisničkih zahtjeva* u sklopu ovog istraživanja obavljena je ručno, na način da je autor (u ulozi testera) nakon generiranja aplikacije provjerio svaki korisnički zahtjev zasebno, odnosno korisničke priče i njezine kriterije prihvaćanja. Na *Slici 56* prikazan je primjer korištenja, tj. testiranja aplikacije kroz korisničko sučelje u web pregledniku. Za pokretanje generirane aplikacije na računalu (serveru) potrebno je instalirati *Java* virtualnu mašinu, poslužitelj poput *Apache Tomcat*-a na koji instaliramo aplikaciju te kreirati shemu za prvu studiju slučaja na *MySQL* bazi podataka. Kompletna generirana aplikacija, odnosno njezin generirani izvorni programski kod dostupan je u *Dodatku J* ovog doktorskog rada.



Slika 56: Uređivanje narudžbenice u web pregledniku

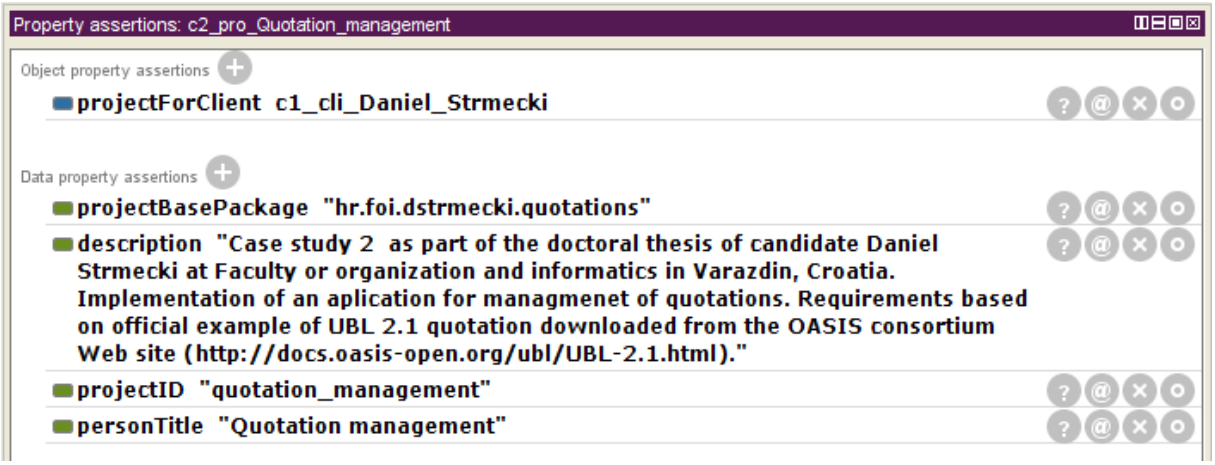
5.2. Studija slučaja 2

5.2.1. Primjer UBL ponude

Druga studija slučaja implementirati će aplikaciju za upravljanje ponudama, prema službenom primjeru preuzetom sa stranice OASIS konzorcija (dostupno u *Dodatku I* ovog rada). Iz navedene XML specifikacije izvučeni su korisnički zahtjevi, odnosno preuzeta su sva polja, tipovi podataka i veze među njima koju UBL ponuda treba sadržavati prema ovom službenom primjeru. Kao i kod prve studije slučaja, ovdje vrijede ista ograničenja. Implementacija aplikacije za upravljanje ponudama podrazumijeva kreiranje web obrazaca s odgovarajućim elementima korisničkog sučelja, unosna polja i odgovarajuće validacije te poslovnu logiku potrebnu za izračun vrijednosti. Osnovne funkcionalnosti, poput prijave / odjave korisnika i definicije korisničkih prava korisnika biti će također uključene u implementaciju. Navedene funkcionalnosti biti će implementirane primjenom dizajniranog okvira te kreiranih artefakata (ontološki model i generatori programskog koda) u sklopu ovog doktorskog rada. Za razliku od prve studije slučaja, u drugoj studiji fokus će biti stavljen na maksimalnu ponovnu iskoristivost ontološke specifikacije i funkcionalnosti dostupnih iz prve studije slučaja. Ovoga puta u fazi *(6) Razvoj novih i specifičnih funkcionalnosti* ne očekujemo nikakve veće zahvate, jer bi sve potrebno za razvoj ovog proizvoda već trebalo biti dostupno unutar modela i generatora, s obzirom da smo već razvili vrlo sličan proizvod iz iste obitelji.

5.2.2. Specifikacija projekta i korisničkih zahtjeva

Prema razvojnom ciklusu dizajniranog okvira, razvoj započinjemo aktivnošću *(1) Specifikacija projekta*. Na *Slici 57* prikazani su atributi kreirane instance projekta koji sadrže generalne podatke o drugoj studiji slučaja. Instanca projekta povezana je sa klijentom za kojeg se projekt izrađuje, dok će sama instanca projekta kasnije poslužiti za povezivanje svih ostalih instanci s odgovarajućim projektom.



Slika 57: Atributi instance projekta studije slučaja 2

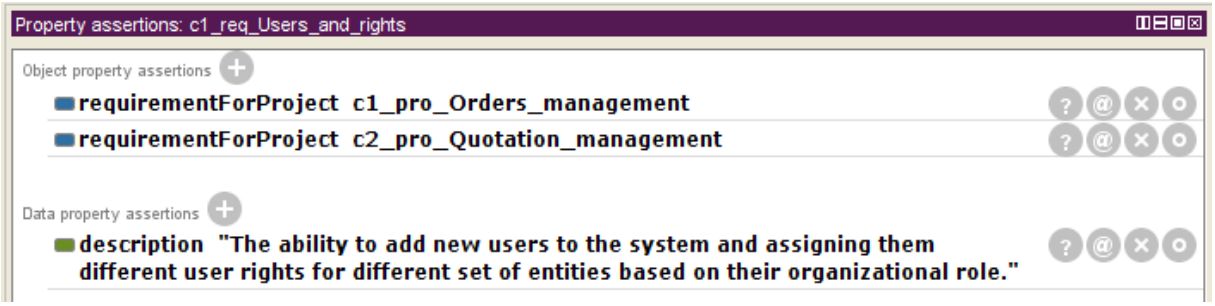
Slijedeća aktivnost po redu je *(2) Specifikacija korisničkih zahtjeva*. Analizom službenog primjera ponude (XML dokumenta iz *Dodatka I* ovog rada) identificirali smo jedanaest web obrazaca koje je potrebno razviti kako bi se omogućilo upravljanje ponudama ovog tipa putem web aplikacije. Analizom korisničkih zahtjeva utvrdili smo da je dobar dio korisničkih zahtjeva jednak onima iz prve studije slučaja, što je i logično jer se radi o proizvodima iz iste obitelji, odnosno različitim dokumentima iz UBL specifikacije. Korisnički zahtjevi za obrasce adrese, artikala, kontakata, država, pravnih osoba, stranaka, poreznih shema, poreznih shema stranka, privatnih osoba i referenci na dokumente identični su onima za prvu studiju slučaja. Razlika je samo u tome da dokument ponude neće sadržavati obrazac *Incoterms*, jer se na ovom tipu dokumenta uvjeti dostave definiraju u formi slobodnog teksta. Novi korisnički zahtjevi i obrasci specifični za drugu studiju slučaja prikazani su u *Tablici 16*. S obzirom da većinu korisničkih zahtjeva već imamo specificiranu u ontološkom modelu, nije potrebno kreirati nove instance za identične korisničke zahtjeve, već ih trebamo samo povezati sa instancom projekta druge studije slučaja pomoću objektnog atributa, kao što je prikazano na primjeru na *Slici 58*. U ovome primjeru ponovno smo iskoristili instancu korisničkog zahtjeva sa prve studije slučaja, na način

da smo postojećoj instanci samo dodali novi objektni atribut koji je povezuje sa instancom projekta druge studije slučaja. Za drugačije, odnosno nove korisničke zahtjeve, ipak moramo kreirati nove instance i povezati ih sa odgovarajućim instancama korisničkih priča i kriterija prihvaćanja. Sve instance korisničkih zahtjeva za drugu studiju slučaja moguće je pregledati tekstualno u *Dodatku G* ili digitalno u *Dodatku J* ovog doktorskog rada.

Tablica 16: Novi korisnički zahtjevi za studiju slučaja 2[26]

Naziv obrasca	Korisnički zahtjev	Napomena
Ponude	Mogućnost dodavanja novih ponuda i uređivanje ili brisanje postojećih ponuda. Ponuda treba sadržavati slijedeća polja: naziv, datum i vrijeme izdavanja, napomena, vrijedi od / do, šifra valute, stranka prodavatelja, stranka inicijatora, adresa dostave, uvjeti dostave, iznos linije, iznos sa porezom, iznos bez poreza i naplativi iznos. Naziv i stranka prodavatelja su obavezna polja. Stranka prodavatelja, stranka inicijatora i mjesto isporuke trebaju biti polja za odabir. Iznos linije, iznos sa porezom, iznos bez poreza i naplativi iznos su polja se automatski izračunavaju iz vrijednosti dodanih stavaka. Iznos sa porezom i naplativi iznos se izračunava prema formuli: ukupni iznos linije + ukupni iznos poreza. Ponude sadržavaju stavke koje trebaju sadržavati slijedeća polja: artikl, količina, mjerna jedinica, iznos poreza i iznos linije. Artikl i količina su obavezna polja za stavke. Ponude osim stavaka, mogu sadržavati i reference na više dokumenta.	Zahtjev sličan kao za studiju slučaja 1, različita su polja i poneki tipovi polja te formule za automatski izračun vrijednosti.

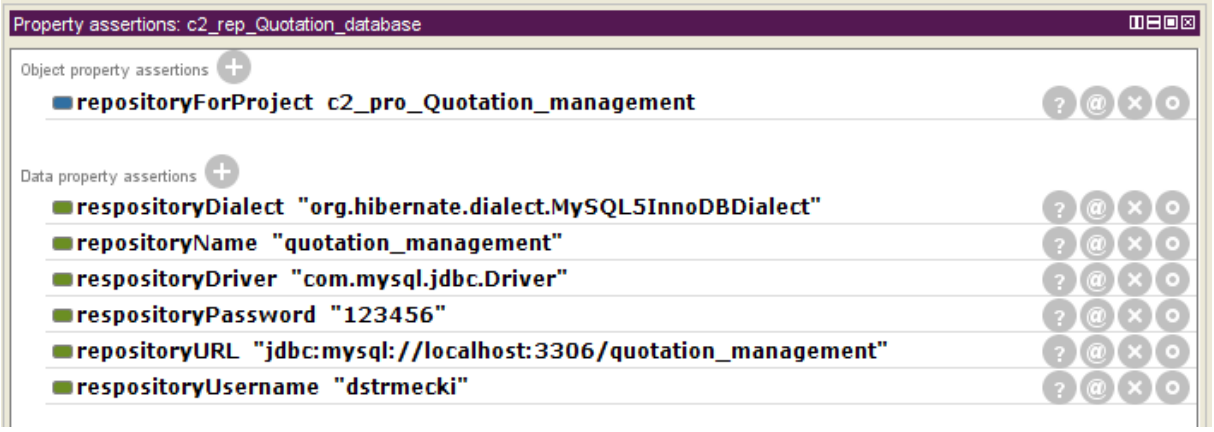
[26] Nisu uključeni korisnički zahtjev koji su identični onima za prvu studiju slučaja, prikazanim u *Tablici 12*



Slika 58: Atributi instance korisničkog zahtjeva za studiju slučaja 2

5.2.3. Specifikacija repozitorija

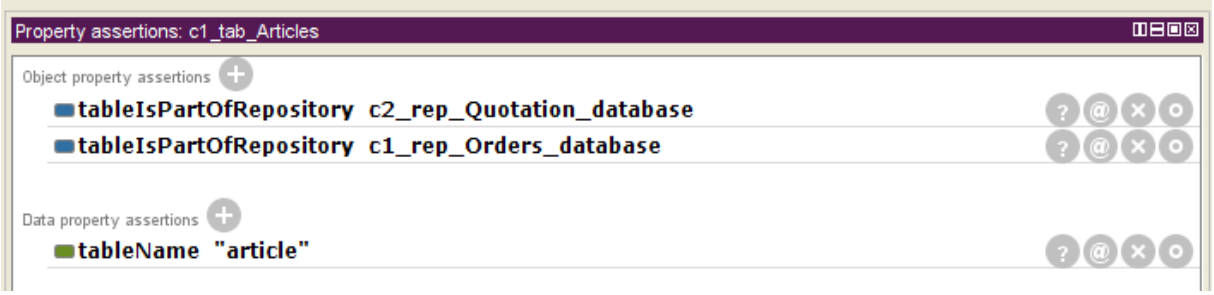
Slijedeća aktivnost, odnosno *(3) Specifikacija repozitorija*, započinje kreiranjem instance koja definira bazu podataka, a čiji atributi su prikazani na *Slici 59*. Instanca repozitorija povezana je sa instancom projekta druge studije slučaja i sadrži podatkovne atribute potrebne za spajanje aplikacije na bazu podataka. Nakon definicije repozitorira sastavljamo popis potrebnih tablica i stupaca za realizaciju ranije definiranih korisničkih zahtjeva. Tablice adrese, artikli, kontakti, kontakti - stranke, kontakti - dostava, države, reference na dokumente, stranke, stranke - pravne osobe, stranke - porezne sheme, privatne osobe, privatne osobe - stranke, porezne sheme i atributi artikala identične su kao i za prvu studiju slučaja. Nove tablice specifične za drugu studiju slučaja prikazane su u *Tablici 17*. Vidimo da se radi samo o tri nove tablice, što znači da većinu tablica možemo ponovno iskoristiti sa prve studije slučaja. To pak zapravo znači da postojeću ontološku specifikaciju tablica možemo jednostavno povezati sa odgovarajućom instancom projekta druge studije slučaja pomoću objektnog atributa, kao što je prikazano na primjeru na *Slici 60*. Za nove tablice, čiji je popis prikazan u *Tablici 17*, moramo kreirati nove instance u ontologiji, kao i za sve nove stupce. Primjer jedne nove instance stupca prikazan je na *Slici 61*. Radi se o stupcu za pohranu iznosa sa porezom, koji pohranjuje decimalne brojeve i povezan je sa odgovarajućom instancom tablice te instancom korisničkog sučelja koja definira poziciju stupca na ekranu korisnika. Čak i prilikom specifikacije novih tablica, možemo ponovno iskoristiti postojeću specifikaciju stupaca, gdje se one podudaraju sa onima definiranim u prvoj studiji slučaja. Tako na primjer, za specifikaciju tablice ponuda možemo ponovno iskoristiti sve stupce sa prve studije slučaja, osim iznosa sa porezom i iznosa bez poreza, jer se radi o novim stupcima koji nisu bili korišteni u prvoj studiji slučaja. Sve ostale instance tablica i stupaca za drugu studiju slučaja moguće je pregledati tekstualno u *Dodatku G* ili digitalno u *Dodatku J* ovog doktorskog rada.



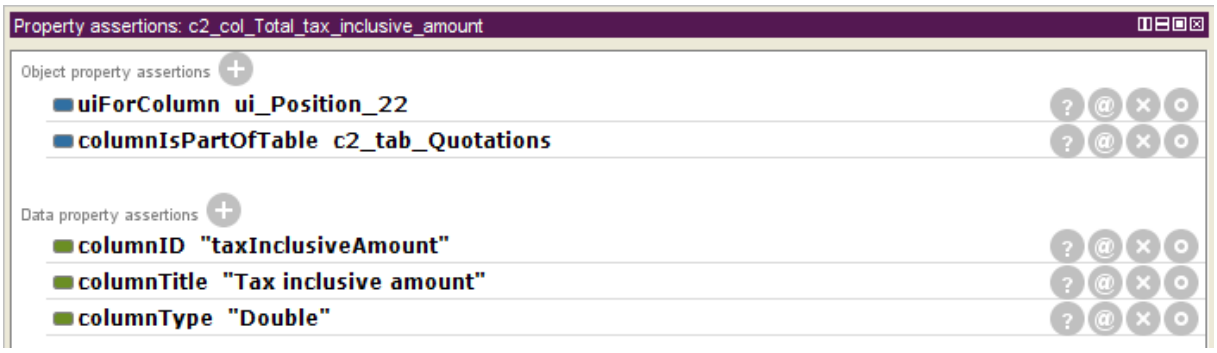
Slika 59: Atributi instance repozitorija studije slučaja 2

Tablica 17: Nove tablice u bazi podataka za realizaciju studije slučaja 2

Naziv tablice	Stupci
Reference - Ponuda	Referenca na dokument i ponuda.
Ponude	ID, naziv, datum i vrijeme izdavanja, napomena, vrijedi od datuma, vrijedi do datuma, šifra valute, stranka prodavatelja, stranka inicijatora, adresa dostave, uvjeti dostave, iznos linije, iznos sa porezom, iznos bez poreza i naplativi iznos.
Ponude - Stavke	ID, artikl, količina, mjerna jedinica, iznos poreza, iznos linije i ponuda.



Slika 60: Atributi instance tablice u bazi podataka za studiju slučaja 2



Slika 61: Atributi instance stupca u bazi podataka za studiju slučaja 2

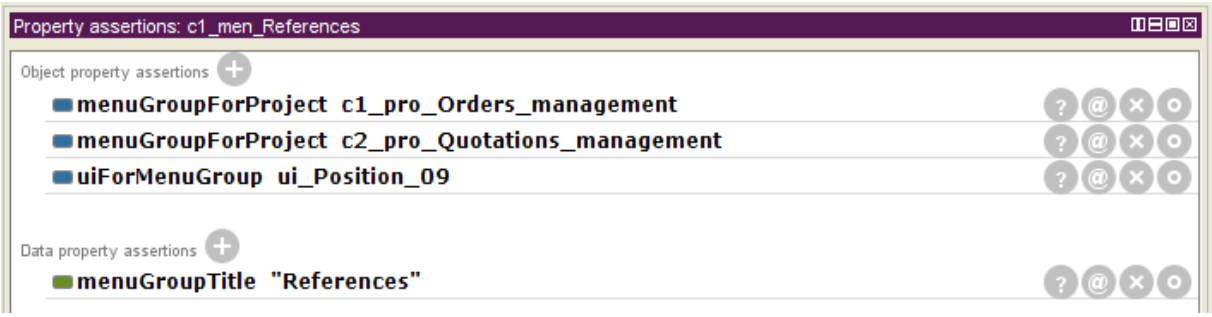
5.2.4. Specifikacija obrazaca i komponenata

Slijedeća aktivnost jest *(3) Specifikacija obrazaca*, koja podrazumijeva specifikaciju web obrazaca, ali i svih komponenata korištenih na pojedinom obrascu. Obrasci za pregled artikla, uređivanje artikala, pregled i uređivanje kontakata, pregled i uređivanje pravnih osoba, pregled i uređivanje poreznih shema, pregled i uređivanje poreznih shema stranaka, pregled i uređivanje privatnih osoba, pregled stranka, uređivanje stranaka, pregled i uređivanje država, pregled adresa, uređivanje adresa te pregled i uređivanje referenci na dokumente sadrže iste komponente kao i odgovarajući obrasci na prvoj studiji slučaja. U *Tablici 18* prikazani su obrasci specifični za drugu studiju slučaja te njihove komponente. Iz tablice vidimo da se radi o manjem broju obrazaca, što znači da značajan broj obrazaca možemo ponovno iskoristiti iz prve studije slučaja. Postojeću ontološku specifikaciju odgovarajuće grupe obrazaca ili pojedinačnog obrasca možemo jednostavno povezati sa novom instancom projekta pomoću objektnog atributa, kao što je prikazano na primjeru na *Slici 62*. Na ovaj način možemo ponovno iskoristiti identičnu specifikaciju pojedine grupe, obrasca ili komponente na više projekata. Međutim, obrasci i komponente iz *Tablice 18* ipak su različiti od onih na prvoj studiji slučaja pa je za njih potrebno kreirati nove instance u ontološkom modelu. *Slika 63* prikazuje primjer jedne nove instance obrasca za drugu studiju slučaja. Na ovom primjeru možemo vidjeti kako se radi o obrascu koji je označen kao početni, što znači da će prikazivati na početnoj stranici kod početnog učitavanja aplikacije. *Slika 64* prikazuje primjer jedne nove instance komponente za drugu studiju slučaja. Konkretno, u ovom primjeru radi se o onemogućenom unosnom polja koje služi samo za prikaz sume ukupnog iznosa linije s porezom (tj. svih stavaka ponude). Na primjeru ove instance unosnog polja vidi se primjena prilagođenog upita na bazu podataka koji se koristi za izračun vrijednosti polja. Osim već standardnih veza komponente sa instancom obrasca, instancom korisničkog sučelja i instancom stupca u bazi podataka, ova komponenta sadrži i dodatne veze na slušače, jer je vrijednost ovog polja potrebno osvježavati kod učitavanja obrasca te nakon ažuriranja vrijednosti stavaka. Sve instance obrazaca i

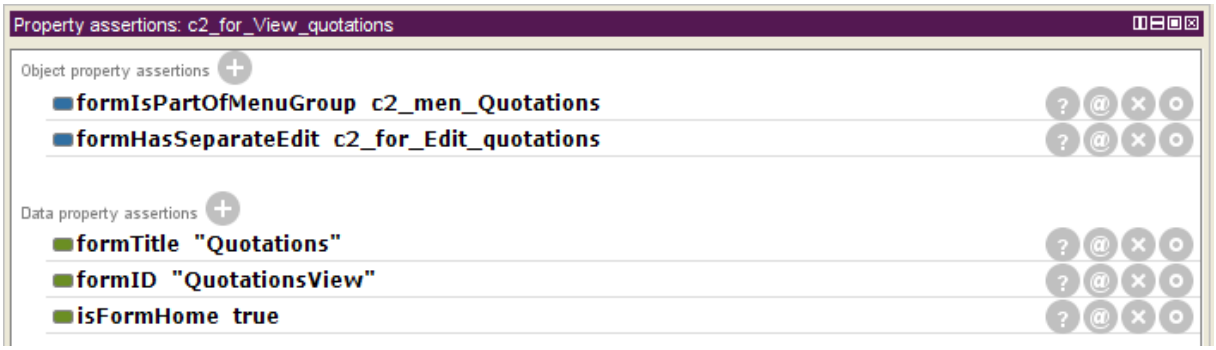
komponenata za drugu studiju slučaja moguće je pregledati tekstualno u *Dodatku G* ili digitalno u *Dodatku J* ovog doktorskog rada.

Tablica 18: Novi obrasci i komponente za studiju slučaja 2

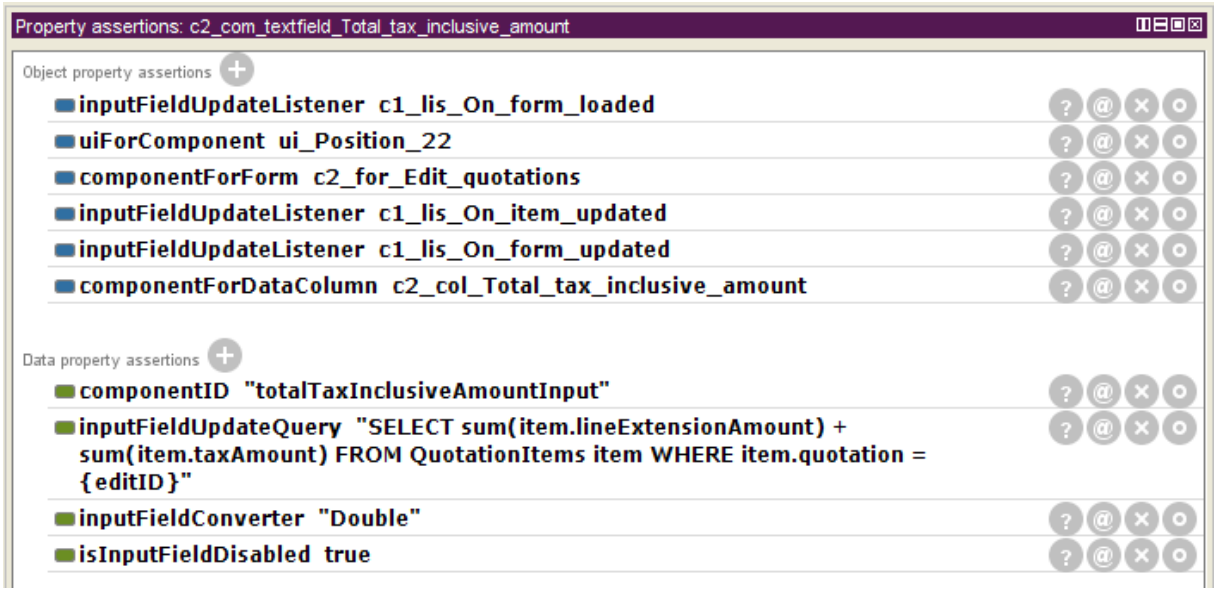
Naziv obrasca	Potrebne komponente
Pregled ponuda	Tablica s paginacijom, gumb za dodavanje novog zapisa i gumb za brisanje označenih zapisa.
Uređivanje ponuda	Tablica s paginacijom, gumb za dodavanje novog zapisa, gumb za brisanje označenih zapisa, gumb za spremanje zapisa, gumb za brisanje zapisa, tekstualno polje za unos naziva, datumsko polje za unos datuma izdavanja, tekstualno polje za unos napomene, datumsko polje za unos datuma važenja od, datumsko polje za unos datuma važenja do, tekstualno polje za unos valute dokumenta, polje za odabir stranke prodavatelja, polje za odabir stranke inicijatora, polje za odabir adrese dostave, tekstualno polje za unos uvjeta dostave, numeričko polje (dvije decimale) za prikaz ukupnog iznosa linije, numeričko polje (dvije decimale) za prikaz ukupnog iznosa sa porezom, numeričko polje (dvije decimale) za prikaz ukupnog iznosa bez poreza, numeričko polje (dvije decimale) za prikaz ukupnog naplativog iznosa, numeričko polje (cjelobrojno) za unos količine, tekstualno polje za unos mjerne jedinice, numeričko polje (dvije decimale) za unos iznosa poreza, numeričko polje (dvije decimale) za unos iznosa linije i polje za odabir artikla.



Slika 62: Atributi instance grupe obrazaca za studiju slučaja 2



Slika 63: Atributi instance obrasca za studiju slučaja 2



Slika 64: Atributi instance onemogućenog polja za studiju slučaja 2

5.2.5. Validacija modela i razvoj novih funkcionalnosti

Aktivnost *(5) Validacija modela* obavljena je pokretanjem zadanih semantičkog argumentatora *FaCT++* i testnim pokretanjem generatora programskog koda. Primjenom argumentatora pronađeno je par jednostavnih pogrešaka u specifikaciji, poput pogrešno definiranih tipova vrijednosti atributa ili nedostatka obaveznih atributa. Nakon ispravljanja grešaka pronađenih primjenom argumentatora slijedi aktivnost *(6) Razvoj novih i specifičnih funkcionalnosti*. Međutim, ovoga puta u ontološkom modelu nije bilo niti jednog atributa koji već nije podržan od strane generatora programskog koda, pa ovu aktivnost možemo jednostavno preskočiti.

5.2.6. Generiranje programskog koda

Aktivnost *(7) Generiranje programskog koda* svodi se na pokretanje generatora programskog koda klikom na gumb te praćenje ispisa poruka u konzoli, kao što je prikazano na *Slici 65*. Za drugu studiju slučaja uspješno je generirano petnaest web obrazaca. Generirani programski kod, odnosno generirane klase ovih web obrazaca prikazane su na *Slici 66*. Samo generirane klase obrazaca (bez klasa entiteta, uslužnih klasa i slično) sadrže više od 6500 linija generiranog programskog koda.

```
Markers  Properties  Servers  Data Source Explorer  Snippets  Problems  Console  Debug  History
<terminated> Application [Java Application] C:\Program Files\Java\jdk1.8.0_151\bin\javaw.exe (2. ruj 2018. 11:32:58)
DEBUG: Ontology model loaded.
DEBUG: Creating root folder...
DEBUG: Root folder created.
DEBUG: Generating project structure...
DEBUG: Project structure generated.
DEBUG: Generating utilities and exstentions...
DEBUG: Utilities and exstentions generated.
DEBUG: Generating data entities...
DEBUG: Generated table persons_join
DEBUG: Generated table party_tax_scheme
DEBUG: Generated table document_reference
DEBUG: Generated table country
DEBUG: Generated table contacts_join
DEBUG: Generated table delivery_contacts_join
DEBUG: Generated table contact
DEBUG: Generated table quotation
DEBUG: Generated table item_property
DEBUG: Generated table document_reference_join
DEBUG: Generated table tax_scheme
DEBUG: Generated table person
DEBUG: Generated table party
DEBUG: Generated table quotation_items
DEBUG: Generated table address
DEBUG: Generated table article
DEBUG: Generated table party_legal_entity
DEBUG: Data entities generated...
DEBUG: Generating forms...
DEBUG: Generated form QuotationsView
DEBUG: Generated form ArticlesView
DEBUG: Generated form ContactsViewEdit
DEBUG: Generated form LegalEntitiesViewEdit
DEBUG: Generated form PartyTaxSchemeViewEdit
DEBUG: Generated form PersonsViewEdit
DEBUG: Generated form PartiesView
DEBUG: Generated form CountriesViewEdit
DEBUG: Generated form AddressView
DEBUG: Generated form TaxSchemasViewEdit
DEBUG: Generated form DocumentReferencesViewEdit
DEBUG: Generated form AddressEdit
DEBUG: Generated form ArticlesEdit
DEBUG: Generated form PartiesEdit
DEBUG: Generated form QuotationsEdit
DEBUG: Forms generated...
DEBUG: Done.
```

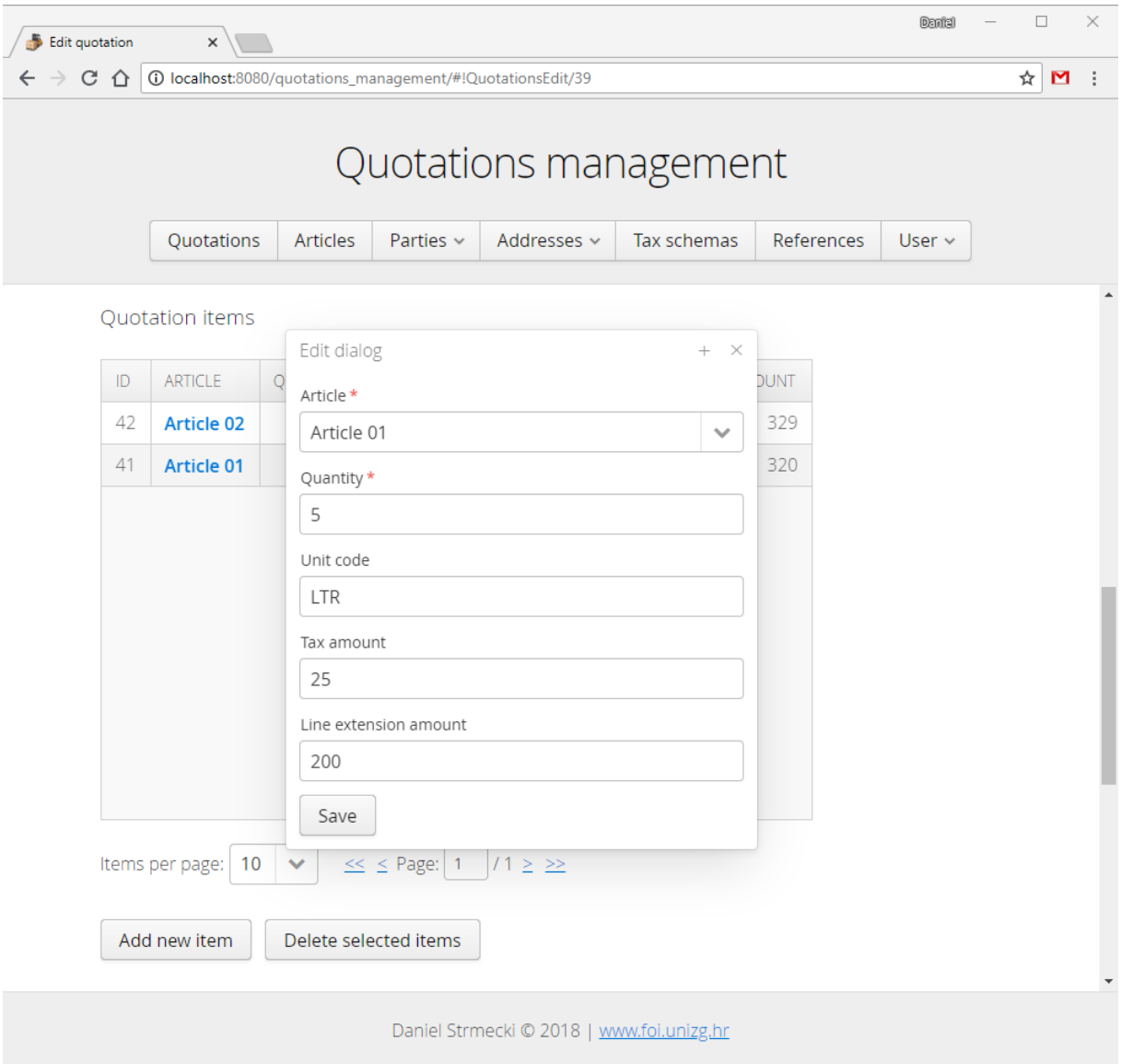
Slika 65: Ispis poruka u Eclipse konzoli prilikom generiranja studije slučaja 2

	Lines of Code
src/main/java/hr/foi/dstrmecki/quotations/form	6.5k
AddressEdit.java	270
AddressView.java	293
ArticlesEdit.java	481
ArticlesView.java	292
ContactsViewEdit.java	303
CountriesViewEdit.java	284
DocumentReferencesViewEdit.java	314
LegalEntitiesViewEdit.java	305
LoginView.java	86
PartiesEdit.java	759
PartiesView.java	273
PartyTaxSchemeViewEdit.java	331
PersonsViewEdit.java	323
QuotationsEdit.java	941
QuotationsView.java	290
TaxSchemasViewEdit.java	284
UserAdminView.java	317
UserRightsView.java	382

Slika 66: Pregled generiranih obrazaca i broja linija koda studije slučaja 2

5.2.7. Validacija korisničkih zahtjeva

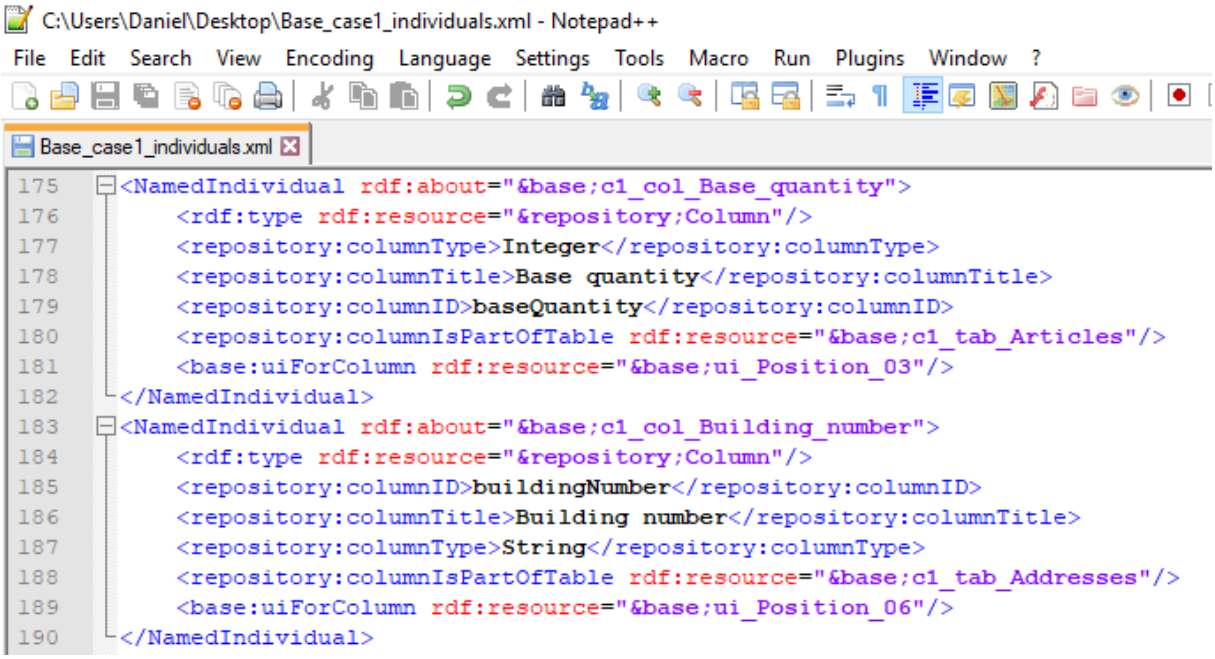
Aktivnost *(7) Validacija korisničkih zahtjeva* u sklopu ovog istraživanja obavljena je ručno, na način da je autor (u roli testera) nakon generiranja aplikacije provjerio svaki korisnički zahtjev, odnosno korisničke priče i njezine kriterije prihvaćanja. Na *Slici 67* prikazan je primjer testiranja aplikacije kroz korisničko sučelje u web pregledniku. U ovom primjeru dodaje se nova stavka na postojeći dokument ponude. Za pokretanje generirane aplikacije na računalu potrebno je instalirati *Javu* i web poslužitelj na koji instaliramo aplikaciju te kreirati shemu za drugu studiju slučaja u bazi podataka. Kompletna generirana aplikacija, odnosno njezin generirani izvorni programski kod dostupan je u *Dodatku J* ovog doktorskog rada.



Slika 67: Dodavanje nove stavke ponudi u web pregledniku

5.3. Analiza i zaključak

Dvije studije slučaja prezentirane u ovom poglavlju provedene su kako bismo potvrdili ili opovrgnuli istraživačke hipoteze H1 i H2, definirane u poglavlju *1.3.2. Ciljevi i hipoteze*. No, kako bismo mogli izvući zaključke iz provedenih studija slučaja, najprije moramo analizirati njihove rezultate. Ontološku specifikaciju možemo analizirati ručno, na način da je izvezemo u RDF / XML formatu te analiziramo korištenjem naprednijeg alata za uređivanje teksta poput *Nodepad++*[27]. S obzirom da su izvezene ontologije spremljene u dobro strukturiranom formatu, sve što trebamo napravili jest izdvojiti instance (odnosno specifikaciju) od modela te izbaciti suvišne linije (poput praznih linija i komentara) iz specifikacije. Primjer ontološke specifikacije spremne za analizu prikazan je na *Slici 68*. Valja napomenuti kako svaki atribut u ovoj datoteci, bilo da se radi o podatkovnom ili objektnom atributu, zauzima jednu liniju u XML datoteci specifikacije. Rezultati analize ontološke specifikacije prikazani su u *Tablici 19* i provjereni su usporedbom sa metrikama dostupnim u alatu *Protege*.



Slika 68: Priprema za analizu ontološke specifikacije

[27] Dostupno 29.07.2018. na https://notepad-plus-plus.org/download/v7.5.8.html

Tablica 19: Analiza ontološke specifikacije

Studija slučaja	Broj linija u specifikaciji	Broj instanci	Broj atributa	Broj podatkovnih atributa	Broj objektnih atributa
Studija slučaja 1	1964	267	1160	470	697
Studija slučaja 2[28]	1892 (306)	257 (23)	1115 (231)	459 (61)	656 (163)
Ukupno	3856	524	2275	929	1353

Kako bismo mogli usporediti specifikaciju i generirani programski kod, najprije moramo statički analizirati programski kod. Zbog velike količine programskog koda analizu nećemo raditi ručno, već ćemo upotrijebiti popularni alat za statičku analizu *Java* koda, *SonarQube*[29]. Ovaj alat se obično upotrebljava za kontinuiranu analizu novog koda te pronalazak greška i ranjivosti, no može se iskoristiti i za jednostavnu, jednokratnu statičku analizu koda kako bismo analizirali broj linija koda, broj klasa, metoda, datoteka, itd. Alat razlikuje ukupan broj linija koda, od stvarnog broja linija koje ne sadrži prazne linije, komentare i slično. Rezultati statičke analize generiranog programskog koda za svaku od studija slučaja prikazani su u *Tablici 20*. Raspored linija koda po Java paketima, prikazan ja ne *Slici 69* za prvu studiju slučaja te na *Slici 70* za drugu studiju slučaja. S obzirom da se radi o generiranom programskom kodu, određena razina duplikacije koda među klasama entiteta i obrazaca ne treba nas zabrinjavati, jer je takav rezultat očekivan s obzirom da je taj generirani kod temeljen na istom predlošku.

Tablica 20: Statička analiza generiranog programskog koda

Studija slučaja	Ukupan broj linija	Broj linija koda	Broj naredbi	Broj metoda	Broj klasa	Broj datoteka
Studija slučaja 1	12910	10204	5688	618	77	52
Studija slučaja 2	12296	9696	5424	582	74	50
Ukupno	25206	19900	11112	1200	151	102

[28] Ubrojene su i ponovno iskorištene instance te atributi s prve studije slučaja. U zagradama je naveden broj novih instanci i atributa dodanih isključivo za potrebe druge studije slučaja.
[29] Dostupno 29.07.2018. na https://www.sonarqube.org/

	Lines of Code	Duplications
orders_management		
pom.xml	199	0.0%
src/main/java/hr/foi/dstrmecki/orders	93	0.0%
src/main/java/hr/foi/dstrmecki/orders/data	2.2k	12.0%
src/main/java/hr/foi/dstrmecki/orders/database	50	0.0%
src/main/java/hr/foi/dstrmecki/orders/ext/pagination	507	3.2%
src/main/java/hr/foi/dstrmecki/orders/form	6.9k	24.7%
src/main/java/hr/foi/dstrmecki/orders/util	257	0.0%

Slika 69: Statička analiza generiranog programskog koda za studiju slučaja 1

	Lines of Code	Duplications
quotations_management		
pom.xml	199	0.0%
src/main/java/hr/foi/dstrmecki/quotations	92	0.0%
src/main/java/hr/foi/dstrmecki/quotations/data	2.1k	8.6%
src/main/java/hr/foi/dstrmecki/quotations/database	50	0.0%
src/main/java/hr/foi/dstrmecki/quotations/ext/pagination	507	3.2%
src/main/java/hr/foi/dstrmecki/quotations/form	6.5k	23.5%
src/main/java/hr/foi/dstrmecki/quotations/util	253	0.0%

Slika 70: Statička analiza generiranog programskog koda za studiju slučaja 2

Prva istraživačka hipoteza H1 fokusirana je na podizanje razine apstrakcije opisa zahtjeva i artefakata, a mjeri se u smanjenju ontološke specifikacije aplikacije u odnosu na generirani programski kod. U prvoj studiji slučaja 1964 linija ontološke specifikacija, bilo je dovoljno za generiranje 10204 linija izvršivog programskog koda, što je 5,19 puta više. U drugoj studiji slučaja 1892 linija ontološke specifikacije generiralo je 9696 linija programskog koda, što je

5,12 puta više. Čak i kada uzmemo u obzir da postoji određena razina duplikacije u generiranom programskom kodu, ona niti u jednoj studiji slučaja ne prelazi 20% te nema tako veliki utjecaj da bi promijenila zaključak provedene analize. Ako broj linija generiranog koda umanjimo za 20% i dalje je preko četiri puta više koda generirano, nego ontološki specificirano. Stoga sa sigurnošću možemo zaključiti kako je prva istraživačka hipoteza H1 potvrđena, tj. primjenom dizajniranog okvira podigli smo razinu apstrakcije pri razvoju obitelji poslovnih web aplikacija.

Druga istraživačka hipoteza H2 fokusirana je na razinu ponovne iskoristivosti ranije razvijenih i ontološki specificiranih artefakata. Za njezinu potvrdu ili odbacivanje upotrijebili ćemo najčešće korištenu metriku generativnog programiranja, postotak ponovnog iskorištenja softvera. Ova metrika definira se kao omjer dijela softvera koji je u određenom projektu ponovno iskorišten u odnosnu na ukupni softver (Vahidov, 2012). Ponovno iskorištavanje analizirati ćemo na razini ontološke specifikacije, a ne generiranog programskog koda. Analiziramo ponovno iskorištavanje ontološke specifikacije iz razloga što smo podizanjem razine apstrakcije ontologiju učinili glavnim artefaktom u razvoju. Usporediti ćemo broj linija i elemenata ontološke specifikacije koju su ponovno iskorišteni sa prve na drugoj studiji slučaja. Konkretno, od 1892 linija u ontološkoj specifikaciji druge studije slučaja, 1586 linija ponovno je iskorišteno, što iznosi 83%. Ako istu metriku primijenimo na broj instanci u ontologiji, od 257 instanci koliko ih je ukupno korišteno na drugoj studiji slučaja, 234 instance su ponovno iskorištene, što iznosi 91%. Ako istu metriku primijenimo na atribute u ontologiji, od 1115 atributa korištenih na drugoj studiji slučaja, njih 884 je ponovno iskorišteno, što iznosi 79%. Ove brojke ukazuju na vrlo visoku razinu ponovne iskoristivosti razvijenih i ontološki specificiranih artefakata sa prve studije slučaja, što potvrđuje drugu istraživačku hipotezu H2. Naravno da je visoki stupanj ponovno iskoristivosti artefakata uvjetovan time da razvijamo proizvode iz iste obitelji te da je određeni postotak korisničkih zahtjeva jednak na obje studije slučaja. Dizajnirani okvir nam osigurava da ne trebamo brinuti o duplikaciji generiranog programskog koda, jer smo podigli razinu apstrakcije pri razvoju. Visoka razina ponovne iskoristivosti ostvaruje se kroz ontološku specifikaciju koja omogućuje povezivanje artefakata sa različitih projekta dodavanjem novih veza pomoću objektnih atributa. Time se izbjegava loša praksa kopiranja programskog koda koja uvelike otežava održavanje.

6. DISKUSIJA I ZAKLJUČAK

6.1. Pregled ostvarenih ciljeva

Provedeno istraživanje fokusirano je na problem razvoja obitelji poslovnih web aplikacije te predlaže inovativno rješenje, temeljeno na ontološki potpomognutom generativnom pristupu razvoja. U ovome poglavlju pogledati ćemo unatrag na provedeno istraživanje i naglasiti rezultate pojedinih poglavlja, odnosno faza istraživanja. Poglavlje *3. Sustavni pregled literature* rezultiralo je bazom znanja koja je poslužila kao smjernica za dizajn razvojnog okvira. U ovome poglavlju identificirali smo različite načine primjene ontologija u automatskom programiranju, ali i nekoliko pristupa srodnih dizajniranom okviru. Analizom pristupa, okvira i prototipa prikupljenih pregledom literature prikupili smo znanje o trenutnom stanju tehnike te identificirali prednosti i nedostatke na koje treba obratiti pozornost prilikom razvoja modela i generatora, ali i dizajna samog razvojnog okvira. U poglavlju *4. Dizajn i razvoj* razvili smo inovativne artefakte: razvojni okvir te ontološki model i generatore programskog koda. Ontološki model razvijen je prema odabranoj metodologiji razvoja ontologija, a generatori programskog koda prema najboljoj praksi temeljenoj na iskustvu autora u razvoju web aplikacija. Nakon što smo razvili potrebne artefakte, dizajnirali smo razvojni okvir koji se sastoji od osam razvojnih faza ili aktivnosti. Način primjene dizajniranog okvira prikazan je pomoću nekoliko dijagrama i na primjeru specifikacije jednostavnog web okvira. U sklopu ovog poglavlja detaljnije smo objasnili način razvoja novih funkcionalnosti, jer je upravo ta mogućnost jedna od glavnih prednosti ovog okvira u usporedbi sa konkurentima. U poglavlju *5. Demonstracija i evaluacija* primijenili smo dizajnirani razvojni okvir za razvoj dvije poslovne aplikacije iz iste obitelji. Na obje studije slučaja prošli smo kroz osam razvojnih aktivnosti okvira i uspješno specificirali te generirali obje aplikacije. Prva studija slučaja demonstrirala je specifikaciju proizvoda od nule te razvoj specifičnih funkcionalnosti. Analizom ontološke specifikacije i generiranog programskog koda zaključili smo da je zadovoljen prvi cilj istraživanja C1, tj. da je dizajnirani okvir uspješno podigao razinu apstrakcije pri razvoju obitelji poslovnih web aplikacija. Druga studija slučaja demonstrirala je specifikaciju druge varijante proizvoda iz iste obitelji, gdje je ponovno iskorišten velik dio postojeće ontološke specifikacije sa prve studije slučaja. Analizom stupnja ponovne

iskoristivosti ontološke specifikacije zaključili smo da je zadovoljen i drugi cilj istraživanja C2, tj. da je primjenom okvira postignut visoki stupanj ponovne iskoristivosti artefakata korištenih na ranijim projektima.

6.2. Ograničenja istraživanja

U provedenom istraživanju može se identificirati nekoliko ograničenja. Pregled literature, koji je proveden u fazi *(3) Formiranje baze znanja*, obavljen je od strane jednog istraživača, odnosno autora ovog doktorskog rada. Sustavni pregled literature definiran je na način da ga izvodi tim istraživača te je stoga proces uklanjanja pristranosti u ovom slučaju kompleksan i dugotrajan. Kako bismo umanjili utjecaj ovog ograničenja definirali smo vrlo uska istraživačka pitanja, koja su se usredotočila samo na nužne teme i strogo smo slijedili preporuke za obavljanje sustavnih pregleda literature u programskom inženjerstvu. Osim toga, ovdje vrijedi istaknuti i bitnu ulogu mentora u smanjenju pristranosti, koji je kontrolirao rezultate u svakoj od provedenih iteracija. U fazi *(4) Dizajn i razvoj* također možemo identificirati nekoliko ograničenja. Osnovno ograničenje dizajniranog okvira jest da je primjenjiv samo za specifičan tip aplikacije, tj. da nije opće primjene za sve poslovne aplikacije. Prije primjene okvira potrebno je razmotriti kakav tip aplikacija će se razvijati te uzeti u obzir broj projekata koji će se razvijati u budućnosti i broj artefakata koji će se moći ponovno iskoristiti. Dizajnirani okvir primjenjiv je za razvoj obitelji poslovnih web aplikacije temeljenih na web obrascima koji pohranjuju podatke u relacijsku bazu podataka. Odluka za primjenu dizajniranog okvira treba temeljiti na dugoročnoj analizi isplativosti. U fazama *(5) Demonstracija* i *(6) Evaluacija* ograničenje predstavlja broj provedenih studija slučaja te uloga autora, koji je ujedno razvio i evaluirao okvir. Broj provedenih studija slučaja u sklopu ovog doktorska rada morao je biti ograničen, kako bi istraživanje bilo provedivo u razumnom vremenskom roku od strane samo jednog istraživača. Subjektivni utjecaj istraživača, kao jednog od glavnih ograničenja provedenog istraživanja, umanjili smo na način da smo primijenili standardnu specifikaciju umjesto prikupljanja specifičnih korisničkih zahtjeva. U provedenim studijama slučaja korištena je UBL specifikacija dokumenata i njihovih atributa za razvoj odgovarajuće programske podrške. Osim toga, unaprijed smo definirali funkcionalnosti koje će biti implementirane u sklopu studija slučaja, dok su ostale funkcionalnosti ostale izvan opsega istraživanja. Uz sva identificirana ograničenja istraživanja, moramo naglasiti kako je istraživanje imalo za glavni cilj dizajnirati inovativni razvojni okvir koji se može koristiti za

rješavanje problema razvoja obitelji poslovnih web aplikacije. Na temelju provedene demonstracije i evaluacije u prethodnom poglavlju, taj je cilj u potpunosti ostvaren.

6.3. Mogućnosti za buduća istraživanja

Kao i svaki okvir ili alat za razvoj aplikacija, niti ovaj dizajniran u sklopu ovog rada nije savršen i postoje brojne mogućnosti za njegova daljnja poboljšanja. U ovome poglavlju ćemo dodatno naglasiti neke od mogućnost za poboljšanje okvira, odnosno potencijalne teme za buduća istraživanja. Ontološki model i generatore programskog koda moguće je proširiti na način da podržavaju dodatne funkcionalnosti. U ovome doktorskom radu razvijene su osnovne funkcionalnosti poput generiranja web obrazaca za uređivanje zapisa u bazi podataka, prijave i odjave korisnika, dodjeljivanja korisničkih prava te specifične funkcionalnosti potrebne za generiranje studija slučaja. Neke od dodatnih funkcionalnosti klasičnih poslovnih web aplikacija s kojima bismo mogli obogatiti kreirane artefakte su mogućnost upravljanje radnim tijekom ili mogućnost upravljanja datotekama. Slijedeća mogućnost za poboljšanje jest proširivanje obuhvata prilagođenog argumentatora. On u trenutnoj verziji provjerava samo da je za svaki tip instance definiran skup obaveznih atributa. Prilagođeni argumentator mogao bi biti proširen na način da provjerava kompletnost cjelokupnog ontološkog modela, što bi pružilo jednu značajno višu razinu zaključivanja. Na primjer, u argumentator se može dodati provjera da je za svaku tablicu u bazi podataka definiran minimalan broj stupaca, ili da je za svaki obrazac definiran osnovni set komponenata. Prilagođeni argumentator koji bi pokrio cijeli ontološki model, tj. provjerio cijelu ontološku specifikaciju prije generiranja programskog koda uvelike bi pridonio ranijem otkrivanju grešaka. U trenutnoj verziji ovakve greške se otkrivaju tek prilikom generiranja programskog koda i često zahtijevaju intervenciju programera. Još jedna identificirana mogućnost za poboljšanje jest automatizacija validacije, odnosno testiranja generiranih aplikacija. Trenutno se u dizajniranom okviru validacija korisničkih zahtjeva obavlja ručno. S obzirom da su korisnički zahtjevi zapisani u ontologiji i da se na temelju njih specificiraju drugi artefakti (poput tablica, obrazaca i komponenata), moguće je automatizirati proces testiranja aplikacije primjenom nekog od okvira za testiranje korisničkog sučelja (npr. *Selenium*). U ovom kratkom pregledu mogućnosti za buduća istraživanja predstavili smo one najznačajnije, tj. one za koje smatramo da bi značajno utjecale na poboljšanje kvalitete dizajniranih i razvijenih artefakata.

6.4. Komunikacija

Primijenjena metodologija istraživanja temeljenog na dizajnu ne završava sa evaluacijom kreiranih artefakta, već sadrži još jednu dodatnu aktivnost na samom kraju istraživanja. Ta aktivnost je *(7) Komunikacija* i podrazumijeva iznošenje problema, njegove važnosti, kreiranih artefakta, njihovog dizajna te učinkovitosti drugim istraživačima i stručnjacima u području od interesa. Prema korištenoj DSRM metodologiji objava cijelog istraživanja ili njegovih dijelova u znanstvenim istraživačkim publikacijama prikladan je način komunikacije rezultata. U skladu sa metodologijom, prilikom pripreme i provedbe ovog istraživanja objavljeno je nekoliko znanstveno istraživačkih radova na engleskom jeziku (u znanstvenim časopisima ili prezentirano na znanstvenim konferencijama):

- *Web Form Generators Design Model*, CECIIS 2015 (Strmečki et al., 2015)
- *An Overview on the use of Ontologies in Software Engineering, Journal of Computer Science* (Strmečki et al., 2016)
- *Towards an Ontological Model for Modelling and Automatic Code Generation of Complete Web Information Systems*, CECIIS 2017 (Strmečki & Magdalenić, 2016)
- *A Systematic Literature Review on the Application of Ontologies in Automatic Programming, International Journal of Software Engineering and Knowledge Engineering* (Strmečki et al., 2018)

Artefakti kreirani u sklopu ovog istraživanja, kao glavni doprinos ovog doktorskog rada, do dana predaje rada nisu prezentirani u drugim znanstveno istraživačkim radovima. Međutim autor zadržava pravo na naknadnu objavu ovih artefakata te drugih dijelova istraživanja na engleskom jeziku kako bi se rezultati istraživanja bolje komunicirali široj publici.

6.5. Zaključak

Dizajnirani razvojni okvir omogućuje razvoj obitelji poslovnih web aplikacija na visokoj razini apstrakcije, jer primjenjuje generativni pristup razvoju i podiže apstrakciju primjenom ontološkog modela. Dizajnom i razvojem okvira te ontološkog modela i generatora programskog koda ostvaren je očekivani znanstveni doprinos istraživanja. Primjenom dizajniranog okvira, prilikom započinjanja novog projekta, ne kreće se od nule, već je moguće ponovno iskoristiti sve artefakte korištene na prijašnjim projektima. Osim modela i generatora programskog koda, moguće je ponovno iskoristiti i specifikaciju, na način da se identični elementi ne specificiraju više puta, već se samo dodaju nove veze među ontološkim elementima. Na ovaj način izbjegava se duplikacija izvornog programskog koda pa čak i duplikacija specifikacije. Također, ovim pristupom smanjuje se broj ponavljajućih zadataka koje programer treba obaviti na projektu i velik broj zadataka moguće je obaviti bez uključivanja programera ili samo povremenog uključivanja programera u roli konzultanta. Kao i kod većine drugih alata ili okvira u programskom inženjerstvu, i ovaj dizajniran u sklopu ovog doktorskog rada ima svoje područje primjene. Stoga je iznimno važno dobro proučiti korisničke zahtjeve i napraviti analizu isplativosti prije primjene okvira za razvoj konkretne obitelji proizvoda. Ukoliko se radi o web aplikacijama baziranim na obrascima i relacijskim bazama podataka sa visokim udjelom ponavljajućih artefakata među varijantama proizvoda, tada primjena dizajniranog razvojnog okvira osigurava visoku razine apstrakcije, kao i visoku razinu ponovne iskoristivosti artefakata, što na kraju rezultira i dugoročnim uštedama. Primjenom ontologija pomaknuli smo fokus razvoja dalje od izvršivog programskog koda i približili poslovnu logiku krajnjim korisnicima, kojima je ontološki model ipak značajno razumljiviji od *Java* ili *C#* programskog koda. Primjenom razvojnog okvira osigurali smo visoki stupanj ponovne iskoristivosti ontološkog znanja, koje je potrebno kako bi se inicijalna investicija u ontologije isplatila i kako bi dugoročno osigurali uštede te učinili poslovni web softver po mjeri pristupačniji malim i srednjim poduzećima.

LITERATURA

Alias, M., & Robin, C. R. R. (2014). An Ontology-Based Software Development Environment Using Upgraded Functionalities of Clojure. *Advances in Intelligent Systems and Computing*, *243*, 619–625. https://doi.org/10.1007/978-81-322-1665-0_61

Andreou, A. S., & Papatheocharous, E. (2015). Automatic Matching of Software Component Requirements using Semi-formal Specifications and a CBSE Ontology. *Evaluation of Novel Approaches to Software Engineering (ENASE)*. https://doi.org/978-989758100-7

Asikainen, T., Männistö, T., & Soininen, T. (2007). Kumbang: A domain ontology for modelling variability in software product families. *Advanced Engineering Informatics*, *21*(1), 23–40. https://doi.org/10.1016/j.aei.2006.11.007

Athanasiadis, I. N., Villa, F., & Rizzoli, A.-E. (2007). Ontologies, JavaBeans and Relational Databases for enabling semantic programming. In *31st Annual International Computer Software and Applications Conference - Vol. 2 - (COMPSAC 2007)* (Vol. 2, pp. 341–346). IEEE. https://doi.org/10.1109/COMPSAC.2007.160

Bachmann, A., Hesse, W., Russ, A., & Kop, C. (2007). OBSE–an approach to Ontology-based Software Engineering in the practice. *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures*, *1*.

Barstow, D. (1984). A Perspective on Automatic Programming. *AI Magazine*, *5*(1), 5–27. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=0A3CFD222AD7A9EBE01360D842CD1682?doi=10.1.1.368.6155&rep=rep1&type=pdf

Batory, D. (2006). A Tutorial on Feature Oriented Programming and the AHEAD Tool Suite. *Generative and Transformational Techniques in Software Engineering*, *4143*, 3–35. https://doi.org/10.1007/11877028_1

Bauer, B., & Roser, S. (2006). Semantic-enabled software engineering and development. *INFORMATIK 2006 - Informatik Fur Menschen, Beitrage Der 36. Jahrestagung Der Gesellschaft Fur Informatik e.V. (GI)*, *2*, 293–296. https://doi.org/10.1.1.92.681

Bevilacqua, L., Furno, A., di Carlo, V. S., & Zimeo, E. (2011). A tool for automatic generation of WS-BPEL compositions from OWL-S described services. In *2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings* (pp. 1–8). IEEE. https://doi.org/10.1109/SKIMA.2011.6090024

Bossche, M. Vanden, Ross, P., MacLarty, I., Van Nuffelen, B., & Pelov, N. (2007). Ontology driven software engineering for real life applications. *Third Int'l Workshop Semantic Web Enabled Software Eng*, 1–5. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.64.9906&rep=rep1&type=pdf

1. Bourque, P., & Fairley, R. E. (2014). *Guide to the Software Engineering - Body of Knowledge. IEEE Computer Society*. Retrieved from www.swebok.org

Brambilla, M., & Fraternali, P. (2014). Large-scale Model-Driven Engineering of web user interaction: The WebML and WebRatio experience. *Science of Computer Programming*, *89*, 1–35. https://doi.org/http://dx.doi.org/10.1016/j.scico.2013.03.010

Calero, C., Ruiz, F., & Piattini, M. (2006). *Ontologies for Software Engineering and Software Technology*. (C. Calero, F. Ruiz, & M. Piattini, Eds.). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-34518-3

Chongchong Zhao, Guoxin Ai, Xiao Yu, & Xiaofeng Wang. (2010). Research on automated testing framework based on ontology and multi-agent. In *2010 Third International Symposium on Knowledge Acquisition and Modeling* (pp. 206–209). IEEE. https://doi.org/10.1109/KAM.2010.5646259

Cui-Mei, B. (2009). Combining Intelligent Agent with the Semantic Web Services for Building an e-Commerce System. In *2009 IEEE International Conference on e-Business Engineering* (pp. 371–376). IEEE. https://doi.org/10.1109/ICEBE.2009.58

Czarnecki, K., & Eisenecker, U. W. (2000). *Generative programming: methods, tools, and applications*. ACM Press/Addison-Wesley Publishing Co.

Damaševičius, R., Štuikys, V., & Toldinas, J. (2008). Domain ontology-based generative component design using feature diagrams and meta-programming techniques. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 5292 LNCS, pp. 338–341). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1.1.140.4093

Dao, C., Xu, C., & Chunlai, C. (2008). Semantic and rules based upon mediator dynamic web service composition in logistics information application. *Management of Innovation and Technology, 2008. ICMIT 2008. 4th IEEE International Conference on*, 532–536. https://doi.org/10.1109/ICMIT.2008.4654421

Dillon, T. S., Chang, E., & Wongthongthain, P. (2008). Ontology-based software engineering-software engineering 2.0. *Aswec 2008: 19Th Australian Software Engineering Conference, Proceedings*, 13–23. https://doi.org/10.1109/ASWEC.2008.77

Djuric, D., & Devedzic, V. (2010). Magic potion: Incorporating new development paradigms through metaprogramming. *IEEE Software*, *27*(5), 38–44. https://doi.org/10.1109/MS.2010.90

Djuric, D., & Devedzic, V. (2012). Incorporating the Ontology Paradigm Into Software Engineering: Enhancing Domain-Driven Programming in Clojure/Java. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(1), 3–14. https://doi.org/10.1109/TSMCC.2011.2140316

Dourandish, R., Zumel, N., & Manno, M. (2007). A design pattern for automatic generation of web services from domain ontologies. In *Proceedings of the Third International Conference on Web Information Systems and Technologies* (pp. 341–348). SciTePress - Science and and Technology Publications. https://doi.org/10.5220/0001265703410348

Duran-Limon, H. A., Garcia-Rios, C. A., Castillo-Barrera, F. E., & Capilla, R. (2015). An Ontology-Based Product Architecture Derivation Approach. *IEEE Transactions on Software Engineering*, *41*(12), 1153–1168. https://doi.org/10.1109/TSE.2015.2449854

García Coria, J. A., Castellanos-Garzón, J. A., & Corchado, J. M. (2014). Intelligent business

processes composition based on multi-agent systems. *Expert Systems with Applications*, *41*(4), 1189–1205. https://doi.org/10.1016/j.eswa.2013.08.003

Gašević, D., Kaviani, N., & Milanović, M. (2009). Ontologies and Software Engineering. In *Handbook on Ontologies* (pp. 593–615). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92673-3_27

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly*, *37*(2), 337–355. https://doi.org/10.25300/MISQ/2013/37.2.01

Hai-tao, Z., & Qing-rui, G. (2010). A dynamic web services composition and realization on the base of semantic. In *2010 2nd International Conference on Future Computer and Communication* (pp. V2-624-V2-627). IEEE. https://doi.org/10.1109/ICFCC.2010.5497552

Happel, H., & Seedorf, S. (2006). Applications of Ontologies in Software Engineering. *In 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006)*, 1–14. https://doi.org/10.1.1.89.5733

Hevner, A., March, S., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1).

Hildisch, A., Steurer, J., & Stolle, R. (2007). HMI generation for plug-in services from semantic descriptions. *Proceedings - ICSE 2007 Workshops: Fourth International Workshop on Software Engineering for Automotive Systems, SEAS'07*. https://doi.org/10.1109/SEAS.2007.6

Insaurralde, C. C. (2013). Software Programmed by Artificial Agents toward an Autonomous Development Process for Code Generation. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 3294–3299). IEEE. https://doi.org/10.1109/SMC.2013.561

Jansen, M., & Girardi, R. (2006). GENMADEM: A Methodology for Generative Multi-agent Domain Engineering. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 4039 LNCS, pp. 399–402). https://doi.org/10.1007/11763864_32

John, S. (2010). Leveraging Traditional Software Engineering Tools to Ontology Engineering under a New Methodology. In *2010 5th International Conference on Future Information Technology* (pp. 1–5). IEEE. https://doi.org/10.1109/FUTURETECH.2010.5482657

Katasonov, A., & Palviainen, M. (2010). Towards ontology-driven development of applications for smart environments. In *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (pp. 696–701). IEEE. https://doi.org/10.1109/PERCOMW.2010.5470523

Kim, S. Il, & Kim, H. S. (2016). Ontology-based open API composition method for automatic mash-up service generation. In *2016 International Conference on Information Networking (ICOIN)* (pp. 351–356). IEEE. https://doi.org/10.1109/ICOIN.2016.7427130

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *EBSE Technical Report*.

Korthaus, A., Schwind, M., & Seedorf, S. (2007). Leveraging Semantic Web technologies for business component specification. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*(2), 130–141. https://doi.org/10.1016/j.websem.2006.11.005

Lee, Y., Li, W., Tsai, W., Son, Y., & Moon, K. (2009). A code generation and execution environment for service-oriented smart home solutions. In *2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)* (Vol. 0, pp. 1–8). IEEE. https://doi.org/10.1109/SOCA.2009.5410277

Li, F. (2009). An Algorithm Concerned Dynamic Composition of Web Services Based on Ontology. In *2009 International Conference on Information Engineering and Computer Science* (pp. 1–4). IEEE. https://doi.org/10.1109/ICIECS.2009.5363604

Li, G., Hai, H., & Hu, Z. (2008). A Flexible Framework for Semi-automatic Web Services Composition. In *2008 IEEE Asia-Pacific Services Computing Conference* (pp. 1258–1262). IEEE. https://doi.org/10.1109/APSCC.2008.152

Li, H., Chen, F., Yang, H., Guo, H., Chu, W. C.-C., & Yang, Y. (2009). An Ontology-Based Approach for GUI Testing. In *2009 33rd Annual IEEE International Computer Software and Applications Conference* (Vol. 1, pp. 632–633). IEEE. https://doi.org/10.1109/COMPSAC.2009.92

Liu, L., Wang, Z., Chen, J., & Li, L. (2008). Interface Semantic Extension for Web Component Service Matching and Composition. In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing* (Vol. 1, pp. 1–4). IEEE. https://doi.org/10.1109/WiCom.2008.2022

Magdalenić, I., Radošević, D., & Orehovački, T. (2013). Autogenerator: Generation and execution of programming code on demand. *Expert Systems with Applications*, *40*(8), 2845–2857. https://doi.org/10.1016/j.eswa.2012.12.003

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266. https://doi.org/10.1016/0167-9236(94)00041-2

March, & Storey. (2008). Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research. *MIS Quarterly*, *32*(4), 725. https://doi.org/10.2307/25148869

Martinez, J., Ziadi, T., Bissyandé, T. F., Klein, J., & Le Traon, Y. (2015). Bottom-up adoption of software product lines. In *Proceedings of the 19th International Conference on Software Product Line - SPLC '15* (pp. 101–110). New York, New York, USA: ACM Press. https://doi.org/10.1145/2791060.2791086

Moser, T., Moser, T., Dürr, G., & Biffl, S. (2010). Ontology-Based Test Case Generation For Ontology-Based Test Case Generation For Simulating Complex Production Automation Systems. *SEKE 2010 - Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering*, (June 2016). https://doi.org/1891706268;978-189170626-4

Musen, M. A. (2000). Ontology-oriented design and programming. *Knowledge Engineering and Agent Technology*, 3–16. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.6062&rep=rep1&amp

;type=pdf

Nasser, V. H. ., Du, W. ., & MacIsaac, D. . (2010). An ontology-based software test generation framework. *SEKE 2010 - Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering*, (1), 192–197. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-79952418392&partnerID=40&md5=30ee81349c35327a764d76ab20a96626

Nguyen, C. D., Perini, A., & Tonella, P. (2008). Ontology-based Test Generation for Multiagent Systems. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3* (pp. 1315–1320). Retrieved from http://dl.acm.org/citation.cfm?id=1402821.1402860

Nguyen, T., Colman, A., & Han, J. (2015). A Feature-Based Framework for Developing and Provisioning Customizable Web Services. *IEEE Transactions on Services Computing*, *1374*(c), 1–1. https://doi.org/10.1109/TSC.2015.2405546

Noy, N., & McGuinness, D. (2001). Ontology Development 101 : A Guide to Creating Your First Ontology. *Knowledge Systems Laboratory*.

Nyulas, C. I., O'Connor, M. J., Tu, S. W., Buckeridge, D. L., AOkhmatovskaia, N., & Musen, M. A. (2008). An ontology-driven framework for deploying JADE agent systems. *Proceedings - 2008 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2008*, *1*(3), 573–577. https://doi.org/10.1109/WIIAT.2008.25

Pahl, C. (2007a). An ontology for software component matching. *International Journal on Software Tools for Technology Transfer*, *9*(2), 169–178. https://doi.org/10.1007/s10009-006-0015-9

Pahl, C. (2007b). Semantic model-driven architecting of service-based software systems. *Information and Software Technology*, *49*(8), 838–850. https://doi.org/10.1016/j.infsof.2006.09.007

Palviainen, M., Kuusijarvi, J., & Ovaska, E. (2014). A semi-automatic end-user programming approach for smart space application development. *Pervasive and Mobile Computing*, *12*, 17–36. https://doi.org/10.1016/j.pmcj.2013.04.002

Parreiras, F. S., Saathoff, C., Walter, T., Franz, T., & Staab, S. (2009). APIs a gogo: Automatic Generation of Ontology APIs. In *2009 IEEE International Conference on Semantic Computing* (pp. 342–348). IEEE. https://doi.org/10.1109/ICSC.2009.90

Peffers, K., Tuunanen, T., Rothenberger, M., & Samir, C. (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–78. https://doi.org/10.2753/MIS0742-1222240302

Peffers, K., Tuure, T., Charles, G., Matti, R., Wendy, H., Ville, V., & Johanna, B. (2006). The design science research process: a model for producing and presenting information systems research. *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology*.

Poggi, A. (2011). A Java and OWL based approach for system interoperability. *Recent Researches in Computer Science - Proceedings of the 15th WSEAS International Conference on Computers, Part of the 15th WSEAS CSCC Multiconference*, 195–200. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-

82955190514&partnerID=40&md5=00b5fe733411a020513ba2fdc53cb06e

Prabhu, S. (2007). Towards Distributed Dynamic Web Service Composition. In *Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07)* (pp. 25–32). IEEE. https://doi.org/10.1109/ISADS.2007.73

Radetzki, U., & Cremers, A. B. (2006). Automatic Discovery and Composition of Services with IRIS. *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, 39. https://doi.org/10.1109/ICDEW.2006.34

Rahmani, T., Oberle, D., & Dahms, M. (2010). An adjustable transformation from OWL to Ecore. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6395 LNCS*(PART 2), 243–257. https://doi.org/10.1007/978-3-642-16129-2_18

Rich, C., & Waters, R. C. (1988). Automatic programming: myths and prospects. *Computer*, *21*(8), 40–51. https://doi.org/10.1109/2.75

Ringert, J. O., Rumpe, B., & Wortmann, A. (2014). Multi-platform generative development of component & connector systems using model and code libraries. *CEUR Workshop Proceedings*, *1281*, 26–35.

Robal, T., Ojastu, D., Kalja, A., & Jaakkola, H. (2015). Managing software engineering competences with domain ontology for customer and team profiling and training. In *2015 Portland International Conference on Management of Engineering and Technology (PICMET)* (pp. 1369–1376). IEEE. https://doi.org/10.1109/PICMET.2015.7273171

Rodrigues da Silva, A. (2015). Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, *43*, 139–155. https://doi.org/10.1016/j.cl.2015.06.001

Roser, S., & Bauer, B. (2008). Automatic Generation and Evolution of Model Transformations Using Ontology Engineering Space. In *Journal on Data Semantics XI* (Vol. 5383, pp. 32–64). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92148-6_2

Shengyang Luo, Yinglin Wang, & Jianmei Guo. (2009). Research on ontology-based usable user interface layout approach. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems* (pp. 234–238). IEEE. https://doi.org/10.1109/ICICISYS.2009.5357748

Sinha, R., Pang, C., Martinez, G. S., Kuronen, J., & Vyatkin, V. (2015). Requirements-Aided Automatic Test Case Generation for Industrial Cyber-physical Systems. *Engineering of Complex Computer Systems (ICECCS), 2015 20th International Conference on*, 198–201. https://doi.org/10.1109/ICECCS.2015.32

Solis, J., Pacheco, H., Najera, K., & Estrada, H. (2013). A MDE Framework for semi-automatic development of web applications. *MODELSWARD 2013 - Proceedings of the 1st International Conference on Model-Driven Engineering and Software Development*, 241–246. https://doi.org/10.5220/0004321302410246

Soylu, A., & De Causmaecker, P. (2009). Merging model driven and ontology driven system development approaches pervasive computing perspective. *2009 24th International Symposium on Computer and Information Sciences*, 730–735.

https://doi.org/10.1109/ISCIS.2009.5291915

Strmečki, D., & Magdalenić, I. (2016). Towards an ontological model for modelling and automatic code generation of complete Web information systems. *Proceedings of the Central European Conference on Information and Intelligent Systems*, 265–271.

Strmečki, D., Magdalenić, I., & Kermek, D. (2016). An Overview on the use of Ontologies in Software Engineering. *Journal of Computer Science*, *12*(12), 597–610. https://doi.org/10.3844/jcssp.2016.597.610

Strmečki, D., Magdalenić, I., & Radosević, D. (2018). A Systematic Literature Review on the Application of Ontologies in Automatic Programming. *International Journal of Software Engineering and Knowledge Engineering*, *28*(5), 559–591. https://doi.org/10.1142/S0218194018300014

Strmečki, D., Radošević, D., & Magdalenić, I. (2015). Web form generators design model, 255–260.

Talanov, M., Krekhov, A., & Makhmutov, A. (2010). Automating programming via concept mining, probabilistic reasoning over semantic knowledge base of SE domain. In *2010 6th Central and Eastern European Software Engineering Conference (CEE-SECR)* (pp. 30–35). IEEE. https://doi.org/10.1109/CEE-SECR.2010.5783147

Tang, L., Li, H., Qiu, B., Li, M., Wang, J., Wang, L., … Tang, S. (2006). WISE: A Prototype for Ontology Driven Development of Web Information Systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 3841 LNCS, pp. 1163–1167). https://doi.org/10.1007/11610113_126

Tichy, W. F., & Koerner, S. J. (2010). Text to software: developing tools to close the gaps in software engineering. *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research (FoSER '10)*, 379–384. https://doi.org/10.1145/1882362.1882439

Toti, D., & Rinelli, M. (2015). Semi-automatic Generation of an Object-Oriented API Framework over Semantic Repositories. In *2015 International Conference on Intelligent Networking and Collaborative Systems* (pp. 446–449). IEEE. https://doi.org/10.1109/INCoS.2015.22

Vahidov, R. (2012). *Design-Type Research in Information Systems*. IGI Global. https://doi.org/10.4018/978-1-4666-0131-4

van Ruijven, L. C. (2013). Ontology for Systems Engineering. *Procedia Computer Science*, *16*, 383–392. https://doi.org/10.1016/j.procs.2013.01.040

Visser, E. (2008). WebDSL: A Case Study in Domain-Specific Language Engineering. *Generative and Transformational Techniques in Software Engineering II*, *5235*, 291–373. https://doi.org/10.1007/978-3-540-88643-3_7

Wang, Y., Bai, X., Li, J., & Huang, R. (2007). Ontology-Based Test Case Generation for Testing Web Services. *Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07)*, 43–50. https://doi.org/10.1109/ISADS.2007.54

Wiebe, A. J., & Chan, C. W. (2012). Ontology driven software engineering. In *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)* (pp. 1–

4). IEEE. https://doi.org/10.1109/CCECE.2012.6334938

Wongthongtham, P., Chang, E., & Dillon, T. (2007). Ontology modelling notations for software engineering knowledge representation. *Proceedings of the 2007 Inaugural IEEE-IES Digital EcoSystems and Technologies Conference, DEST 2007*, 339–345. https://doi.org/10.1109/DEST.2007.371995

Wongthongtham, P., Chang, E., Dillon, T., & Sommerville, I. (2009). Development of a software engineering ontology for multisite software development. *IEEE Transactions on Knowledge and Data Engineering*, *21*(8), 1205–1217. https://doi.org/10.1109/TKDE.2008.209

Wongthongtham, P., Kasisopha, N., Chang, E., & Dillon, T. (2008). A software engineering ontology as software engineering knowledge representation. *Proceedings - 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008*, *2*, 668–675. https://doi.org/10.1109/ICCIT.2008.301

Xiao, H., Zou, Y., Tang, R., Ng, J., & Nigul, L. (2009). An automatic approach for ontology-driven service composition. In *2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA)* (pp. 1–8). IEEE. https://doi.org/10.1109/SOCA.2009.5410470

Xu, J., Lee, Y. H., Tsai, W. T., Li, W., Son, Y. S., Park, J. H., & Moon, K. D. (2009). Ontology-based smart home solution and service composition. *Proceedings - 2009 International Conference on Embedded Software and Systems, ICESS 2009*, 297–304. https://doi.org/10.1109/ICESS.2009.60

Zimmer, C., & Rauschmayer, A. (2004). Tuna : Ontology-Based Source Code Navigation and Annotation. *Workshop on Ontologies as Software Engineering Artifacts*, 1–9.

Zuniga, J. C., Pérez-Alcázar, J. J., & Digiampietri, L. (2010). Implementation Issues for Automatic Composition of Web Services. *Database and Expert Systems Applications (DEXA), 2010 Workshop on*, 201–205. https://doi.org/10.1109/DEXA.2010.54

DODATAK A – POPIS ODABRANIH RADOVA ZA SLR

1) Y. Lee, W. Li, W. Tsai, Y. Son, and K. Moon, "A code generation and execution environment for service-oriented smart home solutions," in 2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), 2009, vol. 00, no. c, pp. 1–8.
2) R. Dourandish, N. Zumel, and M. Manno, "A design pattern for automatic generation of web services from domain ontologies," in Proceedings of the Third International Conference on Web Information Systems and Technologies, 2007, pp. 341–348.
3) Z. Hai-tao and G. Qing-rui, "A dynamic web services composition and realization on the base of semantic," in 2010 2nd International Conference on Future Computer and Communication, 2010, pp. V2–624–V2–627.
4) G. Li, H. Hai, and Z. Hu, "A Flexible Framework for Semi-automatic Web Services Composition," in 2008 IEEE Asia-Pacific Services Computing Conference, 2008, pp. 1258–1262.
5) Poggi, "A Java and OWL based approach for system interoperability," Recent Res. Comput. Sci. - Proc. 15th WSEAS Int. Conf. Comput. Part 15th WSEAS CSCC Multiconference, pp. 195–200, 2011.
6) J. Solis, H. Pacheco, K. Najera, and H. Estrada, "A MDE Framework for semi-automatic development of web applications," Model. 2013 - Proc. 1st Int. Conf. Model. Eng. Softw. Dev., pp. 241–246, 2013.
7) S. Ou, N. Georgalas, M. Azmoodeh, K. Yang, and X. Sun, "A Model Driven Integration Architecture for Ontology-Based Context Modelling and Context-Aware Application Development," in Human-Computer Interaction, 2006, pp. 188–197.
8) M. Palviainen, J. Kuusijarvi, and E. Ovaska, "A semi-automatic end-user programming approach for smart space application development," Pervasive Mob. Comput., vol. 12, pp. 17–36, 2014.
9) L. Bevilacqua, A. Furno, V. S. di Carlo, and E. Zimeo, "A tool for automatic generation of WS-BPEL compositions from OWL-S described services," in 2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings, 2011, no. 1, pp. 1–8.
10) T. Rahmani, D. Oberle, and M. Dahms, "An adjustable transformation from OWL to Ecore," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6395 LNCS, no. PART 2, pp. 243–257, 2010.
11) F. Li, "An Algorithm Concerned Dynamic Composition of Web Services Based on Ontology," in 2009 International Conference on Information Engineering and Computer Science, 2009, pp. 1–4.
12) L. Han, Z. Xu, and Q. Yao, "An Approach to Web Service Composition Based on Service-Ontology," Fuzzy Syst. Knowl. Discov. 2009. FSKD '09. Sixth Int. Conf., vol. 2, pp. 173–177, 2009.
13) H. Xiao, Y. Zou, R. Tang, J. Ng, and L. Nigul, "An automatic approach for ontology-driven service composition," in 2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), 2009, pp. 1–8.
14) H. Li, F. Chen, H. Yang, H. Guo, W. C.-C. Chu, and Y. Yang, "An Ontology-Based Approach for GUI Testing," in 2009 33rd Annual IEEE International Computer Software and Applications Conference, 2009, vol. 1, pp. 632–633.
15) H. A. Duran-Limon, C. A. Garcia-Rios, F. E. Castillo-Barrera, and R. Capilla, "An Ontology-Based Product Architecture Derivation Approach," IEEE Trans. Softw. Eng., vol. 41, no. 12, pp. 1153–1168, Dec. 2015.

16) V. H. . Nasser, W. . Du, and D. . MacIsaac, “An ontology-based software test generation framework,” SEKE 2010 - Proc. 22nd Int. Conf. Softw. Eng. Knowl. Eng., no. 1, pp. 192–197, 2010.
17) C. I. Nyulas, M. J. O’Connor, S. W. Tu, D. L. Buckeridge, N. AOkhmatovskaia, and M. A. Musen, “An ontology-driven framework for deploying JADE agent systems,” Proc. - 2008 IEEE/WIC/ACM Int. Conf. Intell. Agent Technol. IAT 2008, vol. 1, no. 3, pp. 573–577, 2008.
18) F. S. Parreiras, C. Saathoff, T. Walter, T. Franz, and S. Staab, “APIs a gogo: Automatic Generation of Ontology APIs,” in 2009 IEEE International Conference on Semantic Computing, 2009, pp. 342–348.
19) H. Happel and S. Seedorf, “Applications of Ontologies in Software Engineering,” 2nd Int. Work. Semant. Web Enabled Softw. Eng. (SWESE 2006), pp. 1–14, 2006.
20) U. Radetzki and A. B. Cremers, “Automatic Discovery and Composition of Services with IRIS,” Data Eng. Work. 2006. Proceedings. 22nd Int. Conf., p. 39, 2006.
21) S. Andreou and E. Papatheocharous, “Automatic Matching of Software Component Requirements using Semi-formal Specifications and a CBSE Ontology,” Eval. Nov. Approaches to Softw. Eng., 2015.
22) M. Talanov, A. Krekhov, and A. Makhmutov, “Automating programming via concept mining, probabilistic reasoning over semantic knowledge base of SE domain,” in 2010 6th Central and Eastern European Software Engineering Conference (CEE-SECR), 2010, pp. 30–35.
23) B. Cui-Mei, “Combining Intelligent Agent with the Semantic Web Services for Building an e-Commerce System,” in 2009 IEEE International Conference on e-Business Engineering, 2009, pp. 371–376.
24) M. G. Gillespie, D. A. Stacey, and S. S. Crawford, “Designing ontology-driven system composition processes to satisfy user expectations: A case study for fish population modelling,” Commun. Comput. Inf. Sci., vol. 272 CCIS, pp. 290–304, 2013.
25) R. Damaševičius, V. Štuikys, and J. Toldinas, “Domain ontology-based generative component design using feature diagrams and meta-programming techniques,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2008, vol. 5292 LNCS, pp. 338–341.
26) G. A. Silver, O. Al-Haj Hassan, and J. A. Miller, “From domain ontologies to modeling ontologies to executable simulation models,” Proc. - Winter Simul. Conf., pp. 1108–1117, 2007.
27) Hildisch, J. Steurer, and R. Stolle, “HMI generation for plug-in services from semantic descriptions,” Proc. - ICSE 2007 Work. Fourth Int. Work. Softw. Eng. Automot. Syst. SEAS’07, 2007.
28) J. C. Zuniga, J. J. Pérez-Alcázar, and L. Digiampietri, “Implementation Issues for Automatic Composition of Web Services,” Database Expert Syst. Appl. (DEXA), 2010 Work., pp. 201–205, 2010.
29) D. Djuric and V. Devedzic, “Incorporating the Ontology Paradigm Into Software Engineering: Enhancing Domain-Driven Programming in Clojure/Java,” IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev., vol. 42, no. 1, pp. 3–14, Jan. 2012.
30) J. A. García Coria, J. A. Castellanos-Garzón, and J. M. Corchado, “Intelligent business processes composition based on multi-agent systems,” Expert Syst. Appl., vol. 41, no. 4, pp. 1189–1205, Mar. 2014.
31) L. Liu, Z. Wang, J. Chen, and L. Li, “Interface Semantic Extension for Web Component Service Matching and Composition,” in 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008, vol. 1, pp. 1–4.

32) C. C. Insaurralde, “Knowledge-driven software development for Distributed Aircraft Fuel Management,” in 2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC), 2013, pp. 7A5–1–7A5–12.
33) T. Asikainen, T. Männistö, and T. Soininen, “Kumbang: A domain ontology for modelling variability in software product families,” Adv. Eng. Informatics, vol. 21, no. 1, pp. 23–40, Jan. 2007.
34) Korthaus, M. Schwind, and S. Seedorf, “Leveraging Semantic Web technologies for business component specification,” Web Semant. Sci. Serv. Agents World Wide Web, vol. 5, no. 2, pp. 130–141, Jun. 2007.
35) D. Djuric and V. Devedzic, “Magic potion: Incorporating new development paradigms through metaprogramming,” IEEE Softw., vol. 27, no. 5, pp. 38–44, 2010.
36) Soylu and P. De Causmaecker, “Merging model driven and ontology driven system development approaches pervasive computing perspective,” 2009 24th Int. Symp. Comput. Inf. Sci., pp. 730–735, 2009.
37) Bachmann, W. Hesse, A. Russ, and C. Kop, “OBSE–an approach to Ontology-based Software Engineering in the practice,” Proc. 2nd Int. Work. Enterp. Model. Inf. Syst. Archit., vol. 1, 2007.
38) N. Athanasiadis, F. Villa, and A.-E. Rizzoli, “Ontologies, JavaBeans and Relational Databases for enabling semantic programming,” in 31st Annual International Computer Software and Applications Conference - Vol. 2 - (COMPSAC 2007), 2007, vol. 2, no. July, pp. 341–346.
39) M. Vanden Bossche, P. Ross, I. MacLarty, B. Van Nuffelen, and N. Pelov, “Ontology driven software engineering for real life applications,” Third Int’l Work. Semant. Web Enabled Softw. Eng, pp. 1–5, 2007.
40) S. Kamari and M. R. Khayyambashi, “Ontology-based and automatic generation of service interfaces from several choreographies,” J. Theor. Appl. Inf. Technol., pp. 110–116, 2010.
41) S. Roser and B. Bauer, “Ontology-Based Model Transformation,” Satell. Events Model. 2005 Conf., pp. 355–356, 2006.
42) S. Il Kim and H. S. Kim, “Ontology-based open API composition method for automatic mash-up service generation,” in 2016 International Conference on Information Networking (ICOIN), 2016, pp. 351–356.
43) J. Xu, Y. H. Lee, W. T. Tsai, W. Li, Y. S. Son, J. H. Park, and K. D. Moon, “Ontology-based smart home solution and service composition,” Proc. - 2009 Int. Conf. Embed. Softw. Syst. ICESS 2009, pp. 297–304, 2009.
44) T. Moser, T. Moser, G. Dürr, and S. Biffl, “Ontology-Based Test Case Generation For Ontology-Based Test Case Generation For Simulating Complex Production Automation Systems,” SEKE 2010 - Proc. 22nd Int. Conf. Softw. Eng. Knowl. Eng., no. June 2016, 2010.
45) Y. Wang, X. Bai, J. Li, and R. Huang, “Ontology-Based Test Case Generation for Testing Web Services,” Eighth Int. Symp. Auton. Decentralized Syst., pp. 43–50, 2007.
46) D. Nguyen, A. Perini, and P. Tonella, “Ontology-based Test Generation for Multiagent Systems,” in Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3, 2008, pp. 1315–1320.
47) M. Uschold, “Ontology-driven information systems: Past, present and future,” Front. Artif. Intell. Appl., no. January, 2008.
48) R. Sinha, C. Pang, G. S. Martinez, J. Kuronen, and V. Vyatkin, “Requirements-Aided Automatic Test Case Generation for Industrial Cyber-physical Systems,” Eng. Complex Comput. Syst. (ICECCS), 2015 20th Int. Conf., pp. 198–201, 2015.

49) Chongchong Zhao, Guoxin Ai, Xiao Yu, and Xiaofeng Wang, “Research on automated testing framework based on ontology and multi-agent,” in 2010 Third International Symposium on Knowledge Acquisition and Modeling, 2010, pp. 206–209.
50) Y. Zhao, J. Dong, and T. Peng, “Ontology Classification for Semantic-Web-Based Software Engineering,” IEEE Trans. Serv. Comput., vol. 2, no. 4, pp. 303–317, Oct. 2009.
51) Shengyang Luo, Yinglin Wang, and Jianmei Guo, “Research on ontology-based usable user interface layout approach,” in 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009, pp. 234–238.
52) Dao, C. Xu, and C. Chunlai, “Semantic and rules based upon mediator dynamic web service composition in logistics information application,” Manag. Innov. Technol. 2008. ICMIT 2008. 4th IEEE Int. Conf., pp. 532–536, 2008.
53) Pahl, “Semantic model-driven architecting of service-based software systems,” Inf. Softw. Technol., vol. 49, no. 8, pp. 838–850, Aug. 2007.
54) B. Bauer and S. Roser, “Semantic-enabled software engineering and development,” Inform. 2006 - Inform. fur Menschen, Beitrage der 36. Jahrestagung der Gesellschaft fur Inform. e.V., vol. 2, pp. 293–296, 2006.
55) Toti and M. Rinelli, “Semi-automatic Generation of an Object-Oriented API Framework over Semantic Repositories,” in 2015 International Conference on Intelligent Networking and Collaborative Systems, 2015, pp. 446–449.
56) C. C. Insaurralde, “Software Programmed by Artificial Agents toward an Autonomous Development Process for Code Generation,” in 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 3294–3299.
57) W. F. Tichy and S. J. Koerner, “Text to software: developing tools to close the gaps in software engineering,” Proc. FSE/SDP Work. Futur. Softw. Eng. Res. (FoSER ’10), pp. 379–384, 2010.
58) S. Prabhu, “Towards Distributed Dynamic Web Service Composition,” in Eighth International Symposium on Autonomous Decentralized Systems (ISADS’07), 2007, pp. 25–32.
59) Katasonov and M. Palviainen, “Towards ontology-driven development of applications for smart environments,” in 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010, pp. 696–701.
60) L. Tang, H. Li, B. Qiu, M. Li, J. Wang, L. Wang, B. Zhou, D. Yang, and S. Tang, “WISE: A Prototype for Ontology Driven Development of Web Information Systems,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3841 LNCS, no. 60473072, 2006, pp. 1163–1167.
61) S. Roser and B. Bauer, “Automatic Generation and Evolution of Model Transformations Using Ontology Engineering Space,” in Journal on Data Semantics XI, vol. 5383, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 32–64.
62) M. Alias and C. R. R. Robin, “An Ontology-Based Software Development Environment Using Upgraded Functionalities of Clojure,” Adv. Intell. Syst. Comput., vol. 243, pp. 619–625, 2014.
63) M. Jansen and R. Girardi, “GENMADEM: A Methodology for Generative Multi-agent Domain Engineering,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4039 LNCS, 2006, pp. 399–402.

DODATAK B – ONTOLOGIJA KORISNIČKIH ZAHTJEVA


```xml
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.w3.org/2002/07/owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <Ontology
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements"/>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#ac
ceptanceCriteriaForStory">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#AcceptanceCriteria"/>
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#UserStory"/>
    </ObjectProperty>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#em
ployeeOfCompany">
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Company"/>
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#PrivatePerson"/>
    </ObjectProperty>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#pr
ojectForClient">
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Project"/>
    </ObjectProperty>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#re
quirementForProject">
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Project"/>
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Requirement"/>
    </ObjectProperty>
```

```
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#st
oryIsPartOfRequirement">
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Requirement"/>
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#UserStory"/>
    </ObjectProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#cl
ientAdress">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#cl
ientCopyright">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#cl
ientTelephone">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#cl
ientWebsite">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#cl
ientEmail">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#co
mpanyName">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Company"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#de
scription">
```

```
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Project"/>
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Requirement"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#pe
rsonFirstName">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#PrivatePerson"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#pe
rsonLastName">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#PrivatePerson"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#pe
rsonTitle">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#PrivatePerson"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#pr
ojectBasePackage">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Project"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#pr
ojectID">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Project"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#pr
ojectTitle">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Project"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#Ac
ceptanceCriteria">
```

```
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#UserStory"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#Cl
ient"/>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#Co
mpany">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#Pr
ivatePerson">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Client"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#Pr
oject"/>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#Re
quirement"/>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#Us
erStory">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/requirements
#Requirement"/>
    </Class>
</rdf:RDF>
```

DODATAK C – ONTOLOGIJA REPOZITORIJA PODATAKA


```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.w3.org/2002/07/owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <Ontology
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository"/>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#columnIsPartOfTable">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Column"/>
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Table"/>
    </ObjectProperty>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#tableIsPartOfRepository">
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Repository"/>
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Table"/>
    </ObjectProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#columnAllowedValues">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Column"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#columnID">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Column"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#columnJoinByID">
```

```
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#colu
mnTitle">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#colu
mnType">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#isCo
lumnJoinedManyToOne">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#isCo
lumnJoinedOneToMany">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#isCo
lumnValueAnIdentifier">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#isCo
lumnValueGenerated">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#isCo
lumnValueNotNull">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
```

```
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#isCo
lumnValueUnique">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#C
olumn"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#repo
sitoryName">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#R
epository"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#repo
sitoryURL">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#R
epository"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#resp
ositoryDialect">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#R
epository"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#resp
ositoryDriver">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#R
epository"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#resp
ositoryPassword">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#R
epository"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#resp
ositoryUsername">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#R
epository"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#tabl
eName">
```

```
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#T
able"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Colu
mn">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#T
able"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Repo
sitory"/>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#Tabl
e">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/repository#R
epository"/>
    </Class>
</rdf:RDF>
```

DODATAK D – ONTOLOGIJA WEB OKVIRA


```xml
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.w3.org/2002/07/owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <Ontology
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms"/>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#formHasSe
parateEdit">
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
    </ObjectProperty>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#formHasSe
parateView">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
    </ObjectProperty>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#formIsPar
tOfMenuGroup">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#MenuGr
oup"/>
    </ObjectProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#formID">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#formTitle
">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
```

```
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#isFormHom
e">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/
>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#isMenuGro
upHidden">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#MenuGr
oup"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#menuGroup
Title">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#MenuGr
oup"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#Form"/>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#MenuGroup
"/>
</rdf:RDF>
```

DODATAK E – ONTOLOGIJA WEB KOMPONENATA


```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.w3.org/2002/07/owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <Ontology
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components"/>
    <ObjectProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#inpu
tFieldUpdateListener">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#L
istener"/>
    </ObjectProperty>
    <rdf:Description rdf:about="&owl;topObjectProperty">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#P
agedTable"/>
    </rdf:Description>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#comp
onentID">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#C
omponent"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#comp
onentLabel">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#C
omponent"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#date
FieldAutoFill">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#D
ateField"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#date
FieldUseTime">
```

```
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#D
ateField"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#inpu
tFieldConverter">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range>
            <rdfs:Datatype>
                <oneOf>
                    <rdf:Description>
                        <rdf:type rdf:resource="&rdf;List"/>
                        <rdf:first>Double</rdf:first>
                        <rdf:rest>
                            <rdf:Description>
                                <rdf:type rdf:resource="&rdf;List"/>
                                <rdf:first>Integer</rdf:first>
                                <rdf:rest rdf:resource="&rdf;nil"/>
                            </rdf:Description>
                        </rdf:rest>
                    </rdf:Description>
                </oneOf>
            </rdfs:Datatype>
        </rdfs:range>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#inpu
tFieldRequiredLabel">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#inpu
tFieldUpdateQuery">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#inpu
tFieldValidatorMessage">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#inpu
tFieldValidatorRegeEx">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
```

```
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#inpu
tFieldWidth">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#isIn
putFieldDisabled">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#page
dTableIsMain">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#P
agedTable"/>
        <rdfs:range rdf:resource="&xsd;boolean"/>
    </DatatypeProperty>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#AddN
ewItemButton">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#B
utton"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Butt
on">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#C
omponent"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Chec
kBox">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Comb
oBox">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Comp
onent"/>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Date
Field">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
```

```
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Dele
teRecordButton">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#B
utton"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Dele
teSelectedItemsButton">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#B
utton"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Inpu
tField">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#C
omponent"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#List
ener"/>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#OnFo
rmLoaded">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#L
istener"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#OnFo
rmUpdated">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#L
istener"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#OnIt
emUpdated">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#L
istener"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Opti
onGroup">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Page
dTable">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#C
omponent"/>
    </Class>
```

```
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Save
RecordButton">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#B
utton"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Text
Area">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
    </Class>
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/components#Text
Field">
        <rdfs:subClassOf
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/components#I
nputField"/>
    </Class>
</rdf:RDF>
```

DODATAK F – ONTOLOGIJA KORISNIČKOG SUČELJA


```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.w3.org/2002/07/owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <Ontology
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/interface"/>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#backg
roundColor">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#In
terface"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#textC
olor">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#In
terface"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#uiHei
ghtInPixels">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#In
terface"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#uiPos
itionIndex">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#In
terface"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </DatatypeProperty>
    <DatatypeProperty
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#uiWid
thInPixels">
        <rdfs:domain
rdf:resource="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#In
terface"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </DatatypeProperty>
```

```
    <Class
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/interface#Inter
face"/>
</rdf:RDF>
```

DODATAK G – BAZNA ONTOLOGIJA SA INSTANCAMA PROVEDENIH STUDIJA SLUČAJA


```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY base "http://www.foi.hr/danielstrmecki/ontologies/2017/base#"
>
    <!ENTITY forms
"http://www.foi.hr/danielstrmecki/ontologies/2017/forms#" >
    <!ENTITY interface
"http://www.foi.hr/danielstrmecki/ontologies/2017/interface#" >
    <!ENTITY repository
"http://www.foi.hr/danielstrmecki/ontologies/2017/repository#" >
    <!ENTITY components
"http://www.foi.hr/danielstrmecki/ontologies/2017/components#" >
    <!ENTITY requirements
"http://www.foi.hr/danielstrmecki/ontologies/2017/requirements#" >
]>
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
     xml:base="http://www.w3.org/2002/07/owl"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

xmlns:repository="http://www.foi.hr/danielstrmecki/ontologies/2017/reposito
ry#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:base="http://www.foi.hr/danielstrmecki/ontologies/2017/base#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:components="http://www.foi.hr/danielstrmecki/ontologies/2017/componen
ts#"

xmlns:interface="http://www.foi.hr/danielstrmecki/ontologies/2017/interface
#"

xmlns:requirements="http://www.foi.hr/danielstrmecki/ontologies/2017/requir
ements#"
     xmlns:forms="http://www.foi.hr/danielstrmecki/ontologies/2017/forms#">
    <Ontology
rdf:about="http://www.foi.hr/danielstrmecki/ontologies/2017/base">
        <imports rdf:resource="http://localhost:8085/Components.owl"/>
        <imports rdf:resource="http://localhost:8085/Forms.owl"/>
        <imports rdf:resource="http://localhost:8085/Interface.owl"/>
        <imports rdf:resource="http://localhost:8085/Repository.owl"/>
        <imports rdf:resource="http://localhost:8085/Requirements.owl"/>
    </Ontology>
    <AnnotationProperty rdf:about="&base;formForUserStory"/>
    <AnnotationProperty
rdf:about="&components;componentIsChildOfComponent"/>
    <AnnotationProperty rdf:about="&components;componentIsRequired"/>
    <AnnotationProperty rdf:about="&components;componentRequiredLabel"/>
    <AnnotationProperty rdf:about="&components;dateFieldResoution"/>
    <AnnotationProperty rdf:about="&components;inputFieldIsDisabled"/>
    <AnnotationProperty rdf:about="&components;tableRowsPerPage"/>
```

```
<AnnotationProperty rdf:about="&forms;formIsHome"/>
<AnnotationProperty rdf:about="&forms;menuGroupIsHidden"/>
<AnnotationProperty rdf:about="&forms;menuGroupPositionIndex"/>
<AnnotationProperty rdf:about="&requirements;basePackage"/>
<AnnotationProperty rdf:about="&requirements;companyTitle"/>
<ObjectProperty rdf:about="&base;clientForProject">
    <rdfs:domain rdf:resource="&requirements;Client"/>
    <rdfs:range rdf:resource="&requirements;Project"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;componentForDataColumn">
    <rdfs:domain rdf:resource="&components;Component"/>
    <rdfs:range rdf:resource="&repository;Column"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;componentForForm">
    <rdfs:domain rdf:resource="&components;Component"/>
    <rdfs:range rdf:resource="&forms;Form"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;componentForFormItem">
    <rdfs:domain rdf:resource="&components;Component"/>
    <rdfs:range rdf:resource="&forms;Form"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;formForUserStory">
    <rdfs:domain rdf:resource="&forms;Form"/>
    <rdfs:range rdf:resource="&requirements;UserStory"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;formMainEditTable">
    <rdfs:domain rdf:resource="&forms;Form"/>
    <rdfs:range rdf:resource="&repository;Table"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;listenerForForm">
    <rdfs:domain rdf:resource="&components;Listener"/>
    <rdfs:range rdf:resource="&forms;Form"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;menuGroupForProject">
    <rdfs:domain rdf:resource="&forms;MenuGroup"/>
    <rdfs:range rdf:resource="&requirements;Project"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;pagedTableEditForm">
    <rdfs:domain rdf:resource="&components;PagedTable"/>
    <rdfs:range rdf:resource="&forms;Form"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;pagedTableJoinColumn">
    <rdfs:domain rdf:resource="&components;PagedTable"/>
    <rdfs:range rdf:resource="&repository;Column"/>
    <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;pagedTableMainTable">
    <rdfs:domain rdf:resource="&components;PagedTable"/>
    <rdfs:range rdf:resource="&repository;Table"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;pagedTableShowColumn">
    <rdfs:domain rdf:resource="&components;PagedTable"/>
    <rdfs:range rdf:resource="&repository;Column"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;pagedTableSubTable">
    <rdfs:domain rdf:resource="&components;PagedTable"/>
    <rdfs:range rdf:resource="&repository;Table"/>
</ObjectProperty>
<ObjectProperty rdf:about="&base;repositoryForProject">
    <rdfs:domain rdf:resource="&repository;Repository"/>
    <rdfs:range rdf:resource="&requirements;Project"/>
```

```
    </ObjectProperty>
    <ObjectProperty rdf:about="&base;uiForColumn">
        <rdfs:range rdf:resource="&interface;Interface"/>
        <rdfs:domain rdf:resource="&repository;Column"/>
    </ObjectProperty>
    <ObjectProperty rdf:about="&base;uiForComponent">
        <rdfs:domain rdf:resource="&components;Component"/>
        <rdfs:range rdf:resource="&interface;Interface"/>
        <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
    </ObjectProperty>
    <ObjectProperty rdf:about="&base;uiForMenuGroup">
        <rdfs:domain rdf:resource="&forms;MenuGroup"/>
        <rdfs:range rdf:resource="&interface;Interface"/>
    </ObjectProperty>
    <ObjectProperty rdf:about="&components;componentIsChildOfComponent">
        <rdfs:range rdf:resource="&components;Component"/>
        <rdfs:domain rdf:resource="&components;Component"/>
    </ObjectProperty>
    <rdf:Description rdf:about="&components;inputFieldUpdateListener">
        <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
    </rdf:Description>
    <rdf:Description rdf:about="&forms;formHasSeparateEdit">
        <rdfs:range rdf:resource="&forms;Form"/>
        <rdfs:domain rdf:resource="&forms;Form"/>
    </rdf:Description>
    <rdf:Description rdf:about="&forms;formIsPartOfMenuGroup">
        <rdfs:domain rdf:resource="&forms;Form"/>
        <rdfs:range rdf:resource="&forms;MenuGroup"/>
    </rdf:Description>
    <rdf:Description rdf:about="&repository;columnIsPartOfTable">
        <rdfs:domain rdf:resource="&repository;Column"/>
        <rdfs:range rdf:resource="&repository;Table"/>
    </rdf:Description>
    <rdf:Description rdf:about="&repository;tableIsPartOfRepository">
        <rdfs:range rdf:resource="&repository;Repository"/>
        <rdfs:domain rdf:resource="&repository;Table"/>
    </rdf:Description>
    <rdf:Description rdf:about="&requirements;acceptanceCriteriaForStory">
        <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
    </rdf:Description>
    <rdf:Description rdf:about="&requirements;employeeOfCompany">
        <rdfs:range rdf:resource="&requirements;Company"/>
        <rdfs:domain rdf:resource="&requirements;PrivatePerson"/>
    </rdf:Description>
    <rdf:Description rdf:about="&requirements;projectForClient">
        <rdfs:range rdf:resource="&requirements;Client"/>
        <rdfs:domain rdf:resource="&requirements;Project"/>
    </rdf:Description>
    <rdf:Description rdf:about="&requirements;requirementForProject">
        <rdfs:range rdf:resource="&requirements;Project"/>
        <rdfs:domain rdf:resource="&requirements;Requirement"/>
    </rdf:Description>
    <rdf:Description rdf:about="&requirements;storyIsPartOfRequirement">
        <rdfs:range rdf:resource="&requirements;Requirement"/>
        <rdfs:domain rdf:resource="&requirements;UserStory"/>
    </rdf:Description>
    <NamedIndividual rdf:about="&base;c1_acc_Calculation_entity">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>When entity values are updated, the auto-
calculated values like sum are updated and displayed to the
user.</requirements:description>
```

```
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Calculations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Calculation_items">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>When an item is added to an entity, or
its values are update, the auto-calculated values like sum are updated and
displayed to the user.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Calculations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Entity_add">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>New entities can be added. A info message
is shown to the user after a successful save
operation.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_add"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_sto_Item_add"/>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Reference_add"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Entity_delete">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Entity can be deleted unless its
referenced. A info message is shown to the user after a successful
deletion.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_delete"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_sto_Item_delete"/>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Reference_delete"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Entity_edit">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Entities can be edited. A info message is
shown to the user after a successful save
operation.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_edit"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_sto_Item_edit"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Entity_used">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>If entity is used as a reference, then it
cannot be deleted, a warning is shown to the
user.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_delete"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Input_mandatory">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>To add or edit an entity mandatory fields
must be provided (fields cannot be left blank). If mandatory fields are not
provided a warning is shown to the user.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_add"/>
```

```
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_edit"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Input_option">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Option groups are used to select a single
option from a list of predefined values. This values are fixed by the
system and users are not able to add new values.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_add"/>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_edit"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Input_select">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Select boxes are used to select a single
entity from a list of predefined entity values. Users are able to add new
values to the list of predefined entities.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_add"/>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Entity_edit"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Right_assign">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Administrators can assign read, create
update and delete user rights to users for different sets of
entities.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_Rights_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Right_create">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Only users with the can create user right
are allowed to create new entities of specific
types.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_permissions"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Right_delete">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Only users with the can delete user right
are allowed to delete existing enties.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_permissions"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Right_read">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Only users with the can read user right
are allowed to view data of the specific entity
table.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_permissions"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_Right_update">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Only users with the can update user right
are allowed to make changes to existing enties.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_permissions"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_acc_User_add">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Administrators can add new users. A info
message is shown to the administrator after a successful save
operation.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_User_deactivate">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>User accounts can be activated or
deactivated. Deactivated user accounts cannot be used to login on the
system in the future.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_User_delete">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description>Administator can delete existing users. A
info message is shown to the administrator after a successful
deletion.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_acc_User_edit">
        <rdf:type rdf:resource="&requirements;AcceptanceCriteria"/>
        <requirements:description></requirements:description>
        <requirements:description>Administrators can edit existing users
and change their password. A info message is shown to the administrator
after a successful save operation.</requirements:description>
        <requirements:acceptanceCriteriaForStory
rdf:resource="&base;c1_sto_User_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_cli_Daniel_Strmecki">
        <rdf:type rdf:resource="&requirements;PrivatePerson"/>
        <requirements:cllientEmail
rdf:datatype="&xsd;string">daniel.strmecki@gmail.com</requirements:cllientE
mail>
        <requirements:clientAdress>Ulica grada Koblenza 6, 42000
Varaždin</requirements:clientAdress>
        <requirements:personTitle>Mr.</requirements:personTitle>
        <requirements:personFirstName>Daniel</requirements:personFirstName>
        <requirements:clientCopyright>Daniel Strmecki ©
2018</requirements:clientCopyright>

<requirements:clientWebsite>http://www.foi.unizg.hr/</requirements:clientWe
bsite>
        <requirements:personLastName>Strmecki</requirements:personLastName>
        <requirements:employeeOfCompany
rdf:resource="&base;c1_cli_Ecx_io"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_cli_Ecx_io">
        <rdf:type rdf:resource="&requirements;Company"/>
        <requirements:cllientEmail
rdf:datatype="&xsd;string">daniel.strmecki@ecx.io</requirements:cllientEmai
l>
        <requirements:companyName>ecx.io an IBM
company</requirements:companyName>
        <requirements:clientAdress>Međumurska 28, 42000
Varaždin</requirements:clientAdress>
```

```
        <requirements:companyTitle>ecx.io an IBM
company</requirements:companyTitle>

<requirements:clientWebsite>https://www.ecx.io/</requirements:clientWebsite
>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Accounting_cost_code">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Accounting cost
code</repository:columnTitle>
        <repository:columnID>accountingCostCode</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_09"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Additional_street_name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>additionalStreetName</repository:columnID>
        <repository:columnTitle>Additional street
name</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Address">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnID>address</repository:columnID>
        <repository:columnType>Address</repository:columnType>
        <repository:columnTitle>Address</repository:columnTitle>
        <repository:columnJoinByID>address_id</repository:columnJoinByID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Parties"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_legal_entities"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_tax_schemas"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Address_identification">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Address
identification</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>adressIdentification</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Allowance_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>allowanceAmount</repository:columnID>
        <repository:columnType>Double</repository:columnType>
        <repository:columnTitle>Allowance amount</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_18"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Article">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnJoinByID>article_id</repository:columnJoinByID>
        <repository:columnType>Article</repository:columnType>
        <repository:columnTitle>Article</repository:columnTitle>
        <repository:columnID>article</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Properties"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotation_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Base_quantity">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>Integer</repository:columnType>
        <repository:columnTitle>Base quantity</repository:columnTitle>
        <repository:columnID>baseQuantity</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Building_number">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>buildingNumber</repository:columnID>
        <repository:columnTitle>Building number</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Buyer_customer_party">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnTitle>Buyer customer
party</repository:columnTitle>
        <repository:columnID>buyerCustomerParty</repository:columnID>

<repository:columnJoinByID>buyer_customer_party_id</repository:columnJoinBy
ID>
        <repository:columnType>Party</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_10"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Charge_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Charge amount</repository:columnTitle>
        <repository:columnType>Double</repository:columnType>
        <repository:columnID>chargeAmount</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_19"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_col_City_name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>City name</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>cityName</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Code">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Code</repository:columnTitle>
        <repository:columnID>code</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Countries"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Company_ID">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Company ID</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>companyID</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_legal_entities"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_tax_schemas"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Contact">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnID>contact</repository:columnID>
        <repository:columnType>Contact</repository:columnType>
        <repository:columnTitle>Contact</repository:columnTitle>
        <repository:columnJoinByID>contact_id</repository:columnJoinByID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Contacts_join"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Delivery_contacts_join"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_10"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Country">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnID>country</repository:columnID>
        <repository:columnTitle>Country</repository:columnTitle>
        <repository:columnType>Country</repository:columnType>
        <repository:columnJoinByID>country_id</repository:columnJoinByID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Country_subentity">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>countrySubentity</repository:columnID>
        <repository:columnType>String</repository:columnType>
```

```
        <repository:columnTitle>Country subentity</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Delivery_location">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnType>Address</repository:columnType>
        <repository:columnTitle>Delivery location</repository:columnTitle>

<repository:columnJoinByID>delivery_location_id</repository:columnJoinByID>
        <repository:columnID>deliveryLocation</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_13"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Delivery_party">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnType>Party</repository:columnType>
        <repository:columnID>deliveryParty</repository:columnID>

<repository:columnJoinByID>delivery_party_id</repository:columnJoinByID>
        <repository:columnTitle>Delivery party</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_16"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Delivery_terms">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>

<repository:columnJoinByID>delivery_terms_id</repository:columnJoinByID>
        <repository:columnType>Incoterms</repository:columnType>
        <repository:columnID>deliveryTerms</repository:columnID>
        <repository:columnTitle>Delivery terms</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_17"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Department">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Department</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>department</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Description">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>description</repository:columnID>
        <repository:columnTitle>Description</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
```

```
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Incoterms"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Tax_schemas"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Document_currency_code">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>documentCurrencyCode</repository:columnID>
        <repository:columnTitle>Document currency
code</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_08"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Document_reference">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:columnID>document_reference</repository:columnID>

<repository:columnJoinByID>document_reference_id</repository:columnJoinByID
>
        <repository:columnType>DocumentReference</repository:columnType>
        <repository:columnTitle>Document reference</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Document_references_join"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Document_references_join"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Document_reference_type">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:columnID>typeOfReference</repository:columnID>
        <repository:columnType>TYPE_OF_REFERENCE</repository:columnType>

<repository:columnAllowedValues>QUOTATION,ORDER,ORIGINATOR,CONTRACT,ADDITIO
NAL</repository:columnAllowedValues>
        <repository:columnTitle>Type of reference</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Document_references"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Electronic_mail">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>electronicMail</repository:columnID>
        <repository:columnTitle>Electronic mail</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Contacts"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_col_Family_name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>familyName</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Family name</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_First_name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>First name</repository:columnTitle>
        <repository:columnID>firstName</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_ID">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueAnIdentifier
rdf:datatype="&xsd;boolean">true</repository:isColumnValueAnIdentifier>
        <repository:isColumnValueGenerated
rdf:datatype="&xsd;boolean">true</repository:isColumnValueGenerated>
        <repository:columnID>id</repository:columnID>
        <repository:columnType>Long</repository:columnType>
        <repository:columnTitle>ID</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Contacts"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Countries"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Incoterms"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_legal_entities"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_tax_schemas"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Tax_schemas"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotation_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_01"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Issue_date">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>issueDate</repository:columnID>
        <repository:columnTitle>Issue date</repository:columnTitle>
        <repository:columnType>Date</repository:columnType>
```

```
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Job_title">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>jobTitle</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Job title</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Line_extension_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>lineExtensionAmount</repository:columnID>
        <repository:columnType>Double</repository:columnType>
        <repository:columnTitle>Line extension
amount</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotation_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Middle_name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Middle name</repository:columnTitle>
        <repository:columnID>middleName</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueUnique
rdf:datatype="&xsd;boolean">true</repository:isColumnValueUnique>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:columnID>name</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Name</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Contacts"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Countries"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Document_references"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Incoterms"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Parties"/>
```

```
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_legal_entities"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_tax_schemas"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Tax_schemas"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Note">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>note</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Note</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Document_references"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Order">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:columnID>order1</repository:columnID>
        <repository:columnType>Order1</repository:columnType>
        <repository:columnTitle>Order</repository:columnTitle>
        <repository:columnJoinByID>order_id</repository:columnJoinByID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Document_references_join"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Originator_customer_party">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnType>Party</repository:columnType>
        <repository:columnTitle>Originator customer
party</repository:columnTitle>
        <repository:columnID>originatorCustomerParty</repository:columnID>

<repository:columnJoinByID>originator_customer_party_id</repository:columnJ
oinByID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_12"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Partial_delivery_indicator">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Partial delivery</repository:columnTitle>
```

```
        <repository:columnID>partialDeliveryIndicator</repository:columnID>
        <repository:columnType>Boolean</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Party">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnJoinByID>party_id</repository:columnJoinByID>
        <repository:columnTitle>Party</repository:columnTitle>
        <repository:columnID>party</repository:columnID>
        <repository:columnType>Party</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Contacts_join"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Delivery_contacts_join"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons_join"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_10"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Party_identification">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Party
identification</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>partyIdentification</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Parties"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Party_legal_entity">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnTitle>Party legal entity</repository:columnTitle>
        <repository:columnID>partyLegalEntity</repository:columnID>

<repository:columnJoinByID>party_legal_entity_id</repository:columnJoinByID
>
        <repository:columnType>PartyLegalEntity</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Parties"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Party_tax_scheme">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnTitle>Party tax scheme</repository:columnTitle>
        <repository:columnType>PartyTaxScheme</repository:columnType>

<repository:columnJoinByID>party_tax_scheme_id</repository:columnJoinByID>
        <repository:columnID>partyTaxScheme</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Parties"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_col_Person">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnTitle>Person</repository:columnTitle>
        <repository:columnJoinByID>person_id</repository:columnJoinByID>
        <repository:columnType>Person</repository:columnType>
        <repository:columnID>person</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Persons_join"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_10"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Postal_zone">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Postal zone</repository:columnTitle>
        <repository:columnID>postalZone</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_09"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Postbox">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>postbox</repository:columnID>
        <repository:columnTitle>Postbox</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_08"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Price_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Price amount</repository:columnTitle>
        <repository:columnID>priceAmount</repository:columnID>
        <repository:columnType>Double</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Property">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Property</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>property</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Properties"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Quantity">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>Integer</repository:columnType>
        <repository:columnTitle>Quantity</repository:columnTitle>
        <repository:columnID>quantity</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotation_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_col_Registration_name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>registrationName</repository:columnID>
        <repository:columnTitle>Registration name</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_legal_entities"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_tax_schemas"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Requested_delivery_from">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Requested delivery
from</repository:columnTitle>
        <repository:columnType>Date</repository:columnType>
        <repository:columnID>requestedDeliveryFrom</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_18"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Requested_delivery_to">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>Date</repository:columnType>
        <repository:columnID>requestedDeliveryTo</repository:columnID>
        <repository:columnTitle>Requested delivery
to</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_19"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Seller_supplier_party">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnType>Party</repository:columnType>
        <repository:columnID>sellerSupplierParty</repository:columnID>

<repository:columnJoinByID>seller_supplier_party_id</repository:columnJoinB
yID>
        <repository:columnTitle>Seller supplier
party</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_11"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Sellers_item_identification">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>String</repository:columnType>
```

```
        <repository:columnTitle>Sellers item
identification</repository:columnTitle>

<repository:columnID>sellersItemIdentification</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Special_terms">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>

<repository:columnJoinByID>special_terms_id</repository:columnJoinByID>
        <repository:columnType>Incoterms</repository:columnType>
        <repository:columnID>specialTerms</repository:columnID>
        <repository:columnTitle>Special terms</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_18"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Standard_item_identification">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Standard item
identification</repository:columnTitle>
        <repository:columnType>String</repository:columnType>

<repository:columnID>standardItemIdentification</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Street_name">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Street name</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>streetName</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Addresses"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Tax_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>Double</repository:columnType>
        <repository:columnID>taxAmount</repository:columnID>
        <repository:columnTitle>Tax amount</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotation_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Tax_scheme">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnTitle>Tax scheme</repository:columnTitle>
        <repository:columnJoinByID>tax_scheme</repository:columnJoinByID>
        <repository:columnType>TaxScheme</repository:columnType>
        <repository:columnID>taxScheme</repository:columnID>
```

```
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Party_tax_schemas"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Telefax">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>telefax</repository:columnID>
        <repository:columnTitle>Telefax</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Contacts"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Telephone">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Telephone</repository:columnTitle>
        <repository:columnID>telephone</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Contacts"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Total_line_extension_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Line extension
amount</repository:columnTitle>
        <repository:columnType>Double</repository:columnType>
        <repository:columnID>lineExtensionAmount</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_21"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Total_payable_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Payable amount</repository:columnTitle>
        <repository:columnID>payableAmount</repository:columnID>
        <repository:columnType>Double</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_24"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Total_tax_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Tax amount</repository:columnTitle>
        <repository:columnID>taxAmount</repository:columnID>
        <repository:columnType>Double</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_22"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Unit_code">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Unit code</repository:columnTitle>
        <repository:columnType>String</repository:columnType>
        <repository:columnID>unitCode</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Articles"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Order_items"/>
```

```
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotation_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Url">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>URL</repository:columnTitle>
        <repository:columnID>url</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Document_references"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Valid_from">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Valid from</repository:columnTitle>
        <repository:columnID>validFrom</repository:columnID>
        <repository:columnType>Date</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Valid_to">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnTitle>Valid to</repository:columnTitle>
        <repository:columnID>validTo</repository:columnID>
        <repository:columnType>Date</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Orders"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_col_Value">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnID>propertyValue</repository:columnID>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Value</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c1_tab_Properties"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_button_Add_new_record">
        <rdf:type rdf:resource="&components;AddNewItemButton"/>
        <components:componentID>addNewButton</components:componentID>
        <components:componentLabel>Add new item</components:componentLabel>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_addresses"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_articles"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_orders"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_parties"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_countries"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_document_references"/>
```

```
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_incoterms"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_party_legal_entities"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_party_tax_schemas"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_tax_schemas"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:componentForForm
rdf:resource="&base;c2_for_View_quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_button_Delete_record">
        <rdf:type rdf:resource="&components;DeleteRecordButton"/>
        <components:componentLabel>Delete</components:componentLabel>
        <components:componentID>deleteRecordButton</components:componentID>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_button_Delete_selected">
        <rdf:type rdf:resource="&components;DeleteSelectedItemsButton"/>

<components:componentID>deleteSelectedButton</components:componentID>
        <components:componentLabel>Delete selected
items</components:componentLabel>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_addresses"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_articles"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_orders"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_parties"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_countries"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_incoterms"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_party_legal_entities"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_party_tax_schemas"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_tax_schemas"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:componentForForm
rdf:resource="&base;c2_for_View_quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_button_Save_record">
        <rdf:type rdf:resource="&components;SaveRecordButton"/>
```

```
        <components:componentID>saveRecordButton</components:componentID>
        <components:componentLabel>Save</components:componentLabel>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_checkbox_Partial_delivery_indicator">
        <rdf:type rdf:resource="&components;CheckBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Partial_delivery_indicator"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Address">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Address"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_party_legal_entities"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_party_tax_schemas"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Article">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <components:inputFieldRequiredLabel>Article is
required</components:inputFieldRequiredLabel>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Article"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForFormItem
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_combobox_Buyer_customer_party">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <components:inputFieldRequiredLabel>Buyer customer party is
required</components:inputFieldRequiredLabel>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Buyer_customer_party"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_09"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Country">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Country"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_09"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Delivery_location">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Delivery_location"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
```

```
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_12"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Delivery_party">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Delivery_party"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_16"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Delivery_terms">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Delivery_terms"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_17"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_combobox_Originator_customer_party">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Originator_customer_party"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_11"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Party_legal_entity">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Party_legal_entity"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Party_tax_scheme">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Party_tax_scheme"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_combobox_Seller_supplier_party">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <components:inputFieldRequiredLabel>Seller supplier party is
required</components:inputFieldRequiredLabel>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Seller_supplier_party"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_10"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_combobox_Special_terms">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Special_terms"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_18"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_com_combobox_Tax_scheme">
        <rdf:type rdf:resource="&components;ComboBox"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Tax_scheme"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_party_tax_schemas"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_datefield_Issue_date">
        <rdf:type rdf:resource="&components;DateField"/>
        <components:dateFieldUseTime
rdf:datatype="&xsd;boolean">true</components:dateFieldUseTime>
        <components:dateFieldAutoFill
rdf:datatype="&xsd;boolean">true</components:dateFieldAutoFill>
        <components:componentLabel>Issue date /
time</components:componentLabel>

<components:dateFieldResoution>second</components:dateFieldResoution>
        <components:inputFieldRequiredLabel>Issue date is
required</components:inputFieldRequiredLabel>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Issue_date"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_datefield_Requested_delivery_from">
        <rdf:type rdf:resource="&components;DateField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Requested_delivery_from"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_13"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_datefield_Requested_delivery_to">
        <rdf:type rdf:resource="&components;DateField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Requested_delivery_to"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_14"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_datefield_Valid_from">
        <rdf:type rdf:resource="&components;DateField"/>
        <components:dateFieldResoution>day</components:dateFieldResoution>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Valid_from"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_datefield_Valid_to">
        <rdf:type rdf:resource="&components;DateField"/>
        <components:dateFieldResoution>day</components:dateFieldResoution>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Valid_to"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
```

```
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_optiongroup_Document_reference_type">
        <rdf:type rdf:resource="&components;OptionGroup"/>
        <components:inputFieldRequiredLabel>Type of document reference is
required</components:inputFieldRequiredLabel>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Document_reference_type"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Addresses">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>addressesTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Address_identification"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Building_number"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_City_name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Country"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Department"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Postbox"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Street_name"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_addresses"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Addresses"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Articles">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>articlesTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Base_quantity"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Description"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Price_amount"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Sellers_item_identification"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Standard_item_identification"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Unit_code"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_articles"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Articles"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Contacts">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>contactsTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Electronic_mail"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Telefax"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Telephone"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Contacts"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Contacts_subtable">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:pagedTableIsMain
rdf:datatype="&xsd;boolean">false</components:pagedTableIsMain>
        <components:componentID>contactsSubtable</components:componentID>
        <components:componentLabel>Contacts</components:componentLabel>
        <base:pagedTableJoinColumn rdf:resource="&base;c1_col_Contact"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Contact"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:pagedTableEditForm
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:pagedTableSubTable rdf:resource="&base;c1_tab_Contacts"/>
        <base:pagedTableMainTable
rdf:resource="&base;c1_tab_Contacts_join"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Countries">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>countriesTable</components:componentID>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Code"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_countries"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Countries"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_pagedtable_Delivery_contacts_subtable">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:pagedTableIsMain
rdf:datatype="&xsd;boolean">false</components:pagedTableIsMain>
        <components:componentLabel>Dellivery
contacts</components:componentLabel>

<components:componentID>deliveryContactsSubtable</components:componentID>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Contact"/>
        <base:pagedTableJoinColumn rdf:resource="&base;c1_col_Contact"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:pagedTableEditForm
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:pagedTableSubTable rdf:resource="&base;c1_tab_Contacts"/>
        <base:pagedTableMainTable
rdf:resource="&base;c1_tab_Delivery_contacts_join"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_pagedtable_Document_references">
        <rdf:type rdf:resource="&components;PagedTable"/>

<components:componentID>documentReferencesTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Document_reference_type"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Note"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Url"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:pagedTableMainTable
rdf:resource="&base;c1_tab_Document_references"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_pagedtable_Document_references_subtable">
        <rdf:type rdf:resource="&components;PagedTable"/>
```

```
        <components:pagedTableIsMain
rdf:datatype="&xsd;boolean">false</components:pagedTableIsMain>
        <components:componentLabel>Document
references</components:componentLabel>

<components:componentID>documentReferencesSubtable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Document_reference"/>
        <base:pagedTableJoinColumn
rdf:resource="&base;c1_col_Document_reference"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:pagedTableEditForm
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:pagedTableSubTable
rdf:resource="&base;c1_tab_Document_references"/>
        <base:pagedTableMainTable
rdf:resource="&base;c1_tab_Document_references_join"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Incoterms">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>incotermsTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Description"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_incoterms"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Incoterms"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Order_items">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentLabel>Order items</components:componentLabel>
        <components:componentID>orderItemsTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Accounting_cost_code"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Article"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Line_extension_amount"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Partial_delivery_indicator"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Quantity"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Requested_delivery_from"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Requested_delivery_to"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Tax_amount"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Unit_code"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Order_items"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Orders">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>ordersTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Buyer_customer_party"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Issue_date"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Note"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Seller_supplier_party"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Total_payable_amount"/>
```

```
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Valid_from"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Valid_to"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_orders"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Orders"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Parties">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>partiesTable</components:componentID>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Address"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Party_identification"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Party_legal_entity"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Party_tax_scheme"/>
        <base:componentForForm rdf:resource="&base;c1_for_View_parties"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Parties"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_pagedtable_Party_legal_entities">
        <rdf:type rdf:resource="&components;PagedTable"/>

<components:componentID>partyLegalEntitiesTable</components:componentID>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Address"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Company_ID"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_party_legal_entities"/>
        <base:pagedTableMainTable
rdf:resource="&base;c1_tab_Party_legal_entities"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Party_tax_schemas">
        <rdf:type rdf:resource="&components;PagedTable"/>

<components:componentID>partyTaxSchemasTable</components:componentID>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Address"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Company_ID"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Tax_scheme"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_party_tax_schemas"/>
        <base:pagedTableMainTable
rdf:resource="&base;c1_tab_Party_tax_schemas"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Persons">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>personsTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Family_name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_First_name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Job_title"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Middle_name"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Persons"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Persons_subtable">
        <rdf:type rdf:resource="&components;PagedTable"/>
```

```
        <components:pagedTableIsMain
rdf:datatype="&xsd;boolean">false</components:pagedTableIsMain>
        <components:componentID>personsSubtable</components:componentID>
        <components:componentLabel>Persons</components:componentLabel>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Person"/>
        <base:pagedTableJoinColumn rdf:resource="&base;c1_col_Person"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:pagedTableEditForm
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:pagedTableSubTable rdf:resource="&base;c1_tab_Persons"/>
        <base:pagedTableMainTable
rdf:resource="&base;c1_tab_Persons_join"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Properties">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>propertiesTable</components:componentID>
        <components:componentLabel>Article
properties</components:componentLabel>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Property"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Value"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Properties"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_pagedtable_Tax_schemas">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>taxSchemasTable</components:componentID>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Description"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:componentForForm
rdf:resource="&base;c1_for_Viewedit_tax_schemas"/>
        <base:pagedTableMainTable rdf:resource="&base;c1_tab_Tax_schemas"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textarea_Description">
        <rdf:type rdf:resource="&components;TextArea"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Description"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_incoterms"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_tax_schemas"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textarea_Note">
        <rdf:type rdf:resource="&components;TextArea"/>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Note"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Accounting_cost_item">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Accounting_cost_code"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
```

```
        <base:uiForComponent rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Accounting_cost_order">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Accounting_cost_code"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_08"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Additional_street_name">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Additional_street_name"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Address_identification">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Address_identification"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Allowance_amount">
        <rdf:type rdf:resource="&components;TextField"/>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Allowance_amount"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_19"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Base_quantity">
        <rdf:type rdf:resource="&components;TextField"/>

<components:inputFieldConverter>Integer</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Base_quantity"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Building_number">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldWidth
rdf:datatype="&xsd;int">150</components:inputFieldWidth>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Building_number"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Charge_amount">
        <rdf:type rdf:resource="&components;TextField"/>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Charge_amount"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_20"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_City_name">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_City_name"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Code">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Code"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_countries"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Company_id">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Company_ID"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_party_legal_entities"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_party_tax_schemas"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Country_subentity">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Country_subentity"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Department">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Department"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Document_currency_code">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Document_currency_code"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_07"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Electronic_mail">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldValidatorRegeEx xml:lang="(([a-za-z0-9-
])+\\.)+([a-za-z0-9]{2,4})+$">^([a-zA-Z0-9_\\.\\-
+])+</components:inputFieldValidatorRegeEx>
        <components:inputFieldValidatorMessage
xml:lang="gmail.com">Incorrect email format. Example:
name_surname</components:inputFieldValidatorMessage>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Electronic_mail"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_contacts"/>
```

```
        <base:uiForComponent rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Family_name">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Family_name"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_First_name">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_First_name"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Job_title">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Job_title"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Line_extension_amount">
        <rdf:type rdf:resource="&components;TextField"/>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Line_extension_amount"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForFormItem
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Middle_name">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Middle_name"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Name">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldRequiredLabel>Name is
required</components:inputFieldRequiredLabel>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Name"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_countries"/>
```

```
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_incoterms"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_party_legal_entities"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_party_tax_schemas"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_persons"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_tax_schemas"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_01"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Party_identification">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Party_identification"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_parties"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Postal_zone">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Postal_zone"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_08"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Postbox">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldWidth
rdf:datatype="&xsd;int">150</components:inputFieldWidth>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Postbox"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_08"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Price_amount">
        <rdf:type rdf:resource="&components;TextField"/>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Price_amount"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Property">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Property"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_articles"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Quantity">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldRequiredLabel>Quantity is
required</components:inputFieldRequiredLabel>

<components:inputFieldConverter>Integer</components:inputFieldConverter>
```

```
        <base:componentForDataColumn rdf:resource="&base;c1_col_Quantity"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForFormItem
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Sellers_item_identification">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Sellers_item_identification"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Standard_item_identification">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Standard_item_identification"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_06"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Street_name">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Street_name"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_addresses"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Tax_amount">
        <rdf:type rdf:resource="&components;TextField"/>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Tax_amount"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForFormItem
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Telefax">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldValidatorRegeEx>^\\+(?:[0-9] ?){6,14}[0-
9]$</components:inputFieldValidatorRegeEx>
        <components:inputFieldValidatorMessage>Incorrect telefax number
format.  Example: +385 123456</components:inputFieldValidatorMessage>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Telefax"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Telephone">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldValidatorMessage>Incorect telephone number
format. Example: +385 123456</components:inputFieldValidatorMessage>
        <components:inputFieldValidatorRegeEx>^\\+(?:[0-9] ?){6,14}[0-
9]$</components:inputFieldValidatorRegeEx>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Telephone"/>
```

```
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_contacts"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Total_line_extension_amount">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:isInputFieldDisabled
rdf:datatype="&xsd;boolean">true</components:isInputFieldDisabled>
        <components:inputFieldIsDisabled
rdf:datatype="&xsd;boolean">true</components:inputFieldIsDisabled>
        <components:inputFieldUpdateQuery>SELECT
sum(ord8.lineExtensionAmount) FROM OrderItems ord8 WHERE ord8.order1 =
{editID}</components:inputFieldUpdateQuery>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Total_line_extension_amount"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_loaded"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_updated"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_item_updated"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_21"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_com_textfield_Total_payable_amount">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:inputFieldIsDisabled
rdf:datatype="&xsd;boolean">true</components:inputFieldIsDisabled>
        <components:isInputFieldDisabled
rdf:datatype="&xsd;boolean">true</components:isInputFieldDisabled>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <components:inputFieldUpdateQuery>SELECT sum(item.taxAmount) +
sum(item.lineExtensionAmount) + order1.allowanceAmount -
order1.chargeAmount FROM OrderItems item, Order1 order1 WHERE item.order1 =
{editID} AND order1 = item.order1</components:inputFieldUpdateQuery>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Total_payable_amount"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_loaded"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_updated"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_item_updated"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_23"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Total_tax_amount">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:isInputFieldDisabled
rdf:datatype="&xsd;boolean">true</components:isInputFieldDisabled>
        <components:inputFieldIsDisabled
rdf:datatype="&xsd;boolean">true</components:inputFieldIsDisabled>

<components:inputFieldConverter>Double</components:inputFieldConverter>
```

```
        <components:inputFieldUpdateQuery>SELECT sum(ord8.taxAmount) FROM
OrderItems ord8 WHERE ord8.order1 =
{editID}</components:inputFieldUpdateQuery>

<components:componentID>totalTaxAmountInput</components:componentID>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Total_tax_amount"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_loaded"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_updated"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_item_updated"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_22"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Unit_code">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Unit_code"/>
        <base:componentForForm rdf:resource="&base;c1_for_Edit_articles"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_orders"/>
        <base:componentForFormItem
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Url">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Url"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_com_textfield_Value">
        <rdf:type rdf:resource="&components;TextField"/>
        <base:componentForDataColumn rdf:resource="&base;c1_col_Value"/>
        <base:componentForFormItem
rdf:resource="&base;c1_for_Edit_articles"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_03"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Edit_addresses">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formTitle>Edit address</forms:formTitle>
        <forms:formID>AddressEdit</forms:formID>
        <forms:formHasSeparateView
rdf:resource="&base;c1_for_View_articles"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Hidden"/>
        <base:formMainEditTable rdf:resource="&base;c1_tab_Addresses"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c2_men_Hidden"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Edit_articles">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formTitle>Edit article</forms:formTitle>
        <forms:formID>ArticlesEdit</forms:formID>
        <forms:formHasSeparateView
rdf:resource="&base;c1_for_View_articles"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Hidden"/>
        <base:formMainEditTable rdf:resource="&base;c1_tab_Articles"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c2_men_Hidden"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_for_Edit_orders">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>OrdersEdit</forms:formID>
        <forms:formTitle>Edit order</forms:formTitle>
        <forms:formHasSeparateView
rdf:resource="&base;c1_for_View_orders"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Hidden"/>
        <base:formMainEditTable rdf:resource="&base;c1_tab_Orders"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Edit_parties">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>PartiesEdit</forms:formID>
        <forms:formTitle>Edit party</forms:formTitle>
        <forms:formHasSeparateView
rdf:resource="&base;c1_for_View_parties"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Hidden"/>
        <base:formMainEditTable rdf:resource="&base;c1_tab_Parties"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c2_men_Hidden"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_View_addresses">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formTitle>Addresses</forms:formTitle>
        <forms:formID>AddressView</forms:formID>
        <forms:formHasSeparateEdit
rdf:resource="&base;c1_for_Edit_addresses"/>
        <forms:formIsPartOfMenuGroup
rdf:resource="&base;c1_men_Addresses"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_View_articles">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formTitle>Articles</forms:formTitle>
        <forms:formID>ArticlesView</forms:formID>
        <forms:formHasSeparateEdit
rdf:resource="&base;c1_for_Edit_articles"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Articles"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_View_orders">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:isFormHome
rdf:datatype="&xsd;boolean">true</forms:isFormHome>
        <forms:formIsHome
rdf:datatype="&xsd;boolean">true</forms:formIsHome>
        <forms:formTitle>Orders</forms:formTitle>
        <forms:formID>OrdersView</forms:formID>
        <forms:formHasSeparateEdit
rdf:resource="&base;c1_for_Edit_orders"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Orders"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_View_parties">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>PartiesView</forms:formID>
        <forms:formTitle>Parties</forms:formTitle>
        <forms:formHasSeparateEdit
rdf:resource="&base;c1_for_Edit_parties"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Parties"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Viewedit_contacts">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>ContactsViewEdit</forms:formID>
        <forms:formTitle>Contacts</forms:formTitle>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Parties"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Viewedit_countries">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>CountriesViewEdit</forms:formID>
        <forms:formTitle>Countries</forms:formTitle>
        <forms:formIsPartOfMenuGroup
rdf:resource="&base;c1_men_Addresses"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Viewedit_document_references">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formTitle>Document references</forms:formTitle>
        <forms:formID>DocumentReferencesViewEdit</forms:formID>
        <forms:formIsPartOfMenuGroup
rdf:resource="&base;c1_men_References"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Viewedit_incoterms">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>IncotermsViewEdit</forms:formID>
        <forms:formTitle>Incoterms</forms:formTitle>
        <forms:formIsPartOfMenuGroup
rdf:resource="&base;c1_men_Codebooks"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c1_for_Viewedit_party_legal_entities">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formTitle>Legal entities</forms:formTitle>
        <forms:formID>LegalEntitiesViewEdit</forms:formID>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Parties"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Viewedit_party_tax_schemas">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>PartyTaxSchemeViewEdit</forms:formID>
        <forms:formTitle>Party tax schemas</forms:formTitle>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Parties"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Viewedit_persons">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>PersonsViewEdit</forms:formID>
        <forms:formTitle>Persons</forms:formTitle>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c1_men_Parties"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_for_Viewedit_tax_schemas">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formTitle>Tax schemas</forms:formTitle>
        <forms:formID>TaxSchemasViewEdit</forms:formID>
        <forms:formIsPartOfMenuGroup
rdf:resource="&base;c1_men_Codebooks"/>
        <forms:formIsPartOfMenuGroup
rdf:resource="&base;c2_men_Tax_schemas"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_lis_On_form_loaded">
        <rdf:type rdf:resource="&components;OnFormLoaded"/>
        <base:listenerForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:listenerForForm rdf:resource="&base;c2_for_Edit_quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_lis_On_form_updated">
        <rdf:type rdf:resource="&components;OnFormUpdated"/>
        <base:listenerForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:listenerForForm rdf:resource="&base;c2_for_Edit_quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_lis_On_item_updated">
```

```
        <rdf:type rdf:resource="&components;OnItemUpdated"/>
        <base:listenerForForm rdf:resource="&base;c1_for_Edit_orders"/>
        <base:listenerForForm rdf:resource="&base;c2_for_Edit_quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_Addresses">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Addresses</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_05"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_Articles">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Articles</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_Codebooks">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Codebooks</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_08"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_Hidden">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupIsHidden
rdf:datatype="&xsd;boolean">true</forms:menuGroupIsHidden>
        <forms:isMenuGroupHidden
rdf:datatype="&xsd;boolean">true</forms:isMenuGroupHidden>
        <forms:menuGroupTitle>Hidden</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_Orders">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Orders</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_01"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_Parties">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Parties</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_04"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_Quotations">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Quotations</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
```

```
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_01"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_men_References">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>References</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_09"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_pro_Orders_management">
        <rdf:type rdf:resource="&requirements;Project"/>
        <requirements:projectTitle>Orders
management</requirements:projectTitle>

<requirements:projectBasePackage>hr.foi.dstrmecki.orders</requirements:proj
ectBasePackage>
        <requirements:projectID>orders_management</requirements:projectID>
        <requirements:description>Case study 1 as part of the doctoral
thesis of candidate Daniel Strmecki at Faculty or organization and
informatics in Varazdin, Croatia. Implementation of an aplication for
managmenet of orders. Requirements based on official example of UBL 2.1
order downloaded from the OASIS consortium Web site (http://docs.oasis-
open.org/ubl/UBL-2.1.html).</requirements:description>
        <requirements:projectForClient
rdf:resource="&base;c1_cli_Daniel_Strmecki"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_rep_Orders_database">
        <rdf:type rdf:resource="&repository;Repository"/>

<repository:respositoryPassword>123456</repository:respositoryPassword>

<repository:respositoryUsername>dstrmecki</repository:respositoryUsername>

<repository:respositoryDriver>com.mysql.jdbc.Driver</repository:respository
Driver>

<repository:repositoryURL>jdbc:mysql://localhost:3306/orders_management</re
pository:repositoryURL>

<repository:respositoryDialect>org.hibernate.dialect.MySQL5InnoDBDialect</r
epository:respositoryDialect>

<repository:repositoryName>orders_management</repository:repositoryName>
        <base:repositoryForProject
rdf:resource="&base;c1_pro_Orders_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Addresses">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new addresses and editing or
deleting existing addresses. A single address can be connected to multiple
parties or orders. Addresses should have editable fields for Name, Address
identification, Street name, Additional street name, Building number,
Department, City name, Postal zone, Postbox, Country and Country subentity.
Name is the only mandatory field. Country is an selection
field.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Articles">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new articles and editing or
deleting existing articles. A single articles can be connected to multiple
orders. Articles should have editable fields for Name, Base quantity, Price
amount, Unit code, Sellers item identification, Standard item
identification and Description. Name is the only mandatory field. Articles
can have serveral property-value pairs connected to
them.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Contacts">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new contacts and editing or
deleting existing contacts. A single contact can be assigned to multiple
parties. Contact should have editable fields for Name, Telephone, Telefax
and Electronic Mail. Name is the only mandatory
field.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Countries">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new countries and editing or
deleting existing countries. A single country can be assigned to multiple
adresses. Country should have editable fields for Identification Code and
Name. Name is the only mandatory field.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Document_references">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new document references and
editing or deleting existing document references. A single document
references can be connected to multiple orders. Document references should
have editable fields for Name, Type of reference, URL and Note. Name and
Type of reference are the only mandatory fields. Type of reference is an
option group field and can be either Qutation, Order, Originator, Contract
or Additional.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Incoterms">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new incoterms and editing or
deleting existing incoterms. A single incoterm can be assigned to multiple
orders. Incoterms should have editable fields for Name and Description.
Name is the only mandatory field.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;c1_req_Legal_entities">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new legal entities and editing or
deleting existing legal entities. A single legal entity can be connected to
multiple parties. Legal entities should have editable fields for Name,
Company ID and Address. Name is the only mandatory field. Address is an
selection field.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Orders">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new orders and editing or deleting
existing orders. Orders should have editable fields for Name, Issue date /
time, Note, Valid from / to, Document currency code, Accounting cost code,
Buyer customer party, Seller supplier party, Originator customer party,
Delivery location, Delivery party, Delivery terms, Special terms, Allowance
amount, Charge amount,  and Description. Name, Buyer customer party and
Seller supplier party are mandatory fields. Buyer customer party, Seller
supplier party, Originator customer party, Delivery location, Delivery
party, Delivery terms and Special terms are selection fields. Line
extension amount, Tax amount and Payable amount are non-editable fields
that are automatically calculated using items data. Payable amount is the
total Line extension amount + Tax amount + Allowance amount - Charge
amount. Orders can have serveral items connected to them. Orders should
have editable fields for Article, Quantity, Unit code, Tax amount, Line
extension amount, Accounting cost code, Partial delivery and Requested
delivery from / to. Article and Quantity are mandatory fields for items.
Article is a selection field and Partial delivery is a checkbox field.
Orders can have references to multiple
Documents.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Parties">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new parties and editing or
deleting existing parties. A single party can be assigned to multiple
orders. Parties should have editable fields for Name, Party Identification,
Address, Party Tax Scheme and Party Legal Entity. Name is the only
mandatory field. Parties can have references to multiple Contacts,
Dellivery contacts and Persons.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Party_tax_schemas">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new party tax schemas and editing
or deleting existing  party tax schemas. A single  party tax schema can be
connected to multiple parties. Party tax schemas should have editable
fields for Name, Company ID, Address and Tax scheme. Name is the only
mandatory field. Address and Tax scheme are selection
fields.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Persons">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new persons and editing or
deleting existing persons. A single person can be assigned to multiple
parties. Person should have editable fields for Name (fulll name), First
Name, Family Name, Middle Name and Job Title. Name is the only mandatory
field.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Tax_schemas">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new tax schemas and editing or
deleting existing  tax schemas. A single  tax schema can be assigned to
multiple orders. Tax schemas should have editable fields for Name and
Description. Name is the only mandatory field.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_req_Users_and_rights">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>The ability to add new users to the
system and assigning them different user rights for different set of
entities based on their organizational role.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c1_pro_Orders_management"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Calculations">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I am able to see auto-
calculated values like sum every time I update an entity or its
items.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Entity_add">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can create a new entity of
this type.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Addresses"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Articles"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Contacts"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Countries"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Document_references"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Incoterms"/>
```

```
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Legal_entities"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Parties"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Party_tax_schemas"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Persons"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Tax_schemas"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Entity_delete">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can delete an existing
entity of this type.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Addresses"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Articles"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Contacts"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Countries"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Document_references"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Incoterms"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Legal_entities"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Parties"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Party_tax_schemas"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Persons"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Tax_schemas"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Entity_edit">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can edit an existing entity
of this type.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Addresses"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Articles"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Contacts"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Countries"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Document_references"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Incoterms"/>
```

```
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Legal_entities"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Parties"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Party_tax_schemas"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Persons"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Tax_schemas"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Item_add">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can create a new item for a
specific entity.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Articles"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Item_delete">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can delete an existing item
for a specific entity.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Articles"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Item_edit">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can edit an existing item
for a specific entity.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Articles"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Reference_add">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can add a reference from one
entity to another.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Parties"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Reference_delete">
        <rdf:type rdf:resource="&requirements;UserStory"/>
```

```
        <requirements:description>As a user, I can remove a reference from
one entity to another.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Parties"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_Rights_management">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As an user in an administrative role I
can assign read, create, update and delete user rights to specific users
for specific sets of entities.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Users_and_rights"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_User_management">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As an user in an administrative role I
can add new users to the systems, deactivate or delete existing
users.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Users_and_rights"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_sto_User_permissions">
        <rdf:type rdf:resource="&requirements;UserStory"/>
        <requirements:description>As a user, I can only perform read,
create, update and delete operations on entities only if I am assigned the
adequete user rights.</requirements:description>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Addresses"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Articles"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Contacts"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Countries"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Document_references"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Incoterms"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Legal_entities"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Orders"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Parties"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Party_tax_schemas"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Persons"/>
        <requirements:storyIsPartOfRequirement
rdf:resource="&base;c1_req_Tax_schemas"/>
        <requirements:requirementForProject
rdf:resource="&base;c2_req_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Addresses">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>address</repository:tableName>
```

```
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Articles">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>article</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Contacts">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>contact</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Contacts_join">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>contacts_join</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Countries">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>country</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Delivery_contacts_join">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>delivery_contacts_join</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Document_references">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>document_reference</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Document_references_join">
        <rdf:type rdf:resource="&repository;Table"/>

<repository:tableName>document_reference_join</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Incoterms">
```

```
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>incoterms</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Order_items">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>order_items</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Orders">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>order1</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Parties">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>party</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Party_legal_entities">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>party_legal_entity</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Party_tax_schemas">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>party_tax_scheme</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Persons">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>person</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Persons_join">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>persons_join</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Properties">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>item_property</repository:tableName>
```

```
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c1_tab_Tax_schemas">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>tax_scheme</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c1_rep_Orders_database"/>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_col_Delivery_terms">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>String</repository:columnType>
        <repository:columnTitle>Delivery terms</repository:columnTitle>
        <repository:columnID>deliveryTerms</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_17"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_col_Quotation">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:isColumnValueNotNull
rdf:datatype="&xsd;boolean">true</repository:isColumnValueNotNull>
        <repository:isColumnJoinedManyToOne
rdf:datatype="&xsd;boolean">true</repository:isColumnJoinedManyToOne>
        <repository:columnTitle>Quotation</repository:columnTitle>
        <repository:columnID>quotation</repository:columnID>
        <repository:columnJoinByID>quotation_id</repository:columnJoinByID>
        <repository:columnType>Quotation</repository:columnType>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Document_references_join"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotation_items"/>
        <base:uiForColumn rdf:resource="&base;ui_Position_02"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_col_Total_tax_exclusive_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>Double</repository:columnType>
        <repository:columnID>taxExclusiveAmount</repository:columnID>
        <repository:columnTitle>Tax exclusive
amount</repository:columnTitle>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_rep_Quotations_database"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_col_Total_tax_inclusive_amount">
        <rdf:type rdf:resource="&repository;Column"/>
        <repository:columnType>Double</repository:columnType>
        <repository:columnTitle>Tax inclusive
amount</repository:columnTitle>
        <repository:columnID>taxInclusiveAmount</repository:columnID>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_rep_Quotations_database"/>
        <repository:columnIsPartOfTable
rdf:resource="&base;c2_tab_Quotations"/>
    </NamedIndividual>
```

```
    <NamedIndividual
rdf:about="&base;c2_com_pagedtable_Document_references_subtable">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:pagedTableIsMain
rdf:datatype="&xsd;boolean">false</components:pagedTableIsMain>
        <components:componentLabel>Document
references</components:componentLabel>

<components:componentID>documentReferencesSubtable</components:componentID>
        <base:pagedTableJoinColumn
rdf:resource="&base;c1_col_Document_reference"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Document_reference"/>
        <base:pagedTableEditForm
rdf:resource="&base;c1_for_Viewedit_document_references"/>
        <base:pagedTableSubTable
rdf:resource="&base;c1_tab_Document_references"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:pagedTableMainTable
rdf:resource="&base;c2_tab_Document_references_join"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_com_pagedtable_Quotation_items">
        <rdf:type rdf:resource="&components;PagedTable"/>

<components:componentID>quotationItemsTable</components:componentID>
        <components:componentLabel>Quotation
items</components:componentLabel>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Article"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Line_extension_amount"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Quantity"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Tax_amount"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Unit_code"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:pagedTableMainTable
rdf:resource="&base;c2_tab_Quotation_items"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_com_pagedtable_Quotations">
        <rdf:type rdf:resource="&components;PagedTable"/>
        <components:componentID>ordersTable</components:componentID>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Issue_date"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Name"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Originator_customer_party"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Seller_supplier_party"/>
        <base:pagedTableShowColumn
rdf:resource="&base;c1_col_Total_payable_amount"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Valid_from"/>
        <base:pagedTableShowColumn rdf:resource="&base;c1_col_Valid_to"/>
        <base:componentForForm
rdf:resource="&base;c2_for_View_quotations"/>
        <base:pagedTableMainTable rdf:resource="&base;c2_tab_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_com_textarea_Delivery_terms">
        <rdf:type rdf:resource="&components;TextArea"/>
        <base:componentForDataColumn
rdf:resource="&base;c2_col_Delivery_terms"/>
```

```
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_17"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c2_com_textfield_Total_line_extension_amount">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:isInputFieldDisabled
rdf:datatype="&xsd;boolean">true</components:isInputFieldDisabled>
        <components:inputFieldUpdateQuery>SELECT
sum(item.lineExtensionAmount) FROM QuotationItems item WHERE item.quotation
= {editID}</components:inputFieldUpdateQuery>

<components:componentID>totalLineExtensionInput</components:componentID>

<components:inputFieldConverter>Double</components:inputFieldConverter>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Total_line_extension_amount"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_loaded"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_updated"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_item_updated"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_20"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c2_com_textfield_Total_payable_amount">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:isInputFieldDisabled
rdf:datatype="&xsd;boolean">true</components:isInputFieldDisabled>
        <components:inputFieldUpdateQuery>SELECT
sum(item.lineExtensionAmount) + sum(item.taxAmount) FROM QuotationItems
item WHERE item.quotation = {editID}</components:inputFieldUpdateQuery>

<components:inputFieldConverter>Double</components:inputFieldConverter>

<components:componentID>totalPayaybleAmountInput</components:componentID>
        <base:componentForDataColumn
rdf:resource="&base;c1_col_Total_payable_amount"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_loaded"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_updated"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_item_updated"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_23"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c2_com_textfield_Total_tax_exclusive_amount">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:isInputFieldDisabled
rdf:datatype="&xsd;boolean">true</components:isInputFieldDisabled>

<components:componentID>totalTaxExclusiveAmountInput</components:componentI
D>
```

```
<components:inputFieldConverter>Double</components:inputFieldConverter>
        <components:inputFieldUpdateQuery>SELECT
sum(item.lineExtensionAmount) FROM QuotationItems item WHERE item.quotation
= {editID}</components:inputFieldUpdateQuery>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_loaded"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_updated"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_item_updated"/>
        <base:componentForDataColumn
rdf:resource="&base;c2_col_Total_tax_exclusive_amount"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_21"/>
    </NamedIndividual>
    <NamedIndividual
rdf:about="&base;c2_com_textfield_Total_tax_inclusive_amount">
        <rdf:type rdf:resource="&components;TextField"/>
        <components:isInputFieldDisabled
rdf:datatype="&xsd;boolean">true</components:isInputFieldDisabled>
        <components:inputFieldUpdateQuery>SELECT
sum(item.lineExtensionAmount) + sum(item.taxAmount) FROM QuotationItems
item WHERE item.quotation = {editID}</components:inputFieldUpdateQuery>

<components:inputFieldConverter>Double</components:inputFieldConverter>

<components:componentID>totalTaxInclusiveAmountInput</components:componentI
D>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_loaded"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_form_updated"/>
        <components:inputFieldUpdateListener
rdf:resource="&base;c1_lis_On_item_updated"/>
        <base:componentForDataColumn
rdf:resource="&base;c2_col_Total_tax_inclusive_amount"/>
        <base:componentForForm
rdf:resource="&base;c2_for_Edit_quotations"/>
        <base:uiForComponent rdf:resource="&base;ui_Position_22"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_for_Edit_quotations">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:formID>QuotationsEdit</forms:formID>
        <forms:formTitle>Edit quotation</forms:formTitle>
        <forms:formHasSeparateView
rdf:resource="&base;c2_for_View_quotations"/>
        <forms:formIsPartOfMenuGroup rdf:resource="&base;c2_men_Hidden"/>
        <base:formMainEditTable rdf:resource="&base;c2_tab_Quotations"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_for_View_quotations">
        <rdf:type rdf:resource="&forms;Form"/>
        <forms:isFormHome
rdf:datatype="&xsd;boolean">true</forms:isFormHome>
        <forms:formID>QuotationsView</forms:formID>
        <forms:formTitle>Quotations</forms:formTitle>
        <forms:formHasSeparateEdit
rdf:resource="&base;c2_for_Edit_quotations"/>
        <forms:formIsPartOfMenuGroup
rdf:resource="&base;c2_men_Quotations"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_men_Hidden">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:isMenuGroupHidden
rdf:datatype="&xsd;boolean">true</forms:isMenuGroupHidden>
        <forms:menuGroupTitle>Hidden</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_last"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_men_Quotations">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Quotations</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_01"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_men_Tax_schemas">
        <rdf:type rdf:resource="&forms;MenuGroup"/>
        <forms:menuGroupTitle>Tax schemas</forms:menuGroupTitle>
        <base:menuGroupForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
        <base:uiForMenuGroup rdf:resource="&base;ui_Position_08"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_pro_Quotations_management">
        <rdf:type rdf:resource="&requirements;Project"/>
        <requirements:projectTitle>Quotations
management</requirements:projectTitle>
        <requirements:description>Case study 2  as part of the doctoral
thesis of candidate Daniel Strmecki at Faculty or organization and
informatics in Varazdin, Croatia. Implementation of an aplication for
managmenet of quotations. Requirements based on official example of UBL 2.1
quotation downloaded from the OASIS consortium Web site (http://docs.oasis-
open.org/ubl/UBL-2.1.html).</requirements:description>

<requirements:projectBasePackage>hr.foi.dstrmecki.quotations</requirements:
projectBasePackage>

<requirements:projectID>quotations_management</requirements:projectID>
        <requirements:projectForClient
rdf:resource="&base;c1_cli_Daniel_Strmecki"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_rep_Quotations_database">
        <rdf:type rdf:resource="&repository;Repository"/>

<repository:repositoryURL>jdbc:mysql://localhost:3306/quotation_management<
/repository:repositoryURL>

<repository:respositoryPassword>123456</repository:respositoryPassword>

<repository:repositoryName>quotation_management</repository:repositoryName>

<repository:respositoryDriver>com.mysql.jdbc.Driver</repository:respository
Driver>

<repository:respositoryDialect>org.hibernate.dialect.MySQL5InnoDBDialect</r
epository:respositoryDialect>

<repository:respositoryUsername>dstrmecki</repository:respositoryUsername>
        <base:repositoryForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
```

```
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_req_Quotations">
        <rdf:type rdf:resource="&requirements;Requirement"/>
        <requirements:description>Adding new quotations and editing or
deleting existing quotations. Quotation should have editable fields for
Name, Issue date / time, Note, Valid from / to, Document currency code,
Seller supplier party, Originator customer party, Delivery location,
Delivery terms and Requested delivery period from / to,  Name and Seller
supplier party are mandatory fields. Seller supplier party, Originator
customer party and Delivery location are selection fields. Line extension
amount, Tax inclusive amount, Tax exclusive amount and Payable amount are
non-editable fields that are automatically calculated using items data. Tax
inclusive amount and Payable amount is the total Line extension amount +
total Tax amount. Quotations can have serveral items connected to them.
Quotations should have editable fields for Article, Quantity, Unit code,
Tax amount and Line extension amount. Article and Quantity are mandatory
fields for items. Article is a selection field and Partial delivery is a
checkbox field. Quotations can have references to multiple
Documents.</requirements:description>
        <requirements:requirementForProject
rdf:resource="&base;c2_pro_Quotations_management"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_tab_Document_references_join">
        <rdf:type rdf:resource="&repository;Table"/>

<repository:tableName>document_reference_join</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_tab_Quotation_items">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>quotation_items</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;c2_tab_Quotations">
        <rdf:type rdf:resource="&repository;Table"/>
        <repository:tableName>quotation</repository:tableName>
        <repository:tableIsPartOfRepository
rdf:resource="&base;c2_rep_Quotations_database"/>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_01">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">1</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_02">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">2</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_03">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">3</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_04">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">4</interface:uiPositionIndex>
    </NamedIndividual>
```

```
    <NamedIndividual rdf:about="&base;ui_Position_05">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">5</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_06">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">6</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_07">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">7</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_08">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">8</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_09">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">9</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_10">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">10</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_11">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">11</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_12">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">12</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_13">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">13</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_14">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">14</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_15">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">15</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_16">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">16</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_17">
```

```
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">17</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_18">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">18</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_19">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">19</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_20">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">20</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_21">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">21</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_22">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">22</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_23">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">23</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_24">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">24</interface:uiPositionIndex>
    </NamedIndividual>
    <NamedIndividual rdf:about="&base;ui_Position_last">
        <rdf:type rdf:resource="&interface;Interface"/>
        <interface:uiPositionIndex
rdf:datatype="&xsd;int">99</interface:uiPositionIndex>
    </NamedIndividual>
</rdf:RDF>
```

DODATAK H – SLUŽBENI PRIMJER UBL NARUDŽBENICE


```xml
<?xml version="1.0" encoding="UTF-8"?>
<Order xmlns="urn:oasis:names:specification:ubl:schema:xsd:Order-2"

xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComp
onents-2"

xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponen
ts-2">
  <cbc:UBLVersionID>2.1</cbc:UBLVersionID>

<cbc:CustomizationID>urn:www.cenbii.eu:transaction:biicoretrdm001:ver1.0</c
bc:CustomizationID>
  <cbc:ProfileID schemeAgencyID="BII" schemeID="Profile"
    >urn:www.cenbii.eu:profile:BII01:ver1.0</cbc:ProfileID>
  <cbc:ID>34</cbc:ID>
  <cbc:IssueDate>2010-01-20</cbc:IssueDate>
  <cbc:IssueTime>12:30:00</cbc:IssueTime>
  <cbc:Note>Information text for the whole order</cbc:Note>
  <cbc:DocumentCurrencyCode>SEK</cbc:DocumentCurrencyCode>
  <cbc:AccountingCostCode>Project123</cbc:AccountingCostCode>
  <cac:ValidityPeriod>
    <cbc:EndDate>2010-01-31</cbc:EndDate>
  </cac:ValidityPeriod>
  <cac:QuotationDocumentReference>
    <cbc:ID>QuoteID123</cbc:ID>
  </cac:QuotationDocumentReference>
  <cac:OrderDocumentReference>
    <cbc:ID>RjectedOrderID123</cbc:ID>
  </cac:OrderDocumentReference>
  <cac:OriginatorDocumentReference>
    <cbc:ID>MAFO</cbc:ID>
  </cac:OriginatorDocumentReference>
  <cac:AdditionalDocumentReference>
    <cbc:ID>Doc1</cbc:ID>
    <cbc:DocumentType>Timesheet</cbc:DocumentType>
    <cac:Attachment>
      <cac:ExternalReference>
        <cbc:URI>http://www.suppliersite.eu/sheet001.html</cbc:URI>
      </cac:ExternalReference>
    </cac:Attachment>
  </cac:AdditionalDocumentReference>
  <cac:AdditionalDocumentReference>
    <cbc:ID>Doc2</cbc:ID>
    <cbc:DocumentType>Drawing</cbc:DocumentType>
    <cac:Attachment>
      <cbc:EmbeddedDocumentBinaryObject mimeCode="application/pdf"

>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi</cbc:EmbeddedDocumentBina
ryObject>
    </cac:Attachment>
  </cac:AdditionalDocumentReference>
  <cac:Contract>
    <cbc:ID>34322</cbc:ID>
    <cbc:ContractType>FrameworkAgreementID123</cbc:ContractType>
  </cac:Contract>
  <cac:BuyerCustomerParty>
    <cac:Party>
```

```
      <cbc:EndpointID schemeAgencyID="9"
schemeID="GLN">7300072311115</cbc:EndpointID>
      <cac:PartyIdentification>
        <cbc:ID schemeAgencyID="9" schemeID="GLN">7300070011115</cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyIdentification>
        <cbc:ID>PartyID123</cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyName>
        <cbc:Name>Johnssons byggvaror</cbc:Name>
      </cac:PartyName>
      <cac:PostalAddress>
        <cbc:ID schemeAgencyID="9" schemeID="GLN">1234567890123</cbc:ID>
        <cbc:Postbox>PoBox123</cbc:Postbox>
        <cbc:StreetName>Rådhusgatan</cbc:StreetName>
        <cbc:AdditionalStreetName>2nd floor</cbc:AdditionalStreetName>
        <cbc:BuildingNumber>5</cbc:BuildingNumber>
        <cbc:Department>Purchasing department</cbc:Department>
        <cbc:CityName>Stockholm</cbc:CityName>
        <cbc:PostalZone>11000</cbc:PostalZone>
        <cbc:CountrySubentity>RegionX</cbc:CountrySubentity>
        <cac:Country>
          <cbc:IdentificationCode>SE</cbc:IdentificationCode>
        </cac:Country>
      </cac:PostalAddress>
      <cac:PartyTaxScheme>
        <cbc:RegistrationName>Herra Johnssons byggvaror
AS</cbc:RegistrationName>
        <cbc:CompanyID>SE1234567801</cbc:CompanyID>
        <cac:RegistrationAddress>
          <cbc:CityName>Stockholm</cbc:CityName>
          <cac:Country>
            <cbc:IdentificationCode>SE</cbc:IdentificationCode>
          </cac:Country>
        </cac:RegistrationAddress>
        <cac:TaxScheme>
          <cbc:ID schemeID="UN/ECE 5153" schemeAgencyID="6">VAT</cbc:ID>
        </cac:TaxScheme>
      </cac:PartyTaxScheme>
      <cac:PartyLegalEntity>
        <cbc:RegistrationName>Johnssons Byggvaror AB</cbc:RegistrationName>
        <cbc:CompanyID schemeID="SE:ORGNR">5532331183</cbc:CompanyID>
        <cac:RegistrationAddress>
          <cbc:CityName>Stockholm</cbc:CityName>
          <cbc:CountrySubentity>RegionX</cbc:CountrySubentity>
          <cac:Country>
            <cbc:IdentificationCode>SE</cbc:IdentificationCode>
          </cac:Country>
        </cac:RegistrationAddress>
      </cac:PartyLegalEntity>
      <cac:Contact>
        <cbc:Telephone>123456</cbc:Telephone>
        <cbc:Telefax>123456</cbc:Telefax>
        <cbc:ElectronicMail>pelle@johnsson.se</cbc:ElectronicMail>
      </cac:Contact>
      <cac:Person>
        <cbc:FirstName>Pelle</cbc:FirstName>
        <cbc:FamilyName>Svensson</cbc:FamilyName>
        <cbc:MiddleName>X</cbc:MiddleName>
        <cbc:JobTitle>Boss</cbc:JobTitle>
      </cac:Person>
```

```
    </cac:Party>
    <cac:DeliveryContact>
      <cbc:Name>Eva Johnsson</cbc:Name>
      <cbc:Telephone>1234356</cbc:Telephone>
      <cbc:Telefax>123455</cbc:Telefax>
      <cbc:ElectronicMail>eva@johnsson.se</cbc:ElectronicMail>
    </cac:DeliveryContact>
  </cac:BuyerCustomerParty>
  <cac:SellerSupplierParty>
    <cac:Party>
      <cbc:EndpointID schemeAgencyID="9"
schemeID="GLN">7302347231111</cbc:EndpointID>
      <cac:PartyIdentification>
        <cbc:ID>SellerPartyID123</cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyName>
        <cbc:Name>Moderna Produkter AB</cbc:Name>
      </cac:PartyName>
      <cac:PostalAddress>
        <cbc:ID schemeAgencyID="9" schemeID="GLN">0987654321123</cbc:ID>
        <cbc:Postbox>321</cbc:Postbox>
        <cbc:StreetName>Kungsgatan</cbc:StreetName>
        <cbc:AdditionalStreetName>suite12</cbc:AdditionalStreetName>
        <cbc:BuildingNumber>22</cbc:BuildingNumber>
        <cbc:Department>Sales department</cbc:Department>
        <cbc:CityName>Stockholm</cbc:CityName>
        <cbc:PostalZone>11000</cbc:PostalZone>
        <cbc:CountrySubentity>RegionX</cbc:CountrySubentity>
        <cac:Country>
          <cbc:IdentificationCode>SE</cbc:IdentificationCode>
        </cac:Country>
      </cac:PostalAddress>
      <cac:PartyLegalEntity>
        <cbc:RegistrationName>Moderna Produkter AB</cbc:RegistrationName>
        <cbc:CompanyID schemeID="SE:ORGNR">5532332283</cbc:CompanyID>
        <cac:RegistrationAddress>
          <cbc:CityName>Stockholm</cbc:CityName>
          <cbc:CountrySubentity>RegionX</cbc:CountrySubentity>
          <cac:Country>
            <cbc:IdentificationCode>SE</cbc:IdentificationCode>
          </cac:Country>
        </cac:RegistrationAddress>
      </cac:PartyLegalEntity>
      <cac:Contact>
        <cbc:Telephone>34557</cbc:Telephone>
        <cbc:Telefax>3456767</cbc:Telefax>
        <cbc:ElectronicMail>lars@moderna.se</cbc:ElectronicMail>
      </cac:Contact>
      <cac:Person>
        <cbc:FirstName>Lars</cbc:FirstName>
        <cbc:FamilyName>Petersen</cbc:FamilyName>
        <cbc:MiddleName>M</cbc:MiddleName>
        <cbc:JobTitle>Sales manager</cbc:JobTitle>
      </cac:Person>
    </cac:Party>
  </cac:SellerSupplierParty>
  <cac:OriginatorCustomerParty>
    <cac:Party>
      <cac:PartyIdentification>
        <cbc:ID schemeAgencyID="9" schemeID="GLN">0987678321123</cbc:ID>
      </cac:PartyIdentification>
```

```
    <cac:PartyName>
      <cbc:Name>Moderna Produkter AB</cbc:Name>
    </cac:PartyName>
    <cac:Contact>
      <cbc:Telephone>346788</cbc:Telephone>
      <cbc:Telefax>8567443</cbc:Telefax>
      <cbc:ElectronicMail>sven@moderna.se</cbc:ElectronicMail>
    </cac:Contact>
    <cac:Person>
      <cbc:FirstName>Sven</cbc:FirstName>
      <cbc:FamilyName>Pereson</cbc:FamilyName>
      <cbc:MiddleName>N</cbc:MiddleName>
      <cbc:JobTitle>Stuffuser</cbc:JobTitle>
    </cac:Person>
  </cac:Party>
</cac:OriginatorCustomerParty>
<cac:Delivery>
  <cac:DeliveryLocation>
    <cac:Address>
      <cbc:ID schemeAgencyID="9" schemeID="GLN">1234567890123</cbc:ID>
      <cbc:Postbox>123</cbc:Postbox>
      <cbc:StreetName>Rådhusgatan</cbc:StreetName>
      <cbc:AdditionalStreetName>2nd floor</cbc:AdditionalStreetName>
      <cbc:BuildingNumber>5</cbc:BuildingNumber>
      <cbc:Department>Purchasing department</cbc:Department>
      <cbc:CityName>Stockholm</cbc:CityName>
      <cbc:PostalZone>11000</cbc:PostalZone>
      <cbc:CountrySubentity>RegionX</cbc:CountrySubentity>
      <cac:Country>
        <cbc:IdentificationCode>SE</cbc:IdentificationCode>
      </cac:Country>
    </cac:Address>
  </cac:DeliveryLocation>
  <cac:RequestedDeliveryPeriod>
    <cbc:StartDate>2010-02-10</cbc:StartDate>
    <cbc:EndDate>2010-02-25</cbc:EndDate>
  </cac:RequestedDeliveryPeriod>
  <cac:DeliveryParty>
    <cac:PartyIdentification>
      <cbc:ID schemeAgencyID="9" schemeID="GLN">67654328394567</cbc:ID>
    </cac:PartyIdentification>
    <cac:PartyName>
      <cbc:Name>Swedish trucking</cbc:Name>
    </cac:PartyName>
    <cac:Contact>
      <cbc:Name>Per</cbc:Name>
      <cbc:Telephone>987098709</cbc:Telephone>
      <cbc:Telefax>34673435</cbc:Telefax>
      <cbc:ElectronicMail>bill@svetruck.se</cbc:ElectronicMail>
    </cac:Contact>
  </cac:DeliveryParty>
</cac:Delivery>
<cac:DeliveryTerms>
  <cbc:ID schemeAgencyID="6" schemeID="IMCOTERM">FOT</cbc:ID>
  <cbc:SpecialTerms>CAD</cbc:SpecialTerms>
  <cac:DeliveryLocation>
    <cbc:ID>STO</cbc:ID>
  </cac:DeliveryLocation>
</cac:DeliveryTerms>
<cac:AllowanceCharge>
  <cbc:ChargeIndicator>true</cbc:ChargeIndicator>
```

```
    <cbc:AllowanceChargeReason>Transport
documents</cbc:AllowanceChargeReason>
    <cbc:Amount currencyID="SEK">100</cbc:Amount>
  </cac:AllowanceCharge>
  <cac:AllowanceCharge>
    <cbc:ChargeIndicator>false</cbc:ChargeIndicator>
    <cbc:AllowanceChargeReason>Total order value
discount</cbc:AllowanceChargeReason>
    <cbc:Amount currencyID="SEK">100</cbc:Amount>
  </cac:AllowanceCharge>
  <cac:TaxTotal>
    <cbc:TaxAmount currencyID="SEK">100</cbc:TaxAmount>
  </cac:TaxTotal>
  <cac:AnticipatedMonetaryTotal>
    <cbc:LineExtensionAmount
currencyID="SEK">6225</cbc:LineExtensionAmount>
    <cbc:AllowanceTotalAmount
currencyID="SEK">100</cbc:AllowanceTotalAmount>
    <cbc:ChargeTotalAmount currencyID="SEK">100</cbc:ChargeTotalAmount>
    <cbc:PayableAmount currencyID="SEK">6225</cbc:PayableAmount>
  </cac:AnticipatedMonetaryTotal>
  <cac:OrderLine>
    <cbc:Note>Freetext note on line 1</cbc:Note>
    <cac:LineItem>
      <cbc:ID>1</cbc:ID>
      <cbc:Quantity unitCode="LTR">120</cbc:Quantity>
      <cbc:LineExtensionAmount
currencyID="SEK">6000</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="SEK">10</cbc:TotalTaxAmount>
      <cbc:PartialDeliveryIndicator>false</cbc:PartialDeliveryIndicator>
      <cbc:AccountingCostCode>ProjectID123</cbc:AccountingCostCode>
      <cac:Delivery>
        <cac:RequestedDeliveryPeriod>
          <cbc:StartDate>2010-02-10</cbc:StartDate>
          <cbc:EndDate>2010-02-25</cbc:EndDate>
        </cac:RequestedDeliveryPeriod>
      </cac:Delivery>
      <cac:OriginatorParty>
        <cac:PartyIdentification>
          <cbc:ID schemeAgencyID="ZZZ" schemeID="ZZZ">EmployeeXXX</cbc:ID>
        </cac:PartyIdentification>
        <cac:PartyName>
          <cbc:Name>Josef K.</cbc:Name>
        </cac:PartyName>
      </cac:OriginatorParty>
      <cac:Price>
        <cbc:PriceAmount currencyID="SEK">50</cbc:PriceAmount>
        <cbc:BaseQuantity unitCode="LTR">1</cbc:BaseQuantity>
      </cac:Price>
      <cac:Item>
        <cbc:Description>Red paint</cbc:Description>
        <cbc:Name>Falu Rödfärg</cbc:Name>
        <cac:SellersItemIdentification>
          <cbc:ID>SItemNo001</cbc:ID>
        </cac:SellersItemIdentification>
        <cac:StandardItemIdentification>
          <cbc:ID schemeAgencyID="6" schemeID="GTIN">1234567890123</cbc:ID>
        </cac:StandardItemIdentification>
        <cac:AdditionalItemProperty>
          <cbc:Name>Paint type</cbc:Name>
          <cbc:Value>Acrylic</cbc:Value>
```

```
        </cac:AdditionalItemProperty>
        <cac:AdditionalItemProperty>
          <cbc:Name>Solvant</cbc:Name>
          <cbc:Value>Water</cbc:Value>
        </cac:AdditionalItemProperty>
      </cac:Item>
    </cac:LineItem>
  </cac:OrderLine>
  <cac:OrderLine>
    <cbc:Note>Freetext note on line 2</cbc:Note>
    <cac:LineItem>
      <cbc:ID>2</cbc:ID>
      <cbc:Quantity unitCode="C62">15</cbc:Quantity>
      <cbc:LineExtensionAmount
currencyID="SEK">225</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="SEK">10</cbc:TotalTaxAmount>
      <cbc:PartialDeliveryIndicator>false</cbc:PartialDeliveryIndicator>
      <cbc:AccountingCostCode>ProjectID123</cbc:AccountingCostCode>
      <cac:Delivery>
        <cac:RequestedDeliveryPeriod>
          <cbc:StartDate>2010-02-10</cbc:StartDate>
          <cbc:EndDate>2010-02-25</cbc:EndDate>
        </cac:RequestedDeliveryPeriod>
      </cac:Delivery>
      <cac:OriginatorParty>
        <cac:PartyIdentification>
          <cbc:ID schemeAgencyID="ZZZ" schemeID="ZZZ">EmployeeXXX</cbc:ID>
        </cac:PartyIdentification>
        <cac:PartyName>
          <cbc:Name>Josef K.</cbc:Name>
        </cac:PartyName>
      </cac:OriginatorParty>
      <cac:Price>
        <cbc:PriceAmount currencyID="SEK">15</cbc:PriceAmount>
        <cbc:BaseQuantity unitCode="C62">1</cbc:BaseQuantity>
      </cac:Price>
      <cac:Item>
        <cbc:Description>Very good pencils for red paint.</cbc:Description>
        <cbc:Name>Pensel 20 mm</cbc:Name>
        <cac:SellersItemIdentification>
          <cbc:ID>SItemNo011</cbc:ID>
        </cac:SellersItemIdentification>
        <cac:StandardItemIdentification>
          <cbc:ID schemeAgencyID="6" schemeID="GTIN">123452340123</cbc:ID>
        </cac:StandardItemIdentification>
        <cac:AdditionalItemProperty>
          <cbc:Name>Hair color</cbc:Name>
          <cbc:Value>Black</cbc:Value>
        </cac:AdditionalItemProperty>
        <cac:AdditionalItemProperty>
          <cbc:Name>Width</cbc:Name>
          <cbc:Value>20mm</cbc:Value>
        </cac:AdditionalItemProperty>
      </cac:Item>
    </cac:LineItem>
  </cac:OrderLine>
</Order>
```

DODATAK I – SLUŽBENI PRIMJER UBL PONUDE


```xml
<?xml version="1.0" encoding="UTF-8"?>
<Quotation xmlns="urn:oasis:names:specification:ubl:schema:xsd:Quotation-2"
xmlns:cac="urn:oasis:names:specification:ubl:schema:xsd:CommonAggregateComp
onents-2"
xmlns:cbc="urn:oasis:names:specification:ubl:schema:xsd:CommonBasicComponen
ts-2">
  <cbc:UBLVersionID>2.0</cbc:UBLVersionID>
  <cbc:CustomizationID>OIOUBL-2.1</cbc:CustomizationID>
  <cbc:ProfileID schemeAgencyID="320" schemeID="urn:oioubl:id:profileid-
1.2">Procurement-QuoSim-1.0</cbc:ProfileID>
  <cbc:ID>QIY7655</cbc:ID>
      <cbc:CopyIndicator>false</cbc:CopyIndicator>
      <cbc:UUID>4D07786B-DA6D-439F-82D1-6FFFC7F4E3B1</cbc:UUID>
      <cbc:IssueDate>2008-05-01</cbc:IssueDate>
  <cbc:IssueTime>11:32:26.0Z</cbc:IssueTime>
  <cbc:Note languageID="da-dk">Bestilling af computere</cbc:Note>
  <cac:ValidityPeriod>
            <cbc:StartDate>2008-05-01</cbc:StartDate>
            <cbc:EndDate>2008-05-06</cbc:EndDate>
      </cac:ValidityPeriod>
      <cac:RequestForQuotationDocumentReference>
            <cbc:ID>G867B</cbc:ID>
            <cbc:UUID>93T5G3G5-HYA3-7267-BVG3-GS46SW44WG53</cbc:UUID>
            <cbc:IssueDate>2008-04-19</cbc:IssueDate>
      </cac:RequestForQuotationDocumentReference>
  <cac:SellerSupplierParty>
    <cbc:CustomerAssignedAccountID>LEV00123</cbc:CustomerAssignedAccountID>
    <cac:Party>
      <cbc:EndpointID schemeID="DK:CVR">DK18296799</cbc:EndpointID>
      <cac:PartyIdentification>
        <cbc:ID schemeID="DK:CVR">DK18296799</cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyName>
        <cbc:Name>Delcomputer A/S</cbc:Name>
      </cac:PartyName>
      <cac:PostalAddress>
        <cbc:AddressFormatCode listAgencyID="320"
listID="urn:oioubl:codelist:addressformatcode-
1.1">StructuredDK</cbc:AddressFormatCode>
        <cbc:StreetName>Arne Jacobsens Allé</cbc:StreetName>
        <cbc:BuildingNumber>15</cbc:BuildingNumber>
        <cbc:CityName>København S</cbc:CityName>
        <cbc:PostalZone>2300</cbc:PostalZone>
        <cac:Country>
          <cbc:IdentificationCode>DK</cbc:IdentificationCode>
        </cac:Country>
      </cac:PostalAddress>
      <cac:PartyTaxScheme>
        <cbc:CompanyID schemeID="DK:SE">DK18296799</cbc:CompanyID>
        <cac:TaxScheme>
          <cbc:ID schemeAgencyID="320" schemeID="urn:oioubl:id:taxschemeid-
1.1">63</cbc:ID>
          <cbc:Name>Moms</cbc:Name>
        </cac:TaxScheme>
      </cac:PartyTaxScheme>
      <cac:PartyLegalEntity>
        <cbc:RegistrationName>Delcomputer A/S</cbc:RegistrationName>
```

```
        <cbc:CompanyID schemeID="DK:CVR">18296799</cbc:CompanyID>
      </cac:PartyLegalEntity>
    </cac:Party>
  </cac:SellerSupplierParty>
  <cac:OriginatorCustomerParty>
    <cac:Party>
      <cbc:EndpointID schemeAgencyID="9"
schemeID="GLN">5798000416604</cbc:EndpointID>
      <cac:PartyIdentification>
        <cbc:ID schemeAgencyID="9" schemeID="GLN">5798000416604</cbc:ID>
      </cac:PartyIdentification>
      <cac:PartyName>
        <cbc:Name>Gentofte Kommune</cbc:Name>
      </cac:PartyName>
      <cac:PostalAddress>
        <cbc:AddressFormatCode listAgencyID="320"
listID="urn:oioubl:codelist:addressformatcode-
1.1">StructuredDK</cbc:AddressFormatCode>
        <cbc:StreetName>Bernstorffsvej</cbc:StreetName>
        <cbc:BuildingNumber>161</cbc:BuildingNumber>
        <cbc:CityName>Charlottenlund</cbc:CityName>
        <cbc:PostalZone>2920</cbc:PostalZone>
        <cac:Country>
          <cbc:IdentificationCode>DK</cbc:IdentificationCode>
        </cac:Country>
      </cac:PostalAddress>
      <cac:PartyTaxScheme>
        <cbc:CompanyID schemeID="DK:SE">DK12345678</cbc:CompanyID>
        <cac:TaxScheme>
          <cbc:ID schemeAgencyID="320" schemeID="urn:oioubl:id:taxschemeid-
1.1">63</cbc:ID>
          <cbc:Name>Moms</cbc:Name>
        </cac:TaxScheme>
      </cac:PartyTaxScheme>
      <cac:PartyLegalEntity>
        <cbc:RegistrationName>Gentofte Kommune</cbc:RegistrationName>
        <cbc:CompanyID schemeID="DK:CVR">DK12345678</cbc:CompanyID>
      </cac:PartyLegalEntity>
      <cac:Contact>
        <cbc:ID>12345678</cbc:ID>
        <cbc:Name>Sille Schyberg</cbc:Name>
      </cac:Contact>
    </cac:Party>
  </cac:OriginatorCustomerParty>
  <cac:Delivery>
    <cac:DeliveryAddress>
      <cbc:AddressFormatCode listAgencyID="320"
listID="urn:oioubl:codelist:addressformatcode-
1.1">StructuredDK</cbc:AddressFormatCode>
      <cbc:StreetName>Bernstorffsvej</cbc:StreetName>
      <cbc:BuildingNumber>161</cbc:BuildingNumber>
      <cbc:CityName>Charlottenlund</cbc:CityName>
      <cbc:PostalZone>2920</cbc:PostalZone>
      <cac:AddressLine>
        <cbc:Line>IT-afdelingen</cbc:Line>
      </cac:AddressLine>
      <cac:AddressLine>
          <cbc:Line>1. sal</cbc:Line>
      </cac:AddressLine>
      <cac:Country>
        <cbc:IdentificationCode>DK</cbc:IdentificationCode>
```

```
      </cac:Country>
    </cac:DeliveryAddress>
    <cac:RequestedDeliveryPeriod>
      <cbc:StartDate>2008-05-06</cbc:StartDate>
      <cbc:StartTime>09:30:47.0Z</cbc:StartTime>
      <cbc:EndDate>2008-05-10</cbc:EndDate>
      <cbc:EndTime>09:30:47.0Z</cbc:EndTime>
    </cac:RequestedDeliveryPeriod>
  </cac:Delivery>
  <cac:DeliveryTerms>
    <cbc:SpecialTerms>1% reduktion i kontraktsummen pr. dags forsinkelse
jf. SKI kontrakt</cbc:SpecialTerms>
  </cac:DeliveryTerms>
      <cac:QuotedMonetaryTotal>
    <cbc:LineExtensionAmount currencyID
="DKK">197750.00</cbc:LineExtensionAmount>
    <cbc:TaxExclusiveAmount currencyID
="DKK">49437.50</cbc:TaxExclusiveAmount>
    <cbc:TaxInclusiveAmount currencyID
="DKK">247187.50</cbc:TaxInclusiveAmount>
    <cbc:PayableAmount currencyID ="DKK">247187.50</cbc:PayableAmount>
  </cac:QuotedMonetaryTotal>
  <cac:QuotationLine>
    <cbc:ID>1</cbc:ID>
    <cbc:Note>Computer</cbc:Note>
    <cac:LineItem>
      <cbc:ID>DELL1052665</cbc:ID>
      <cbc:Quantity unitCode="EA">35</cbc:Quantity>
      <cbc:LineExtensionAmount
currencyID="DKK">150500.00</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="DKK">37625.00</cbc:TotalTaxAmount>
      <cac:Price>
        <cbc:PriceAmount currencyID="DKK">4300.00</cbc:PriceAmount>
        <cbc:BaseQuantity unitCode="EA">1</cbc:BaseQuantity>
      </cac:Price>
      <cac:Item>
        <cbc:Description>Stationær computer</cbc:Description>
        <cbc:Name>Dell PrecisionTM  T3400</cbc:Name>
      </cac:Item>
    </cac:LineItem>
  </cac:QuotationLine>
  <cac:QuotationLine>
    <cbc:ID>2</cbc:ID>
    <cbc:Note>Skærm</cbc:Note>
    <cac:LineItem>
      <cbc:ID>DELL2363463</cbc:ID>
      <cbc:Quantity unitCode="EA">35</cbc:Quantity>
      <cbc:LineExtensionAmount
currencyID="DKK">43750.00</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="DKK">10937.50</cbc:TotalTaxAmount>
      <cac:Price>
        <cbc:PriceAmount currencyID="DKK">1250.00</cbc:PriceAmount>
        <cbc:BaseQuantity unitCode="EA">1</cbc:BaseQuantity>
      </cac:Price>
      <cac:Item>
        <cbc:Description>Fladskærm</cbc:Description>
        <cbc:Name>FP/BL 1908WFP</cbc:Name>
      </cac:Item>
    </cac:LineItem>
  </cac:QuotationLine>
  <cac:QuotationLine>
```

```
    <cbc:ID>3</cbc:ID>
    <cbc:Note>Mus</cbc:Note>
    <cac:LineItem>
      <cbc:ID>DELL2367452</cbc:ID>
      <cbc:Quantity unitCode="EA">35</cbc:Quantity>
      <cbc:LineExtensionAmount
currencyID="DKK">1750.00</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="DKK">437.50</cbc:TotalTaxAmount>
      <cac:Price>
        <cbc:PriceAmount currencyID="DKK">50.00</cbc:PriceAmount>
        <cbc:BaseQuantity unitCode="EA">1</cbc:BaseQuantity>
      </cac:Price>
      <cac:Item>
        <cbc:Description>Mus</cbc:Description>
        <cbc:Name>Dell Quietkey USB-tastatur, sort - Dansk
(QWERTY)</cbc:Name>
      </cac:Item>
    </cac:LineItem>
  </cac:QuotationLine>
  <cac:QuotationLine>
    <cbc:ID>4</cbc:ID>
    <cbc:Note>Tastatur</cbc:Note>
    <cac:LineItem>
      <cbc:ID>DELL8436783</cbc:ID>
      <cbc:Quantity unitCode="EA">35</cbc:Quantity>
      <cbc:LineExtensionAmount
currencyID="DKK">1750.00</cbc:LineExtensionAmount>
      <cbc:TotalTaxAmount currencyID="DKK">437.50</cbc:TotalTaxAmount>
      <cac:Price>
        <cbc:PriceAmount currencyID="DKK">50.00</cbc:PriceAmount>
        <cbc:BaseQuantity unitCode="EA">1</cbc:BaseQuantity>
      </cac:Price>
      <cac:Item>
        <cbc:Description>Tastatur</cbc:Description>
        <cbc:Name>Dell Quietkey USB-tastatur, sort - Dansk
(QWERTY)</cbc:Name>
      </cac:Item>
    </cac:LineItem>
  </cac:QuotationLine>
</Quotation>
```

DODATAK J – CD SA IZVORNIM PROGRAMSKIM KODOM GENERATORA, ONTOLOGIJAMA I PROVEDENIM STUDIJAMA SLUČAJA[30]

[30] Alternativno dostupno i putem weba na lokaciji: https://drive.google.com/drive/folders/1vaxUtaCWwLyb0vEvMHOZeBrdBfCsXGWx

ŽIVOTOPIS

Daniel Strmečki rođen je Osijeku 11.05.1989. godine. Rano djetinjstvo proveo je u malom Baranjskom mjestu Čemincu. Početkom Domovinskog rata preselio se s obitelji u Varaždin gdje kreće u osnovnu školu. Krajem rata, zajedno s obitelji vraća se najprije u Osijek, gdje završava dva razreda osnovne škole te potom u Čeminac, gdje i završava osnovnu školu. Nakon toga upisuje Elektrostrojarsku školu u Varaždinu, smjer Računalni tehničar. Srednju školu provodi u Učeničkom domu u Varaždinu te po završetku upisuje preddiplomski studij u istom gradu, na Fakultetu organizacije i informatike, smjer Informacijski sustavi. Na istom fakultetu 2011. godine upisuje i diplomski studij, smjer Informacijsko i programsko inženjerstvo. Preddiplomski i diplomski studij završava sa pohvalama *cum laude*, odnosno *magna cum laude* te dekanovom nagradom za izvrsnost u akademskoj godini 2011/2012. Za vrijeme studija počinje se honorarno baviti programiranjem, odnosno razvojem web i mobilnih aplikacija za tvrtke Hrvatski Telekom d.d., Adonai Studio d.o.o. i Conecto d.o.o. Po završetku studija zapošljava se na poziciji programera u tvrtki Evolva d.o.o. gdje radi na razvoju poslovnih web aplikacija i Java razvojnog okvira do 2016. godine. U međuvremenu, 2014. godine upisuje doktorski studij na Fakultetu organizacije i informatike, smjer Informacijske znanosti i počinje se baviti znanstvenim istraživanjem na području programskog inženjerstva, osobito disciplinom generativnog programiranja. U kolovozu 2016. zapošljava se na poziciji višeg Java programera u tvrtki ecx.io, članici IBM iX grupacije. Na toj poziciji radi i danas, konkretno na razvoju poslovnih web rješenja u *Adobe Experience Manager* razvojnom okviru. Trenutno vodi tim od sedam razvojnih programera za klijenta *H&M*, jedne od najvećih tvrtki za prodaju odjeće u svijetu.

POPIS OBJAVLJENIH ZNANSTVENIH RADOVA

1) Strmečki, Bernik, Radošević: *Gamification in E-Learning: Introducing Gamified Design Elements into E-Learning Systems, Journal of Computer Science,* 2016
2) Strmečki, Magdalenić, Kermek: *An Overview on the use of Ontologies in Software Engineering, Journal of Computer Science, 2016*
3) Bernik, Radošević, Strmečki: *Research on Efficiency of Applying Gamified Design into University's e-Courses: 3D Modeling and Programming, Journal of Computer Science,* 2017
4) Strmečki, Magdalenić, Radošević: *A Systematic Literature Review on the Application of Ontologies in Automatic Programming, International Journal of Software Engineering and Knowledge Engineering,* 2018

SUDJELOVANJE NA ZNANSTVENIM KONFERENCIJAMA

1) Strmečki, Radošević, Magdalenić,: *Web Form Generators Design Model,* CECIIS 2015
2) Novak, Strmečki, Oreški: *Linked Open Data: Realization, Trends and Application Overview,* CEEEGov Days 2016
3) Kermek, Strmečki, Novak, Kaniški: *Preparation of a Hybrid e-Learning Course for Gamification,* MIPRO 2016
4) Strmečki, Magdalenić: *Towards an Ontological Model for Modelling and Automatic Code Generation of Complete Web Information Systems,* CECIIS 2017

SUDJELOVANJE NA STRUČNIM KONFERENCIJAMA

1) Strmečki, Tomašković: *Hybrid mobile apps - Deploy Java web application on Android to run offline*, JavaCRO 2014
    - Nagrada za najbolju prezentaciju u web i mobilnoj kategoriji
2) Strmečki: *Generative Programming: AspectJ and GUI form generators*, JavaCRO 2015
3) Strmečki: *How to develop your own DSLs in Eclipse using Xtend and* Xtext, JavaCRO 2016
4) Strmečki: *Test-drive your Web components development*, JavaCRO 2017
    - Nagrada za najbolju prezentaciju u kategoriji metodologije i alati
5) Strmečki: *A decade of clean code: Tips and tools for modern professional developers*, JavaCRO 2018
    - Nagrada za najboljih pet prezentacija, prosječna ocjena 4,82 od maksimalno 5

POLOŽENI CERTIFIKATI

1) *Oracle Certified Associate, Java SE 8 Programmer*, 2015
2) *Oracle Certified Professional, Java SE 8 Programmer*, 2017
3) *Adobe Certified Expert - Adobe Experience Manager 6 Developer*, 2017
4) *Adobe Certified Expert - AEM Dev/Ops Engineer*, 2018
5) *Adobe Certified Expert - AEM 6 Architect*, 2018

ŽIVOTOPIS MENTORA

Ivan Magdalenić rođen je 17. travnja 1977. godine u Čakovcu. Nakon završetka prirodoslovno-matematičke gimnazije 1995. godine upisao je diplomski studij na Fakultetu elektrotehnike i računarstva, Sveučilište u Zagrebu. Diplomirao je 2000. godine na smjeru Telekomunikacije i informatika. Magistrirao je 2003. godine s temom magistarskog rada "Elektronička razmjena poslovnih dokumenata". Od 2000. do 2004. godine radio je kao znanstveni novak na Fakultetu elektrotehnike i računarstva Sveučilište u Zagrebu. Od 2004. godine radi kao asistent na Fakultetu organizacije i informatike Sveučilište u Zagrebu. Doktorirao je na Fakultetu elektrotehnike i računarstva u znanstvenom polju Računarstvo s temom doktorske disertacije "Dinamičko generiranje ontološki podržanih usluga Weba za dohvat podataka". Izabran je u znanstveno nastavno zvanje docent 2012. godine, a u znanstveno-nastavno zvanje izvanredni profesor 2018. godine.

Područje znanstvenog istraživanja Ivana Magdalenića uključuje automatsko i generativno programiranje, elektroničko poslovanje, Semantički web i druge napredne tehnologije weba. Ivan Magdalenić je autor sveučilišnog udžbenika, objavio je poglavlje u znanstvenoj knjizi te je autor devetnaest radova objavljenih u znanstvenim časopisima te dvadeset i dva rada objavljenih na međunarodnim konferencijama.

Ivan Magdalenić aktivno sudjeluje u znanstvenim i stručnim projektima uvođenja elektroničkog poslovanja u Republici Hrvatskoj. Također je član nacionalnih tijela kojima je zadatak podizanje svijesti o elektroničkom poslovanju i uvođenje elektroničkog poslovanja u gospodarstvo i javnu upravu i to: Član Nacionalnog više dioničkog Foruma za e-račun, Član Nacionalnog vijeća za elektroničko poslovanje, Član Sektorskog odbora za akreditaciju davatelja usluga certificiranja u području primjene e-potpisa pri Hrvatskoj akreditacijskoj agenciji te Član povjerenstva za e-račun i voditelj Tehničkog odbora projekta e-Račun pri Ministarstvo gospodarstva, rada i poduzetništva. Sudjelovao je na nizu stručnih konferencija na tematiku podizanja znanja i popularizaciji elektroničkog poslovanja e-biz i CUC.

Ivan Magdalenić obnašao je dužnost pročelnika Katedre za informatičke tehnologije i računarstvo na Fakultetu organizacije i informatike Sveučilište u Zagrebu u razdoblju od 2013.-2017. Ivan Magdalenić je dobitnik nagrade za mladog znanstvenika na Fakultetu organizacije i informatike Sveučilište u Zagreb 2012. godine.