

Univerza
v Ljubljani

Fakulteta
*za gradbeništvo
in geodezijo*



VLADO STANKOVSKI

Vlado Stankovski

**TEHNOLOGIJE SEMANTIČNE MREŽE
PRI RAČUNALNIŠKO INTEGRIRANI GRADITVI**

DOKTORSKA DISERTACIJA

Ljubljana, 2009

Univerza
v Ljubljani

Fakulteta
*za gradbeništvo
in geodezijo*



*MEDNARODNI PODIPLOMSKI
ŠTUDIJ GRADBENE
INFORMATIKE*

DOKTORSKI ŠTUDIJ

Kandidat:

Mag. VLADO STANKOVSKI, univ. dipl. inž. rač. in inf.

Vlado Stankovski

**TEHNOLOGIJE SEMANTIČNE MREŽE
PRI RAČUNALNIŠKO INTEGRIRANI GRADITVI**

Doktorska disertacija štev.: 192

**SEMANTIC GRID TECHNOLOGIES
IN COMPUTER INTEGRATED CONSTRUCTION**

Doctoral thesis No.: 192

Temo doktorske disertacije je odobril Senat Univerze v Ljubljani na svoji 2. seji dne
15. novembra 2005 in imenoval
mentorja prof.dr. Žigo Turka.

Ljubljana, 8. april 2009

Univerza
v Ljubljani

Fakulteta
*za gradbeništvo
in geodezijo*



Komisijo za oceno ustreznosti teme doktorske disertacije v sestavi

prof.dr. Žiga Turk,
prof.dr. Saša Divjak, UL FRI,
prof.dr. Janez Duhovnik,

Vlado Stankovski

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 19. redni seji dne
20. aprila 2005.

Komisijo za oceno doktorske disertacije v sestavi

prof.dr. Žiga Turk,
prof.dr. Janez Duhovnik,
prof.dr. Danijel Rebolj, UM FG,

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 19. redni seji dne
2. julija 2008.

Komisijo za zagovor doktorske disertacije v sestavi

prof. dr. Bojan Majes, dekan, predsednik
prof.dr. Žiga Turk,
prof.dr. Janez Duhovnik,
prof.dr. Danijel Rebolj, UM FG,

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 26. redni seji dne
25. marca 2009.

Univerza
v Ljubljani

Fakulteta
*za gradbeništvo
in geodezijo*



~~IZJAVA O AVTORSTVU~~ Vlado Stankovski

Podpisani **mag. VLADO STANKOVSKI**, univ. dipl. inž. rač. in inf., izjavljam, da sem avtor doktorske disertacije z naslovom: **“TEHNOLOGIJE SEMANTIČNE MREŽE PRI RAČUNALNIŠKO INTEGRIRANI GRADITVI”**.

V Ljubljani, 8. aprila 2009

.....

(podpis)

Рупрехт

Magister doctissime! Поймите,
перед вами одним из соблазнённых
вашим же учением.
Скажите прямо, что же такое магия:
истина или заблуждение,
наука или нет?

Агриппа

Истинная магия есть наука наук,
объяснение всех тайн, явленное магам
разных веков,
разных стран и народов.

*Dialog iz opere Ognjeni angel Sergeja Prokofjeva
po noveli Valerija Briusova (prevod v prilogi)*

ERRATA ET CORRIGENDA

Stran, vrstica

Namesto

Naj bo

BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK	004.7:624(043.3)
Avtor	Mag. Vlado Stankovski, univ. dipl. inž. rač. in inf.
Mentor	Prof. dr. Žiga Turk, univ. dipl. inž. gradb.
Naslov	Tehnologije semantične mreže pri računalniško integrirani graditvi
Obseg in oprema	217 str., 4 pregl., 21 sl.
Ključne besede	semantika, ontologija, mreža, medobratovalnost

Izvilleček

Pomemben cilj računalništva in informatike pri gradbeniških projektih je omogočiti dinamično sodelovanje med podjetji, izboljšave storitev in izkoristek obstoječih programov, podatkov, informacij in znanj. Gradbeništvo ima glede računalniških aplikacij tudi nekaj specifičnih zahtev, ki izhajajo iz neponovljivosti in obsega posameznih gradbeniških izdelkov, procesov ter krogov sodelujočih. Medmrežne tehnologije so osnova za povezovanje procesov v vseh fazah graditve in vodijo v računalniško integrirano graditev. Računalniška mreža ali preprosto mreža (angl. grid) je storitvena infrastruktura, ki nastaja z namenom, da omogoči neomejeno in lahkotno deljenje široko porazdeljenih, heterogenih virov in na ta način prispeva k reševanju kompleksnih inženirskih problemov. Hipoteza dela je, da mreža lahko postane primerna osnova za računalniško integrirano graditev, če pri razvoju mrežnih tehnologij uporabljamo semantične tehnologije. To so predvsem ontologije, metapodatkovne, informacijske, ontološke mrežne storitve in posredniki virov.

Hipotezo smo preizkusili tako, da smo izdelali ontologijo, ki definira koncept mrežnega vira ter omogoča opisovanje konkretnih virov, ki nastopajo v mrežnem okolju. Izdelali smo tudi arhitekturo in načrt mrežnega sistema in implementirali testno okolje, v katerega smo vključili omenjene mrežne storitve. Testno okolje omogoča izvajanje kompleksnih mrežnih aplikacij, ki imajo obliko delotokov. Pokazali smo, da so ontologija in metapodatki, ki predstavljajo opise mrežnih virov, koristni pri vključevanju, odkrivanju, izbiri in dinamični integraciji virov na mreži. Na ta način smo dosegli vrsto izboljšav glede na obstoječe sisteme: višjo raven abstrakcije pri izdelavi in izvajanju inovativnih in zmogljivih inženirskih aplikacij, večjo prožnost, boljši izkoristek virov in varnost, kar je za dinamično sodelovanje v okviru virtualnih organizacij zelo pomembno.

BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION

UDC	004.7:624(043.3)
Author	Vlado Stankovski, M.Sc.
Mentor	Prof. Žiga Turk, Ph.D.
Title	Semantic Grid Technologies in Computer Integrated Construction
Notes	217 p., 4 tab., 21 fig.
Key words	semantics, grid, construction, interoperability

Abstract

Important goal of computer science in civil engineering projects is to facilitate dynamic collaboration among the companies, improvements of services and reuse of programs, data, information and knowledge. Civil engineering has some specific requirements concerning computer applications, which arise from the irrepeatability and scale of particular civil engineering products, processes and collaborating groups. Internet technologies are basis for linking processes in all construction phases, which leads to computer integrated construction. Computing grid, or shortly grid is a service infrastructure, which is being developed to facilitate infinite and seamless sharing of widely distributed, heterogeneous resources, hence, contributing towards the solution of complex engineering problems. A hypothesis of this work is that the grid can become viable platform for computer integrated construction, if semantic technologies are used for its development, i.e. ontologies and metadata, information, ontology and resource broker grid services.

The hypothesis is tested by developing an ontology that defines the concept of a grid resource to describe specific resources in a grid environment. The aforementioned grid services are included in the design of a grid system, and are developed and deployed in a test bed. The test bed allows for the execution of complex grid applications, which take the form of workflows. It is shown that the ontology and the metadata about grid resources are useful when enabling, discovering, selecting and dynamically integrating resources on the grid. This approach yields several improvements against existing systems: a higher level of abstraction when developing and executing innovative and powerful engineering applications, greater flexibility, resource utilization and security, which is very important for dynamic collaboration within virtual organizations.

ZAHVALA

Za pomoč pri nastajanju doktorske disertacije se zahvaljujem mentorju prof. dr. Žigi Turku. Doc. dr. Matiji Maroltu se zahvaljujem za nasvete glede uporabe strokovnih terminov in mag. Jerneju Trnkoczyju za koristne pripombe v zaključni fazi dela.

Hvala tudi vsem na Katedri za gradbeno informatiko, ki so mi omogočili prijetno delovno ozračje.

Navsezadnje se zahvaljujem svoji družini za vso pozitivno energijo in podporo.

KAZALO VSEBINE

1	UVOD.....	1
1.1	Izzivi kompleksnih inženirskih problemov	1
1.2	Potencial mrežnih in semantičnih tehnologij.....	3
1.3	Hipoteza in cilji	8
1.4	Struktura dela.....	12
2	SEMANTIČNA MREŽA.....	14
2.1	Medmrežje in njegova prihodnost.....	14
2.1.1	<i>Razvojni trendi</i>	<i>16</i>
2.2	Storitveno orientirana arhitektura.....	18
2.3	Mreža	21
2.3.1	<i>Definicija mreže</i>	<i>21</i>
2.3.2	<i>Pogledi na mrežne sisteme.....</i>	<i>24</i>
2.3.3	<i>Mrežne arhitekture</i>	<i>26</i>
2.3.4	<i>Obstoječe mrežno prilagojeno sistemsko programje in storitve.....</i>	<i>34</i>
2.3.5	<i>Tehnologije uporabniških vmesnikov</i>	<i>37</i>
2.3.6	<i>Dosedanji razvoj mrežnih infrastruktur v Sloveniji, Evropi in po svetu</i>	<i>39</i>
2.4	Semantični splet	42
2.4.1	<i>Definicija semantičnega spleta.....</i>	<i>42</i>
2.4.2	<i>Jeziki semantičnega spleta.....</i>	<i>44</i>
2.4.3	<i>O ontologijah.....</i>	<i>46</i>
2.4.4	<i>Orodja in tehnologije za upravljanje z metapodatki.....</i>	<i>48</i>
2.5	Razvoj standardov na področju medmrežnih tehnologij.....	50
2.5.1	<i>Standardizacijske organizacije.....</i>	<i>51</i>
2.6	Druge raziskave na področju semantičnih mrež.....	56
3	E-GRADBENIŠTVO	59
3.1	Grozdi kot oblika povezovanja podjetij	60
3.2	Virtualne organizacije	63
3.3	Problem medobratovalnosti in vloga ontologij	65
3.4	Najnovejše raziskave na področju semantičnih tehnologij v kontekstu gradbeništva	69

3.5	Ontologije na področju gradbeništva	72
3.5.1	Ontologija virtualne organizacije.....	72
3.5.2	Model zgradb IFC	73
3.5.3	IFC-zapis v OWL.....	74
3.5.4	Ontologija gradbene informatike	74
3.5.5	Ontologija spletnih storitev	75
3.6	Potencialna področja uporabe mrežnih sistemov	76
3.6.1	Analize fizikalnih veličin zgradb	76
3.6.2	Seizmična analiza zgradb	77
3.6.3	Sanitarno inženirstvo	78
3.6.4	Problem upravljanja s projektnimi dokumentacijami	78
3.6.5	Združene digitalne knjižnice	80
3.7	Primeri mrežnih sistemov na področju gradbeništva	80
3.7.1	Sistem NEESGrid.....	81
3.7.2	Semantične mrežne storitve in orodja IntelliGrid	82
4	ANALIZA ZAHTEV	84
4.1	Proces definicije zahtev	85
4.2	Primer uporabe	85
4.3	Uporabniške zahteve	87
4.4	Aplikacijske in systemske zahteve	91
4.5	Netehnične zahteve.....	94
4.6	Sinteza zahtev, ki se nanašajo na uporabo semantičnih tehnologij.....	95
5	ONTOLOGIJA ZA SISTEM	97
5.1	Mrežni viri.....	98
5.1.1	Podatki	100
5.1.2	Izvršljivi programi.....	101
5.1.3	Ljudje (uporabniki mrežnega sistema in njihove pravice).....	103
5.2	Vključevanje domenskih konceptualnih modelov	103
6	ARHITEKTURA SISTEMA.....	104
6.1	Plast virov.....	108
6.2	Plast storitev semantične mreže	108
6.2.1	Posrednik virov	108
6.2.2	Storitev integracije informacij.....	109

6.2.3	<i>Podatkovne in metapodatkovne storitve mreže</i>	110
6.2.4	<i>Ontološke storitve</i>	111
6.3	Plast odjemalcev in uporabniških vmesnikov	111
6.4	Virtualna organizacija in realni svet	111
7	PROTOTIP SISTEMA	113
7.1	Implementacija mrežnih storitev.....	113
7.2	Testno mrežno okolje	115
7.2.1	<i>Pravice, varnost in zasebnost v virtualni organizaciji</i>	117
7.3	Uporabniški vmesniki in odjemalci	117
7.4	Enostavno vključevanje obstoječih virov zlasti izvršljivih programov v mrežno okolje	119
7.5	Odkrivanje in dinamična integracija mrežnih virov.....	123
7.6	Vpogled v delovanje sistema	128
8	EVALVACIJA	132
8.1	Prožnost za razvoj novih, inovativnih mrežnih aplikacij	132
8.1.1	<i>Izdelava kompleksnih mrežnih aplikacij</i>	132
8.1.2	<i>Izdelava mrežnih portalov</i>	133
8.2	Upoštevanje pravic uporabnikov	135
8.3	Povečanje zmogljivosti, hitrosti in nadgradljivosti sistema	136
8.4	Nižje- in višjenivojske funkcionalnosti izdelanega prototipa.....	137
8.5	Povzetek lastnosti izdelanega prototipa	138
9	DISKUSIJA	140
9.1	Prednosti in slabosti pristopa v primerjavi z drugimi študijami	140
9.1.1	<i>Uporaba in razvoj mrežnih infrastruktur</i>	142
9.1.2	<i>Ontologija versus ontologije</i>	144
9.1.3	<i>Infrastrukturni elementi semantične mreže in izkoristek virov</i>	147
9.1.4	<i>Tehnologije delotokov in uporabniški vmesniki</i>	148
9.1.5	<i>Prilagajanje obstoječe programske opreme za izvajanje na mreži</i>	149
9.2	Analiza paradigme semantične mreže.....	149
9.3	Dejavniki pri vpeljavi mrežnih sistemov.....	151
9.4	Standardizacija na področju tehnologij semantične mreže	152
9.5	Odprta tehnološka vprašanja in nadaljnje delo	153

10	ZAKLJUČEK	155
11	POVZETEK.....	159
12	SUMMARY	165
VIRI.....		171
Uporabljeni viri.....		171
Spletni viri.....		181
PRILOGI.....		189
Priloga A	Ontologija mrežnih virov	189
Priloga B	Prevod citata na začetni strani.....	217

KAZALO PREGLEDNIC

Preglednica 7-1: Osnovna konfiguracija testnega okolja.....	117
Preglednica 7-2: Algoritem za razvrščanje posrednika mrežnih virov	131
Preglednica 8-1: Pohitritev sistema	137
Preglednica 8-2: Nadgradljivost sistema	137

KAZALO SLIK

Slika 1-1: Vizija souporabe porazdeljenih heterogenih virov.....	8
Slika 2-1: Razvoj računskih in mrežnih tehnologij.....	15
Slika 2-2: Mrežne storitve, ki jih predvideva arhitektura OGSA	30
Slika 3-1: Ogdenov trikotnik. Medobratovalnost med heterogenimi orodji je omogočena s pristopanjem k sporazumni ontologiji.....	69
Slika 5-1: Razmerja med koncepti: ontologija, gradbeniško okolje (realni svet), storitev semantične mreže in mrežni vir	97
Slika 5-2: Koncept izvršljivih programov.....	101
Slika 6-1: Načrt semantične mreže	106
Slika 7-1: Testno mrežno okolje.....	116
Slika 7-2: Uporabniški vmesnik za sestavljanje delotokov	119
Slika 7-3: Splošni podatki o izvršljivem programu	120
Slika 7-4: Metapodatki, povezani z izvajanjem.....	121
Slika 7-5: Opis vhodnih podatkov	121
Slika 7-6: Spremenljivke okolja in druge systemske zahteve.....	122
Slika 7-7: Obrazec za nalaganje izvršljivih programov	123
Slika 7-8: Raziskovalec aplikacij (modul A).....	124
Slika 7-9: Nastavitev opcij mrežne aplikacije (modul C).....	125
Slika 7-10: Metapodatki, povezani s potekom izvajanja (modul D)	128
Slika 7-11: Metapodatki, povezani s provenienco (modul E).....	128
Slika 7-12: Izvajanje multiposlov v mrežnem sistemu	129
Slika 8-1: Mrežna aplikacija s področja sanitarnega inženirstva.....	133
Slika 9-1: Obravnavane semantične in mrežne tehnologije.....	141

OKRAJŠAVE IN SIMBOLI

Kratica	Izpisano	Prevod, pojasnilo
AIM	AOL Instant Messaging	Aplikacija za takojšnje sporočanje podjetja AOL
API	Application Programming Interface	Aplikacijski programski vmesnik, nabor deklaracij funkcij, ki jih omogoča knjižnica ali storitev.
ARPANET	Advanced Research Projects Agency Network	Podatkovno omrežje, zasnovano na protokolnem skladu NCP
B2B	Business-to-Business	Elektronske transakcije med podjetji
BLIS-XML	Building Lifecycle Interoperable Software – XML	Metodologija za zapis modelov EXPRESS v jeziku XML
BPEL	Business Process Execution Language	Jezik za formalno specifikacijo poslovnih procesov in protokolov za poslovno sodelovanje
BPEL4WS	Business Process Execution Language for Web Services	Jezik za formalno specifikacijo poslovnih procesov in protokolov za poslovno sodelovanje z uporabo spletnih storitev
BTP	Baza tehničnih podatkov	Baza tehničnih podatkov
C	C	Programski jezik C
CA	Certificate Authority	Organizacija za overjanje digitalnih potrdil
CAD	Computer Aided Design	Računalniško podprto načrtovanje
CDL	Choreography Description Language	Jezik za opisovanje sporočil, ki si jih izmenjujejo sodelujoče storitve
CGO	Core Grid Ontology	Ontologija mrežnih virov projekta CoreGrid
CIB W78	Conference on Information Technology in Construction	Konferenca o informacijskih tehnologijah v gradbeništvu

Kratica	Izpisano	Prevod, pojasnilo
CIM	Common Information Model	Skupni informacijski model
CORBA	Common Object Request Broker Architecture	Skupna arhitektura posredovanja zahtev po objektih
CRISP-DM	CRoss Industry Standard Process for Data Mining	Specifikacija procesov za podatkovno rudarjenje, ki je neodvisna od platforme in proizvajalca računalniške opreme
CSF	Community Scheduler Framework	Razvrščevalac poslov
DAG	Directed Acyclic Graph	Usmerjeni aciklični graf
DAML-S	DARPA Agent Markup Language	Ontologija spletnih storitev, predhodnik OWL-S
DC	Dublin Core	Shema za opisovanje dokumentov
DCOM	Distributed Component Object Model	Objektni model porazdeljenih komponent
DEISA	Distributed European Infrastructure for Supercomputing Applications	Porazdeljena evropska infrastruktura za superračunalniške aplikacije
DMF	Data Management Forum	Forum za upravljanje s podatki
DMTF	Distributed Management Task Force	Delovna skupina za porazdeljeni menedžment
DNS	Domain Name Server	Imenski strežnik
DTD	Document Type Definition	Jezik za opisovanje dokumentov ali delov dokumentov
EAD	Encoded Archival Description	DTD-shema za opisovanje podatkov
ECPPM	European Conference on Product and Process Modeling	Evropska konferenca za modeliranje produktov in procesov
EDG	European Data Grid	Evropski projekt Data Grid
EDM	EXPRESS Data Manager	Objektno orientirana podatkovna baza, ki omogoča zanesljivo hranjenje in izmenjavo tehničnih podatkov o produktih
EGA	Enterprise Grid Alliance	Združenje za podjetniško mrežo

Kratica	Izpisano	Prevod, pojasnilo
EGEE	Enabling Grids for E-science	Vzpostavitev mreže za e-znanost
ESB	Enterprise Service Bus	Vodilo za podjetniške storitve
ETSI	European Telecommunications Standards Institute	Evropski institut za telekomunikacijske standarde
EXPRESS	EXPRESS	Jezik za modeliranje informacij
FGCS	Future Generation Computer Systems	Časopis Revija s področja tehnologij mreže
FIRE	Future Internet Research and Experimentation	Poročilo skupine strokovnjakov Evropske unije o prihodnosti iInterneta
FMEA	Failure Modes Effects and Criticality Analysis	Efekti Učinki porušitvenih stanj in analiza kritičnosti
FTP	File Transfer Protocol	Internetni protokol za prenos datotek
GASS	Global Access to Secondary Storage	Globalni dostop do sekundarne shrambe, del orodjarne GT4
GEODISE	Grid Enabled Optimisation and Design Search for Engineering	Raziskovalni projekt Optimizacija na mreži in iskanje načrtov
GFD	Grid Final Document	Tip dokumenta, ki ga izdaja organizacija OGF
GGF	Global Grid Forum	Globalni forum za mrežo
GIS	Geografski informacijski sistem Geographic Information System	Geografski informacijski sistem
GLUE	Grid Laboratory Uniform Environment	Shema/ontologija za opisovanje mrežnih infrastruktur
GMA	Grid Monitoring Architecture	Arhitektura za spremljanje in nadzor mreže
GOM	Grid Organizational Memory	Organizacijski spomin mreže, ki so ga razvili v okviru projekta K-WfGrid
GRAM	Globus Resource Allocation and Management	Storitev za dodelitev in upravljanje z viri orodjarne Globus
GridFTP	Grid File Transfer Protocol	Mrežni protokol za prenos datotek

Kratica	Izpisano	Prevod, pojasnilo
GRMS	GridLab Resource Management System	Sistem za upravljanje z viri orodjarne GridLab
GRO	Grid Resource Ontology	Ontologija mrežnih virov, ki smo jo zgradili pri tem delu
GSI	Grid Security Infrastructure	Mrežna varnostna infrastruktura
GT4	Globus Toolkit version 4	Orodjarna Globus verzija 4
HEP	High Energy Physics	Fizika visokih energij
HPC	High Performance Computing	Visoko zmogljivo računalništvo
HTC	Hi Throughput Computing	Visoko prepustno računalništvo
HTML	Hypertext Markup Language	Jezik za označevanje hHipertekstni jezika
HTTP	Hypertext Transport Protocol	Protokol za prenos hiperteksta
IAI	International Alliance for Interoperability	Mednarodno zruženje za medobratovalnost
ICH	Interoperability Clearinghouse	Katalog terminov s področja medobratovalnosti
ICQ	I seek you	Aplikacija za takojšnje sporočanje
IDL	Interface Definition Language	Jezik za opis vmesnikov, na katerem temelji WSDL
IETF	The Internet Engineering Task Force	Standardizacijska organizacija
IFC	Industry Foundation Classes	Temeljni razredi za industrijo
IKPIR	Institut za potresno inženirstvo in računalništvo	
IP	Internet Protocol	Internetni protokol
IPv6	Internet Protocol version 6	IPv6 ima veliko večji naslovni prostor kot njegov predhodnik IPv4.
ISTforCE	Intelligent Services and Tools for Concurrent Engineering	Inteligentne storitve in orodja za sodobno inženirstvo
JDL	Job Description Language	Jezik za opisovanje poslov, ki so ga definirali na projektu Condor

Kratica	Izpisano	Prevod, pojasnilo
JSDL	Job Submission Description Language	Jezik za opisovanje poslov, ki ga razvijajo v okviru OGF
KDD	Knowledge Discovery in Databases	Odkrivanje znanja iz podatkovnih zbirk
KGI	Katedra za gradbeno informatiko	
LAN	Local Area Network	Lokalno omrežje
LCG	LHC Computing Grid	Računska mreža velikega hadronskega trkalnika
LHC	Large Hadron Collider	Veliki hadronski trkalnik
MARC	MAchine Readable Cataloging	Format za zapisovanje in izmenjavo bibliografskih in sorodnih podatkov
MDS	Monitoring and Discovery Services	Storitve za spremljanje/nadzor in odkrivanje
MPI	Message Passing Protocol	Vmesnik za posredovanje sporočil
NCP	Network Control Program	Omrežni kontrolni program
NEES	Network for Earthquake Engineering Simulation	Mreža za simulacije pri potresnem inženirstvu
NEESit	NEES Cyberinfrastructure Center	Center za računalniško infrastrukturo organizacije NEES
NG4J	Named Graphs for Jena	Imenovani grafi za Jeno
OAI	Open Archives Initiative	Iniciativa odprtih arhivov
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting	Protokol za zajemanje metapodatkov iz odprtih arhivov
OASIS	Organization for the Advancement of Structured Information Standards	Organizacija za razvoj strukturnih informacijskih standardov
OGF	Open Grid Forum	Odprti forum za mrežo
OGSA	Open Grid Services Architecture	Odprta arhitektura mrežnih storitev
OGSA-DAI	Open Grid Services Architecture - Data Access and Integration	Arhitektura odprtih mrežnih storitev za dostop do podatkov in integracijo
OGSI	Open Grid Services Infrastructure	Odprta infrastruktura mrežnih storitev

Kratica	Izpisano	Prevod, pojasnilo
OMII	Open Middleware Infrastructure Institute	Institut za odprtokodno prilagojeno sistemsko programje
OSI	Open Systems Interconnection	Osnovni arhitekturni model za komunikacijo med računalniki
OWL	Web Ontology Language	Jezik spletnih ontologij
OWL-G	Web Ontology Language for Grid Services	Ontologija mrežnih storitev
OWL-S	Web Ontology Language for Web Services	Ontologija spletnih storitev
P2P	Peer-to-Peer	Enak z enakim
PIS	Poslovni informacijski sistem	
PKI	Public Key Infrastructure	Infrastruktura javnih ključev
RAM	Random Access Memory	Bralno-pisalni pomnilnik
RBAC	Role Based Access Control	Model kontrole dostopa glede na vlogo
RDF	Resource Description Framework	Jezik za opisovanje spletnih virov, ki omogoča tvorjenje usmerjenih in označenih grafov
RDFS	RDF Schema	Shema RDF
RFC	Request for Comments	Priporočila, ki jih izdaja organizacija IETF
RFT	Reliable File Transfer	Storitev zanesljivega prenosa datotek, WS-različica GridFTP
R-GMA	Relational Grid Monitoring Architecture	Relacijska arhitektura za spremljanje in nadzor mreže
RMI	Remote Method Invocation	Oddaljeni klic metod
RPC	Remote Procedure Call	Oddaljeni klic procedur
RSL	Resource Specification Language	Jezik za specifikacijo virov, ki ga uporablja storitev GRAM
SAML	Security Assertion Markup Language	Jezik za opisovanje varnostne politike

Kratica	Izpisano	Prevod, pojasnilo
SciX	Open, self organising repository for scientific information exchange	Odprta, samoorganizirajoča zbirka za izmenjavo znanstvenih informacij
SHOE	Simple HTML Ontology Extensions	Razširitev jezika HTML, ki omogoča ontološke specifikacije
SMI	Storage Management Initiative	Iniciativa za upravljanje shramb
SNIA	Storage Networking Industry Association	Asocijacija industrijZveza podjetij za omrežno shrambo
SOA	Service Oriented Architecture	Storitveno orientirana arhitektura
SOAP	Simple Object Access Protocol	Osnovni protokol za dostop do objektov
S-OGSA	Semantic Open Grid Services Architecture	Odprta arhitektura semantičnih mrežnih storitev
SOKU	Service Oriented Knowledge Utilities	Storitveno orientirani pripomočki znanja
SQL	Structured Query Language	Strukturirani povpraševalni jezik
STEP	STandard for Exchange of Product	Standard za izmenjavo produktnih modelov
TCP	Transmission Control Protocol	Protokol kontrole prenosa
TCP/IP	Transmision Control Protocol/Internet Protocol	Internetni sklad protokolov
TLS	Transport Layer Security	Standard za zagotavljanje varnosti pri transportu sporočil
UDDI	Universal Description, Discovery and Integration	Univerzalno opisovanje, odkrivanje in integracija spletnih storitev
URI	Uniform Resource Identifier	Uniformni identifikator vira
UTF-8	Unicode Transformation Format – no. 8	Transformacijski format Unicode – št. 8
VOMS	Virtual Organization Membership Service	Sistem za upravljanje pripadnosti virtualni organizaciji
W3C	World Wide Web Consortium	Konzorcij svetovnega spleta

Kratica	Izpisano	Prevod, pojasnilo
WAN	Wide Area Network	Široko-prostrano omrežje
WBEM	Web-Based Enterprise Management	Upravljanje organizacij prek spleta
Web 2.0	Web 2.0	Trend povečanja kreativnosti, deljenja informacij in sodelovanja med uporabniki na svetovnem spletu ter pri načrtovanju spletnih aplikacij
Web 3.0	Web 3.0	Splet prihodnosti, semantični splet
WG	Working group	Delovna skupina
WS	Web Service	Spletna storitev
WSDL	Web Services Description Language	Jezik za opisovanje spletnih storitev
WSDM	Web Service Distributed Management	Standard za upravljanje porazdeljenih spletnih storitev
WSFL	Web Services Flow Language	XML-jezik za opis kompozicij toka spletnih storitev
WS-I	Web Services Interoperability Organization	Organizacija, ki skrbi za medobratovalnost spletnih storitev
WSMF	Web Service Modelling Framework	Okolje za modeliranje spletnih storitev
WSMO	Web Service Modelling Ontology	Ontologija za modeliranje spletnih storitev
WSN	Web Services Notification	Standard za obveščanje spletnih storitev
WSRF	Web Services Resource Framework	Infrastruktura spletnih virov
WWW	World Wide Web	Svetovni splet
XACML	Extensible Access Control Markup Language	Razširljiv jezik za opisovanje kontrole dostopa
XHTML	eXtensible Hipertext Markup Language	HTML, zapisan v sintaksi jezika XML
XML	eXtensible Markup Language	Razširljiv označevalni jezik

SLOVENSKO-ANGLEŠKI SLOVAR MANJ ZNANIH IZRAZOV

Slovenski izraz	Angleški izraz
Delegacija poverilnic, delegacija elektronskega certifikata, s katerim se dokazuje istovetnost uporabnika na določeni ravni zaščite informacijskega sistema	Credentials Delegation
Delotok	Workflow
Dnevnik napak	Error Log
Enak z enakim	Peer to peer
Enotna prijava, enkratni vpis	Single Sign-on
Gruča	Cluster
Informacijske storitve	Information Services
Informacijski protokol	Information Protocol
Informacijski vir – dokument, publikacija, podatkovna zbirka, kjer lahko uporabnik dobi informacijo; prim. vir informacije	Information source, information resource
Internet stvari	Internet of Things
Izjava, RDF-izjava	Statement, RDF statement
Izvajalno okolje	Common Runtime
Izvor	Provenance
Izvršljiv računalniški program	Executable
Kolektivna plast	Collective Layer
Končna točka	Endpoint
Krmiljenje poslov	Job control
Lastniška programska oprema	Proprietary software
Medmrežje, medmrežne tehnologije	Internet, Internet technologies
Medobratovalnost, medsebojna obratovalnost, skupna obratovalnost, združljivost (med sistemi)	Interoperability

Slovenski izraz	Angleški izraz
Mreža, računska mreža	Grid, Computing Grid
Mreže naslednje generacije	Next Generation Grids
Mrežna varnostna infrastruktura	Grid Security Infrastructure
Mrežni posel	Grid Job
Mrežno prilagojeno sistemsko programje	Grid middleware
Nadgradljivost	Scale-up
Nadzorna sled	Monitoring Log
Nadzorne storitve	Monitoring Services
Namizne aplikacije	Standalone applications
Naslavljanje spletnih storitev	WS-Addressing
Oddaljena prisotnost	Telepresence
Oddaljeno nadzorovanje	Telecontrol
Oddaljeno opazovanje	Teleobservation
Orodjarna	Toolkit
Osnovna plast	Fabric layer
Osnovne mrežne storitve	Core Grid Services
Paketni, neinteraktivni posel	Batch Job
Plast virov	Resource Layer
Podajalniki različnih vsebin	Content Syndication Feeds
Podatkovna kopija	Data Replica
Podatkovna mreža	Data Grid
Podatkovne storitve	Data Services
Podatkovni tok	Data Stream
Podatkovno rudarjenje	Data Mining
Pohitritev	Speed-up
Politika	Policy
Ponudniki storitev	Service Providers

Slovenski izraz	Angleški izraz
Posel – v celoti opisana enota dela, ki jo mora opraviti računalnik	Job
Posrednik virov	Resource Broker
Povezovalna plast	Connectivity Layer
Prijavljanje	Log-in
Prilagojeno sistemsko programje	Middleware
Pripomočkovno računalništvo	Utility Computing
Pripravni prenos	Stage-in
Protokol za upravljanje	Management Protocol
Računanje na zahtevo	On Demand Computing
Računska mreža	Computational Grid
Razvrščevalec poslov	Job Scheduler
Semantična mreža	Semantic Grid
Semantični splet	Semantic Web
Sestavljenka – spletna aplikacija v kateri so združeni podatki iz več spletnih virov	Mash-up
Socialno mreženje	Social networking
Sočasno razvrščanje	Co-scheduling
Spletne storitve	Web Services
Storitev integracije informacij	Information Integrator Service
Storitve spremljanja, nadzorne storitve	Monitoring services
Storitve upravljanja izvajanja	Execution Management Services
Storitveno orientirani pripomočki znanja	Service Oriented Knowledge Utilities
Tehnologije mreže	Grid technologies
Testno okolje	Test bed
Upravljalca poslov	Job Manager
Upravljanje izvajanja opravil	Execution Management

Slovenski izraz	Angleški izraz
Upravljanje podatkov	Data Management
Upravljanje virov	Resource Management
Urejevalnik/upravljalac delotokov	Workflow Editor/Manager
Varnostne storitve	Security Services
Varnostno potrdilo	Security Certificate
Vir	Resource
Vir brez stanja	Stateless Resource
Vir s stanjem	Stateful Resource
Vir spletne storitve	WS-Resource
Virtualna organizacija	Virtual Organization
Visoko prepustno računalništvo	High Throughput Computing
Visoko zmogljivo računalništvo	High Performance Computing
Vsebnik	Container
Zajemanje	Harvesting
Zaključni prenos	Stage-out
Zasebnost	Privacy

1 Uvod

1.1 Izzivi kompleksnih inženirskih problemov

Gradbena podjetja lahko na zelo zahtevnem trgu konkurirajo samo, če izboljšajo svojo inovativnost, sposobnost reševanja novih, bolj kompleksnih inženirskih problemov, dinamičnost in navsezadnje prilagodljivost tržnim razmeram. Narava poslovanja v sodobnih delovnih okoljih, podatki in računalniške aplikacije pa pogosto zahtevajo zmožnosti, ki jih konvencionalne informacijsko-komunikacijske infrastrukture ne morejo zagotoviti (Turk, 1992 in 2007; Cerovšek, 2003; Dolenc, Stankovski in Turk, 2005; Dolenc in sod., 2007 in 2008). V nadaljevanju naštevamo nekatere probleme na področju gradbeništva, ki so izziv pri načrtovanju in izdelovanju računalniških aplikacij.

Integracija virov je eden ključnih problemov na področju gradbeništva, ki je postal še bolj izražen z razvojem Interneta (Turk in sod., 2004, a in b). V odprtem informacijskem sistemu gradbenega podjetja nastopajo zelo različni viri, kot so obilica računalniške strojne in programske opreme (npr. računalniški programi za risanje, programi za izračun statike), podatki, informacije in znanje (npr. načrti, ki temeljijo na različnih podatkovnih modelih, znanje o delovnih procesih) ter človeški viri. Pri reševanju konkretnih problemov pa ponavadi potrebujemo le podmnožico natanko določenih virov. Načini, ki bodo omogočali integracijo virov, pa morajo ponujati dodano vrednost, npr. bolj učinkovito ali hitrejšo integracijo. Uporabna primera sta npr. integracija poslovnih procesov ali združevanje in prevzemi podjetij.

V gradbeništvu je pereč tudi problem **medobratovalnosti** (angl. interoperability). Pri različnih fazah delovnih procesov v gradbeništvu se uporablja množica računalniških programov, kot so npr. produkti podjetij SIMULIA (produkti ABAQUS, CAD-Integrated FEA), SOFiSTiK (SofiCAD), Autodesk (AutoCAD), GRAPHISOFT (ArchiCAD) (spletni

viri) ipd. Vhodni in izhodni podatkovni formati pri vseh teh programih so zelo različni in niso nujno medsebojno kompatibilni.

V dinamičnem svetu, v katerem živimo, so ljudje in organizacije, ki sodelujejo pri reševanju gradbeniških problemov, običajno na **geografsko dislociranih lokacijah**, kar zahteva še bolj kompleksne sisteme, ki lahko izrabijo infrastrukturo, ki je porazdeljena prek različnih administrativnih domen in ureja pravice vseh udeležencev pri izrabi raznorodnih virov, dostopnih tudi prek informacijskih in telekomunikacijskih omrežij. Namenska in specializirana strojna in programska oprema je trenutno pogosto povezana z določeno fizično lokacijo. Pri nekaterih industrijskih aplikacijah pa je čedalje bolj izražena tudi zahteva, da se podatki in programi prenesejo na katerokoli izbrano lokacijo na omrežju na prožen, enostaven, učinkovit in varen način ter se tam namestijo, podatki pa obdelajo (Stankovski in sod., 2004, 2008a in 2008b).

Z razvojem obstoječih informacijskih sistemov, ki postopoma postajajo vse bolj kompleksni in sofisticirani, se njihovi uporabniki in menedžerji soočajo tudi s skokovitim naraščanjem **količine obstoječih široko porazdeljenih podatkov**, projektnih dokumentacij in tudi znanja, ki ga potrebujejo v določenih situacijah. Zato je še posebej izpostavljena zahteva za predelavo masivnih, povsod dostopnih podatkov in znanja v informacije, ki so ključne pri vsakodnevnih in strateških odločitvah gradbenih podjetij (Trnkoczy, Turk in Stankovski, 2006). Tovrstne obdelave podatkov pa zahtevajo bolj sofisticirane algoritme, večjo podatkovno prepustnost omrežja, več pomnilnika ter večjo procesorsko moč.

Pojavlja se tudi problem uporabe **visoko strukturiranih podatkov**, ki dobijo pravi pomen oziroma uporabnost v določenem trenutku delovnega procesa, npr. pri izdelavi dela kompleksnih produktov (stavb, mostov, avtocest). Proizvajalca programske opreme za tovrstne aplikacije sta npr. podjetji EPM Technology (EDM) in Data Design System (DDS-CAD) (spletni viri).

Uporaba računalništva in informatike na področju gradbeništva ni omejena samo na aplikacije, ki omogočajo dinamično sodelovanje različnih akterjev gradbenega procesa. Pri

mnogih inženirskih nalogah je nujna uporaba **strogo specializiranih aplikacij**, ki včasih zahtevajo veliko časa za namestitev in poznejšo uporabo. Na določenih področjih gradbeništva je nujno potrebna uporaba **visoko zmogljivih računalniških aplikacij** (Hartmann, 2006). Kot primer takih aplikacij lahko navedemo pogosto uporabljano metodo končnih elementov (Dolenc, 2001), ki omogoča natančne izračune fizikalnih veličin zgradb.

Navsezadnje, pri dinamični integraciji virov so vprašanja **varnosti** in **zaupanja** še posebno pomembna (InteliGrid, spletni vir). Treba pa je tudi zagotoviti načine za hitro dogovarjanje in sklepanje poslov na mreži, ki bodo upoštevali dogovorjeno **kakovost storitev**.

Tako lahko sklenemo, da je delovno okolje podjetij, ki pokrivajo različne faze gradbenega procesa, eno najkompleksnejših, dinamično spreminjajočih se delovnih okolij. Rešitev omenjenih potreb po dinamični integraciji in souporabi heterogenih virov kot tudi potreb po izvajanju visoko zmogljivih računalniških aplikacij so lahko tehnologije mreže. Potencial, ki ga te tehnologije nudijo, podrobneje obravnavamo v naslednjem podpoglavju.

1.2 Potencial mrežnih in semantičnih tehnologij

Ena od poti k povečanju inovativnosti oziroma sposobnosti reševanja novih, bolj kompleksnih inženirskih problemov, dinamičnosti in navsezadnje prilagodljivosti tržnim razmeram vodi prek razvoja novih infrastruktur in njihovega ponujanja podjetnikom, raziskovalcem, skupinam razvijalcev ter zasebnikom. Tako je bilo z razvojem železniške, električne, telekomunikacijske mreže in, navsezadnje, tako je tudi z razvojem Interneta – največjega svetovnega omrežja računalnikov. V današnjih modernih ekonomijah se produktivnost po zaslugi informacijsko-komunikacijskih infrastruktur povečuje za približno polovico, zato so nenadomestljiv substrat družb, ki temeljijo na informacijah in znanju. Računalniške arhitekture, tehnologije in infrastrukture, ki so jih nedavno predlagali in razvili, obljublajo, da bodo odigrale glavno vlogo pri porazdeljenih aplikacijah prihodnosti. To so predvsem: (1) storitveno orientirana arhitektura in spletne storitve, (2) tehnologije mreže ter (3) tehnologije semantičnega spleta. Podrobno jih obravnavamo v drugem poglavju dela.

Storitveno orientirana arhitektura (Service-Oriented Architecture, SOA) je konceptualna arhitektura poslovanja, ki promovira deljenje geografsko porazdeljenih poslovnih funkcij na prožen način. Pri tem so poslovne funkcije predstavljene kot storitve, ki jih uporabniki lahko delijo in ponovno uporabljajo preko medmrežja. Z ekonomskega vidika pa je SOA pomembna smernica za izboljšanje učinkovitosti poslovanja podjetij in njihovega konkurenčnega položaja (Jurič, 2007). V ekonomiji se pogosto pojavljata tudi termina storitveno usmerjeno računalništvo in storitveno usmerjena ekonomija. Vrednost SOA pride do izraza pri procesno usmerjenem pristopu, pri katerem so eksplicitno definirane povezave in sodelovanje storitev v poslovnih procesih. Razvoj storitev in njihova orkestracija v procesnem smislu organizaciji prinese bolj učinkovito in bolj prilagodljivo informacijsko infrastrukturo ter s tem prispeva k lažji in hitrejši realizaciji poslovnih ciljev.

Glavne prednosti sistemov, temelječih na SOA, so rahla povezanost sistemskih komponent, enostavnost in prožnost pri ponovni rabi, nadgradljivost, medobratovalnost ter abstrakcija od nižjenivojskih tehnologij (Plaszczak in Wellner, 2006), ki se doseže z uporabo spletnih in mrežnih storitev (Marks in Bell, 2006). Zaradi omenjenih lastnosti velika večina podjetij danes vidi prihodnost ravno v prehodu njihovih informacijskih sistemov na SOA. Pri tem velja pripomniti, da je še vedno odprto vprašanje, kako uspešno preorientirati obstoječe informacijske sisteme (SOA Institute, spletni vir).

Spletne storitve (angl. Web services) so modularne aplikacije, ki jih lahko objavimo, poiščemo in prikličemo prek spleta. V trenutku, ko je spletna storitev narejena in nameščena na kakšnem spletnem strežniku, jo lahko poiščejo in zaženejo tudi druge aplikacije oziroma storitve. Same spletne storitve opravljajo funkcije, ki so lahko karkoli od preprostih zahtev za seštevanje števil do kompleksnih poslovnih procesov. Ključno pa je to, da imajo spletne storitve natančno definirano funkcionalnost in prikažejo svoj opis na zunanjo zahtevo. V tem trenutku se zdi, da obstaja soglasje, da se bo sodelovanje med oddaljenimi entitetami v prihodnje opravljalo prek spletnih storitev. Zaradi rahle povezanosti pa je vidik medsebojne obratovalnosti pomemben kriterij pri analizi zahtev za prihodnje infrastrukture.

Mednarodni raziskovalni projekti s področja gradbeništva, kot sta npr. projekta ISTforCE in SciX (spletni viri), ki sta potekala na Katedri za gradbeno informatiko (KGI) ter na Inštitutu za konstrukcijsko in potresno inženirstvo in računalništvo (IKPIR), so pripomogli k definiciji smernic nadaljnjega razvoja na področju gradbene informatike. Na primer, članek Katranuschkova, Schererja in Turka (2001) predlaga arhitekturo informacijskega sistema, ki temelji na uporabi spletnih storitev.

Računalniška mreža ali preprosto **mreža** (angl. grid) je storitvena infrastruktura, ki nastaja z namenom, da omogoči neomejeno in lahkotno deljenje široko porazdeljenih, heterogenih virov, na primer: računalnikov, pomnilnikov, aplikacij, podatkov itd. Foster (2002, spletni vir) je formalno definiral mrežo kot paralelni in porazdeljeni sistem, ki omogoča deljenje, izbiro ter združevanje geografsko porazdeljenih in avtonomnih virov, dinamično v času izvajanja aplikacij in v odvisnosti od dostopnosti virov, njihovih zmožnosti, zmogljivosti, cene in drugih zahtev. Dostopnost virov pa lahko izboljša inovativnost in posledično gospodarsko rast, če jih le znamo uporabiti pri reševanju kompleksnih industrijskih in znanstvenih problemov.

Tehnologije mreže temeljijo na odprti arhitekturi mrežnih storitev (Open Grid Services Architecture, OGSA, spletni vir), katere cilj je prispevati k medsebojni obratovanosti heterogenih mrežnih sistemov, tako da lahko različne vrste storitev komunicirajo med seboj in si delijo informacije. OGSA je arhitektura za porazdeljeno računanje in interakcije, ki temelji na principih SOA in definira koncept mrežnih storitev. Mrežne storitve pa so posebne vrste spletnih storitev, na podlagi katerih je mogoče modelirati in upravljati stanja heterogenih mrežnih virov. S pomočjo mrežnih storitev lahko zgradimo mrežno infrastrukturo, ki omogoča izvajanje porazdeljenih aplikacij.

Velik potencial tehnologij mreže so prepoznali v Evropski uniji, Združenih državah Amerike, v Indiji, na Kitajskem, Japonskem, v Avstraliji in tudi drugod po svetu. Pri tem mnogi verjamejo, da bo mrežna infrastruktura dodatno prispevala h globalizaciji trgov in razvitim državam v boju na svetovnih trgih zagotavljala pomembno prednost. Danes pa še vedno gojimo veliko pričakovanj glede mrežne infrastrukture prihodnosti. Želimo si, da bi

tehnologije mreže odprle nove priložnosti za industrijo, delo, študij in raziskave, izboljšale elektronsko poslovanje, povečale socialno vključenost ljudi in imele tudi druge pozitivne socioekonomske vplive na vrsto različnih panog in sektorjev.

Vendar pa veliki porazdeljeni aplikacijski scenariji postavljajo visoke zahteve in ni pričakovati, da bo v kratkem razvito prilagojeno sistemsko programje, ki jim bo vsem kos. Kljub temu obstajajo poskusi v to smer, npr. evropski projekti EGEE, DEISA in NorduGrid (spletni viri), kot tudi tehnološko razvojni infrastrukturni projekti OMII-Europa, HPC-Europa (spletni viri) in njihova pričakovana nadaljevanja. Pri tem pa bi bilo zelo koristno, če bi razvijalci zaradi doseganja medobratovnosti med porajajočimi se infrastrukturami upoštevali nastajajoče standarde na področju spletnih in mrežnih storitev.

Določeni vidiki mreže bodo zahtevali veliko razvojnega časa, da pridemo do ustreznih tehnoloških rešitev. Predvsem so izpostavljeni problemi dinamične integracije virov, dostopnih na mreži, medsebojne obratovnosti mrežnih aplikacij in pripadajočih podatkov, varnost pri deljenju virov, vprašanja zaupanja, provenience pri izvajanju mrežnih aplikacij, kakovost storitev itd. Z nekaterimi od omenjenih odprtih problemov se ukvarjamo tudi v pričujočem delu. Pri tem predpostavljamo, da je zaradi visoke stopnje kompleksnosti, ki vlada v mrežnem okolju, nujna uporaba semantičnih tehnologij, ki se že nekaj časa uvajajo na področju svetovnega spleta. Le-te bodo pomembno prispevale k nemotenemu razvoju inovativnih, prožnih in razširljivih mrežnih sistemov in aplikacij. Tako nam prihajajoče tehnologije in infrastrukture obetajo še veliko možnosti za sproščanje človeške kreativnosti in inovativnosti pri reševanju najbolj zahtevnih znanstvenih in inženirskih problemov.

Tehnologije semantičnega spleta. Semantika je veda, ki preučuje pomen znakov, besed ali drugih oblik predstavnosti ter medsebojne pomenske odnose v nekem sistemu. Berners-Lee, Hendler in Lassila (2001) ter Shadbald, Berners-Lee in Hall (2006) pa so predstavili novo obliko spletnih dokumentov, v kateri je definirana semantika njihove vsebine, in tako dosegli vrsto izboljšav (npr. izboljšano brskanje po spletnih dokumentih). Odkritje so enoznačno poimenovali tehnologije semantičnega spleta.

Pri semantičnem spletu so še zlasti pomembne tako imenovane ontologije. Ontologijo kot filozofsko disciplino, ki skuša bivajoče določiti v njegovi identiteti, sta prva definirala Platon in Aristotel (*ontos* v stari grščini pomeni biti, *logos* pa pomeni beseda). V računalništvu pa se ontologije uporabljajo za definiranje različnih konceptov, na primer različnih tipov entitet in njihovih lastnosti (Staab /ur./ in Studer /ur./, 2004). S pomočjo ontologije lahko enoumno definiramo entitete, kot so različni programi, storitve, naprave, uporabniki, podatkovni viri itd., ki nastopajo na področju, ki ga ontologija opisuje. Poleg enoumne definicije entitet ontologije določajo tudi različne odnose, ki nastopajo med njimi, predpisujejo pa tudi aksiome nad lastnostmi teh entitet, ki morajo biti univerzalno veljavni.

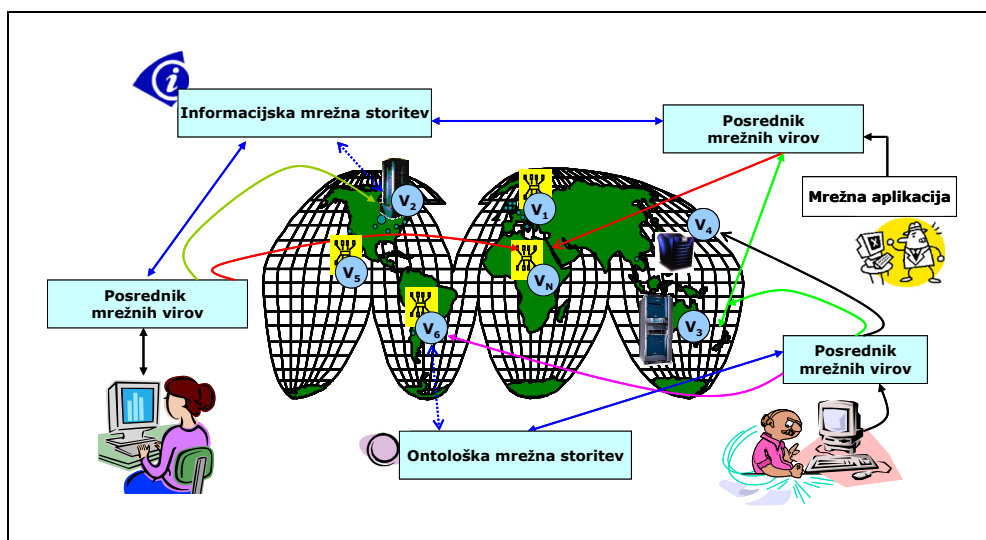
Potreba po dinamični integraciji in medsebojni obratovalnosti heterogenih virov, dostopnih na medmrežju, narekuje uporabo kompleksnih konceptualnih modelov. Ti naj bi omogočali medsebojno razumevanje med različnimi entitetami (udeleženci, programi, storitve, naprave itd.), ki sodelujejo v procesu reševanja kompleksnih problemov. Staab (ur.) in Studer (ur.) (2004) ugotavljata, da uporaba ontologij na semantičnem spletu omogoča:

- povečanje razumevanja okolja in razmerij, ki vladajo med njegovimi deli,
- olajšanje različnih vrst poizvedovanja; poizvedbe lahko opravljajo tako ljudje kot tudi avtomatizirani programi, ki jim pravimo agenti,
- olajšanje interakcije med ljudmi oziroma umetnimi agenti in entitetami,
- zmožnost semantičnega odkrivanja entitet na spletu,
- zmožnost interakcije »novih« entitet (entitet, ki sledijo /pripadajo/ drugim ontologijam) s sistemom na preprost način,
- preverjanje, ali so opisi določenih entitet konsistentni z aksiomi, ki so definirani v ontologiji, in
- zmožnost določanja pravil za kontekstno občutljivo obnašanje različnih entitet s strani ljudi/agentov.

Na gradbenem področju je na primer Turk (2006) predstavil ontologijo gradbene informatike, kar je formalna definicija znanstvenega področja. V tem konkretnem primeru

ontologija omogoča celovit vpogled v področje in tematske sklope, s katerimi se ukvarja gradbena informatika.

Vizija, ki jo delimo z drugimi raziskovalnimi skupinami po svetu, je, da bodo **tehnologije mreže spremenile današnji Internet v globalni superračunalnik, tehnologije semantičnega spleta pa bodo prispevale h konvergenci sistemov, ki temeljijo na visoko strukturiranih informacijah in znanju**. Na sliki 1-1 je prikazana vizija svetovne semantične mreže, ki ima potencial, da omogoči reševanje kompleksnih inženirskih problemov. V mrežni sistem je vgrajena metapodatkovna infrastruktura, ki temelji na sporazumnem konceptualnem modelu (ontologiji) in je neodvisna od posameznih mrežnih aplikacij.



Slika 1-1: Vizija souporabe porazdeljenih heterogenih virov

Fig. 1-1: Vision of sharing of distributed heterogeneous resources

1.3 Hipoteza in cilji

Domnevamo, da lahko ontologija, ki opisuje določeno porazdeljeno in nehomogeno okolje, na primer mrežno okolje, bistveno prispeva k njegovemu nemotenemu delovanju. Za nadaljnji razvoj mreže pa je treba reševati predvsem probleme:

- avtomatskega odkrivanja in združevanja različnih virov pri reševanju kompleksnih problemov,
- medobratovalnosti med različnimi viri in
- kontekstualne »zavednosti« aplikacij.

Hipoteza disertacije je, da je za dinamično integracijo heterogenih in porazdeljenih virov na mreži nujna večja uporaba semantičnih tehnologij, kot so ontologije in mehanizmi – metapodatkovne, informacijske, ontološke mrežne storitve ter posredniki mrežnih virov. To je tisto, kar je treba še razviti, da bi lahko mreža postala primerna osnova za reševanje kompleksnih gradbeniških problemov.

Pri dokazovanju naše hipoteze bomo razvili:

- Ontologijo, ki bo definirala nekatere pomembne vire, ki nastopajo v mrežni infrastrukturi. Na podlagi ontoloških konceptov bo omogočeno opisovanje konkretnih virov, npr. izvršljivih programov, računalnikov itd. Ontologija in opisi virov (metapodatki) skupaj predstavljajo bazo znanja, ki je lahko koncentrirana na določenih strežnikih ali pa porazdeljena v omrežju in je uporabna v najrazličnejše namene.
- Na podlagi principov SOA in OGSA bomo izdelali arhitekturo in načrt mrežnega sistema, ki bo vseboval metapodatkovne, informacijske in ontološke mrežne storitve ter posrednika mrežnih virov. Namen prvih treh mrežnih storitev je, da podpirajo avtomatizirano sklepanje in odločanje o izbiri virov na podlagi obstoječih opisov virov (metapodatkov). Namen posrednika mrežnih virov pa je, da omogoči izvajanje kompleksnih inženirskih aplikacij, ki imajo obliko delotokov. Z uporabo ontologije, metapodatkov o virih ter omenjenih mrežnih storitev želimo doseči, da bo testno mrežno okolje neodvisno od posameznih mrežnih aplikacij.
- Obstoječi izvršljivi programi, podatki, računalniki, računalniške gruče itd. so mrežni viri, ki jih projekti InteliGrid, OntoGrid, K-WfGrid in drugi niso podrobno konceptualno modelirali. Predvsem nas zanima, kako na podlagi ontologije omogočiti preprosto vključevanje obstoječih virov v mrežno okolje, zlasti

vključevanje obstoječih izvršljivih programov in njihovo nadaljnjo uporabo pri izdelavi kompleksnih mrežnih aplikacij.

- Preverili bomo, ali oziroma koliko nam opisi mrežnih virov pomagajo pri semantičnem odkrivanju virov na mreži, njihovem dinamičnem povezovanju, razumevanju kompleksnosti mreže in razmerij, ki vladajo med njenimi deli, lažšanju različnih vrst poizvedovanja, ki jih lahko opravljajo tako ljudje kot tudi mrežne storitve, enostavnosti pri vključevanju obstoječih izvršljivih programov v mrežni sistem, dinamični izdelavi uporabniških vmesnikov, povečanju prožnosti, prilagodljivosti, razširljivosti, zmogljivosti in nadgradljivosti mrežnih aplikacij.
- Izdelana ontologija naj bi omogočala tudi preprosto vključevanje domenskih ontologij, npr. s področja gradbeništva, podatkovnega rudarjenja, varnosti itd., ki se lahko uporabijo za dodatno opisovanje mrežnih virov, določanje omejitev pri njihovi uporabi itd. Danes obstaja že vrsta ontologij, ki pokrivajo različna področja industrijskih in znanstvenih aplikacij. Na primer, konceptualni model s področja gradbeništva, ki se nanaša predvsem na gradbeniške produkte, je znan pod imenom Industry Foundation Classes (IFC, spletni vir) in postaja standardni model za zapis inženirskih informacij, medtem ko je na področju podatkovnega rudarjenja uveljavljen konceptualni model Cross Industry Standard Process for Data Mining (CRISP-DM, spletni vir; Shearer, 2000), ki omogoča zapisovanje procesov podatkovnega rudarjenja neodvisno od platforme in proizvajalca računalniške opreme.
- Naš cilj je pokazati, da testno mrežno okolje, ki temelji na ontologiji mrežnih virov ter na omenjenih mrežnih storitvah omogoča vrsto izboljšav v primerjavi z obstoječimi sistemi: inovativne in zmogljive inženirske aplikacije, višjo stopnjo abstrakcije, večjo prožnost, boljši izkoristek virov in varnost, kar je za dinamično sodelovanje v okviru virtualnih organizacij zelo pomembno.
- Če se bo izkazalo, da je ontologija mrežnih virov koristna pri dinamični integraciji heterogenih virov, imajo semantične tehnologije potencial, da prispevajo k večji konkurenčnosti gradbenih podjetij. Poleg tega je naš splošni cilj, da prispevamo k analizi uporabnosti semantičnih tehnologij pri mrežnih tehnologijah in infrastrukturah.

- Osrednji del študije tvori analiza primerov uporabe na področju gradbeništva. Bolj natančno študija predstavlja podrobno analizo motivacije za to, da predelamo večje število gradbeniških aplikacij za izvajanje v mrežnem okolju. Zanima nas tudi odgovor na vprašanje, kaj vse je potrebno za realizacijo kompleksnih mrežnih aplikacij.
- Pokazali bomo tudi, da je mrežni sistem, ki temelji na ontologiji mrežnih virov, prožen in omogoča razvoj različnih mrežnih aplikacij, npr. izdelali bomo mrežno aplikacijo, ki omogoča iskanje dokumentov iz porazdeljenih podatkovnih zbirk. Take podatkovne zbirke se na področju gradbeništva pogosto pojavljajo. K razvoju tehnologij za odprt dostop do podatkovnih zbirk z namenom omogočanja izmenjave (znanstvenih) informacij je v preteklih letih pomembno prispeval evropski projekt SciX. Tukaj smo imeli za cilj, da dosežke projekta SciX nadgradimo z možnostjo, da metapodatke iz javno dostopnih zbirk preberemo, obdelamo na mreži z naprednimi algoritmi ter rezultate teh obdelav ponudimo uporabnikom v obliki združene metapodatkovne zbirke. Obdelava podatkov iz geografsko porazdeljenih in velikih podatkovnih zbirk je ponavadi računsko zelo zapletena, pri računskih operacijah pa nam lahko pomaga mrežni sistem. Na primer, vanj bi lahko vključili in uporabili programe za avtomatsko učenje ontologij oziroma druge programe, ki omogočajo obdelovanje metapodatkov (Trnkoczy, Turk in Stankovski, 2006).

Z doslednim upoštevanjem odprtokodnih tehnologij in standardov želimo zagotoviti širšo uporabnost naših rezultatov pri različnih prilagojenih sistemskih programjih oziroma pri večjem številu mrežnih okolij, namenjenih reševanju kompleksnih problemov, ter njihov nadaljnji razvoj.

Odprtokodne tehnologije. Semantična mreža naj bi temeljila na uveljavljenih odprtokodnih tehnologijah, kot so Protégé, Jena, Globus Toolkit, Triana, GridBus, CSF in OGSA-DAI (spletni viri). Pri tem je naš namen zmanjšati stroške razvoja in izboljšati medobratovalnost z drugimi aplikacijami.

Uporaba standardov. Odločili smo se, da bomo upoštevali pomembne svetovne standarde, kot so jeziki XML, RDF in OWL, standardno odprto arhitekturo mrežnih storitev OGSA, nastajajoči standard na področju mrežnih storitev WSRF in druge standarde. Ta pristop nam bo omogočil čim širšo uporabnost novih razvitih tehnologij.

1.4 Struktura dela

V uvodnem poglavju smo predstavili problematiko, hipotezo in cilje dela.

V drugem poglavju podrobno obravnavamo medmrežne tehnologije, storitveno orientirano arhitekturo ter trenutno stanje na področju dveh ključnih medmrežnih tehnologij: tehnologij mreže in tehnologij semantičnega spleta. Upoštevanje svetovnih standardov oziroma standardov, ki trenutno nastajajo, in obstoječih odprtokodnih rešitev je princip našega dela, zato podajamo pregled standardizacijskih organizacij in obstoječih protokolov ter skladov tehnologij. V okviru tega poglavja smo pripravili tudi pregled rezultatov pomembnih tekočih in ravnokar zaključenih mednarodnih projektov in drugih raziskav, ki pokrivajo različne vidike semantične mreže.

V tretjem poglavju obravnavamo različne probleme na področju gradbeništva, npr. pri tvorjenju dinamičnih virtualnih organizacij, potresnem in konstrukcijskem inženirstvu, upravljanju s projektno dokumentacijo, pa tudi širše v inženirski stroki, npr. pri ekološkem modeliranju in simulacijah.

Četrto poglavje vsebuje generični primer uporabe ter sintezo uporabniških, aplikacijskih in sistemskih zahtev, ki smo jih upoštevali pri izgradnji ontologije mrežnih virov in sistemskega načrta.

V petem poglavju podajamo ontologijo mrežnih virov, kar je eden ključnih izvirnih prispevkov dela.

Ontologijo mrežnih virov smo upoštevali pri izdelavi arhitekture mrežnega sistema, ki je predstavljena v šestem poglavju. Ključne visokonivojske mrežne storitve, ki smo jih

vgradili v arhitekturo sistema, so posrednik virov, metapodatkovne, informacijske in ontološke mrežne storitve.

V sedmem poglavju predstavljamo delovanje testnega okolja (mrežnega sistema), ki smo ga zgradili z uporabo odprtokodnih rešitev. V testno okolje smo torej vgradili metapodatkovno infrastrukturo, ki temelji na ontologiji mrežnih virov in je neodvisna od posameznih mrežnih aplikacij. Tako smo zgradili sistem, ki temelji na principih semantične mreže.

V osmem poglavju predstavljamo rezultate evalvacije prototipa mrežnega sistema, zlasti glede na obravnavane zahteve, ki so tipične na področju gradbeništva.

Na podlagi dobljenih rezultatov v devetem poglavju diskutiramo o problemih, ki smo jih reševali, načinih reševanja, dosežkih, prednostih in slabostih pristopa ter drugih odprtih vprašanj in možnosti za nadaljnje delo.

Deseto poglavje podaja pregled ključnih izvirnih prispevkov dela in zaključek.

Zaradi obsežne uporabe različnih kratic smo na uvodnih straneh dela dodali preglednico z njihovo razlago in slovensko-angleški slovarček, ki bralcu nazorno prikaže angleške različice slovenskih strokovnih izrazov.

2 Semantična mreža

Namen tega poglavja je podati pregled nad definicijami, koncepti in tehnologijami medmrežja, mreže ter semantičnega spleta. Najnovejši razvoj na področju mrežnih tehnologij kaže na nujno potrebo po posvojitvi tehnologij semantičnega spleta v mrežne sisteme. Skupek teh področij raziskav se pogosto pojavlja pod pojmom semantična mreža (angl. Semantic Grid) (De Roure, Jennings in Shadbolt, 2003; Chen, Finin in Joshi, 2003; Fox, 2003; Johnston, 2004; Cannataro in Talia, 2004; Zhuge, 2004; Turk in sod., 2004, a in b; Turk in sod., 2005). Predlagano je bilo, da je v mrežo treba vgraditi nova semantična orodja, semantične storitve, znanje o virtualnih organizacijah, mrežnih aplikacijah, delotokih, mrežnih infrastrukturah in specifičnih področij aplikacij (npr. gradbeništva, medicine). V okviru tega poglavja smo pripravili tudi pregled rezultatov pomembnih tekočih in ravnokar zaključenih mednarodnih projektov in drugih raziskav, ki pokrivajo različne vidike semantične mreže.

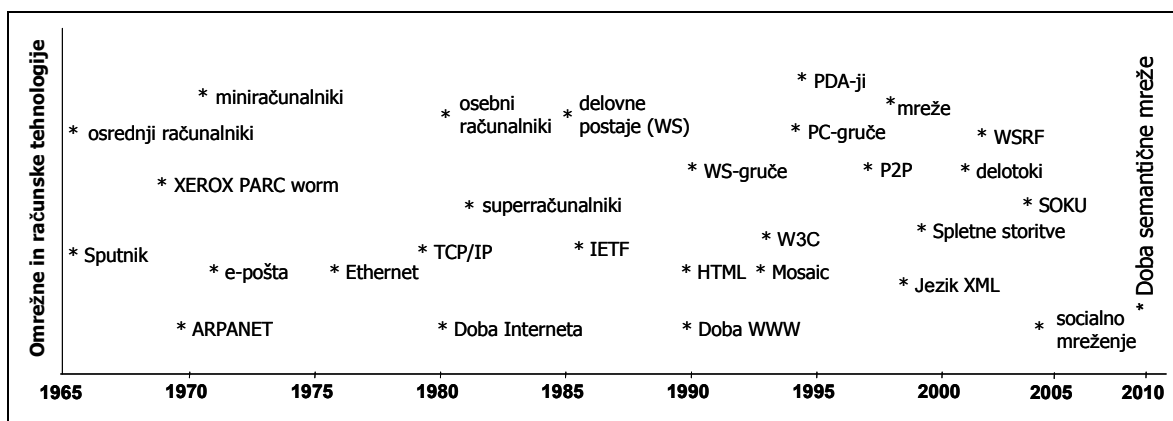
2.1 Medmrežje in njegova prihodnost

Medmrežje oziroma internet (skrajšano iz angleške besedne zveze »inter-network«) je v splošnem smislu računalniško omrežje, ki povezuje več omrežij. Internet je tudi lastno ime za javno razpoložljiv mednarodno povezan sistem računalnikov, vključno z informacijami in uslugami za uporabnike. Sistem uporablja način paketno preklopljivih komunikacijskih protokolov TCP/IP. V razširjenem pomenu, se internet velikokrat nanaša tudi na storitve, kot so svetovni splet (World Wide Web, WWW), elektronska pošta itd.

Internet se še vedno razvija. Njegova velikost in dogajanje na njem sta danes povsem drugačna kot pred desetimi leti in prav gotovo bo v naslednjih letih doživel nadaljnje velike spremembe. Danes je Internet telekomunikacijska in informacijska infrastruktura, brez katere si ni mogoče predstavljati konkurenčnega in trajnostnega razvoja niti sodobnega življenja na splošno – leta 2007 je število internetnih strežnikov preseglo 370 milijonov.

Tehnologije mreže temeljijo na medmrežnih tehnologijah. Pri njihovem razvoju pa istočasno igrajo pomembno vlogo računske in omrežne tehnologije in s tem povezani standardi (slika 2-1).

V zgodovinskem razvoju obstaja nekaj pomembnih trenutkov ali faz: razvoj omrežij na paketno preklapljanje (od leta 1969 naprej), sklad protokolov TCP/IP (l. 1983), svetovni splet (l. 1990), pojav jezika XML (l. 2000). Pojav mrežnih standardov WSRF (l. 2005) pa bi tudi lahko predstavljal enega od pomembnejših mejnikov v razvoju interneta.



Slika 2-1: Razvoj računskih in mrežnih tehnologij

Fig. 2-1: Development of computing and networking technologies

Živimo v obdobju velikega razmaha iskalnih orodij in brskalnikov po spletnih vsebinah. Nadaljnji razvoj omogoča nastajanje programskih jezikov, neodvisnih od platforme, kot je na primer jezik Java. Nekatere ključne tehnologije, ki se še vedno razvijajo, pa temeljijo na pomembnih dosežkih zadnjega desetletja, to so predvsem jezik XML, tehnologije spletnih storitev in najnovejši mrežni standardi WSRF.

Poznavanje teh tehnologij, arhitektur in standardov je nujno potrebno za izgradnjo mrežnih sistemov. Zato jih tukaj podrobneje obravnavamo. Preden se lotimo natančne analize omenjenih tehnologij, pa si v nadaljevanju na kratko oglejmo nekatere nove trende, ki se pojavljajo na medmrežju.

2.1.1 Razvojni trendi

Raziskave in razvoj na področju medmrežnih tehnologij, kot tudi računalništva in informatike na splošno, napredujejo z neustavljivim tempom. V tem trenutku so nekateri ključni tehnološki trendi naslednji: prehod k zelo širokim pasovnim omrežjem in IPv6, povsod navzoče brezžične tehnologije, tehnologije mreže, tehnologije semantičnega spleta, Internet stvari, nove arhitekture in topologije omrežij, storitve, temelječe na spletnih tehnologijah, tehnologije uporabniških vmesnikov, elektronske vsebine, ki jih ustvarjajo posamezniki na mreži, kot tudi socialno mreženje, Web 2.0, 3.0 itd. Po eni strani gre za paleto novih tehnologij in pristopov, po drugi pa včasih za zelo majhne razlike med pojmovanjem modnih terminov ali pa za različne strokovne interpretacije istih terminov. Prihaja tudi do konvergence tehnologij predvsem zaradi nujne potrebe po upoštevanju svetovnih standardov – to pa zaradi zagotavljanja globalne medobratovalnosti infrastruktur. Pri tem je najbolj očitna konvergenca informacijskih in telekomunikacijskih tehnologij, spletnih in mrežnih tehnologij itd.

V Evropi, Ameriki in na drugih kontinentih so v preteklih letih razvili vrsto testnih okolij (angl. test beds). Preglede takih okolij so pripravile tudi skupine izvedencev Evropske komisije (Future for European Grids, spletni vir). Glede na organizacijske oblike obstoječa testna okolja pokrivajo ves spekter od velikih, tesno povezanih infrastruktur na kontinentalni ravni do posameznih storitev majhnih organizacij, od nacionalnih testnih infrastruktur do omrežij tipa enak z enakim (Peer to Peer, P2P). Viri, ki so danes dostopni na Internetu, so zelo različni, zato se v tem trenutku razvijajo nove tehnologije in programske paradigme, ki bi omogočile dinamično povezovanje in uporabo še tako različnih testnih okolij. Pri tem lahko pomagajo tehnologije delotokov, virtualizacija, socialna udeležba pri razvoju aplikacij, socialno mreženje in sodelovanje ter storitveno orientirani pripomočki znanja (Service Oriented Knowledge Utilities, SOKU).

Delotoki. Delotok je zaporedje povezanih storitev, ki se izvaja v primerno orkestriranih korakih (Taylor in sod., 2007). Delotoki imajo obliko različnih domensko-specifičnih grafov, ki predstavljajo informacijske tokove. Delotoki so bistveno bolj učinkovito sredstvo za deljenje znanja, procesov, komunikacije, shrambe ali vsebine kot posamezne storitve, ki

so bolj elementarni gradniki. Obstoječi sistemi, ki omogočajo urejanje, izvajanje in nadzor nad delotoki, kot so Triana, Taverna, Kepler, Pipes (spletni viri) itd., že danes učinkovito povezujejo porazdeljene podatkovne vire, analitična orodja in računske procese. V tem trenutku je v znanstvenih in industrijskih krogih eden ključnih razvojnih trendov odkrivanje in deljenje znanja, ki ga vsebujejo kompleksni delotoki, ter njihovo avtomatizirano sestavljanje glede na uporabniške zahteve. Uporabniki proizvedejo vsak dan več podatkov oziroma digitalnih vsebin, nastaja tudi veliko novih pomožnih storitev (angl. utility services), ki so dostopne na medmrežju, zato sta se zelo povečali kompleksnost in dinamičnost informacijskih tokov. Ključna prednost uporabe delotokov je, da so učinkovit in priljubljen način za abstrakcijo kompleksnosti porazdeljenih podatkovno-računskih virov in pomožnih storitev.

Virtualizacija. Virtualizacija je širok pojem, ki se nanaša na abstrakcijo virov. V medmrežnem svetu je virtualno lahko karkoli: ljudje, računski in podatkovni viri, programi, delotoki, ravno tako pa tudi organizacije. V preteklih letih je bilo vloženo veliko truda v raziskave in razvoj prilagojenih sistemskih programij, ki se nanašajo na deljenje in rabo porazdeljenih računalniških virov v velikem merilu. Spletne storitve omogočajo virtualizacijo raznorodnih računalniških virov, objavljanje in deljenje delotokov pa ima potencial, da virtualizira kompleksne naloge računalniških aplikacij.

Socialna udeležba pri razvoju aplikacij. Obstoječe infrastrukture omogočajo kombiniranje storitev različnih ponudnikov na vrsto različnih načinov. Na žalost pa jih v tem trenutku lahko izkorišča le manjšina specializiranih interesnih skupin, ki znajo uporabljati porazdeljene infrastrukture, nove programske paradigme in obstoječa testna okolja, ki omogočajo reševanje kompleksnih problemov. Velika večina učenih posameznikov, ki imajo velike potrebe po deljenju znanja in virov, sodelovanju in ustvarjanju storitev z dodano vrednostjo, pa nima možnosti, da bi bili vključeni. Obstoječe interesne skupine, kot so gradbeniki, drugi inženirji, medicinci, znanstveniki itd. želijo generirati nove storitve z dodano vrednostjo na podlagi omenjenih pristopov za deljenje porazdeljenih virov, vendar jim manjkajo primerna, uporabniško prijazna sredstva. V zadnjih letih je trend Web 2.0 omogočil evolucijo spleta v ekosistem podatkov in storitev,

kjer so uporabniki istočasno porabniki in ponudniki informacij in storitev. Gre za globalno, porazdeljeno aplikacijsko platformo, ki temelji na miselnosti, da je celotna programska oprema sestavljena samo iz spletnih storitev. Te tehnologije omogočajo, da uporabniki povezujejo pravočasno in samostojno prispevane vsebine in da si med seboj delijo vire z uporabo orodij, kot so na primer Pipes. Tovrstno osvobajanje vsebin in razvoj aplikacij ustvarja živahen socialni učinek in močno pospešuje zmogljivosti aplikacij zaradi učinka mreže interesne skupnosti. Npr. delotoki so, sami po sebi, sestavljenke (mash-ups) in podajalniki različnih vsebin (angl. Content Syndication Feeds).

Socialno mreženje in sodelovanje. Razvoj modelov, ki spodbujajo deljenje virov pri reševanju kompleksnih problemov v okviru specializiranih interesnih skupin, danes predstavlja velik izziv. Vzpon sociospleta (angl. Socio-Web) in sociomreže (angl. Socio-Grid) je močno svarilo, da so ljudje tisti, ki ustvarjajo in delijo znanje in vire ter povzročajo pozitivne učinke mreženja. Danes cvetijo blogi, wikiji, storitve deljenih značk (angl. tags), takojšnje sporočanje (ICQ, AIM itd.), delo in socialno mreženje ljudi, semantično opisovanje razmerij med podatki itd. Z deljenimi značkami je najprej začel del.icio.us, ki jih je poimenoval socialni zaznamki.

Storitveno orientirani pripomočki znanja (Service Oriented Knowledge Utilities, SOKU). To je vizija (FIRE, spletni vir), ki jo predstavljajo povsod navzoče in zelo dinamične semantične storitvene infrastrukture, ki omogočajo virtualizacijo heterogenih in dinamičnih virov, kot so podatki, algoritmi in znanje, ki jih imajo njihovi lastniki, posamezniki (fizične) ali pravne osebe. SOKU zvišuje raven abstrakcije za programske modele ter omogoča dinamično in prožno kreiranje sestavljenih sposobnosti iz porazdeljenih storitev v okviru posameznih interesnih skupin, istočasno z upoštevanjem specifičnih zahtev teh skupin.

2.2 Storitveno orientirana arhitektura

Storitveno orientirana arhitektura (Service Oriented Architecture, SOA) je aplikacijska arhitektura, ki s tehničnega vidika predstavlja množico splošno sprejetih

smernic in postopkov za načrtovanje in implementacijo informacijskih sistemov. SOA temelji na naslednjih principih:

- **Vse funkcije aplikacije so definirane kot storitve.**
- **Vse storitve so neodvisne.** Storitve, ki je priklicana, vrne pričakovani rezultat. Komponenta sistema, ki to storitev kliče, ne dela nikakršnih predpostavk v zvezi z notranjo logiko storitve.
- **Prikličemo lahko natančno določene vmesnike storitve.** Pri tem ni pomembna lokacija vmesnikov (npr. ali so lokalni ali oddaljeni), kateri povezovalni protokol bo uporabljen za njihov priklic oziroma katere infrastrukturne komponente sistema so zahtevane za vzpostavitev povezave.

Z uporabo SOA si podjetja obetajo celovito podporo poslovnim procesom, ki je prilagodljiva in omogoča večjo prožnost poslovanja ter hitrejše odzivanje na spremembe. Tako bi gradbeno podjetje hitreje vzpostavilo povezave z novimi partnerji, ki so za vsak projekt drugi in s tem povečalo svojo konkurenčnost na trgu storitev. Podjetja lahko dosežejo omenjene cilje tako, da prestrukturirajo obstoječe informacijske sisteme in aplikacije. Marks in Bell (2006, str. 159–161) predlagata naslednje korake:

- Izpostavitev funkcionalnosti obstoječih aplikacij v obliki storitev.
- Integracija storitev na storitvenem vodilu organizacije (Enterprise Service Bus, ESB). Tukaj gre predvsem za zadovoljitev zahteve, da so storitve povezljive in večnamensko uporabne.
- Kompozicija storitev v posamezne poslovne procese. V ta namen se najpogosteje uporablja jezik BPEL. Z uporabo BPEL lahko eksplicitno definiramo povezave in sodelovanje storitev v poslovnih procesih ter sodelovanje procesov v poslovnih protokolih. Na tem mestu se SOA neposredno naslanja na poslovanje podjetja, na poslovne procese in na njihovo upravljanje ter izvajanje. Uporaba jezika BPEL odpira možnosti tudi za ponovno uporabo delov procesov oziroma znanja v zvezi s preteklimi procesi. Na ta način bi lahko gradbeno podjetje izboljšala sposobnost za *ad hoc* reševanje gradbeniških problemov (Cerovšek in Katranuschkov, 2006).

Tehnološki gradniki za realizacijo platforme SOA že obstajajo in so dosegli zadostno zrelost za uporabo v velikih, kompleksnih sistemih (Jurič, 2007). Sem uvrščamo tehnologije spletnih storitev, sklad dopolnilnih specifikacij spletnih storitev, storitvena vodila in predvsem jezik BPEL. V kontekstu razvoja tehnologij semantične mreže pa v prihodnosti lahko pričakujemo še dodatno podporo opisu sodelovanj med organizacijami oziroma podporo tvorjenju virtualnih organizacij. Že danes pa je jasno, da vpeljava SOA, razvoj storitev in njihova kompozicija v procesnem smislu organizaciji prinaša bolj učinkovito in bolj prilagodljivo informacijsko infrastrukturo in s tem prispeva k lažji in hitrejši realizaciji poslovnih ciljev. V tem smislu ponuja SOA dejanske in oprijemljive koristi za podjetja. Jurič, Heričko in Rozman (2007) so ugotovili, da je mogoče v obdobju petih let po implementaciji večine (vsaj 70 odstotkov) poslovnih procesov prihraniti vsaj deset odstotkov sredstev, potrebnih za informatiko ter hkrati povečati obseg in kakovost storitev (npr. IT-storitev). V to niso vštete posredne koristi SOA, kot so večja dostopnost informacij, povečana kakovost poslovnih odločitev, boljša informiranost partnerjev ter navsezadnje tudi večja prožnost in povezljivost poslovnih procesov organizacije z drugimi organizacijami v globalnem merilu.

2.2.1.1 Sklad tehnologij spletnih storitev in dopolnilne specifikacije

Sklad tehnologij spletnih storitev podpira razvoj sistemov, temelječih na SOA. Osnovne specifikacije sklada so: protokol za izmenjavo XML-sporočil prek računalniških omrežij Simple Object Access Protocol (SOAP), jezik za opisovanje spletnih storitev Web Service Description Language (WSDL) in register Universal Description, Discovery and Integration (UDDI). Nad temi pa pri mrežnih sistemih igra pomembno vlogo sklad dopolnilnih specifikacij spletnih storitev: WS-Reliable Messaging, WS-Security, WS-Routing, WS-Addressing, WS-Coordination, WS-Transaction, WS-Inspection, WS-Policy in WS-Eventing. Vse omenjene tehnologije so osnova za orkestracijo in koreografijo spletnih storitev v smislu podpore poslovnim procesom. Temu je v naboru tehnologij spletnih storitev namenjen jezik Business Process Execution Language for Web Services (BPEL4WS). BPEL4WS je deklarativni jezik, s pomočjo katerega lahko določimo zaporedje proženja spletnih storitev v smislu opisa izvajanja določenega poslovnega procesa (t. i. orkestracija). Poleg proženj posameznih spletnih storitev omogoča BPEL4WS številne dodatne možnosti, kot so specifikacije zank, pogojnih vejitev, lastnosti, korelacije sporočil,

vzpostavitev povezav itn. BPEL4WS s tem posega na področje upravljanja poslovnih procesov in predstavlja standardiziran jezik za specifikacijo procesov. Procesi, specificirani v BPEL4WS, se lahko izvajajo v orkestracijskih strežnikih: Microsoft BizTalk 2004, Collaxa BPEL Server 2.0, IBM BPWS4J, OpenStorm Service Orchestrator in Vergil VCAB Composer. Tako kaže, da BPEL4WS uživa precejšnjo podporo v industriji.

2.3 Mreža

V nadaljevanju bomo predstavili sodobno definicijo mreže in opisali ključne lastnosti, ki naj bi jih imel vsak moderen mrežni sistem. Opis teh značilnosti bo pripomogel k boljšemu razumevanju pojma mreže kot tudi virov, ki nam jih lahko ponudi mrežni sistem. Sodobni mrežni sistemi temeljijo na storitveno orientirani arhitekturi, tehnologiji spletnih storitev in sorodnih standardov, ki smo jih obravnavali v prejšnjem poglavju.

2.3.1 Definicija mreže

Zelo ambiciozno definicijo mreže sta prva postavila Foster in Kesselman (1997, 1998): **računska mreža** (angl. computational grid) je strojna in programska infrastruktura, ki zagotavlja enostaven dostop do porazdeljenih računalniških zmogljivosti. To definicijo je navdihnila v tistem času prevladujoča uporaba mrežnih sistemov za namene deljenja procesorske moči. Pozneje so z razvojem prilagojenega systemskega programja Globus (Foster, Kesselman in Tuecke, 2001; Foster in sod., 2002; Foster, 2005) to definicijo razširili in zapisali, da tehnologije mreže omogočajo koordinirano, prožno in varno deljenje virov, ki služi reševanju kompleksnih nalog in poteka znotraj dinamično oblikovanega nabora končnih uporabnikov, organizacij in ponudnikov virov, ki mu pravimo virtualna organizacija. Foster in sod. (2002) razumejo virtualno organizacijo kot povezovanje virov in posameznikov ali celih skupin ljudi, s skupnimi cilji in pod določenimi pravili medsebojnega sodelovanja.

Tako definirana mreža naj bi omogočala neposreden dostop do geografsko porazdeljenih računalnikov, programske opreme, podatkov, senzorjev ter drugih naprav in virov, ki so

povezani v porazdeljen sistem. Osnovni pogoji, ki jih mora mrežni sistem izpolnjevati (Foster, 2002, spletni vir), so:

- **Sistem koordinira sredstva, ki niso centralno nadzirana.** Tehnologije mreže poenotijo in koordinirajo vire in uporabnike, ki pripadajo različnim nadzornim domenam in rešujejo probleme varnosti, politike rabe virov, plačila, članstva itd. Če ta pogoj ni izpolnjen, imamo opravka s sistemom z lokalnim upravljanjem, ki ni mrežni sistem.
- **Sistem zagotavlja netrivialno kakovost storitev.** Mrežni protokoli omogočajo uporabo virov na koordiniran način in tako omogočajo različne ravni kakovosti storitev, ki se nanašajo na parametre, kot so odzivni čas, dostopnost, varnost oziroma sočasno razvrščanje (angl. co-scheduling) različnih virov.
- **Sistem uporablja standardne, odprte, splošno namenske protokole in vmesnike.** Mrežni protokoli in vmesniki, ki se nanašajo na osnovne probleme avtentikacije, avtorizacije, odkrivanja virov, dostopa in dodeljevanja virov itd., morajo biti standardizirani in odprti. V nasprotnem primeru imamo opravka s sistemom, ki je omejen na določeno aplikacijo, in ni mrežni sistem. To je izjemno pomembna zahteva, ki omogoča medobratovalnost, razširljivost mrežnih sistemov kot tudi njihovo uporabnost v bolj inovativne namene.

Skupna lastnost različnih mrežnih sistemov je deljenje virov v porazdeljenih, raznolikih in dinamično spreminjajočih se okoljih. Pri tem si vire delijo udeleženci z različnimi stopnjami predhodnega zaupanja, poleg tega pa se število udeležencev s časom spreminja, zato morajo obstajati tudi mehanizmi, ki omogočajo nadzor nad uporabo virov. Med ponudniki virov in uporabniki mora biti natančno določeno, kateri viri in pod kakšnimi pogoji so na voljo določenemu uporabniku, dobro pa mora biti poskrbljeno tudi za varnost.

Zgodovinsko gledano se tehnologije za vzporedno in porazdeljeno računanje razvijajo že nekaj desetletij. Mednje sodijo npr. Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA), Remote Method Invocation (RMI), Remote Procedure Call (RPC) ter Message Passing Protocol (MPI). Te tehnologije

omogočajo izdelavo aplikacij za deljenje geografsko oddaljenih virov, vendar morajo razvijalci vse potrebne višjenivojske funkcionalnosti razviti sami. Pri tem pa je ključno poznavanje tudi notranje zgradbe (koncept, entitete in relacije med oddaljenimi objekti). Naštete tehnologije namreč ponujajo le nizkonivojsko infrastrukturo, ne zagotavljajo pa višjenivojskih storitev in programskih modulov, ki so potrebni pri mrežnih aplikacijah. V večji meri se namreč osredotočajo na izmenjavo podatkov oziroma dialog med oddaljenimi programskimi objekti, komponentami ali funkcijami. Tehnologije te vrste so tako lahko le osnova, na kateri je mogoče zasnovati mrežno prilagojeno sistemsko programje.

Danes mrežno računalništvo pokriva velik problemski prostor različnih računsko, podatkovno, informacijsko in organizacijsko intenzivnih aplikacij. Standardizacija na tem področju še vedno poteka. Zato se tudi pogledi posameznih raziskovalnih, akademskih in industrijskih organizacij še vedno precej razlikujejo. Pogosto se tudi dogaja, da se določene definicije mreže nekoliko prilagodijo z namenom boljše prepoznavnosti. V povezavi z mrežnim računalništvom se pogosto uporabljajo tudi drugi pojmi, kot so avtonomno računalništvo (angl. Autonomic Computing), pripomočkovno računalništvo (angl. Utility Computing), računanje na zahtevo (angl. On Demand Computing) ali tudi adaptivno računalništvo (angl. Adaptive Computing), oblakovno računalništvo (angl. Cloud Computing) itd.

Na področju mreže so znane mrežne infrastrukture EGEE, NorduGrid itd. V okviru evropskih projektov InteliGrid, OntoGrid, Provenance, K-WfGrid, SIMDAT, XtremOS ter DataMiningGrid (spletni viri) pa so obstoječi infrastrukturi dodali nabore dodatnih mrežnih protokolov in storitev. Tako so pokazali velik potencial mrežnih tehnologij pri razvoju zmogljivih in geografsko široko porazdeljenih aplikacij.

Danes obstoječe orodjarne in prilagojena sistemska programja, ki se uporabljajo za izgradnjo mrežnih aplikacij, še ne izpolnjujejo vseh treh navedenih pogojev, torej predstavljajo le boljši ali slabši približek temu, kar naj bi ponujal mrežni sistem prihodnosti. Tudi standardi na tem področju še vedno nastajajo. Izboljšati pa je treba tudi stopnjo zagotavljanja kakovosti storitev v mnogih takih sistemih.

2.3.2 Pogledi na mrežne sisteme

Glede na zahteve posameznih aplikacij oziroma njihovih tehnoloških značilnosti obstaja tudi vrsta različnih pogledov na mrežne sisteme. Nekatere od njih predstavljamo v nadaljevanju.

Računske mreže omogočajo izvajanje mrežnih aplikacij, ki potrebujejo veliko procesorsko moč in/ali veliko delovnega spomina. Pri našem delu smo ugotovili, da so posebej primerne aplikacije tiste, ki omogočajo izvajanje tako imenovanih parametričnih študij pri katerih gre za izvršitev velikega števila med seboj popolnoma neodvisnih poslov. Predvsem gre za aplikacije, ki omogočajo simulacije matematičnih modelov, ki so pogoste na področju gradbeništva.

Podatkovne mreže zahtevajo upravljanje z velikimi količinami geografsko porazdeljenih podatkov. Podatki so v geografsko porazdeljenih podatkovnih bazah (npr. relacijskih ali XML-podatkovnih bazah) in datotečnih sistemih, za hranjenje katerih se uporabljajo različni podatkovni modeli (t. i. sheme). Prenosi podatkov iz raznolikih izvorov in njihovo preoblikovanje v ustrezno zahtevano obliko se izvedejo na avtomatiziran način. Sistemi te vrste skrbijo tudi za izdelavo in upravljanje kopij (replik) podatkov, na ta način pa povečujejo dostopnost do podatkov in zmanjšujejo porabo pasovne širine mreže.

Sodelovalne mreže omogočajo medčloveško in medaplikacijsko interakcijo med različnimi znanstvenimi in poslovnimi procesi (načrtovanje, projektiranje itd.), zagotavljajo različne načine medsebojnega komuniciranja udeležencev, kot so video in govorne konference, deljenje aplikacij, skupna raba podatkov, računskih zmogljivosti itn. Ena ključnih značilnosti sodelovalnih mrež je, da posamezni uporabniki in ponudniki storitev nenehno vstopajo in izstopajo iz procesa sodelovanja, in je zato pri teh sistemih poudarek na zagotavljanju varnostnih funkcij.

Določene mrežne aplikacije sledijo ekonomskemu modelu **dostopa na zahtevo**. To so aplikacije, ki potrebujejo dostop do virov v kratkem časovnem okviru, zato zasebna last virov ni komercialno upravičena. Tako je bolj smiselno najeti vire po potrebi. Tipičen

primer takšne aplikacije je dostop do zelo drage merilne opreme ali pa izredno zmogljivih računalniških virov (npr. superračunalnikov).

Semantične mreže so mrežni sistemi, v katere je vgrajena metapodatkovna infrastruktura, ki temelji na sporazumnih konceptualnih modelih (ontologijah). Tehnologije semantične mreže omogočajo, da postanejo metapodatki prvorazredni viri pri mrežnih sistemih. Tako naj bi semantična mreža resnično povišala raven abstrakcije pri razvoju aplikacij ter podpirala tvorjenje in deljenje rešitev kompleksnih industrijskih in znanstvenih problemov. Uporaba sporazumnega konceptualnega modela oziroma ontologije pri mrežnem sistemu naj bi omogočalo zbiranje in izkoriščanje metapodatkov z namenom doseganja večje prožnosti, varnosti, zaupanja, sledenja provenience, priporočila pri uporabi konkretnih procesov in rešitev, dinamični izdelavi uporabniških vmesnikov itn.

Na tem mestu je treba omeniti, da obstoječih mrežnih sistemov običajno ne moremo strogo razvrstiti v eno od naštetih kategorij. Tako vsak mrežni sistem, namenjen računsko zahtevnim aplikacijam, potrebuje tudi podatkovne storitve, ki skrbijo za prenos vhodnih in izhodnih podatkov, ljudi, ki sodelujejo pri reševanju kompleksnih problemov, izvršljivih računalniških programov in orodij ter ustreznih programskih knjižnic itd. Ostre ločnice med posameznimi skupinami torej ni mogoče potegniti.

Ključni parameter, ki vpliva na uporabnost računalniške mreže pa je komunikacija med računskimi posli. Če posamezni posli potrebujejo veliko medsebojne komunikacije in sinhronizacije, lahko postane mrežni sistem precej zamuden, celo neuporaben. Ta vrsta aplikacij je najmanj primerna za izvajanje v mrežnem sistemu, saj mora biti komunikacija med posamičnimi posli zanesljiva in izredno hitra, kar pa je v geografsko široko razpršenih sistemih težko zagotoviti. Tako lahko sklenemo, da niso vse vrste računsko in prostorsko intenzivnih aplikacij enako primerne za izvajanje v mrežnem okolju. V tem trenutku se s temi vprašanji ukvarjajo v okviru evropskega projekta QoSGrid (spletni vir), ki primerja kakovost storitev, ki jih zagotavljajo mrežni sistemi s konvencionalnimi superračunalniki.

2.3.3 Mrežne arhitekture

2.3.3.1 Načela zasnove arhitektur mrežnih sistemov

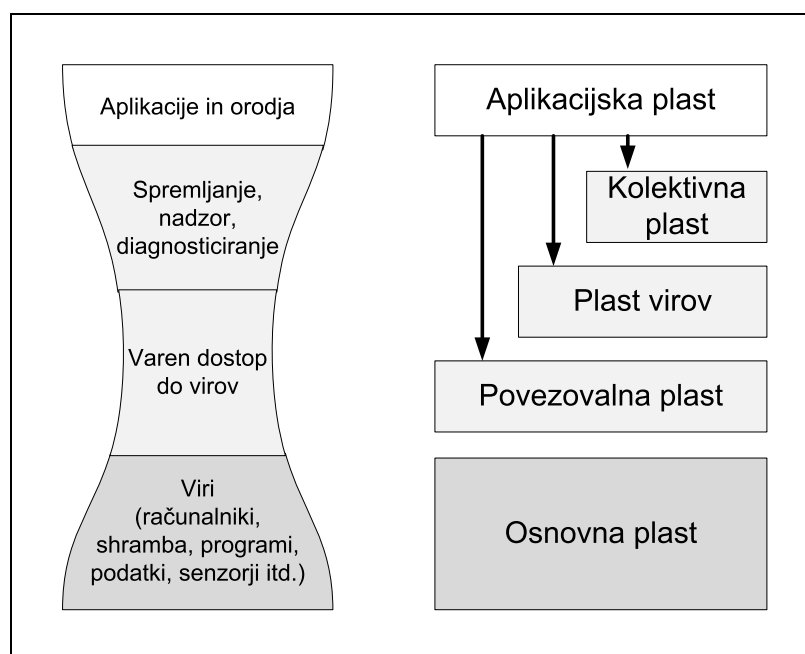
Arhitektura mrežnih sistemov je podobna arhitekturi sklada internetnih protokolov TCP/IP (oziroma modela Open Systems Interconnection /OSI/) in je predstavljena z določenim številom plasti. Razlog in osnovni cilj plastenja je, da zagotavlja strukturiranje sorodnih problemov, ki predstavljajo funkcionalno vsebino posamezne plasti. Tako za vsako plast arhitekture celovitega mrežnega sistema veljajo določene splošne lastnosti. Včasih govorimo tudi o funkcionalni homogenosti plasti, kar pomeni, da plast združuje sorodne storitve.

SOA je predvsem konceptualna arhitektura poslovanja, ki definira povezave in sodelovanje storitev v poslovnih procesih in se nanaša na najvišjo plast v arhitekturi mrežnih sistemov. Toda ker je model OSI prav tako storitveno naravnani, je mogoče na podlagi principov SOA modelirati tudi nižjenivojske plasti (glej sliko spodaj).

Naštejmo nekaj načel, ki formalno opredeljujejo plasti:

- Plast vzpostavimo, ko identificiramo skupino storitev, ki zahtevajo specifično obravnavo.
- Vsaka plast mora pokrivati dobro opredeljene storitve, kar pomeni, da mora biti funkcionalnost plasti zelo natančno opredeljena.
- Meje plasti so opredeljene tako, da se minimalizira pretok informacij prek vmesnika storitvene pristopne točke, to je med sosednjimi plastmi.
- Število plasti mora biti dovolj veliko, da lahko smiselno razvrščamo sorodne funkcije v določeno plast, hkrati pa dovolj majhno, da arhitektura komunikacijskega sistema ostane pregledna.
- Funkcionalnost posamezne plasti oziroma postopek opredeljevanja posameznih storitev upošteva dejstvo, da se predlog plasti vključi tudi v mednarodne standardizacijske aktivnosti.
- Komponente določene plasti imajo tako skupne karakteristike, uporabljajo pa storitve, ki jih nudijo spodaj ležeče plasti.

Foster, Kesselman in Tuecke (2001) so opredelili nekaj funkcionalnih celot mrežnih sistemov, ki jih lahko obravnavamo kot plasti (slika spodaj), in sicer: osnovna plast, povezovalna plast, plast virov, kolektivna plast in aplikacijska plast. Do njih so prišli tako, da so poskušali vgraditi vse zahtevane funkcionalnosti mrežnega sistema. Zato vsaka plast na sliki predstavlja komponente s podobno funkcionalnostjo. Arhitektura plasti sledi modelu peščene ure, v katerem vrat ure predstavlja relativno majhen nabor komponent iz plasti virov in povezovalne plasti. Protokoli teh dveh plasti so načrtovani tako, da jih lahko uporabimo pri velikem številu različnih tipov virov in nad njimi ustvarimo širok nabor višjenivojskih mrežnih storitev ter aplikacij.



Slika: Razplastena arhitektura mrežnega sistema (Foster, Kesselman in Tuecke, 2001)

Fig.: Layered architecture of a grid system (Foster, Kesselman, Tuecke, 2001)

V nadaljevanju se bomo najprej posvetili vprašanju, kako se pretakajo podatki oziroma informacije med sosednjima plastema, šele nato nas bo zanimala tudi vsebina posameznih mrežnih storitev ter njihove realizacije (npr. posrednik virov, informacijska, metapodatkovna, ontološka storitev). Posamezne plasti arhitekture mrežnega sistema zagotavljajo naslednje funkcionalnosti.

Osnovna plast (angl. Fabric Layer) zagotavlja lokalni dostop do virov. Običajno jo predstavlja vmesnik do lokalnega nadzornega sistema vira (npr. datotečnega sistema). Plastem, ki so nad njo, zagotavlja lokalni dostop do virov. Komponente osnovne plasti implementirajo lokalne, za vir specifične operacije, ki potekajo na samem viru. Te ponavadi zajemajo mehanizme za povpraševanje po lastnostih vira (trenutno stanje vira, zmožnosti vira itd.) in mehanizme za dostop do vira. Večinoma te funkcionalnosti podpirajo že viri sami (npr. operacijski sistem vira), v nekaterih primerih pa jih mora zagotoviti mrežno prilagojeno sistemsko programje.

Povezovalna plast (angl. Connectivity Layer) omogoča enostavno in varno komunikacijo med viri. Plast definira osnovne komunikacijske in avtentikacijske protokole. Komunikacijski protokoli omogočajo izmenjavo podatkov in sporočil med viri ter vključujejo TCP/IP-protokole za transport in usmerjanje. Avtentikacijski protokoli zagotavljajo kriptografsko varne mehanizme za preverjanje identitete uporabnikov in virov. Zagotavljajo funkcionalnosti, kot so enkratno prijavljanje (angl. Single Sign-on), delegacija uporabniških pravic, integracija z lokalnimi varnostnimi rešitvami, mehanizmi za zavarovanje prometa, avtorizacijske odločitve itd.

Plast virov (angl. Resource Layer) uporabnikom omogoča pravično dodeljevanje vira. Definira mehanizme in operacije za pogajanja, vzpostavitev, nadzor ter zaračunavanje, ki se izvajajo lokalno na viru. Protokoli plasti virov se nanašajo na posamezen vir in se ne ukvarjajo z upravljanjem virov na ravni celotnega mrežnega sistema. V to plast sta vključeni dve skupini protokolov: informacijski protokoli (angl. Information Protocols), ki se uporabljajo za pridobivanje informacij o stanju vira (npr. konfiguraciji, trenutni obremenjenosti itd.), in upravljalni protokoli (angl. Management Protocols), prek katerih se viru poda seznam operacij, ki naj jih izvrši skupaj z njihovimi zahtevami.

Kolektivna plast (angl. Collective Layer) je plast, v kateri so mehanizmi za koordinacijo virov na ravni celotnega mrežnega sistema. Predstavlja storitve, kot so na primer metapodatkovne in ontološke storitve, informacijske/direktojske storitve, storitve razvrščanja poslov oziroma posredniki virov, storitve za nadzor in diagnosticiranje, storitve

za manipulacijo podatkovnih kopij, storitve upravljanja obremenitve, storitve zaračunavanja itd.

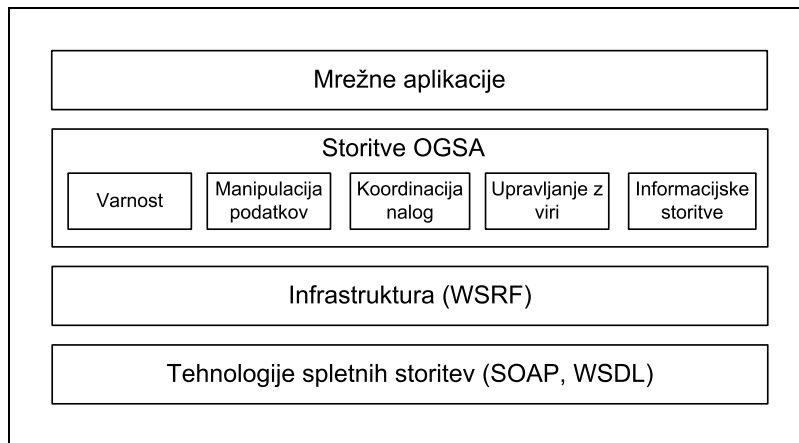
Aplikacijska plast (angl. Applications Layer) je najvišja plast v arhitekturi. Sestavljajo jo uporabniške aplikacije, ki se izvajajo v mrežnem sistemu.

Opisana razplastena mrežna arhitektura je zelo splošna in v nobenem pogledu ne predpisuje tehnologij in vmesnikov komponent, ki se uporabljajo v mrežnem sistemu. Implementacije posameznih komponent plasti so tako lahko zelo različne, kar pomeni, da se s samo definicijo razplastene arhitekture še ne doseže medobratovalnost. Za ta namen je treba predpisati standardne funkcionalnosti in njihove vmesnike, včasih pa je treba določiti tudi nizkonivojske tehnologije, na podlagi katerih so te funkcionalnosti implementirane. Za to pa je bila razvita odprta arhitektura mrežnih storitev, ki je opisana v nadaljevanju.

2.3.3.2 Odprta arhitektura mrežnih storitev

Odprta arhitektura mrežnih storitev (Open Grid Services Architecture, OGSA) je arhitektura za porazdeljeno računanje in interakcije, ki temelji na konceptu mrežnih storitev. OGSA omogoča medobratovalnost heterogenih sistemov, saj lahko različni tipi virov komunicirajo in izmenjujejo informacije. Za izgradnjo sistemov, temelječih na OGSA, so zahtevane storitve, ki imajo stanje. Primer takih storitev so spletne storitve s stanjem (angl. Stateful Web Services).

Ključne storitve, ki jih predvideva OGSA, so prikazane na sliki 2-2. Arhitektura OGSA pa je še vedno v razvoju. Z njo so se v preteklih letih ukvarjali tudi številni raziskovalni projekti (npr. InteliGrid in OntoGrid), ki so poleg obstoječih predlagali tudi dodatne, generične funkcionalnosti. Na primer, projekt OntoGrid je razvil ontološke storitve in predlagal arhitekturo S-OGSA, medtem ko je projekt InteliGrid razvil ontološke storitve in jih uporabil pri izgradnji inženirskih virtualnih organizacij.



Slika 2-2: Mrežne storitve, ki jih predvideva arhitektura OGSA

Fig. 2-2: Grid services according to the OGSA

Varnost. Pomembna lastnost mrežnih sistemov je, da se razprostirajo prek večjega števila administrativnih domen, vsaka od njih pa ima svojo politiko zagotavljanja varnosti. Zato mora mrežni sistem poleg dogovorjene politike virtualne organizacije (medorganizacijski poslovni dogovor, konzorcijska pogodba) upoštevati tudi politike posameznih administrativnih domen. Tako storitve za zagotavljanje varnosti omogočajo izvajanje z varnostjo povezanih politik v virtualni organizaciji. Varnostne politike so lahko različne, predpisujejo pa npr. mehanizme za zagotavljanje integritete sporočil, zaupnosti, avtentikacije sodelujočih entitet, varnega prijavljanja, detekcije vdorov, izolacije aplikacij itd. Namen varnostnih storitev OGSA je torej integracija in poenotenje pogosto uporabljenih varnostnih modelov, mehanizmov in protokolov, z namenom zagotavljanja varnega sodelovanja med različnimi sistemi, kar je ključnega pomena za poslovne procese. To se doseže npr. z delegacijo poverilnic (angl. Credentials Delegation). Delegacija poverilnic omogoča enkratno prijavljanje na mreži, tako neka storitev deluje v imenu drugih uporabnikov in/ali storitev.

Manipulacija podatkov. Podatkovne storitve (angl. Data Services) se nanašajo na upravljanje, dostop, prenos, začasno shranjevanje in repliciranje podatkov, povpraševanje po podatkih, posodabljanje podatkovnih virov itd. Podatkovne storitve lahko skrbijo tudi za federacijo porazdeljenih podatkov, tako da jih npr. pred dostopom preoblikujejo v skupni format in potem še združijo. Poleg tega zagotavljajo določene mehanizme za upravljanje z

metapodatki, ki se uporabljajo za lažje iskanje podatkov. Mrežni sistem običajno vsebuje veliko število raznorodnih podatkovnih virov, ki morajo biti podprti s podatkovnimi storitvami. Primeri so datotečni sistemi različnih tipov, podatkovni tokovi (angl. Data Streams), različne podatkovne baze (relacijske, XML in objektne), katalogi in podobno. Velika raznolikost podatkovnih virov pomeni tudi veliko število različnih podatkovnih storitev.

Koordinacija nalog in upravljanje z viri. Posrednik mrežnih virov (angl. Resource Broker, Meta-Scheduler) je definiran kot entiteta, ki predstavlja most med ponudbo in povpraševanjem po mrežnih virih (npr. Venugopal, Buyya in Winton, 2004). Z drugimi besedami, gre za načrtovanje in krmiljenje (mrežnih) poslov (angl. jobs), ki zahtevajo tudi fizičen prenos virov (npr. podatkov, knjižnic, programov) po omrežju. Posrednike mrežnih virov razvijamo s ciljem avtomatizacije naslednjih operacij:

- **Ujemanje (angl. matching) ponudbe in povpraševanja med mrežnimi viri.** Vsak računalniški posel lahko zahteva vrsto mrežnih virov, kot so pomnilnik, prostor na mrežnih napravah za shranjevanje, izvršljivi programi in drugi viri, ki jih potrebuje pri izvajanju. Posrednik virov mora poiskati take vire, ki se ujemajo z zahtevami računalniškega posla – to pa lahko stori le, če sta ponudba in povpraševanje predstavljena na uniformen način.
- **Razvrščanje poslov.** Ko obstaja večje število zahtev po mrežnih virih ali večja ponudba ustreznih virov, mora posrednik virov uporabiti različne politike razvrščanja (angl. scheduling policies). Njihov cilj je povečati iztržek pri izvajanju določenih funkcij. Kot primer navajamo minimizacijo latence pri izvajanju določenega posla pri različnih ponudnikih virov, ki je lahko izražena denarno.
- **Pripravni prenos (angl. stage-in) – dostava podatkov in programov na mesto izvajanja.** Preden se začne izvajanje določenega posla, mora posrednik virov ponavadi dostaviti podatke, programe ter zahtevane knjižnice na vse tiste računalnike, ki jih je prej določil. Posrednik virov orkestrira celoten proces izvajanja posla, pri katerem se običajno razume, da mora biti zaključen po sistemu »vse ali nič«. Šele ko je prenos uspešno končan, lahko posrednik virov začne izvajati posel.

- **Spremljanje in nadzor izvajanja poslov.** Običajno se stanje posla med izvajanjem spreminja. Računalniški posel ima lahko različna stanja, npr. lahko je v čakalni vrsti (angl. pending), aktiven (angl. active), končan (angl. completed), neuspešen (angl. failed) itn. Spremembe stanja morajo biti uporabniku predstavljene na uporabniško prijazen način, ki omogoča tudi opravljanje ustreznih popravkov, npr. ponovitev posla (poslov) ali preklic izvajanja.
- **Zaključni prenos (angl. stage-out) – dostava rezultatov uporabniku.** V mrežnem okolju je povsem običajno, da so računalniki, kjer dejansko poteka izvajanje računalniških poslov, locirani zunaj administrativne domene tiste osebe, ki posle dejansko izvaja. Zato je pomembno, da se po končanem izvajanju rezultati prenesejo na pomnilniško napravo, ki zagotavlja trajno hranjenje podatkov (angl. persistent storage) in se hkrati izbrišejo vse sledi izvajanja na računalnikih zunaj domače administrativne domene. Ta postopek vključuje brisanje vseh podatkov, programov ter spremljajočih knjižnic, ki so bili preneseni v fazi pripravnega prenosa.

Mrežne aplikacije, ki so usmerjene k reševanju določenih inženirskih problemov, se lahko razlikujejo od običajnih mrežnih aplikacij, ki se ukvarjajo z upravljanjem podatkov ali z navadnim procesiranjem. Pogosto gre tudi za večje zbirke podatkov oziroma dokumentov (npr. projektne dokumentacije), ki jih ne moremo prenašati po omrežju, in moramo zato omogočiti, da namesto podatkov na mesto obdelave prenesemo programe. Včasih se tudi zgodi, da podatki zaradi večje varnosti (npr. strah pred konkurenco) ne smejo prestopiti meje administrativne domene organizacije.

Informacijske storitve. Informacijske storitve nudijo dinamično informacijo o trenutnem stanju virov in storitev v mrežnem sistemu. Mreža predstavlja dinamično okolje, v katerem ni mogoče vzpostaviti statičnega registra, ki bi vseboval vse potrebne informacije o stanju sistema. Na spremembe v okolju se je torej treba odzivati dinamično.

Tipičen primer uporabe teh storitev je uporaba v namene odkrivanja ustreznih virov, ki se trenutno nahajajo na mreži, ali pa spremljanje in nadzorovanje pravilnega delovanja

sistema. Odjemalci teh storitev so običajno druge storitve, na primer: storitve, ki omogočajo iskanje ustreznih virov, storitve zaračunavanja, storitve za rezervacijo virov itd.

Arhitektura OGSA ne predpisuje podatkovnega modela teh storitev niti jezika za povpraševanje po informacijah. Informacija je tako lahko shranjena v različnih oblikah. Najpogosteje se za shranjevanje uporabljajo bolj ali manj centralizirani registri (različne podatkovne baze), podatki pa se osvežujejo z objavno/prijavnimi mehanizmi. Storitve, ki temeljijo na uporabi registra, lahko v osnovi razdelimo na tiste, ki uporabljajo XML in XPath/XQuery (npr. MDS4), in tiste, ki uporabljajo relacijski model podatkovne baze (npr. R-GMA, ki so ga predstavili Cooke in sod., 2003). Alternativa registrom je razpršeno povpraševanje po porazdeljeni informaciji o stanju sistema.

Informacijske storitve imajo v mrežnih sistemih ključno vlogo, saj jih uporablja večina drugih storitev v sistemu. Pri njihovi implementaciji je torej treba zagotoviti visoko dostopnost, toleranco do delnih odpovedi in hiter odziv.

2.3.3.3 Sklad specifikacij Web Services Resource Framework

Web Services Resource Framework (WSRF) je skupek specifikacij, ki podpirajo SOA in OGSA ter predlagajo standarden način za pridruževanje mrežnih virov k spletnim storitvam. WSRF predstavlja poskus sprave med tehnologijami spletnih storitev in nekoliko zastarelimi mrežnimi tehnologijami. V osnovi WSRF definira načine modeliranja in upravljanja s stanjem spletne storitve. Zato nanj lahko gledamo kot na novo različico odprte infrastrukture mrežnih storitev (Open Grid Service Infrastructure, OGSi) verzija 1.0, ki je bila implementirana v prilagojenem sistemskem programju Globus Toolkit 3.x. Glavne novosti, ki jih prinaša WSRF v primerjavi s specifikacijo OGSi, vključujejo uporabo standardnih XML- in WSDL-shem, opustitev modela tovarna/primerek (angl. factory/instance model) in eksplicitno ločitev med storitvijo in entitetami stanja, ki jih lahko storitev uporablja in spreminja med delovanjem. Storitve, ki smo jih zgradili oziroma uporabili pri našem delu, v celoti upoštevajo standard WSRF. Specifikacije WSRF verzija 1.2 so bile sprejete za standard pri organizaciji OASIS leta 2006. Sklad specifikacij WSRF vsebuje naslednje specifikacije: WS-ResourceProperties, WS-ResourceLifetime, WS-ServiceGroup ter WS-BaseFaults.

2.3.3.4 Specifikacije in tehnologije, ki podpirajo varnost in delegacijo

Mrežni sistem omogoča dostop do geografsko porazdeljenih mrežnih virov, ki so zasebne narave. Zato je zahtevana večja varnost pri dostopu oziroma prenosu virov po fizičnem omrežju. Pri tem so predvsem koristne specifikacije, ki podpirajo varnost in delegacijo pravic: WS-Security, WS-SecureConversation ter WS-Delegation. Zaradi svoje narave so sporočila SOAP enostavno berljiva za človeka, ravno tako je enostavno tudi njihovo spreminjanje. Z namenom omogočanja večje varnosti specifikacija WS-Security naslavlja področja zagotavljanja integritete, zaupnosti, avtentikacije in enkripcije sporočil ter zagotavlja varnost od točke oblikovanja SOAP-sporočila do končnega prejemnika. V ta namen uporablja WS-Security tehnologijo digitalnega podpisa XML-dokumentov, enkripcijo XML-dokumentov in časovno žigosanje. Omogoča navezavo sporočil na ustrezne varnostne žetone. Le-ti so lahko binarni, certifikati X.509, Kerberos vstopnice ipd.

Danes obstaja več mehanizmov za zagotavljanje varnosti v mrežnih sistemih, ki so čedalje bolj uporabniško prijazni (Siebenlist in sod., 2004; Cannon in sod., 2003; DataGrid, spletni vir; Keahey in sod., 2002 in 2003; Lorch in Cowles, 2004; Welch in sod., 2003; Pearlman in sod., 2001; ter Barton in sod., 2006). Eden najbolj sodobnih pristopov, ki podpirajo tudi socialno mreženje, pa je Sibboleth (spletni vir) organizacije SymLabs.

2.3.4 Obstoječe mrežno prilagojeno sistemsko programje in storitve

V nadaljevanju največ pozornosti posvečamo prilagojenemu sistemskemu programju Globus Toolkit 4, ki smo ga uporabili kot osnovo pri našem delu, obravnavamo pa tudi druga programja, orodja in storitve, ki so uporabna pri izgradnji mrežnih sistemov in aplikacij. To so predvsem OGSA-DAI, Condor, UniGrids, gLite itn. Poseben poudarek je namenjen analizi obstoječih posrednikov virov in metapodatkovnim mrežnim storitvam.

2.3.4.1 Prilagojeno sistemsko programje Globus Toolkit 4

Globus Toolkit verzija 4 (GT4, spletni vir; Sotomayor in Childers, 2006) je odprtokodna orodjarna, ki jo je razvila Globus Alliance in omogoča razvoj mrežnih sistemov. Orodjarna GT4 implementira standarde OGSA, WSRF, WS-Management, Job Submission Description Language (JSDL), Distributed Resource Management Application API (DRMAA), WS-

Management, WS-BaseNotification, SOAP, WSDL, Grid Security Infrastructure (GSI) in druge. V osnovi je GT4 implementacija nekaterih ključnih protokolov, ki jih je definirala organizacija OGF, in sicer protokol za upravljanje z viri (Grid Resource Allocation & Management Protocol, GRAM), storitve za spremljanje in odkrivanje (Monitoring and Discovery Service, MDS), varnostne storitve (Grid Security Infrastructure, GSI), storitve za globalni dostop do shramb (Global Access to Secondary Storage, GASS) in mrežna storitev za prenos datotek (Grid File Transfer Protocol, GridFTP).

Projekt OntoGrid je predlagal organizaciji OGF, da bi v orodjarno GT4 vključili tudi njihove novo zgrajene semantične storitve (Missier in sod., 2007). To so predvsem storitve za označevanje, upravljanje z metapodatki, sklepanje ter delo z ontologijami. Predlagali so tudi dodatek k arhitekturi OGSA, ki je poimenovan Semantic – OGSA (S-OGSA). Pri našem delu smo uporabili njihove ontološke storitve.

2.3.4.2 Računalniške gruče

Računalniške gruče omogočajo izvajanje široko prepustnih računalniških aplikacij (High Throughput Computing, HTC). Za razliko od mreže tukaj gre za povezovanje obstoječih računalnikov v gruče v okviru posameznih organizacij (Litzkow in Livny, 1990). Tiste, ki temeljijo na programju Condor (spletni vir), so nameščene po vsem svetu ter znanstvenikom in inženirjem omogočajo reševanje različnih računsko zahtevnih problemov.

2.3.4.3 Prednosti in pomanjkljivosti obstoječih posrednikov virov

Posrednik mrežnih virov pregleduje in hrani stanje virov in njihovih zmožnosti v mrežnem okolju, usklajuje zahteve aplikacij po izvajanju določenih poslov z obstoječimi viri (npr. računskimi, podatkovnimi itd.), dodeljuje opravila razpoložljivim in primernim virom ter inicializira in nadzoruje izvajanje teh poslov. Pri našem delu nismo mogli identificirati nobenega obstoječega posrednika virov, ki bi izpolnjeval vse systemske zahteve (ali vsaj večino). V nadaljevanju je povzetek naših ugotovitev v raziskavi, ki smo jo opravili in je obravnavala obstoječe posrednike virov (Kravtsov in sod., 2006).

Sistem GridLab Resource Management System (Allen in sod., 2002; GRMS, spletni vir) je mrežno okolje za posredovanje virov in razvrščanje poslov. Za vzpostavitev mrežnega

okolja morajo uporabniki najprej namestiti skupek (sveženj) storitev GridLab (spletni vir), med katerimi je tudi storitev GRMS. Na tako zgrajeni mreži lahko izvajajo različna opravila. GRMS temelji na dinamičnem odkrivanju in izbiri virov, zmožen pa je tudi ugoditi določenim izzivom na področju posredovanja virov. V resnici GRMS upravlja s celotnim procesom razvrščanja in krmiljenja poslov na porazdeljenih sistemih. Četudi lahko deluje kot samostojna storitev, njegova moč prihaja v celoti do izraza v kombinaciji s celotnim skupkom (svežnjem) storitev GridLab. V času, ko smo načrtovali to študijo, storitev GRMS ni temeljila na standardu WSRF, zato je bila kakršnakoli interakcija z GT4 nemogoča. GRMS ne podpira izvajanja parametričnih študij, ki so zelo pomembne pri večjem številu gradbeniških aplikacij, pa tudi drugih inženirskih in znanstvenih aplikacij na splošno.

Na projektu EGEE so do nedavnega uporabljali posrednik virov LCG-2 (spletni vir), ki je bil nameščen na centralnem računalniku. LCG-2 prejema zahteve in se nato odloča o razvrščanju poslov glede na sistemske in druge zahteve. Zdaj so ga zamenjali s posrednikom virov gLite, ki ni usklajen z WSRF in ne podpira izvajanja parametričnih študij.

Cactus (spletni vir) je okolje za reševanje numeričnih problemov za znanstvenike. Do mrežnih storitev lahko dostopa prek mehanizma MPICH-G. Vendar morajo biti aplikacije za okolje Cactus napisane v MPI, kar nakazuje visoko stopnjo zahtevnosti pri prilagajanju obstoječih aplikacij za izvajanje v mrežnem okolju. Če računalniki, ki se uporabljajo pri izvajanju aplikacij, ne podpirajo MPI, je treba na vsak oddaljeni računalnik namestiti paket MPICH. Cactus ne temelji na standardu WSRF in ne omogoča nobenih funkcionalnosti za delo s podatki.

Nimrod/G (spletni vir) je specializirano okolje za parametrične študije. Pristop k razvrščanju poslov, ki ga uporablja Nimrod/G, ima za cilj optimizirati parametre, ki jih določijo uporabniki, izključno pri računsko intenzivnih poslih. Nimrod/G ne vsebuje nobenih metod za dostop do oddaljenih podatkovnih repozitorijev (shramb) ali za optimizacijo podatkovnih prenosov. Nimrod/G ravno tako ne vsebuje nobenih mehanizmov

za avtomatizirano agregacijo virov. Posrednik virov GridBus (spletni vir) pa je nadgradnja posrednika virov Nimrod/G in je namenjen ne samo računsko, temveč tudi podatkovno intenzivnim nalogam. Posrednik virov GridBus tudi razširja jezik Nimrod/G za parametrično modeliranje s podporo dinamičnim parametrom (parametri, katerih vrednosti se določijo med izvajanjem). Izvirna implementacija GridBus pa ne podpira avtomatiziranega odkrivanja virov, medobratovalnosti (omogočeno je izvajanje izključno na operacijskih sistemih Unix/Linux) niti ne upošteva standarda WSRF. GridBus prav tako nima možnosti za optimizacijo podatkovnih prenosov, ki so zelo pomembni pri delu z velikimi in/ali porazdeljenimi podatkovnimi zbirkami.

2.3.4.4 Metapodatkovne storitve mreže

Cilj metapodatkovnih oziroma informacijskih storitev mreže je priskrbeti informacije o dostopnih mrežnih virih in o njihovem statusu. Uporabljajo se pri nadzoru (npr. ko nas zanima uporaba in dostop do virov) in odkrivanju (ko dodeljujemo računske in druge posle) storitev in virov, dostopnih na mreži. Primer takih storitev je sistem za spremljanje in nadzor (Monitoring and Discovery System 4, MDS4), ki je vključen v GT4. MDS4 ima zmožnosti za zbiranje in hranjenje informacij iz množice porazdeljenih informacijskih virov. Vsebuje tri agregacijske storitve, ki vključujejo storitev za arhiviranje informacijskih virov MDS-Archiver (v persistentno podatkovno bazo, do katere lahko dostopajo odjemalci), za izvajanje zahtevanih nalog ter vrsto različnih vmesnikov in orodij (npr. spletni pregledovalnik, odjemalci, ki delujejo iz komandne vrstice), ki omogočajo poizvedbe v podatkovni bazi. MDS4 se lahko uporablja pri poizvedovanju storitev WSRF v zvezi z lastnostmi virov, pri izvajanju mrežnih aplikacij, pridobivanju podatkov ali omogočanju poizvedbe drugih sistemov za nadzor in smo ga zato uporabili pri izgradnji našega prototipa.

2.3.5 Tehnologije uporabniških vmesnikov

Mrežno okolje je storitvena infrastruktura, ki je že po svoji definiciji zelo kompleksna. Pri snovanju odjemalcev za mrežne sisteme pa je nujno treba upoštevati zahteve končnih uporabnikov. Velika večina (inženirji, raziskovalci in drugi strokovnjaki, ki bi pri svojem delu potrebovali mrežne storitve) namreč ni seznanjena s tehnološkimi posebnostmi mreže.

Zato je zelo koristno, da je odjemalec konceptualno preprost in da ne pomeni dodatne obremenitve za uporabnika.

Do mrežnih storitev lahko dostopamo prek lahkih ali debelih odjemalcev, kar je odvisno od tega, koliko funkcionalnosti naj bi bilo na voljo končnim uporabnikom, neodvisno od oddaljenih mrežnih storitev. Medtem ko so lahki odjemalci ponavadi enostavni in namenjeni vnaprej določeni aplikaciji, so debeli praviloma veliko bolj prožni in omogočajo dostop do storitev mrežnega sistema, ravno tako pa tudi tvorjenje novih mrežnih aplikacij. Napredne tehnologije so urejevalniki delotokov (npr. Triana, Kepler, Taverna /spletni viri/ ipd.), ki omogočajo prožen dostop do kompleksnih storitvenih infrastruktur in njihovo uporabo, kot tudi tehnologije za hitro izdelavo mrežnih portalov (npr. Vine /spletni vir/).

Triana je okolje za reševanje kompleksnih problemov, ki je napisano v programskem jeziku Java in temelji na odprti kodi (Churches in sod., 2006). Njene prednosti so, da je zelo prožna in da se lahko uporablja v veliko različnih scenarijih in na veliko različnih ravneh. En primer uporabe je npr. kot upravljalec delotokov za mrežne aplikacije ali kot sistem za analizo slikovnih, signalnih in tekstovnih podatkov v P2P-okoljih. Triana se lahko uporablja tudi za urejanje skript za ustvarjanje vrste nalog – grafov ali skriptnih formatov, kot npr. Web Services Flow Language (WSFL), Directed Acyclic Graph (DAG), Business Process Execution Language for Web Services (BPEL4WS), formati Petrinet in podobno. Prvotno so jo razvili za hitre vpogleds in analizo podatkov pri projektu GEO600. Podobni urejevalniki delotokov so Kepler in Taverna (OMII). Medtem ko je Kepler po svoji generični zasnovi podoben Triani, je Taverna specializirana za biomedicinske aplikacije.

Razvoj lahkih odjemalcev, na primer mrežnih portalov, je podprt z napredno, modularno in razširljivo orodjarno Vine, ki so jo razvili avtorji projekta GridSphere GridPortlets (spletni vir). Omogoča izdelavo aplikacij kot skupkov virov in storitev, ki izkoriščajo te vire, omogoča hierarhično organizacijo virov in tako tudi reprezentacijo ene ali večjega števila virtualnih organizacij. Omogoča tudi varnostne mehanizme za avtentikacijo končnih uporabnikov in avtorizacijo pri njihovi uporabi virov v okviru določene domene (pravice dostopanja, področje uporabe). Druge ključne funkcije so npr. razširljiv model za izvajanje

poslov in transparentna podpora persistentnih informacij o virih ter podpora izvajanju poslov v povezavi z relacijskimi bazami.

2.3.6 Dosedanji razvoj mrežnih infrastruktur v Sloveniji, Evropi in po svetu

V Sloveniji so v preteklih letih financirali približno ducat projektov, ki so se ukvarjali z razvojem mrežnih infrastruktur. To so projekti:

- »GridForum.SI: Grid infrastruktura za virtualne organizacije«, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo,
- »Tehnologija grid kot standardna komunikacijsko-računska infrastruktura«, Institut Jožef Stefan,
- »Ekstrakcija virtualnega znanja iz obsežnih podatkovnih baz z metodami mehkega računanja v grid tehnologijah«, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko,
- »AGENTGRID: načrtovanje večagentnih sistemov s pomočjo okolja grid«, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko,
- »Računalniški servis na grid infrastrukturi«, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko,
- »Razvoj, standardizacija in implementacija protokolov in servisov grid v kontekstu evropskega projekta EGEE«, Institut Jožef Stefan,
- »Tehnološka infrastruktura s poudarkom na tehnologijah grid – infrastruktura navideznih skupnosti«, Univerza v Ljubljani, Fakulteta za elektrotehniko,
- »Semantično grid okolje za ekološko modeliranje«, Institut Jožef Stefan,
- »Optimalno izkoriščanje podatkovnih virov pri podatkovnem rudarjenju«, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
- »Analiza standardov, priprava strateških usmeritev in izdelava prototipnega porazdeljenega prostorskega sistema«, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, ter
- »SIGNET – Razvoj in implementacija tehnologij grid v evropskem projektu EGEE s prenosom v slovenski e-prostor«, Institut Jožef Stefan.

Projekt GridForum.Si je povezal vse te projekte in omogočil, da se znanje o mrežnih tehnologijah v Sloveniji diseminira (Stankovski in sod., 2005). Projekt AgentGrid je uporabil agente pri mrežnih sistemih (Privošnik in sod., 2006), projekt SiGNet pa omogočil priklop računalniške infrastrukture na Institutu Jožef Stefan na mrežno infrastrukturo projekta vzpostavitve mreže za e-znanost (Enabling Grids for E-science, EGEE).

V Evropi obstaja nekaj mrežnih infrastruktur, od katerih so najbolj znane EGEE, DEISA in NorduGrid (spletni viri).

Na tem mestu smo zaradi njene pomembnosti izločili in nekoliko podrobneje analizirali mrežno infrastrukturo EGEE. Pri razvoju infrastrukture EGEE je bilo pomembno doseči združitev nacionalnih, regijskih in tematskih iniciativ v enotno panevropsko mrežno infrastrukturo. Gre torej za izgradnjo infrastrukture, podprte z medsebojno kompatibilnimi mrežnimi storitvami in protokoli. Le-ta naj bi omogočala podporo aplikacijam, ki prihajajo z zelo različnih področij znanosti. Pri tem si na projektu EGEE prizadevajo za prenos tehnologij in storitev, razvitih v dosedanjih evropskih mrežnih projektih, v produkcijsko okolje. Osnovni zahtevi projekta sta podpora virtualnim organizacijam ter deljenje porazdeljenih računskih in podatkovnih virov.

EGEE je danes največji večnamenski mrežni sistem na svetu, ki se razprostira čez meje Evropske unije in bi lahko postala osnova za nastanek prve svetovne mreže. V infrastrukturo je trenutno vključenih okoli 20.000 računalniških procesorjev in okoli 5 PB shranjevalnega prostora, omenjeni viri pa so porazdeljeni na okoli 180 lokacijah v 32 državah po Evropi in svetu. V vsakem trenutku na tej infrastrukturi teče več kot 1.000 vzporednih računskih poslov, sistem pa redno uporablja več kot 60 virtualnih organizacij.

Trenutno se na infrastrukturi EGEE izvajajo aplikacije s šestih različnih znanstvenih področij, in sicer: s področja fizike osnovnih delcev (High-Energy Physics, HEP), v okviru katerega poteka aplikacija LHC (spletni vir); biomedicine, v okviru katerega tečejo aplikacije s področja obdelave medicinskih slik, bioinformatike in odkrivanja novih zdravil; zemeljskih znanosti, kot so hidrologija, spremljanje atmosferskega onesnaženja in

meteorologija; geofizike, ki vključuje aplikacije s področja potresne analize in raziskovanja vulkanskega delovanja; astrofizike, v okviru katerega potekajo poskusi Planck, MAGIC in Fusion; računske kemije, na katerem potekajo različne simulacije kemijskih procesov. Obstaja tudi natančen opis vseh mrežnih aplikacij, ki trenutno tečejo na infrastrukturi EGEE (Poročilo DNA4.1, spletni vir).

V tako širokem spektru znanstvenih področij se zahteve posameznih aplikacij močno razlikujejo. Tako je na primer pri obdelavi biomedicinskih podatkov v ospredju varovanje osebnih podatkov pacientov in interaktivno izvajanje analiz podatkov. Očitno je, da če vztrajamo pri generičnosti v smislu vključevanja poljubnih izvršljivih programov v mrežne aplikacije, se posledično poveča tudi kompleksnost sistema. Poglavitna lastnost vseh aplikacij ki potekajo na infrastrukturi EGEE je, da mora biti specifična programska oprema vnaprej nameščena na vse oddaljene računalnike, še pred izvajanjem mrežne aplikacije. Pri našem delu pa smo si prizadevali, da z uporabo semantičnih tehnologij omogočimo večjo prožnost pri tvorjenju poljubnih mrežnih aplikacij.

Infrastruktura EGEE je na začetku temeljila na prilagojenem sistemskem programju Globus Toolkit verzija 2, ki ga je pozneje zamenjal gLite. Trenutno stanje programja gLite in storitev, ki jih ponuja, je nazorno opisano v literaturi (Poročilo DJRA1.1, spletni vir). Programje gLite je zgrajeno na principih SOA, ni pa skladno s skladom specifikacij WSRF. Standard WSRF še vedno nastaja, zato ga pri projektu EGEE niso upoštevali. V delu smo se odločili, da ga upoštevamo in tako tudi preizkusimo.

Evropski projekt InteliGrid (spletni vir), ki je potekal na KGI, je pomembno prispeval k nadaljnjemu razvoju področja gradbene informatike, saj se je ukvarjal s problemi vzpostavitve virtualnih organizacij na kompleksni mrežni infrastrukturi. Na tem projektu smo razvili: (1) konceptualni model za uporabo ontologij v mrežnem okolju; (2) ontološke definicije, ki omogočajo modeliranje virtualnih organizacij, vključno z delovnimi procesi, storitvami in viri; (3) osnovne in napredne ontološke storitve, ki so bile implementirane v Javi in nameščene kot mrežne storitve. Tako je projekt InteliGrid skupaj z drugimi projekti, kot so OntoGrid, ProvenanceGrid, K-WFGrid (spletni viri) itd., pomembno prispeval k

razvoju tehnologij semantične mreže. Pri našem delu smo upoštevali tudi dosežke nacionalnih projektov, kot so GridForum.si, AgentGrid in SiGNet (spletni viri), ki so prispevali k prenosu znanja o mreži v gospodarstvo.

Leta 2004 je organizacija Network for Earthquake Engineering Simulation (NEES, NEESit, spletna vira) predstavila prve primere uporabe mrežnih tehnologij na gradbenem področju. Osrednji namen zasnovane mrežne infrastrukture je omogočanje enotnega opravljanja poskusov na področju potresnega inženirstva v Združenih državah Amerike. Mreža povezuje raziskovalce in seizmologe z zmogljivo informacijsko-komunikacijsko infrastrukturo ter omogoča skupno uporabo potresnih naprav in računskih komponent za oblikovanje in preverjanje seizmoloških modelov. Omogoča tudi objavo vseh poskusov in interpretacij podatkov v okviru mreže in tako tudi učinkovito izmenjavo raziskovalnih podatkov med zainteresiranimi uporabniki.

2.4 Semantični splet

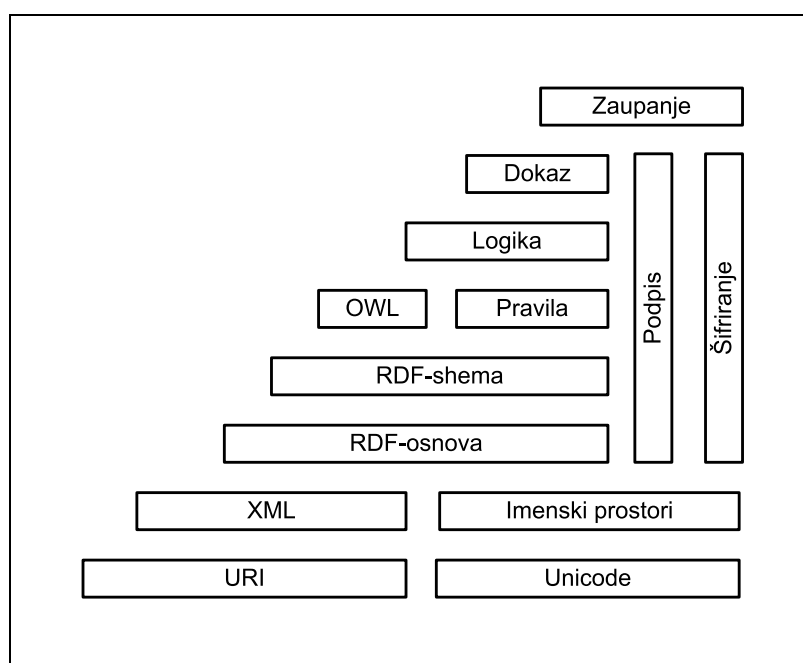
V nadaljevanju opisujemo trenutno stanje na področju semantičnega spleta in analiziramo, kako lahko obstoječe tehnologije pripomorejo k reševanju semantičnih vprašanj in iskanj na spletu. Naš namen je, da te tehnologije uporabimo pri izdelavi mrežnega sistema, ki bi lahko zaradi tega postal bolj prožen in zmogljiv in bi omogočal kompleksnejše mrežne aplikacije. Zato smo najprej opravili analizo trenutnega stanja, jezikov in obstoječih orodij semantičnega spleta, navsezadnje pa analizirali tudi ključno komponento semantične mreže – ontologije.

2.4.1 Definicija semantičnega spleta

Naslednja generacija spleta, ki sta jo Berners–Lee in Fischetti (1999) poimenovala semantični splet, ima za cilj izboljšati obstoječi splet s pomočjo semantičnih (ontoloških) beležk (angl. annotations). Tak splet naj bi nudil kvalitativno novo raven storitev in reševal vprašanja oziroma omogočal iskanja po spletu, ki so na ravni znanja. K razvoju spleta znanja pa naj bi prispevala predvsem **eksplicitna predstavnost** semantike podatkov, ki jo spremljajo domenske teorije (oz. ontologije). Ideja je omogočiti veliko več avtomatiziranih

storitev, ki temeljijo na strojni obdelavi semantičnih beležk, kot tudi hevristik, ki izkoriščajo metapodatke.

Uporabnost ontologij pri doseganju semantične medobratovalnosti na spletu so obravnavali v velikem številu objavljenih študij (Berners–Lee in sod., 2001; Staab /ur./ in Studer /ur./, 2004; Herman, spletni vir; Rossiter, spletni vir). Hierarhična pozicija ontoloških specifikacij in njihova razmerja do drugih specifikacij, ki se nanašajo na svetovni splet, je nazorno predstavljena na sliki (spodaj), ki smo jo povzeli po Hermanu (spletni vir).



Slika: Sklad semantičnega spleta se nenehno izboljšuje (Herman, spletni vir)
Fig.: The Semantic Web stack is continuously improved (Herman, web resource)

Ta pristop zahteva obogatitev obstoječega spleta z ontologijami, ki zajemajo najbolj raznolike domenske teorije. Ročna izgradnja ontologij zahteva veliko znanja in truda (Corcho, Fernandez–Lopez in Gomez–Perez, 2003), njihov primanjkljaj pa je v tem trenutku ozko grlo semantičnega spleta. Zato nekateri sodobni pristopi uporabljajo metode za avtomatsko učenje ontologij (angl. ontology learning) iz obstoječih podatkov.

Tehnologije semantičnega spleta imajo velik potencial pri tkanju globalne mreže, ki bo povezovala velike segmente človeškega znanja in se bo dopolnjevala z možnostmi za strojno obdelavo in učenje. Tako bi te tehnologije lahko uporabili ne samo pri semantičnem opisovanju podatkov, temveč tudi pri semantičnem opisovanju spletnih in mrežnih storitev, računalniških infrastruktur, programske in strojne opreme, domene aplikacije itd. Nekatere domenske ontologije, ki imajo velik pomen na področju gradbeništva, predstavljamo v naslednjem poglavju.

2.4.2 Jeziki semantičnega spleta

Pomemben pogoj za doseganje semantične medobratovalnosti mrežnih sistemov in storitev je sposobnost izkoriščanja bogatih, standardiziranih računalniških jezikov, kot so eXtensible Markup Language (XML), Resource Description Framework (RDF) in Web Ontology Language (OWL) (spletni viri). RDF in OWL temeljita na jeziku XML in omogočata opisovanje različnih entitet, ki se pojavljajo v računalniškem okolju, pa tudi v življenju ljudi na splošno. OWL so razvili nedavno, kot logično nadaljevanje obstoječih standardiziranih jezikov XML in RDF. Zato je število obstoječih aplikacij, temelječih na OWL, v tem trenutku še vedno zelo omejeno. V nadaljevanju si bomo nekoliko bolj natančno ogledali hierarhijo oziroma razlike med omenjenimi jeziki.

Jezik XML, kot tudi XML DTD in XML schema, je zadosten za izmenjavo podatkov med različnimi strankami (ali sistemi), ki se predhodno dogovorijo za določeno množico skupnih definicij (npr. koncepti: »most«, »vrata«, »okno« itd.). Jeziku XML manjkajo nekatere specifikacije, ki bi jih lahko uporabili za opisovanje globljega pomena teh definicij (npr. izjava »okno ima bodisi kvadratno bodisi ovalno obliko in ni večje kot 2 metra«). Ker takih specifikacij ni, avtomatizirani programi ne morejo preveriti in upoštevati globljega pomena. Na primer, ko se pripravi nova XML-shema, se lahko zgodi, da se isti koncept uporabi z včasih zelo subtilnim, vendar različnim pomenom v različnih kontekstih (npr. arhitekt je predvidel ovalna okna, gradbena firma pa je naročila okna v obliki kvadrata), ali pa se lahko zgodi, da se uporabita dva različna termina za stvari, ki imata isti pomen (npr. uporaba dveh različnih jezikov, slovenščine in angleščine pri poimenovanju konceptov: »most« in »bridge«).

RDF in RDF Schema (RDFS)¹ ta problem rešujeta tako, da omogočata dodajanje preproste semantike različnim oznakam. RDFS omogoča definiranje razredov, ki imajo več pod- in nadrazredov, kot tudi lastnosti, ki imajo lahko pod- in nadlastnosti, domene in območja/razpon. Tako na RDFS lahko gledamo kot na preprost ontološki jezik. Raziskovalci predvidevajo, da je za doseganje medobratovalnosti med množičnimi, avtonomno razvitimi in upravljanimi sistemi potrebna še bolj izrazna semantika. Na primer, RDFS ne more izraziti dejstva, da sta razreda »oseba« in »hiša« disjunktna ali da ima gradbeniška firma natanko štiri podizvajalce. Čeprav je RDFS sam po sebi zelo uporaben, ne podpira določenih ontoloških konceptov, kot so ekvivalenca, inverzne relacije in omejitve kardinalnosti razredov. V teh primerih je treba namesto RDFS uporabiti ontološki jezik OWL.

Kontinuiteta razvoja RDFS in OWL omogoča določeno stopnjo medobratovalnosti med temi jeziki. Na primer, tako kot RDFS tudi jezik OWL definira osnovna ontološka pojma »razred« in »lastnost«. Če povzamemo, je OWL veliko bolj izrazen pri opisovanju razredov in njihovih lastnosti, med drugim relacij med razredi (npr. disjunktnost), kardinalnosti (npr. izjava »natančno štiri«), enakosti med koncepti, bogatejšem opisovanju lastnosti, karakteristik lastnosti (npr. simetričnost) in števnosti pri razredih.

Ontologije, zapisane v jeziku OWL, so glede na OWL-GUIDE (spletni vir) kategorizirane v tri vrste: OWL-Lite, OWL-DL in OWL-Full. Značilnost, ki določa vsako od teh vrst, je jezikovna izraznost (ekspresivnost). OWL-Lite je najmanj ekspresiven, medtem ko je OWL-Full najbolj ekspresivna vrsta. Ekspresivnost vrste OWL-DL je med OWL-Lite in OWL-Full. OWL-DL je razširitev vrste OWL-Lite, medtem ko je OWL-Full razširitev vrste OWL-DL. Podrobnejši opis treh vrst jezika OWL je dostopen na OWL-FEATURES (spletni vir).

OWL-Lite je sintaktično najenostavnejša vrsta. Uporablja se v situacijah, ko je potrebna enostavna razredna hierarhija in enostavne omejitve. Omogoča hiter prestop iz obstoječih

¹ V nadaljevanju uporabljamo kratico RDF(S) za »istočasno RDF in RDFS«.

tezavrov¹ (angl. thesauri) in drugih pojmovno enostavnih hierarhij (OWL-GUIDE, spletni vir).

OWL-DL je veliko bolj ekspresivna vrsta od OWL-Lite in sloni na opisni logiki (Descriptive Logic, DL). Opisna logika je odločljiv² del logike prvega reda, in je zato dovezna (uporabna) za avtomatsko sklepanje (OWL-FEATURES, spletni vir). Tako je možno avtomatsko izračunati klasifikacijsko hierarhijo in kontrolirati vse morebitne nedoslednosti v ontologiji, ki pripada vrsti OWL-DL.

OWL-Full je najbolj ekspresivna vrsta jezika OWL. Nameravana uporaba jezika OWL-Full je v situacijah, kjer je zelo visoka ekspresivnost pomembnejša od zajamčene odločljivosti ali računalniške popolnosti jezika. Tako ni mogoče izvajati avtomatiziranih sklepanj na podlagi ontologij, ki pripadajo vrsti OWL-Full.

Na izbiro vrste jezika OWL, ki bi jo uporabili v konkretnem primeru, lahko vpliva veliko dejavnikov. Kljub temu pa obstajata dve enostavni pravili, ki smo jih upoštevali:

- Izbira med vrstama OWL-Lite in OWL-DL lahko temelji na zadostnosti uporabe jezika OWL-Lite pri konkretni aplikaciji.
- Izbira med vrstama OWL-DL in OWL-Full lahko temelji na odločitvi o tem, katera zahteva je pomembnejša, ali je to zahteva po avtomatskem sklepanju ali pa zahteva po visoko ekspresivnih in močnih možnostih za modeliranje, kot je uporaba metarazredov (razredi razredov).

2.4.3 O ontologijah

Standardni jeziki za semantično opisovanje različnih objektov, konceptov in entitet imajo potencial, da omogočijo semantično medobratovalnost podatkov, računalniških infrastruktur

¹ Slovarji, ki vsebujejo sinonime in včasih tudi antonime.

² Logika je odločljiva, če se izračun oziroma algoritem, ki temelji na logiki, izteče v končnem času.

in sistemov, vendar le v primeru vzpostavitve domenske ontologije. V nasprotju z definicijo ontologije, ki sta jo podala Platon in Aristotel, je na področju računalništva pogosto citirana razširjena Gruberjeva (1995) definicija: **ontologija** je formalna, eksplicitna specifikacija deljene konceptualizacije. Ta razširjena oblika formalno pomeni, da mora biti ontologija berljiva za stroje in da jo stroji lahko tudi interpretirajo; eksplicitno pa pomeni, da vsebuje jasne, nedvoumne, asertivne definicije različnih vrst pojmov in omejitev, ki modelirajo cilj obravnave; deljeno pa pomeni, da se uporablja z namenom definiranja skupnega standarda na področju. Termin konceptualizacija se nanaša na objekte, koncepte in druge entitete, za katere se domneva, da obstajajo na določeni domeni, in na relacije, ki obstajajo med temi objekti, koncepti in entitetami (Genesereth in Nilsson, 1987).

Že desetletje so ontologije vroča tema na različnih področjih. Staab /ur./ in Studer /ur./ (2004) sta v knjigi *Handbook on Ontologies* predstavila pregled formalizmov za zapisovanje ter sklepanje na podlagi ontologij, inženiring ontologij (npr. metodologije izgradnje ontologij, učenje ontologij itd.), ontološko infrastrukturo (npr. ontologije kot podpora za reševanje problemov, metode za združevanje in preslikave ontologij, metode za vizualizacijo, raziskovanje in priklic informacij, ki temeljijo na ontologijah) ter vrsto aplikacij ontologij na različnih področjih (npr. za upravljanje z znanjem, upravljanje z vsebinami znotraj virtualne organizacije, pri izdelavi sistemov za svetovanje, izdelavi portalov znanja, izdelavi elektronske trgovine, na področju bioinformatike, nadaljnji aplikaciji ontologij na spletu /Decker in sod., 2000/ kot tudi za doseganje semantične medobratovalnosti).

Tako ontologije predstavljajo enote znanja, ki omogočajo formalno reprezentacijo skupne konceptualizacije določene domene. Pri našem delu smo imeli cilj izdelati ontologijo virov, ki nastopajo v kompleksnem mrežnem okolju, in jo nato uporabiti za lajšanje raznovrstnih poizvedb oziroma pri dinamični integraciji virov na mreži.

2.4.4 Orodja in tehnologije za upravljanje z metapodatki

2.4.4.1 Orodja za urejanje ontologij

Za modeliranje ontologij obstaja vrsta okolij, npr. OilEd, Protégé (spletna vira) itd., med katerimi je Protégé velikokrat citiran v literaturi. Omogoča izgradnjo ontologij, njihovo vizualizacijo kot tudi zajemanje (eksplicitnega) znanja o določeni domeni. Z uporabo Protégéja inženirji ontologij lahko konstruirajo domenske ontologije, prikrojijo vmesnike za vnašanje podatkov in vnašajo podatke. Protégé je razširljiva platforma, ki vključuje grafične komponente, kot so grafi, tabele in multimedijske vsebine, na primer zvok, video, ter različne formate hranjenja metapodatkov, kot so na primer OWL, RDF, XML in tudi HTML. Protégé API drugim aplikacijam omogoča, da uporabljajo, dostopajo in prikazujejo podatke iz baze znanja, ki je bila kreirana prek Protégé.

2.4.4.2 Obravnava pristopov in tehnologij za upravljanje z metapodatki

Trenutno stanje na področju upravljanja z metapodatki so analizirali Missier in sod. (2007). Ugotovili so, da so skoraj edine uporabne tehnologije tiste, ki omogočajo hranjenje in poizvedbe po RDFS-podatkih. Takšni sistemi so npr. Jena (spletni vir), Sesame2 (Broekstra, Kampman in van Harmelen, 2002), KAON2, Kowari Metastore (spletna vira), ki so hkrati tudi najbolj znani in najbolj široko uporabljeni. Ti sistemi, ki slonijo na tradicionalni zmožnosti relacijskih podatkovnih baz, omogočajo bogate, zelo podrobne aplikacijske programske vmesnike za manipuliranje in dostop do RDF-podatkov kot tudi za poizvedbe z uporabo različnih jezikov, vključno z jezikom Sparql (spletni vir), ki ga promovira W3C. Oracle 10g RDF (Axelsson in sod., spletni vir) je najnovejša različica, ki podpira velike podatkovne zbirke in bogate zmožnosti poizvedovanja ter istočasno kombinira moči poizvedovalnih jezikov RDF z (angl. native) relacijskimi poizvedbami, ki so specifične za Oracle. Nedavno so razvili tudi dodatke k osnovnim zmožnostim na področjih kontekstualizacije, porazdeljenosti in razširljivosti, ki so bili deležni pozornosti.

Nadaljnji primer sistema za upravljanje z ontologijami in pripadajočimi bazami znanja je KAON2 (spletni vir), ki temelji na odprti kodi, uporablja pa se tudi v poslovne namene. KAON2 je zbirka orodij, ki omogočajo preprosto ustvarjanje in upravljanje ontologij, in predstavlja okolje za razvoj aplikacij, temelječih na ontologijah. KAON2 je optimiziran tudi

za učinkovito sklepanje (angl. reasoning) pri večjih ontologijah in pripadajočih bazah znanja. Razvoj sistema KAON2 je pripeljal tudi do nekaterih zmogljivih komercialnih produktov za upravljanje z metapodatki, ki jih trži zasebna družba Ontoprise (spletni vir).

Kowari Metastore (spletni vir) je metapodatkovna baza, temelječa na odprti kodi. Gre za sistem za hranjenje in dostop do metapodatkov, ki so hranjeni v obliki tripletov subjekt-predikat-objekt.

Aplikacijska programska vmesnika NG4J in Sesame2 (spletna vira) omogočata grupiranje RDF-izjav v imenovane grafe, ki temeljijo na izbrani kontekstualni informaciji. Adamku in Stuckenschmidt sta predstavila (2005) članek, kjer sta razvila mediacijsko plast, ki omogoča transparenco za odjemalce, ki bi radi dostopali do večjega števila metapodatkovnih zbirk, izdelanih v tehnologiji Sesame2. Cai in Frank (2004) pa sta razvila način za dostop do porazdeljenih RDFS-podatkov z uporabo P2P-protokolov. To so samo nekateri primeri sistemov, ki so jih razvili za osnovne operacije upravljanja z metapodatki.

Četudi omenjeni pristopi omogočajo robustno hranjenje in povpraševanje po RDFS-podatkih, so vidiki pridruževanja metapodatkov k storitvam¹ (angl. service encapsulation), varnosti in kontroli dostopa, transakcij, podvojevanja, evolucije metapodatkov, strategije za propagacijo sprememb trenutno deležni minimalne pozornosti. Nedavno smo opazili nekatere začetke dela na tem področju v iniciativah, kot je IBM-ova Boca RDF-shramba (spletni vir) ali W3C-jev protokol Sparql (spletni vir). Boca ponuja tudi dostop do RDFS-shramb (podatkovnih zbirk) prek spletne storitve z možnostmi za replikacije, transakcije, specifikacije verzije (angl. versioning), zgodovino sprememb (angl. change-tracking) in naročanjem na obvestila (angl. notification). W3C-jev protokol Sparql pa je zelo preprost vmesnik za spletne storitve, tako da lahko delamo poizvedbe v RDF-podatkovnih zbirkah. Missier in sod. (2007) so ugotovili, da četudi se te tehnologije zelo intenzivno razvijajo in testirajo, pa realne, industrijske aplikacije ne izkoriščajo vseh njihovih zmožnosti za upravljanje z metapodatki.

¹ Metapodatki so semantični opisi visokonivojskih konceptov, ki so pridruženi k nižjenivojskim storitvam (angl. service encapsulation).

Obstajajo tudi različni modeli, ki omogočajo povezovanje obstoječih virov, dostopnih na omrežju z eksplicitnimi metapodatki, ki te vire opisujejo, vendar slonijo na *ad hoc* mehanizmih. Eden od najbolj razširjenih načinov za vključevanje metapodatkov v HTML-ali XHTML-dokumente je npr. z uporabo elementa <meta> kjerkoli v dokumentu, tako kot to opisujeta Adida in Birbeck (spletni vir). Ta element omogoča opisovanje lastnosti dokumenta prek parov lastnosti in njihovih vrednosti. Normativne lastnosti vključujejo avtorja, datum izteka veljavnosti, seznam ključnih besed itd., lahko pa uporabljamo tudi druge lastnosti, ki jih ponudniki spletne vsebine definirajo sami, npr. tiste iz nabora Dublin Core. V preteklosti je bil ta način uporabe precej razširjen, npr. Simple HTML Ontology Extensions (SHOE).

XHTML 2.0 (Chong in sod., 2005) predlaga alternativni način povezovanja entitet in njihovih opisov, tako da so metapodatki ločeni od vsebine samega dokumenta. Dokumenti, ki vsebujejo metapodatke, se lahko dodajo k XHTML-dokumentom tako, da se v glavo dokumenta vgradi element <link>. Lastnosti *rel*, *about* in *href* omogočajo reference na vire, ki so zunaj samega dokumenta (Adida in Birbeck, spletni vir).

2.4.4.3 Programi za avtomatsko sklepanje

Program za avtomatsko sklepanje (angl. reasoning) omogoča sklepanje na podlagi množice dejstev ali aksiomov. Tovrstni programi uporabljajo bogato množico mehanizmov. Pravila izpeljave so pogosto specificirana na podlagi ontološkega jezika. Mnogi programi za avtomatsko sklepanje uporabljajo predikatno logiko prvega reda pri sklepanju, obstajajo pa tudi druge, hevristične variante. Delo z ontologijami OWL-DL podpirajo naslednji programi: Pellet, KAON2, RacerPro, Jena, FaCT ter FaCT++. Vsi razen FaCT in FaCT++ podpirajo tudi poizvedovalni jezik Sparql.

2.5 Razvoj standardov na področju medmrežnih tehnologij

Narava medmrežnih sistemov je takšna, da je množica različnih informacijskih virov razpršena in med seboj povezana s transportnim sistemom – omrežjem. Glede na obseg svetovnega omrežja Internet je torej očitno, da iz istih storitev lahko nastajajo različne aplikacije, ki morajo biti medsebojno kompatibilne. Npr. aplikacija elektronske pošte, ki jo

ponuja Microsoft, mora biti kompatibilna z aplikacijo iLane (spletni vir) podjetja Intelligent Mechatronic Systems Inc., ki vozniku v avtomobilu prebira elektronsko pošto. Pri doseganju svetovne kompatibilnosti aplikacij je standardizacija storitev in njihovih komunikacijskih vmesnikov bistvenega pomena. Na medmrežju standardizacija na splošno poteka v dveh vzporednih tokovih, in sicer na način *de iure* (iz teorije v prakso) ali *de facto* (iz prakse v teorijo).

Standardi *de facto* nastajajo neodvisno od mednarodno priznanih organizacij za standardizacijo. V takem primeru postane standard najuspešnejši izdelek določenega proizvajalca. To pomeni, da se postopek standardizacije začne oblikovati v okviru močnih proizvajalcev informacijske opreme ali popolnoma neodvisnih virov, npr. univerz. Na tak način je nastal operacijski sistem Unix, sklad protokolov TCP/IP, specifikacije WSRF itd.

Standardi *de iure* nastajajo pod pokroviteljstvom pooblaščenih agencij za standardizacijo in se uveljavljajo skozi nacionalne in mednarodne standardizacijske politike. Primer standarda *de iure* je sklad protokolov OSI. Tudi IFC nastaja po tej poti.

2.5.1 Standardizacijske organizacije

Priporočila, standardi in specifikacije na področju medmrežnih tehnologij se trenutno oblikujejo znotraj več standardizacijskih organizacij, forumov in teles. Pri posameznih sorodnih aktivnostih te organizacije med seboj tudi sodelujejo, npr. prek agencij Združenih narodov. Najpomembnejše organizacije, katerih delovanje smo obravnavali, so predvsem: Open Grid Forum (OGF), Organization for the Advancement of Structured Information Standards (OASIS), World Wide Web Consortium (W3C), Distributed Management Task Force (DMTF), Storage Networking Industry Association (SNIA), Web Services Interoperability Organization (WS-I), European Telecommunications Standards Institute (ETSI), Internet Engineering Task Force (IETF) in International Alliance for Interoperability (IAI).

V nadaljevanju si bomo ogledali področja delovanja ključnih standardizacijskih organizacij, ki so povezane s predmetom naših raziskav.

2.5.1.1 Open Grid Forum (OGF)

OGF ustanovljen leta 1998 kot forum za mrežo, je najpomembnejša organizacija za standarde na področju mrežnega računalništva. Združuje svetovno skupnost najpomembnejših organizacij, konzorcijev ter posameznikov na področju mrežnih tehnologij, katerih cilj je razvoj odprtih standardov, ki naj bi zagotovili globalno medobratovalnost mrežnih sistemov. Tako člani OGF sistematično zbirajo zahteve mrežnih aplikacij, spremljajo, razvijajo ter optimizirajo kompleksne mrežne aplikacije, razvijajo predloge specifikacij ter so povezani in sodelujejo z drugimi standardizacijskimi telesi.

Cilj OGF je vsesplošna posvojitev mrežnih sistemov v znanosti in industriji, to pa naj bi bilo omogočeno prek razvoja standardov na tem področju in združevanja v mednarodno skupnost za razvoj mreže. Naloge OGF so naslednje:

- Definicija mrežnih specifikacij, ki vodijo v široko posvojene standarde in medobratovalno programsko opremo.
- Združevanje v mednarodno skupnost, ki si izmenjuje ideje, izkušnje in zahteve.
- Promocija dela prek partnerjev, sponzorjev in medijev.

Organizacija OGF opravlja nekaj funkcij. Najpomembnejša se nanaša na standardizacijo. Standardizacijska funkcija je naprej razdeljena na podpodročja, ta pa na delovne skupine (Working Groups, WG), ki izdajajo specifikacije in priporočila. Standardizacijska funkcija obsega naslednja podpodročja: aplikacije, arhitektura, računski viri, podatkovni viri, infrastruktura, povezovanje z drugimi standardizacijskimi organizacijami, upravljanje ter varnost. Nanaša se torej tudi na sodelovanje z drugimi standardizacijskimi organizacijami, kot sta W3C in OASIS (v nadaljevanju).

Nekatere dokumente, ki so pripravljene v okviru OGF, predstavijo zgolj na konferencah ali jih objavijo kot članke v revijah s področja mrežnega računalništva (npr. *The Journal of Grid Computing*, *Future Generation Computer Systems*, Elsevier). Manjše število dokumentov pa pošljejo tudi drugim standardizacijskim organizacijam, te pa jih lahko tudi formalno potrdijo. Tako so bili nekateri dokumenti že objavljeni kot priporočila RFC, ki jih

izdaja organizacija IETF (npr. GridFTP, X.509 Proxy Certificate Profile itd). Pri tem je RFC zbirka dokumentov, ki opisuje različne dejanske in priporočene praktike, ki se nanašajo na Internet, in le majhen delež RFC-dokumentov je sprejetih kot dejanski standardi.

OGF je prve standarde na področju mrežnih sistemov promoviral po principu *de facto*, saj so prav njegovi člani oblikovali trenutno veljavno vizijo mreže, izdali arhitekturne specifikacije, izdelali referenčne implementacije ter potegnili vzporednice s standardi s področja spletnih tehnologij. Člani OGF so sodelovali tudi pri oblikovanju pomembnih priporočil in standardov pri drugih organizacijah: X.509 Proxy Certificate Profile, WS-Agreement, Web Services Description Language 2.0 (WSDL 2.0), Web Service Distributed Management (WSDM), Security Assertion Markup Language (SAML), eXtensible Access Control Markup Language (XACML) idr.

2.5.1.2 Organization for the Advancement of Structured Information Standards (OASIS)

OASIS je neprofitna mednarodna organizacija, ki razvija in oglašuje industrijske standarde za elektronsko poslovanje. Izdaja standarde na področju spletnih storitev, ki se nanašajo predvsem na višjenivojske funkcionalnosti, kot so varnost ter z njo povezani avtentikacija in avtorizacija, izvajanje poslovnih procesov, zanesljivo sporočanje (angl. messaging) itd. Z mrežo povezano področje, za katero skrbijo, je modeliranje stanja virov s spletnimi storitvami (angl. Stateful Resources). Skrbijo torej za specifikacije WSRF, ki skupaj s specifikacijami za obveščanje spletnih storitev (Web Services Notification, WSN /Niblett in Graham, 2005/) predstavljajo nadgradnjo spletnih storitev, ki na ta način izpolnjujejo zahteve odprte arhitekture mrežnih storitev.

2.5.1.3 World Wide Web Consortium (W3C)

Organizacija W3C je mednarodna organizacija, ustanovljena leta 1994 z namenom razvoja in oglaševanja medobratovalnih protokolov s področja spletnih tehnologij, npr. standardov eXtensible Markup Language (XML), Hypertext Transport Protocol (HTTP), Hypertext Markup Language (HTML) in drugih. Organizacija je izdala prve specifikacije s področja spletnih storitev. W3C se osredotoča predvsem na nizkonivojske jedrne funkcionalnosti

spletnih storitev, kot so standardi SOAP, WSDL, mehanizmi za naslavljanje spletnih storitev (WS-Addressing), Web Services Choreography Description Language (WS-CDL) itd. Te specifikacije so za mrežo pomembne, saj pomenijo osnovo, na kateri je zgrajena večina današnjih mrežnih sistemov. W3C na področju standardov spletnih storitev sodeluje z organizacijama OGF in OASIS.

2.5.1.4 Distributed Management Task Force (DMTF)

DMTF je industrijska organizacija, ki so jo ustanovili leta 1992. Razvijajo standarde za upravljanje informacijske infrastrukture podjetij in integracijo medmrežnih tehnologij v podjetja. Razvili so standarda Common Information Model (CIM) in Web-Based Enterprise Management (WBEM), ki opisujeta delovanje informacijskih sistemov za upravljanje strojne ter programske opreme in storitev na način, ki je neodvisen od platform in tehnologij. Organizacija se je leta 2003 povezala z organizacijo OGF z namenom standardizacije pristopa k zagotavljanju, deljenju in upravljanju mrežnih virov in storitev.

2.5.1.5 Storage Networking Industry Association (SNIA)

SNIA je organizacija, ki skrbi za standarde, povezane s shranjevanjem podatkov na mreži. Za mrežne sisteme sta najpomembnejši iniciativi Storage Management Initiative (SMI) in Data Management Forum (DMF). Specifikacije SMI opisujejo standardne vmesnike za odkrivanje, nadzor in upravljanje komponent za shranjevanje podatkov. Specifikacija DMF definira, implementira in priporoča zanesljive metode za obvarovanje in upravljanje življenjskega cikla elektronskih podatkov in informacij. Sodelovanje med organizacijama SNIA in OGF torej poteka predvsem na področju, ki je povezano z upravljanjem podatkov, dostopnih na mreži.

2.5.1.6 Web Services Interoperability Organization (WS-I)

WS-I je bila ustanovljena leta 2002 z namenom zagotavljanja medobratovalnosti med različnimi implementacijami spletnih storitev. Cilj organizacije je integracija obstoječih standardov, in ne razvoj novih. Izdajajo t. i. profile, ki opisujejo, kakšne zahteve mora spletna storitev izpolnjevati, da jo lahko deklariramo kot storitev v skladu z določenim profilom. Mrežne tehnologije temeljijo na spletnih storitvah, torej so za medobratovalnost mrežnih storitev povezave z organizacijo WS-I ključnega pomena.

2.5.1.7 European Telecommunications Standards Institute (ETSI)

ETSI je standardizacijska organizacija s področja telekomunikacijskih tehnologij. Nedavno so vzpostavili novo tehnično skupino z imenom STF 331: TC-GRID (ETSI firms up on GRID, spletni vir), katere cilj je izdajanje standardov za zagotavljanje medobratovalnosti mrežnih in telekomunikacijskih tehnologij. V prihodnosti naj bi bila ta tehnična skupina odgovorna za konvergenco mrežnih standardov na področju informacijskih tehnologij in telekomunikacij.

2.5.1.8 Open Middleware Infrastructure Institute (OMII)

OMII priskrbi in podpira razvoj odprtokodnih programov s ciljem trajnostnega razvoja raziskovalne skupnosti. Na spletnih straneh OMII je bogat katalog programske opreme. Zaradi svoje postavitve v Združenem kraljestvu pa je organizacija zelo vplivna pri standardizacijskih organizacijah, kot sta OASIS in OGF. Na spletnih straneh OMII lahko med drugim najdemo tudi prilagojeno sistemsko programje OGSA-DAI (Antonioletti in sod., 2005), ki omogoča postavitev podatkovnih mrežnih storitev.

2.5.1.9 Internet Engineering Task Force (IETF)

Organizacija IETF skrbi za standarde iz protokolnega sklada TCP/IP, npr. RFC 793, RFC 1122 ter RFC 1323.

2.5.1.10 International Alliance for Interoperability (IAI)

Združenje IAI je zelo pomembno za področje gradbeništva. Vizija združenja, ki danes povezuje več kot 600 članov iz 17 držav, je izboljšanje komunikacije, produktivnosti, skrajšanje procesov, znižanje stroškov ter zagotavljanje oziroma povečanje kakovosti skozi celotni življenjski cikel zgradbe, in sicer z generacijo modela zgradb IFC, ki predstavlja *de iure* standard za formulacijo in elektronsko izmenjavo informacij na področju gradbeništva. Člane združenja IAI so arhitekti, inženirji, izvajalci gradbenih del, lastniki, upravljalci, raziskovalci, državni uradniki, razvijalci programske opreme itd.

2.6 Druge raziskave na področju semantičnih mrež

Predvidevajo, da se bo v prihodnosti mreža razvila v kompleksno infrastrukturo, ki bo omogočala dostop do velike količine heterogenih virov (Next Generation Grids 2, spletni vir). Dinamična in prožna uporaba virov pri mrežnih aplikacijah pa zahteva vse več informacij o funkcionalnosti in dostopnosti virov, javnih vmesnikih in komponentah. Posledica je vse večji pomen metapodatkov o sistemu in virih. Da bi bili metapodatki uporabni, morajo imeti dogovorjeno strukturo, ki omogoča ne samo človeško razumevanje, temveč tudi avtomatizirano računalniško interpretacijo. V nadaljevanju podajamo pregled najnovejših dosežkov na področju semantične mreže ter njihove pomanjkljivosti glede na cilje, ki jih imamo pri tem delu. Več o uporabi semantičnih in mrežnih tehnologij na področju gradbeništva pa je na voljo v tretjem poglavju.

Missier in sod. (2007) so predstavili mrežne storitve za upravljanje z ontologijami in metapodatki, ki omogočajo tudi avtomatsko sklepanje. Te storitve smo tudi uporabili in ugotovili, da je možno iskanje ontoloških konceptov ter njihovih lastnosti po ključnih besedah, upravljanje z ontološkimi instancami, avtomatsko preverjanje konsistentnosti metapodatkovne zbirke itd. Predlagane storitve so zelo generične, kar je istočasno tudi njihova največja pomanjkljivost, saj niso bile preizkušene pri konkretnih industrijskih aplikacijah. Te nove semantične storitve so vključili v arhitekturo S-OGSA.

Semantične tehnologije so lahko uporabne tudi pri dinamičnem strukturiranju kompleksnih mrežnih aplikacij, ki jih je mogoče izdelovati z uporabo tehnologij delotokov. Bubak in sod. (2005) so predstavili prototip sistema, ki omogoča avtomatsko sestavljanje delotokov iz spletnih in mrežnih storitev. Uporabniški vmesnik zgrajenega prototipa podpira uporabnika pri sestavljanju delotokov iz semantičnih in sintaktičnih opisov storitev, ki so shranjeni v organizacijskem spominu mreže (Grid Organizational Memory, GOM). To je omogočeno na podlagi visokonivojskih opisov uporabniških zahtev, kar zahteva celostno razumevanje storitev ter njihovo semantično medobratovalnost. Pri tem se zdi pomembno opozoriti na dejstvo, da sistem ne omogoča sestavljanja delotokov iz obstoječih namiznih (izvršljivih) programov. Potrebne so torej nadaljnje raziskave o teoriji in tehnikah opisovanja in logičnega sklepanja, o semantiki in obnašanju storitev ter njihovega združevanja v

kompleksne delotoke. Pri našem delu smo se omejili na praktično raven in uporabili obstoječi preizkušeni sistem za urejanje in upravljanje delotokov Triana (Churches in sod., 2006).

Stankovski in sod. (2008b) so razvili sistem, ki omogoča razvoj, postavitve in izvajanje aplikacij za podatkovno rudarjenje v mrežnem okolju. Količina dostopnih podatkov in programov, ki jih hranijo posamezniki in organizacije, v zadnjem času strmo narašča, zato naraščajo tudi potrebe po mrežnih aplikacijah za podatkovno rudarjenje (Stankovski in Dubitzky, 2007). Pri tem so se ukvarjali s problemom predelave obstoječih programov za podatkovno rudarjenje za uporabo v mrežnih sistemih ter njihovo vgrajevanje v mrežne aplikacije, kar je pogosto kompleksen in zahteven postopek.

Provenienca pri mrežnih sistemih je še vedno zelo neraziskano področje. Eden poglavitnih ciljev evropskega projekta ProvenanceGrid (spletni vir) je bil omogočiti večje zaupanje v odprtih okoljih, kot je mreža, z uporabo tehnik za shranjevanje metapodatkov, ki jih generira mrežna infrastruktura. Tehnike, ki so jih razvili, omogočajo, da se shranjeni podatki pregledujejo, preverjajo, na podlagi teh metapodatkov pa naj bi bilo omogočeno tudi avtomatsko sklepanje npr. v zvezi z okvarami sistema. Obravnavani problemi pa se nanašajo bolj na nižjenivojsko infrastrukturo in so zanimivi predvsem za administratorje mrežnih sistemov. S problemom provenience pri izdelavi kompleksnih produktov se je ukvarjal tudi Petrinja (2007), ki je predstavil konceptualni model za sledenje provenience ter sistem za upravljanje z metapodatki, ki se nanašajo na provenienco.

Skupna lastnost vseh pristopov je, da temeljijo na večji uporabi ontologij, ki omogočajo opisovanje področja aplikacije z enoumno definicijo konceptov (npr. koncept aplikacije, storitve, naprave, uporabnika, podatkovnega vira, delotoka itd.), in njihovih medsebojnih odnosov v obliki aksiomov, pravil itd. Skupaj z računalniško berljivimi metapodatki o instancah konceptov ontologije tako tvorijo posebne vrste baz znanja. Vse ontologije, ki so jih doslej zgradili in poskušajo opisati kompleksna razmerja, ki vladajo v mrežnem sistemu, pa so precej osnovne. V primeru izdelave produkcijskih sistemov bi jih bilo treba ustrezno nadgraditi.

Na tem mestu lahko sklenemo, da nam semantične tehnologije lahko koristijo pri ustvarjanju in uporabi tako **znanja za mrežno infrastrukturo** kot tudi **znanja za mrežne aplikacije**, pri čemer je slednje, glede na praktično naravnost študije, za nas bolj zanimivo.

3 e-Gradbeništvo

Informacijske tehnologije bistveno spodbujajo razvoj inovativnosti in napredek tudi v gradbeni industriji. S hitrim razvojem računalniških in informacijskih tehnologij pa se nujno spreminja način inženirskega dela, možne postajajo tudi takšne aplikacije, ki si jih prej ni bilo mogoče predstavljati. To velja pri izračunavanju konstrukcij, načrtovanju, simulaciji gradnje, upravljanju projektne dokumentacije in na splošno pri učinkovitejšem načinu projektne dela.

Ob obravnavanju problematike uvajanja računalniških metod v gradbeništvo so Turk (1992, 2001, 2007), Hannus (1992), Eastman, Chase in Assal (1993), Turk in Duhovnik (1995), Ekholm (1996), Cerovšek (2003) in Eastman in sod. (2008) identificirali fenomen enkratnosti:

- **Enkraten proces.** Proces načrtovanja, projektiranja, gradnje in vzdrževanja je zapleten in dolgotrajen in nikoli ni identičen s predhodnimi.
- **Enkraten produkt.** Stavbe, ceste in drugi inženirski objekti so praviloma unikatni, s tem pa tudi spremljajoča dokumentacija in njena uporaba.
- **Enkraten krog sodelujočih.** Od projekta do projekta se spreminja tako projektna skupina (arhitektov, statikov, gradbenikov, strojnih in drugih inženirjev) kot tudi investitorji, izvajalci, podizvajalci in dobavitelji, ravno tako pa tudi državne ustanove, ki dodeljujejo različna dovoljenja in predpisujejo zahteve za nadaljnje zgradbe.
- **Enkraten obseg.** Odvisno od številnih dejavnikov, kot so npr. zahtevnost projekta, lokacija, razpoložljiva finančna sredstva, delovna sila itd., se razlikuje tudi obseg zgornjih treh kategorij tako po kvantiteti kot tudi po kakovosti.

Glede na definicijo imajo tehnologije mreže velik potencial, da pripomorejo k prilagajanju fenomenu enkratnosti. Omogočijo lahko deljenje in sočasno uporabo podatkov (npr. modelov zgradb), specializiranih programov za analizo, kompleksnih delotokov in drugih mrežnih aplikacij, delovnih procesov, integracijo naprav in storitev ter bolj inovativne pristope k uporabi obstoječih aplikacij. Pomembna značilnost tega dela je, da ponuja analizo kompleksnih problemov na področju gradbeništva, ki bi bili rešljivi z uporabo mrežnih sistemov in aplikacij. Tako smo raziskovali problem medobratovalnosti v virtualnih organizacijah, možnosti dinamične integracije naprav in storitev na področju potresnega inženirstva, probleme upravljanja projektnih dokumentacij kot tudi simuliranja in modeliranja na področju sanitarnega inženirstva.

Analiza večjega števila različnih primerov uporabe nam bo omogočila definicijo ključnih arhitekturnih komponent in tako pripomogla k razvoju načrta semantične mreže. Tako zasnovana mreža naj bi omogočala uporabo že obstoječih gradbeniških in drugih aplikacij, ne da bi bilo treba porabiti veliko razvojnega truda za njihovo predelavo. Razvoj popolnoma novih mrežnih aplikacij, ki bi bile sposobne izkoriščati prednosti, ki jih ponujajo SOA, OGSA in pripadajoča odprta mrežna okolja, pa naj bi bil čim bolj enostaven. Razvojne rešitve in lekcije, ki smo se jih naučili pri delu na evropskih in domačih raziskovalnih projektih s področja mrežnih tehnologij InteliGrid, DataMiningGrid, GridForum.si, SiGNet in AgentGrid, smo tudi uporabili pri snovanju naprednega sistema, temelječega na tehnologijah semantične mreže.

V nadaljevanju predstavljamo motivacijo, tehnično-tehnološke zahteve ter nekatera področja aplikacij mrežnih tehnologij v gradbeništvu. Predstavljamo tudi problem medobratovalnosti in vlogo ontologij pri reševanju tega problema ter obstoječe domenske ontologije, ki so še posebno zanimivi za naše delo.

3.1 Grozdi kot oblika povezovanja podjetij

Na področju gradbeništva večinoma nastopajo majhna in srednja podjetja. Zahteve globalnega trga po čim večji učinkovitosti pa jih silijo predvsem v specializacijo, zunanje

izvajanje in povezovanje z drugimi podjetji (Medved, 2004). Prednost povezanega podjetja je, da na trgu doseže več, kot če bi nastopalo samostojno (Bergman in Feser, spletni vir).

Grozdjenje je način, kako lahko mala in srednja podjetja uspešno tekmujejo na globalnem trgu. Vendar mora obstajati skupni ekonomski interes in medsebojno zaupanje, da lahko grozdi obstajajo in delujejo. Značilni za podjetja, povezana v grozdu, so npr. skupne tehnologije, skupni kupci in prodajni kanali, skupna baza delovne sile itd.

Porter (2000) kot razloge za povezovanje izpostavlja lažji dostop do specializiranih virov v lokalnem okolju z manjšimi stroški, lažji in hitrejši dostop do informacij, medsebojno dopolnjevanje, dostop do javnih institucij in javnih finančnih virov ter medsebojno primerjanje in učenje. Med negativnimi lastnostmi povezovalja so izguba identifikacije, prožnosti, izguba know-howa, odvisnost, razpršitev odgovornosti in znižani dobiček.

Dejavniki, ki so pomembni pri vzpostavljanju grozdov (Bergman in Feser, spletni vir), so sam proces oblikovanja povezav, izbira oblike povezovanja ter izvajanje in realizacija povezovanja. Podjetniške povezave pa se v največji meri dotikajo pretoka virov med podjetji, ki so v veliki meri dostopni tudi na omrežju. Za tako povezovanje je ključno, da obstaja preizkušena in dokazano uporabna informacijsko-komunikacijska infrastruktura. Sodelovanje zahteva skupen jezik, kompatibilno komunikacijsko opremo, informacije pa morajo biti natančne in pravočasne.

Povečanje inovativnosti gradbenih podjetij je lahko posledica izboljšav infrastrukture, delovnih procesov, oblikovanja novih izdelkov, drugačnih prodajnih poti in drugih aktivnosti, s katerimi se pridobiva konkurenčna prednost. Inovativnost je tako tesno pogojena z investiranjem v infrastrukturo, ki prinaša informacije, znanje, nova sredstva in omogoča trženje. Pri tem nove tehnologije največkrat omogočajo razvoj novih idej na področju načrtovanja produktov, proizvodnje in trženja.

Z novimi tehnologijami se razvijajo tudi inovativna nova podjetja. Zato grozdi pritegujejo precejšnjo pozornost gospodarske politike, izobraževalnih in znanstvenih institucij.

Podporne institucije (vladne organizacije, izobraževalne in raziskovalne institucije, svetovalci) pa imajo pomembno vlogo pri uvajanju novih infrastruktur.

V tem trenutku se zdi, da imajo tehnologije mreže potrebni potencial, ki lahko podpre zahteve grozdov po izboljšani infrastrukturi, ki bi lahko reševala kompleksne industrijske probleme. Prek mrežne infrastrukture bi se lahko podjetja grozda povezovala in oblikovala t. i. virtualna podjetja. Tako lahko pojmujeemo mrežo kot informacijsko tehnologijo, ki bi v prihodnje lahko omogočila nadaljnji razvoj grozdov.

V gradbeni panogi je nadvse pomembno sodelovanje vseh udeležencev, npr. inženirjev, arhitektov, izvajalcev, naročnikov, državnih ustanov. Danes je to sodelovanje še zelo nepovezano. S skupno virtualno bazo raznorodnih virov bi komunikacija stekla precej hitreje, prehod od ideje k zgrajenemu objektu bi bil hitrejši in rezultat boljši. Z ustrezno mrežno infrastrukturo bi v gradbeniški panogi lahko zagotovili večjo stopnjo gospodarskega razvoja. Nekatera večja slovenska podjetja bi prav tako lahko uporabljala tehnologije mreže pri poenotenju proizvodnje in pridobivanju konkurenčnih prednosti na ciljnih tujih trgih.

Na ta način bi lahko dosegli hitrejše reakcije na želje kupcev, specializacijo za ključne pristojnosti, uporabo dopolnjujočega know-how, boljšo sprejetost na trgu, zagotavljanje obstoja z dolgoročno povezanostjo, osredotočenost na lastne prednosti in razpoložljive vire, razvijanje zmožnosti in odnosov za hitro in uspešno reševanje problemov. Podjetja, ki sledijo tem trendom, so v prednosti zaradi skupnih razpoložljivih virov v grozdu, delujejo v spodbudnem okolju, kjer vlada medsebojno zaupanje.

Gradbena podjetja bi lahko izkoriščala velike računske zmogljivosti, ki jih potrebujejo inženirji (npr. statiki), oziroma specializirane skupne podatkovne vire in druge storitve ali aplikacije, ki so dostopne na mreži.

Tega se zavedajo tudi vlade vse več držav, ki z različnimi državnimi programi podpirajo podjetniško povezovanje. Taki programi so potekali v približno dvajsetih državah, med drugim tudi v Sloveniji, kjer sta ministrstvo za gospodarstvo in pospeševalni center za

mlado gospodarstvo izvajala projekt Oblikovanje lokalnih podjetniških grozdov. Tako je bilo spodbujanje medpodjetniškega povezovanja eno prednostnih področij v razvojnem programu za povečanje konkurenčnosti slovenske industrije v zadnjih letih. Spodbujalo se je povezovanje podjetij, specializacije v proizvodnih verigah in skupni razvoj mednarodnih trgov po sistemu grozdov.

Slovenija pri spodbujanju razvoja grozdov izhaja iz izkušenj drugih držav, ki kažejo, da je povečanje konkurenčnosti posameznega podjetja odvisno od povečanja konkurenčnosti njenega lokalnega okolja. Slovenija ima prednosti predvsem v visoki tehnološki ravni in visoki izobrazbi zaposlenih, slabosti pa so v šibkem prenosu znanja, nizkem zavedanju o prednostih grozdov in pomanjkanju tržne usmeritve izobraževalnih ustanov. Največja nevarnost pri oblikovanju podjetniških grozdov pa je slaba pripravljenost za medpodjetniško sodelovanje, ki je posledica pomanjkanja zaupanja med podjetniki (Slak, 2002).

Kot rezultat omenjenega projekta so se v Sloveniji oblikovali grozdi, med njimi tudi gradbeni (leta 2004), v katerega se je vključilo štirinajst slovenskih gradbenih podjetij. Njegov cilj je, da s svojimi članicami postane mreža podjetij, ki bo specializirana in konkurenčna dobaviteljica celovitih rešitev na področju graditve v Evropski uniji, ki bo dolgoročno konkurenčnost gradila na temeljih tehnološke in organizacijske inovativnosti, razvoja in kakovosti (Ministrstvo za gospodarstvo, 2004).

3.2 Virtualne organizacije

Virtualne organizacije so relativno nova oblika povezovanja organizacij in bi lahko postale osnova za dinamično povezovanje podjetij s ciljem izdelave kompleksnih produktov. Danes so prednostna organizacijska oblika pri prilagajanju fenomenu enkratnosti in odpravljanju razdrobljenosti na gradbenem področju.

Virtualna organizacija (VO) (Camarinha–Matos, Afsarmanesh in Lima, 1999) je »začasno zavezništvo podjetij, ki se združijo z namenom deljenja svojih spretnosti ali osrednjih pristojnosti ter virov z namenom, da bolje odgovorijo na poslovne priložnosti, njihovo

sodelovanje pa je podprto z računalniškimi omrežji«. Projekt Globemen definira virtualno organizacijo kot »sistem, ki strankam ponuja rešitve na podlagi informacijsko-telekomunikacijske infrastrukture, ki sestavlja oziroma agregira ključne pristojnosti na začasen in rekonfigurabilen način« (Karvonen in sod., 2003).

Zgornji dve in tudi novejša definicija, ki govori o mreži sodelujočih podjetij (Camarinha-Matos in Afsarmanesh, 2005), poudarjajo pomen informacijskih in komunikacijskih tehnologij za učinkovito delovanje virtualnih organizacij. Obstaja tudi veliko drugih vidikov, ki razlikujejo virtualne organizacije od tradicionalnih, na primer razlike pri skupinskem delu, stilu vodenja, načinu izkoriščanja virov itd.

Virtualno podjetje je torej podjetje, ki sicer formalno ne obstaja, obstajajo pa povezave med različnimi podjetji, ki na podlagi soglasno opredeljenih pristojnosti kot enovita veriga vrednosti izvajajo poslovne procese. Kot virtualno podjetje tako pojmujejo mrežo formalno samostojnih, vendar gospodarsko povezanih podjetij s skupnimi cilji. Prednost takšnega povezovanja je predvsem v hitri reakciji na zahteve trga, saj se vzpostavljajo samo tiste povezave, ki so potrebne za doseg posameznega cilja. Za te povezave so značilne tudi zabrisane podjetniške meje navzven in navznoter.

Učinkovitost nameščene informacijske in telekomunikacijske infrastrukture pa je široko prepoznaven pogoj za delovanje virtualnih organizacij. Današnje tipične Business-to-Business (B2B) rešitve so usmerjene predvsem na izmenjavo informacij, večinoma prek centraliziranih strežnikov. Razvoj tehnologij prilagojenega systemskega programja, kot so Distributed Component Object Model (DCOM) podjetja Microsoft, Common Object Request Broker Architecture (CORBA) organizacije Object Management Group, Enterprise JavaBeans (EJB) podjetja Sun, .NET podjetja Microsoft, ter razvoj spletnih storitev sicer omogočata deljenje virov znotraj različnih sistemov in med njimi, vendar te tehnične rešitve, ki omogočajo medobratovalnost, temeljijo na »zastarelih« principih, kot so npr. centralizirane podatkovne zbirke, prevajalniki različnih datotečnih formatov, centralizirani sistemi za upravljanje z dokumenti, sistemi za projektno vodenje itd. Vsi omenjeni delčki sicer lahko sestavljajo celoto, vendar pa vsak potrebuje posebno namestitev, dostop,

nastavitve, vzdrževanje in znanje, nujno za uporabo. Torej lahko povzamemo, da tehnologije, ki so trenutno v uporabi v podjetjih in organizacijah, ne zadovoljujejo dinamičnih potreb po deljenju virov, ki jih potrebujemo za vzpostavitev virtualnih organizacij.

3.3 Problem medobratovalnosti in vloga ontologij

Povezovanje računalnikov s hitrimi komunikacijskimi kanali je samo prvi korak pri vzpostavljanju virtualnih organizacij. Medobratovalnost povezanih sistemov pa je še vedno med najbolj kritičnimi spornimi vprašanji. Rešitev tega problema lahko prispeva k uspešnemu razvoju in rasti paradigme virtualnih organizacij. Doseganje medobratovalnosti, ki bi omogočala tudi deljenje informacij, pa zahteva uporabo zelo naprednih računalniških tehnologij, ki šele nastajajo.

Medobratovalnost je definirana kot sposobnost informacijskih sistemov, da delujejo povezani med seboj, kar obsega združljivost strojne opreme, komunikacijskih protokolov, aplikacij in podatkov (ICH Glossary, spletni vir). V zgodovinskem razvoju računalništva so najprej poskrbeli za združljivost strojne opreme in komunikacijskih protokolov, trenutno pa je še vedno pereč problem združljivosti aplikacij in podatkov, ki imajo največji pomen za uspešno sodelovanje znotraj virtualne organizacije. Medobratovalnost podatkov je npr. pogoj za ustvarjanje skupnega jezika za interakcije med raznorodnimi partnerji in storitvami znotraj virtualne organizacije. Medobratovalnost obstoječih aplikacij pa je pogoj za njihovo sestavljanje v nove, inovativne aplikacije za reševanje kompleksnih problemov.

Podatkovno medobratovalnost lahko razčlenimo na tri podplasti, in sicer sintaktično, strukturno in semantično medobratovalnost (DC, spletni vir; Katranuschkov, Scherer in Turk, 2001):

- **Sintaktično medobratovalnost** dosežemo z označevanjem podatkov na podoben način, tako da lahko delimo podatke in da stroji lahko uvozijo/izvozijo podatke, ki so razčlenjeni (npr. spletna vira Encoded Archival Description /EAD/; Machine Readable Cataloging /MARC/).

- **Strukturno medobratovalnost** dosežemo z uporabo strukturiranih definicij metapodatkov, t. i. metapodatkovnih shem, ki omogočajo deljenje podatkov, npr. RFD.
- **Semantično medobratovalnost** pa dosežemo prek dogovorov o standardih za opisovanje vsebine (angl. content description standards) (npr. spletna vira Dublin Core /DC/ in Anglo-American Cataloguing /AAC/).

Medtem ko je koncept medobratovalnosti na področju virtualnih organizacij nekaj samoumevnega, je način, kako naj se ta interpretira in implementira na praktični ravni, predmet nenehnih kontroverznih diskusij, to pa tudi zaradi tega, ker je doseganje učinkovite medobratovalnosti razumljeno kot ključni dejavnik za številne vitalne lastnosti virtualnih organizacij. V nadaljevanju naštevamo pet ključnih zahtev pri virtualnih organizacijah, ki se nanašajo na doseganje medobratovalnosti:

- **Prilagodljive strukture in metode medobratovalnosti.** Trg je gonilna sila za sodelovanje znotraj virtualnih organizacij. Pri tem so sekundarnega pomena obstoječi kanali, na katerih je že vzpostavljena medobratovalnost, primarnega pa poslovne priložnosti, ki dinamično določajo sodelovalno strukturo virtualne organizacije. V teh primerih je raznorodnost konzorcijev nekaj samoumevnega in je neločljivo povezana z reševanim problemom. Izpostavljena pa je zahteva po uporabi prilagodljivih struktur in metod medobratovalnosti, ki so zmožne integracije večjega števila poslovnih domen in vej, tako vertikalno znotraj domene kot tudi horizontalno med domenami.
- **Tehnično-tehnološka rodnost in koncertacija.** Komplementarnost organizacij, ki sodelujejo pri virtualni organizaciji. Vsak partner se odlikuje v določenem podprocesu in/ali ima ključna znanja o trgu. Medobratovalnost mora pri tem zagotoviti tehnično-tehnološko rodnost in koncertacijo oziroma znanje in storitve, ki jih prinaša vsak od partnerjev, morajo biti dostopni za učinkovito izrabljanje.
- **Dinamična udeležba partnerjev pri virtualni organizaciji** zahteva, da se lahko organizacije dinamično priklopijo na omrežje virtualne organizacije in tudi izklopijo. Zaradi tega je tudi pomembno, da so v pravem trenutku (angl. just-in-

time) dostopne informacije o razpoložljivih storitvah, akterjih, virih itd. kot tudi to, da se te informacije redno ažurirajo.

- **Souporaba procesov in virov** (angl. process and resource sharing) pomeni velik izziv pri zagotavljanju medobratovalnosti znotraj virtualne organizacije. Ker so meje posameznih partnerskih organizacij zabrisane, partnerji lahko soustvarjajo in integrirajo poslovne procese, souporablajo podatke, informacije in vire znanja.
- **Notranje organizacijske strukture partnerjev niso poenotene.** Organizacijska struktura in informacijski tokovi so odvisni od narave poslovanja in značilnosti posameznih partnerjev. Ta organizacijski polimorfizem je naslednji večji izziv pri določanju splošno uporabnih metod za zagotavljanje medsebojne obratovalnosti.

Množica kompleksnih nerešenih vprašanj na tem področju nakazuje, da morajo biti vse metode, storitve in strukturne definicije, ki naj bi zagotavljale oziroma predstavljale okvir za medsebojno obratovalnost znotraj določene virtualne organizacije, načrtovane kot prilagodljive entitete, krepke in tolerantne pri okvarah, s posebnim poudarkom na možnosti za vzdrževanje.

Sedanje rešitve se problema lotevajo predvsem s tehnološke perspektive. Na voljo so določene metode in orodja, ki omogočajo poenotenje procesa dostopa do javnih vmesnikov, opisovanje in odkrivanje storitev ter opisovanje vhodno-izhodnih parametrov. Opise vmesnikov, storitev in parametrov lahko tolmačijo tako ljudje kot tudi umetni agenti, tako se doseže avtomatizacija dostopa. Vendar z medsebojno obratovalnostjo na opisani tehnični ravni dosežemo le kompatibilnost pri dostopanju do podatkov in drugih virov. Za prožnejše upravljanje z informacijami (in njihovim pomenom) pa je ključnega pomena semantična medobratovalnost. Le-to lahko dosežemo izključno na podlagi splošno sprejetega temeljnega besedišča konceptov in njihovih medsebojnih relacij, kar predstavlja ontološko raven. Alternativnih pristopov v literaturi nismo zasledili. S tem bi dosegli naslednje:

- Vmesniki so osvobojeni bremena definiranja specializiranih funkcij s fiksnimi vhodnimi/izhodnimi parametri, ki so uporabni izključno pri določeni nalogi, torej, povezovanje dveh partnerjev ali storitev znotraj virtualne organizacije ne sme biti

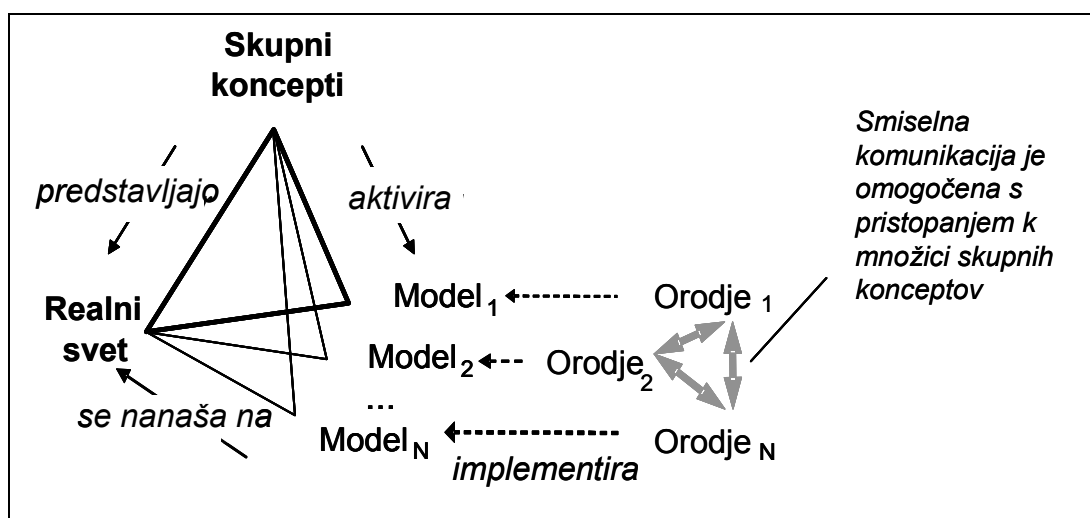
predmet vzpostavljanja specializiranih komunikacijskih poti, ampak postaja izziv deljenje pomena vsebine.

- Tolmačenje sporočene vsebine je ločeno od procesa komunikacije, kar omogoča maksimalno prožnost in robustnost pri pridobivanju potrebnih podatkov in informacij, ki jih je mogoče tudi nadalje tolmačiti.
- Prožna integracija poslovnih procesov postaja možna z vzpostavljanjem ekspresivnega in temeljnega modela poslovnih procesov, ki je lahko podprt s koherentno metodologijo in storitvami za integracijo poslovnih procesov (npr. jezika BPEL in BPEL4WS).

Slika 3-1, ki so jo predstavili Gehre in sod. (2005) ponazarja, kako splošno sprejeta ontologija (t. i. sporazumna ontologija) lahko olajša integracijo med semantično heterogenimi orodji. Na visoki ravni predstavimo razmerje med termini, modeli in realnostjo s pomočjo tako imenovanega Ogdenovega trikotnika (Ogden in Richards, 1994). Na področju računalništva so termini, ki se uporabljajo, pravzaprav simboli, ki označujejo koncepte, npr. termin »stavba« označuje koncept »stavbe«, ki predstavlja artefakt iz realnega sveta. Problem, ki se pojavlja, je, da različna orodja tipično uporabljajo različne modele, ki se nanašajo na artefakte iz realnega sveta. To pa je odvisno od njihovih specifičnih ciljev in domen, ki jih pokrivajo npr. strukturna analiza, izračuni stroškov itd. Tako za vsako od orodij lahko narišemo Ogdenov trikotnik, ki mu pripada. S pristopanjem k skupni množici konceptov se lahko orodja medsebojno »razumejo« in komunicirajo v zvezi s skupno domeno zanimanja. Pri tem ni nujno, da vsako posamezno orodje interpretira ontološke koncepte na enak način, pomembno pa je, da so dejanja, ki jih lahko opazujemo, skladna z definicijami v ontologiji.

Cena, ki mora biti plačana zaradi prednosti, ki jih omogočajo sporazumne ontologije, je velik trud, predvsem pri razvoju kompleksnih algoritmov za interpretacijo in formulacijo vsebine interakcij. Trajnostni razvoj ontologij, kopičenje metapodatkov, povezanih z ontologijami, in doseganje soglasja zanje pa je drugi izziv.

Analiza prednosti in slabosti različnih pristopov pri tvorjenju virtualnih organizacij nazorno kaže, zakaj je nujno treba izdelati primeren okvir za doseganje semantične medobratovalnosti znotraj virtualnih organizacij. Pri tem moramo zadovoljiti zahteve, ki izhajajo iz petih ključnih lastnosti virtualnih organizacij, ki smo jih omenili zgoraj. Poleg tega smo bili pri naši analizi pozorni tudi na dragocene namige, ki so nam omogočili, da v nadaljevanju pripravimo primerno razvojno metodologijo in realistično osnovo za delo.



Slika 3-1: Ogdenov trikotnik. Medobratovalnost med heterogenimi orodji je omogočena s pristopanjem k sporazumni ontologiji

Fig. 3-1: Ogden triangle. Interoperability among heterogeneous tools is facilitated by committing to a consensual ontology

3.4 Najnovejše raziskave na področju semantičnih tehnologij v kontekstu gradbeništva

Danes je v praktično orientiranih raziskavah in razvoju zelo malo znanih sistemov, ki izrabljajo ontologije za doseganje semantične medobratovalnosti, v praksi pa taki sistemi ne obstajajo. Poudarek je še vedno na razvoju orodij in okolij za splošno tehnološki dostop do ontologij in sklepanju ter na praktičnih, poslovno naravnanih vprašanjih in problemih.

K praktičnemu reševanju industrijskih problemov se najpogosteje pristopa s konceptualnim modeliranjem z uporabo XML-shem, npr. sistem Express Data Manager (EDM) podjetja

EPM Technology, sistem DDS-CAD podjetja Data Design System itd. Nedavni razvoj na področju gradbeništva je pripeljal do konceptualnega modela xmlIFC, ki naj bi ga v prihodnje upoštevali razvijalci programske opreme CAD.

V nadaljevanju podajamo pregled nekaterih pomembnih projektov in rezultatov raziskav, ki smo jih zasledili v literaturi in ki uvajajo semantične tehnologije na področju gradbeništva.

Prve korake pri konceptualnem modeliranju na področju gradbeništva sta naredila npr. Björk (1989) z razvojem osnovne strukture modela zgradb in Ekholm (1996), ki je razvil konceptualni model gradbeniških del, ki je uporaben pri klasificiranju. Tudi Turk (2001) je predstavil fenomenološke osnove za konceptualno modeliranje gradbeniških produktov. V naslednjih letih so na projektu e-COGNOS (Lima in sod., 2002, a in b; Bourdeau in sod., 2001; Ferneley, 2002) definirali in razvili odprto infrastrukturo in vrsto orodij, ki omogočajo upravljanje z znanjem v dinamičnem, sodelujočem gradbeniškem okolju. Izdelana ontologija je zapisana v jeziku Daml+Oil, ki je predhodnik jezika OWL. Projekt je bil omejen na vprašanja upravljanja z znanjem na področju gradbeništva, ni se pa ukvarjal s problemom medobratovalnosti mrežnih virov, struktur in sistemov.

K uvajanju semantičnih tehnologij je pomembno prispeval tudi projekt ISTforCE, ki je razvil orodja in aplikacije, ki izrabljajo inženirske ontologije in omogočajo izdelavo primernih uporabniških vmesnikov ter ustreznih pretvorb produktivnih modelov (Katranuschkov in sod., 2001). Gehre in sod. (2004) so nadaljevali in razširili to delo ter predlagali P2P-infrastrukturo in okolje umetnih agentov, ki aktivno podpirajo sodelovanje med gradbeniški podjetji. Pri tem uporabljajo metapodatke z namenom upravljanja z viri in gradbeniškim timom v celoti. P2P-sistemi pa ne omogočajo učinkovitega spremljanja in nadzora nad delovanjem sistema, zato je njihova uporabnost pri reševanju določenih problemov na področju gradbeništva omejena.

Yurchyshyna in sod. (2007) so predstavili sistem, ki preverja, ali je določen gradbeniški projekt v skladu s trenutno zakonodajo. Izdelano računalniško orodje sprejme dva vhodna podatka: gradbeniški projekt v formatu IFC in trenutno dostopno regulativo v obliki pravil, zapisanih v RDF. Poskrbeli so za avtomatsko pretvorbo gradbeniškega projekta iz formata

IFC v format RDF. Pri tem pa uporabljajo samo nekatere koncepte iz modela IFC. Nekaterih pravil iz regulative ni mogoče preverjati¹, orodje pa je še vedno v izdelavi in ni bilo evalvirano (Yurchyshyna in sod., 2007).

Laaroussi in sod. (2007) so predstavili izvirno metodo, ki temelji na analizi Failure Modes Effects and Criticality Analysis (FMEA), kar je analitična metoda. Izdelali so metapodatkovno bazo, ki hrani podatke o analizah, in omogoča njihovo ponovno uporabo. Tako so omogočili pospešitev pri novih študijah, saj ni potrebno ponovno izvajanje zahtevnih analiz, če so bile narejene v preteklosti. Za modeliranje stavb niso uporabili produktnega modela IFC.

Beetz, de Vries in van Leeuwen (2005) so predstavili sistem, ki iz kompleksnih produktnih modelov IFC izlušči parcialni model, ki vsebuje le minimalno količino informacij, ki jih potrebuje ciljna aplikacija. Dobljeni RDF-graf istočasno vsebuje informacijo o provenienci. Podoben sistem, ki ima za cilj delo z metapodatki o provenienci pri produktnih modelih, so predstavili tudi Petrinja, Stankovski in Turk (2007).

Brooke in Parkin (2008) sta predstavila sistem Alliance, ki predstavlja varnostno infrastrukturo in ontologijo, ki definira vloge in procese v okviru virtualne organizacije. Struktura Alliance je ločena od mrežne infrastrukture in omogoča dinamično znanstveno sodelovanje.

Najnovejše dosežke pri uvajanju tehnologij semantične mreže na področju gradbeništva je prinesel projekt InteliGrid, ki se je končal leta 2007. Zaradi njegovega velikega pomena za področje gradbeništva ga obravnavamo podrobneje na koncu poglavja.

Kljub vsemu trudu, ki je bil vložen v omenjene raziskave, pa lahko sklenemo, da so danes vsi dosežki pri uvajanju semantičnih tehnologij na področju gradbeništva zelo fragmentarni

¹ Avtorji prispevka so to utemeljili na podlagi rezultatov projekta ISTforCE.

in imajo omejeno praktično vrednost. Tako ostaja še veliko prostora za nadaljnje raziskave in razvoj na tem področju.

3.5 Ontologije na področju gradbeništva

Vzporedno z razvojem računalniških infrastruktur in sistemov se na področju gradbeništva razvijajo tudi konceptualni modeli. V nadaljevanju podajamo pregled nekaterih najpomembnejših.

3.5.1 Ontologija virtualne organizacije

Pri generičnem delu ontoloških definicij, ki naj bi bile uporabne pri poslovanju v okviru virtualnih organizacij, je bilo doslej zelo malo narejenega. V preteklosti so bili na področju računalništva zelo uporabni modeli, kot je Role Based Access Control (RBAC) (Ferraiolo in Kuhn, 1992; Sandhu in sod., 1996). Na projektu InteliGrid so razvili razširljiv osnovni nabor konceptov, ki se nanašajo na medobratovalnost v okviru virtualne organizacije. Omenjena ontologija predstavlja skupni jezik za logistiko, deljenje virov in storitve integracije virov. Ontologija zajema informacije za splošne entitete, ki nastopajo v okviru virtualne organizacije, opisuje vse od akterjev in drugih strukturnih konceptov do administrativne informacije ter virov in storitev v okviru virtualne organizacije. Pri tem so posebej izpostavljeni mehanizmi za agregacijo storitev, ki so opisani glede na ontologijo OWL-S, vse to pa omogoča koncepte za ontološko podprto avtomatsko integracijo kompleksnih poslovnih procesov.

Najnovejši pristopi v zvezi z upravljanjem virtualnih organizacij pa so Shibboleth, GridShib in myvocs (Barton in sod., 2006; Welch in sod., 2007; Gemmill in sod., 2008; Scavo in Welch, 2008). Ti pristopi dajejo večji pomen minimalni uporabi semantičnih tehnologij, kar se v praksi izkaže kot zelo koristno.

3.5.2 Model zgradb IFC

Temeljni razredi za industrijo (Industry Foundation Classes, IFC) so *de iure* standard na področju produktnih modelov v gradbeništvu. Temeljijo na procesnem modelu IDEF0, kar ni obvezno, a je priporočljivo, metodologiji STEP ter zapisu informacijskega modela v jeziku EXPRESS. Nekateri prispevki s področja produktnega modeliranja (Cerovšek, 2003) IFC ne opisujejo kot standarda zaradi pomanjkljivosti, ki bi jih bilo treba odpraviti za takšno oznako. Standard je po eni od definicij dokument, ki nastane s konsenzom in ga odobri priznani organ, ki določa smernice, pravila in lastnosti dejavnosti ali izdelkov oziroma rezultatov (*de iure* standard). Z druge strani pa se model zgradb IFC v literaturi večkrat označuje kot *de facto* standard.

Model zgradb IFC so razvili z namenom, da podjetjem omogoči neodvisno izmenjavo podatkov oziroma produktnih modelov (npr. med različnimi gradbeniškimi programskimi orodji). Iniciativa iz leta 1994, ki je poleg ugodnega odziva v celotnem gradbeniškem sektorju pritegnila tudi številne razvijalce programske opreme, je botrovala ustanovitvi mednarodnega združenja IAI. Uspešnost združenja se kaže v hitrem širjenju organizacije ter v petih izdanih različicah modela zgradb IFC (prva je bila izdana leta 1996) ter njegovi implementaciji v programski opremi, ki se uporablja na področju gradbeništva.

Pazlar in Turk (2008) sta se ukvarjala s problemom medobratovalnosti med tremi široko uporabljanimi aplikacijami za arhitekturno načrtovanje. Izvajala sta teste z izmenjavo različnih risb in ugotovila, da njihovi vmesniki ne delujejo pravilno. Pri tem sta tudi odkrila, da v različnih primerih prihaja do spreminjanja in/ali izgube informacij na ravni entitet ali atributov. Sklenila sta, da bi bilo v prihodnje koristno vlagati v razvoj boljših IFC-vmesnikov.

Howard in Björk (2007) pa sta predstavila analizo različnih pogledov na model IFC, ki kažejo na njegovo precejšnjo kompleksnost, zaradi katere še ni uveljavljen v praksi.

3.5.3 IFC-zapis v OWL

Konceptualni model zgradb IFC (spletni vir) so različne interesne skupine predelale in zapisale v jeziku XML, na primer, ifcXML, aecXML, BLIS-XML in bcXML (spletni viri). Takoj po standardizaciji jezika OWL pa so se pojavile (Beetz, van Leeuwen in de Vries, 2005; Schevers in Drogemuller, 2005; Dolenc in sod., 2007 in 2008) pobude za razširitev tega modela s koncepti in pravili, ki so specifični za ontologije. Vse te iniciative so danes še vedno v povojih. Praktično ni nobenih znanih produkcijskih sistemov, ki bi temeljili na teh ontologijah. Prevladujoča ideja pa je, da bi na podlagi ontologije IFC tvorili metapodatke (oziroma informacije) avtomatsko ali pa ročno z dostopanjem do strežnikov IFC-produktnih modelov z uporabo in ustrezno pretvorbo. Takšen postopek lahko močno olajša nadgrajevanje obstoječih IFC-produktnih modelov v resnične baze znanja (ontologije skupaj z instancami konceptov), ki bi imele tudi realno praktično uporabno vrednost.

Predelava modela IFC iz obstoječih formatov XML ali EXPRESS v OWL pa je očitno zelo zapletena in v tem trenutku še ni dokončana. Pretvorba konceptualnega modela IFC v OWL pa je izjemno zanimivo vprašanje (npr. za nadaljnje delo).

3.5.4 Ontologija gradbene informatike

Turk (2006) je objavil članek, ki ponuja celovit vpogled v področje in tematske sklope, s katerimi se ukvarja gradbena informatika. Razkriva, da je eden od načinov uporabe ontologij povečanje razumevanja področja in razmerij, ki vladajo med njegovimi deli. V članku je predstavljena ontologija gradbene informatike, vključno z metodologijo, epistemologijo in aksiologijo, kar je formalna definicija tega znanstvenega področja. Ontologija ima hierarhično obliko, ki na najvišji ravni razlikuje med (1) osrednjimi in (2) podpornimi temami. Osrednje teme kreirajo znanje, ki se nanaša bodisi na (a) aktivnosti za obdelavo informacij, (b) aktivnosti komunikacije/ koordinacije ali (c) skupne infrastrukture. Podporne teme pa se nanašajo na procese prenosa znanja in vključujejo (a) raziskovalne potrebe, (b) implementacijo dosežkov ter (c) analizo vpliva raziskav. Namen našega dela je predvsem prispevek na področju 1, c.

3.5.5 Ontologija spletnih storitev

Danes se mnogi strinjajo, da bo prihodnost e-poslovanja temeljila na spletnih storitvah. Tehnologije spletnih storitev pa so še vedno v razvoju. Zato moramo obravnavati tudi ta vidik medopratovalnosti, če želimo analizirati zahteve gradbeniških aplikacij glede prihodnjih mrežnih infrastruktur. V tem trenutku sta najbolj znani dve ontologiji – ontologija spletnih storitev (Web Ontology Language for Web Services, OWL-S) in ontologija za modeliranje spletnih storitev (Web Service Modeling Ontology, WSMO), katerih cilj je dodati pomen običajnim spletnim storitvam. Ontologija OWL-S je bila razvita v okviru organizacije W3C, medtem ko je WSMO plod raziskav v okviru delovne skupine ESSI WSMO, ki jo predstavljajo evropski projekti SEKT, DIP Knowledge Web in ASG.

Četudi je OWL po naravi generičen, je bil razvit z namenom, da podpira opisovanje, oglaševanje in odkrivanje spletnih storitev. OWL je zares priročen pri opisovanju lastnosti in funkcionalnosti spletnih storitev. Pomembna iniciativa W3C v tem kontekstu je tudi razvoj OWL-S, ki je evolvirala iz predhodnika DAML-S. OWL-S daje ponudnikom spletnih storitev osnovni nabor konstruktov za opisovanje lastnosti in zmožnosti (njihovih spletnih storitev) na način, ki je nedvoumen in strojno berljiv (angl. computer interpretable). Cilj ontologije OWL-S, ki se lahko uporablja pri semantičnem opisovanju spletnih storitev, je omogočiti avtomatizacijo nalog, ki jih izvajajo spletne storitve, ki vključujejo avtomatizirano odkrivanje storitev, izvajanje, medopratovanje, sestavljanje (angl. composition), spremljanje in nadzor nad izvajanjem.

Alternativa OWL-S je WSMO. Ta omogoča opisovanje različnih vidikov, ki se nanašajo na semantične spletne storitve. Ontologija temelji na t. i. okvirju za modeliranje spletnih storitev (Web Service Modeling Framework, WSMF), ki je sestavljen iz štirih osnovnih elementov za opisovanje semantičnih spletnih storitev: (1) ontologije, ki priskrbijo terminologijo, ki jo uporabljajo drugi elementi; (2) cilji, ki definirajo probleme, ki naj bi jih rešile spletne storitve; (3) opisi spletnih storitev, ki definirajo različne vidike spletnih storitev; (4) mediatorji (angl. mediators), ki rešujejo probleme medopratovalnosti.

3.6 Potencialna področja uporabe mrežnih sistemov

Potencialna področja uporabe tehnologij mreže so pri tvorjenju dinamičnih virtualnih organizacij, ki smo jih opisali v prejšnjih poglavjih, izdelavi računsko intenzivnih aplikacij (npr. računanje po metodi končnih elementov, simulacije modelov stavb z namenom ocenitve potresnih tveganj, t. i. parametrične študije) ter podatkovno in informacijsko intenzivnih aplikacij (npr. upravljanje s projektnimi dokumentacijami ali simulacije na področju sanitarnega inženirstva). V nadaljevanju se podrobneje posvečamo pomembnim uporabniškim zahtevam in predstavljamo konkretna področja aplikacij, kjer bi lahko razvili in uporabljali mrežne aplikacije.

3.6.1 Analize fizikalnih veličin zgradb

Mrežne tehnologije so lahko koristne pri numerični analizi in različnih simulacijah (Hartmann, 2006; Alonso in sod., 2006). Na tem mestu navajamo dva konkretna primera, kjer bi lahko bile koristne na področju gradbeništva.

Prvi primer je uporaba metode končnih elementov (Prelog, 1975). Potek numerične analize pri tej metodi je ponavadi razdeljen na tri sklope:

- priprava podatkov oziroma izdelava numeričnega modela,
- preračun po metodi končnih elementov oziroma analiza modela in
- predstavitev ter analiza rezultatov.

Tem trem korakom so prirejena tudi programska orodja. Najpomembnejša so tista, ki so namenjena dejanskemu preračunu numeričnega modela. Ta programska orodja so običajno brez uporabniškega vmesnika in ob zagonu le preberejo vhodno datoteko s podatki o numeričnem modelu, po opravljeni analizi pa zapišejo rezultate v izhodno datoteko. Najbolj znani in razširjeni programi za analizo numeričnih modelov po metodi končnih elementov so ABAQUS, ANSYS in MSC.NASTRAN (spletni viri). Poleg njih pa je na voljo še veliko manj znanih tržnih in znanstvenih programov, ki so dostopni tudi na medmrežju (npr. programi OpenSees, spletni vir). Pri njihovem izvajanju lahko tehnologije mreže pripomorejo, da hitreje in bolj zanesljivo pridemo do rezultatov. Na primer, če je

računalniška gruča na Fakulteti za gradbeništvo in geodezijo zasedena za izračune, se lahko uporabijo druge dostopne gruče v okviru virtualne organizacije (npr. večje število fakultet, povezanih skupaj v eno virtualno organizacijo).

Drugi primer je izdelava splošnih neparometričnih modelov pri potresnem inženirstvu (Vamvatsikos in Cornell, 2006), na primer, če želimo izdelati splošni neparometrični model za natančno 30 akcelegramov in natanko en histerezni model. V tem primeru je treba isti izvršljivi program zagnati velikokrat, vsakič z drugimi vhodnimi podatki in drugo nastavitvijo parametrov za izračune.

Pri obeh primerih bi gradbenim inženirjem zelo koristilo, če bi obstajala metoda za enostavno in hitro vključitev obstoječih izvršljivih programov v mrežno okolje in njihovo nadaljnje kombiniranje v kompleksne mrežne aplikacije. V nadaljevanju dela predstavljamo tudi rešitev tega problema, ki temelji na uporabi semantičnih tehnologij pri mrežnih sistemih.

3.6.2 Seizmična analiza zgradb

Raziskovanje potresnih vplivov na zgradbe pripomore k potresno varnejši gradnji. Za tovrstne raziskave se pogosto uporabljajo drage merilne opreme, na primer potresne mize, centrifuge in bazeni za simulacijo valov cunamijev, podatkovni arhivi, računski viri, različna simulacijska orodja ter druga specializirana oprema (NEESit). Zahteve, postavljene mrežnemu sistemu, ki naj bi povezoval omenjene vire, pa so velike. Koristno bi bilo razviti programsko opremo in mrežno infrastrukturo, ki poenostavlja integracijo heterogenih sistemov, ki funkcionirajo na oddaljenih lokacijah, kot so merilni instrumenti in drugi senzorji, računski viri in sistemi za sodelovanje v skupinah, uporabnikom pa ponuditi enostaven uporabniški vmesnik. Pri različnih poskusih je pogosto treba programsko avtomatizirano krmiliti porazdeljene merilne opreme, in sicer med samim izvajanjem računskih simulacij. Včasih je treba tudi sinhronizirati simulacije in merilne instrumente, ki delujejo na fizično ločenih lokacijah. Na oddaljenih lokacijah namreč lahko teče del poskusa, posamezni deli pa medsebojno niso neodvisni. Pomembno je zagotoviti tudi enostavno uporabo obstoječih metod, npr. nelinearne metode za analizo konstrukcij pri

potresni obtežbi (Fafar, 2000), prek mrežnega sistema. Uporabniki mrežnega sistema bi tako lahko med seboj zanesljivo in varno komunicirali, načrtovali in izvajali poskuse na oddaljenih merilnih napravah, za analizo podatkov bi uporabljali napredne metode, rezultate, ki bi bili tako na voljo tudi ostalim pooblaščenim udeležencem, pa bi objavljali kjerkoli na mreži.

3.6.3 Sanitarno inženirstvo

Povečano onesnaževanje vodnih ekosistemov zahteva vse večje napore pri njihovem nadzoru in upravljanju. Pri ekološkem modeliranju pa se pogosto uporabljajo napredne informacijske tehnologije. Matematični modeli ekosistemov omogočajo simulacije pod različnimi pogoji, in tako prispevajo k debati o upravičenosti določenih okoljevarstvenih ukrepov. Pri izgradnji matematičnih modelov je pogosto treba upoštevati kompleksna razmerja v naravi (nelinearna dinamika fenomena, večkratni poskusi pod različnimi pogoji), zato so zahtevani izračuni konceptualno zapleteni in računsko zahtevni. Primer sistema, ki pri ekološkem modeliranju uporablja napredne tehnologije s področja strojnega učenja, so pred leti predstavili Stankovski in sod. (1998). Matematično modeliranje in simulacije pogosto zahtevajo uporabo naprednih orodij in algoritmov, ki so dostopni v obliki izvršljivih (neinteraktivnih) programov.

Mrežni sistem ima potencial, da omogoči prožne, bolj inovativne, hitrejše in nadgradljive aplikacije na tem področju, in sicer pri indukciji modelov vodnih ekosistemov na podlagi obstoječega eksplicitno zapisanega znanja in dostopnih meritev, simulaciji izdelanih modelov glede na spremenjene razmere v okolju itd.

3.6.4 Problem upravljanja s projektnimi dokumentacijami

Naslednji problem s področja gradbeništva, ki smo ga obravnavali pri našem delu, je problem upravljanja s projektnimi dokumentacijami. Gre za podatkovno in informacijsko zahteven problem. Gradbenih projektov je veliko in vsakič se izdelava nova projektna dokumentacija, ki se uporablja za spremljanje stanja na objektu. Življenjski cikel dokumentacije pa je lahko zelo dolg in doživlja številne spremembe (Petrinja, 2007).

Osnovni cilji povezovanja in sledenja izvora dokumentov iz gradbeniške dokumentacije so:

- prenos informacij skozi vse faze uporabe dokumentacije brez potrebe po prepisovanju ali prerinovanju, kar zagotavlja doslednost,
- skupna uporaba enotnega kataloga in drugih standardnih informacij v vseh fazah življenjskega cikla nekega objekta,
- neposredno in trajno povezovanje projektne dokumentacije in sistemov, ki spremljajo objekt med obratovanjem ter hkrati zagotavljajo interaktivnega vpogleda v projektno dokumentacijo.

Zaradi večjega števila podizvajalcev in vzporednih informacijskih sistemov, ki pripadajo ločenim administracijskim domenam, pa pogosto prihaja do težav pri upravljanju s projektno dokumentacijo: nizka ažurnost podatkov, pomanjkanje nadzora nad spremembami, slaba dostopnost dokumentov, podvajanje in neskladnost, različne pravice posameznikov, ki urejajo dokumente. V preteklosti so določene izboljšave dosegli z uporabo centraliziranih sistemov za hranjenje dokumentacije, ki delujejo tudi na daljavo, npr. programska oprema Electronic Computer Aided Design (ECAD, spletni vir). Pri takih sistemih se vsi podatki sproti zapisujejo v eno centralizirano podatkovno bazo. To omogoča povezovanje z ostalimi informacijskimi sistemi, ki služijo za spremljanje stanja na objektih, na primer: geografski informacijski sistemi (GIS), baze tehničnih podatkov (BTP) ter poslovnoinformacijski sistemi (PIS). Izpostavljene pa so težave z medobratovalnostjo sistemov, ki morajo implementirati specifične vmesnike za uvoz in izvoz podatkov.

Številna gradbeniška podjetja pa še danes ažurirajo svoje podatkovne zbirke tako, kot so jih še pred uvajanjem centraliziranih sistemov, kar pomeni ročno vnašanje in ažuriranje podatkov, ki se podvajajo v različnih sistemih. Pogosto so spremembe navedene le v spremljajočih sistemih, ne pa tudi v projektni dokumentaciji. To ločevanje je tipično v sistemih, kjer ni enotnega in usklajenega nadzora nad dokumentacijo skozi celoten življenjski cikel objektov. Posledica so veliki stroški, ki se pojavljajo v obdobjih investicij v obnovo objektov, saj je treba zaradi nezanesljivih informacij, ki bi jih dobili neposredno iz obstoječe dokumentacije, izdelati ponovno analizo obstoječega stanja.

S pomočjo semantičnih tehnologij bi bilo v prihodnje omogočeno samodejno ažuriranje vseh tistih sistemov, kjer je določen podatek potreben. Na ta način se lahko izognemo podvajanju in možnostim napak (Petrijna, Stankovski in Turk, 2007). Tako bi tehnologije semantične mreže lahko prispevale k odpravi problemov (npr. z novimi podatkovnimi, metapodatkovnimi in informacijskimi storitvami), kot so nedostopnost dokumentov, podvajanje podatkov, pomanjkanje nadzora nad spremembami, pravice posameznikov in pravnih oseb itd. Ravno tako nam tehnologije mreže lahko omogočijo, da v sistem vključimo obstoječe programe za obdelavo besedil (Adeva in Calvo, 2006), in tako semantično povežemo porazdeljene zbirke dokumentov v celoto.

3.6.5 Združene digitalne knjižnice

Na medmrežju je vsak dan vse več digitalnih knjižnic, v katerih so shranjeni kompleksno strukturirani dokumenti. Dostop do njih je pogosto omejen zaradi intelektualnih pravic ljudi, ki so jih prispevali. Z druge strani pa so metapodatki o dokumentih praviloma dostopni prek javnih vmesnikov, npr. OAI-PMH. Tehnologije za hitro postavitev digitalnih knjižnic na medmrežju so danes na voljo in so jih prispevali, med drugimi, tudi na projektu SciX. Odkrivanje in uporaba znanja, ki je shranjeno v teh dokumentih, pa je še vedno velik izziv za podjetja in posameznike. Med različnimi možnostmi sta tudi izdelava združenih podatkovnih zbirk (Maly, Zubair in Li, 2005; Trnkoczy, Turk in Stankovski, 2006) in uporaba različnih tehnologij za avtomatsko učenje ontologij. Algoritmi za avtomatsko učenje ontologij pa so ponavadi tudi računsko zelo zapleteni. Pri našem delu smo si zastavili cilj izdelati združeno podatkovno zbirko, ki bi omogočila celovit vpogled v gradbeniške dokumente. Kot podlago pa bi lahko uporabili digitalne knjižnice, izdelane z uporabo tehnologije SciX, ter druge digitalne knjižnice, dostopne na medmrežju.

3.7 Primeri mrežnih sistemov na področju gradbeništva

V nadaljevanju si bomo ogledali dosežke dveh večjih projektov, in sicer NEESGrid in InteliGrid, ki sta uporabila tehnologije mreže pri izdelavi sistemov na področju gradbeništva.

3.7.1 Sistem NEESGrid

Organizacija Network for Earthquake Engineering Simulation (NEES) skrbi za ameriško nacionalno omrežje, ki povezuje raziskovalne organizacije s področij gradbeništva in seizmologije z visoko zmogljivimi računalniški centri in visoko tehnološko raziskovalno opremo, ki so potrebni pri znanstvenih simulacijah in analizah velikega obsega. V okviru organizacije NEES je potekal projekt NEESGrid s ciljem vzpostaviti virtualni raziskovalni laboratorij. NEESGrid je eden redkih projektov, ki je uporabil mrežne tehnologije na področju potresnega inženirstva. Zato je bilo še toliko pomembneje, da te dosežke obravnavamo in jih upoštevamo pri našem delu.

Namen povezovanja posameznih institutov in organizacij je skupinsko raziskovanje in izobraževanje prek integriranega izvajanja znanstvenih poskusov, vse to pa z namenom izboljšanja potresne varnosti. Razvita programska oprema v virtualno organizacijo povezuje 15 raziskovalnih centrov, ki so razpršeni po celotnem ozemlju Združenih držav Amerike. Centri si medsebojno delijo računske in podatkovne vire, sodelujejo v oddaljenih poskusih, nadzirajo njihovo izvajanje, upravljajo raziskovalno opremo, opazujejo potek poskusov in izvajajo različne simulacije. V tem trenutku za mrežno infrastrukturo skrbi organizacija NEESit (spletni vir), ki je del organizacije San Diego Supercomputer Center.

Tehnično gledano je sistem NEESGrid nabor visokonivojskih mrežnih storitev, razvitih z namenom povezovanja oddaljenih naprav in znanstvenikov na posameznih preizkusnih lokacijah. Programska oprema NEESGrid temelji na prilagojenem sistemskem programu Globus Toolkit, pa tudi na programski opremi Condor-G (Frey in sod., 2002). V grobem lahko mrežne storitve, ki so vključene v infrastrukturo NEESGrid, razdelimo v dve skupini. Prva vključuje decentralizirane storitve, ki so nameščene na vseh oddaljenih lokacijah, medtem ko druga zajema le centralizirane storitve, ki tečejo na eni sami lokaciji v sistemu. Storitve, ki se uporabljajo na oddaljenih lokacijah, vključujejo storitve za zbiranje in prenos podatkov, krmiljenje merilnih naprav in nadzor izvajanja poskusov. Centralizirane storitve vključujejo storitve centralne podatkovne zbirke, storitve za spremljanje in nadzor mrežnega sistema, informacijske storitve (Czajkowski in sod., 2001) in storitve za zagotavljanje varnosti (Gehrig, 2004, spletni vir).

Vzpostavljena mrežna infrastruktura NEESGrid poenostavlja integracijo raznolikih sistemov, kot so merilni inštrumenti, računalniki itd., uporabnikom pa ponuja enostaven uporabniški vmesnik. Uporabniki sistema lahko med seboj komunicirajo, načrtujejo in izvajajo poskuse na oddaljenih merilnih napravah, pri analizi podatkov uporabljajo računske vire, rezultate pa objavijo v centralizirani podatkovni zbirki in tako ostalim pooblaščenim udeležencem omogočijo njihovo nadaljnjo uporabo.

3.7.2 Semantične mrežne storitve in orodja InteliGrid

Evropski projekt InteliGrid (2004–2007) (spletni vir) je združil ter nadgradil najnovejša tehnološka in znanstvena spoznanja s področij semantične medobratovalnosti, virtualnih organizacij in tehnologij mreže z namenom zagotoviti inženirsko informacijsko platformo za prilagodljiv, varen, robusten, medobratovalen dostop do informacij, komunikacij in računskih infrastruktur. Turk in sod. (2004, a in b), Stankovski in sod. (2005), Gehre in sod. (2005) ter Dolenc in sod. (2007 in 2008) so predstavili sistemsko arhitekturo izdelane platforme ter njene osnovne komponente. Nekatere ključne tehnične zahteve pri projektu so bile povezane z zagotavljanjem varnosti, enostavnosti, stabilnosti in standardov ter upoštevanjem semantike. Za končnega uporabnika platforme InteliGrid je bistvenega pomena, da so podprti specifični koncepti, ki jih uporablja pri vsakdanjem delu (npr. stena, stavba, graditev itd.), in ne samo informacijski koncepti, ki se običajno povezujejo s tehnologijo semantične mreže. Novost konceptualne arhitekture InteliGrid je v tem, da se koncepti inženirske realnosti in informacijsko-komunikacijski koncepti obravnavajo na enak način. Platforma InteliGrid ponuja veliko prednosti za končne uporabnike in razvijalce, najbolj bistveni pa sta naslednji.

- Z uporabo mrežnih tehnologij ni več potrebna eksplicitna določitev lokacije informacij oziroma storitev. Informacije in storitve so tako dostopne na mreži, in ne na določenem internetnem naslovu.
- Z uporabo semantičnih tehnologij (metapodatkov, konceptov, definiranih v ontologijah, itd.) se zniža raven zahtevanega eksplicitnega znanja o uporabljenih virih (npr. podatkovnih modelov), izboljša dostopnost spletnih in mrežnih storitev itd.

SOA ter spoznanja, pridobljena na uspešno zaključenih sorodnih projektih (Scherer, 1997; Katranuschkov, Scherer in Turk, 2001), so bili osnovne smernice za razvoj konceptualne arhitekture platforme InteliGrid, ki je sestavljena iz štirih plasti: (1) realna plast, (2) konceptualna plast, (3) plast programske opreme in (4) plast osnovnih virov. Pri realni plasti je opis omejen z obravnavano inženirsko panogo, npr. stavba, most, organizacija, inženir, proces itd. Konceptualna plast določa ontologije, ki definirajo koncepte, uporabljene v ostalih plasteh arhitekture. Plast programske opreme predstavlja programsko opremo, ki jo je mogoče namestiti, uporabljati oziroma povezovati. Programska oprema uporablja koncepte, definirane v ontologijah, ki jih določa konceptualna plast. Plast osnovnih virov zajema vire, potrebne za delovanje programske opreme, npr. strojna oprema, računalniška omrežja itd. V okviru projekta so bile razvite številne spletne in mrežne storitve ter uporabniški programi. Celovita predstavitev komponent mrežnega sistema je dostopna na spletni strani projekta InteliGrid.

4 Analiza zahtev

V prejšnjem poglavju smo pokazali, da na področju gradbeništva obstaja potreba po prilagajanju fenomenu enkratnosti in podpori povezovanju med podjetji (npr. doseganju višje stopnje medobratovalnosti). Glavna funkcija mrežnega sistema pa je olajšati deljenje podatkov, programov, procesorjev in naprav za shranjevanje podatkov z namenom izboljšanja obstoječih aplikacij in omogočanja novih. Tak sistem mora upoštevati specifične omejitve in zahteve aplikacij v zvezi s podatki, programi ter njihovimi uporabniki. Omenjeni splošno zastavljeni cilji vodijo k razdelitvi zahtev s perspektive **uporabnikov, aplikacij ter mrežnega sistema**.

Uporabniške zahteve narekujejo potrebe:

- končnih uporabnikov pri definiranju in izvajanju različnih gradbeniških opravil,
- razvijalcev, ki izdelujejo kompleksne mrežne aplikacije,
- sistemskih administratorjev, ki morajo še naprej razvijati, razširjati in vzdrževati produkcijski sistem.

Aplikacijske zahteve so pogojene s tehničnimi dejavniki, kot so:

- vrsta virov in njihova lokacija,
- arhitektura uporabniške programske, systemske in strojne opreme,
- sistemski vmesniki,
- standardi in podobno.

Zaradi celovitosti pristopa smo si kot enega ključnih ciljev dela zastavili identifikacijo zahtev za mrežne sisteme ter načrt, implementacijo in evalvacijo mrežnega sistema, ki zadovoljuje te zahteve. V nadaljevanju po vrsti opisujemo generični primer uporabe mrežnega sistema, ki smo ga izdelali na podlagi obravnavanih primerov uporabe, ter analizo ključnih uporabniških, aplikacijskih in sistemskih zahtev.

4.1 Proces definicije zahtev

Razvoj informacijskih sistemov je logični proces pri katerem običajno sodelujejo sistemski analitiki. Tu predvsem gre za procese zajemanja in definicije uporabniških, aplikacijskih in sistemskih zahtev, načrtovanje in razvoj, vrednotenje izdelanega sistema, izobraževanje ter izbira licenčnega modela. Rezultat teh procesov mora biti informacijski sistem visoke kakovosti, ki ustreza ali presega pričakovanja uporabnikov. Informacijski sistem mora biti izdelan v predvidenem časovnem roku in v okviru predvidenih stroškov, sistem mora biti tudi učinkovito vključen v obstoječo informacijsko infrastrukturo, njegovo vzdrževanje ne sme biti drago in mora biti omogočena tudi njegova nadgradnja. Z razvojem SOA pa postajajo računalniški sistemi vse bolj kompleksni, saj je omogočeno povezovanje večjega števila tradicionalnih sistemov, ki temeljijo na različnih tehnologijah.

Proces razvoja informacijskih sistemov je podprt s strani večjega števila modelov za upravljanje z življenjskim ciklom programske opreme. To so npr. modeli: »slap«, »vodnjak«, »spirala«, »izgradi in popravi«, »stopnjejoči« itd. Pri našem delu smo upoštevali osnovne procese, ki jih določajo ti modeli, npr. definicija primerov uporabe in zajemanja uporabniških, aplikacijskih in sistemskih zahtev.

4.2 Primer uporabe

Na področju gradbeništva kot tudi na drugih industrijskih in znanstvenih področjih najdemo veliko primerov, kjer so uporabni mrežni sistemi. Na področju gradbeništva so Dolenc in sod. (2007, str. 459) na projektu InteliGrid definirali generični primer uporabe, ki smo ga v tem delu razširili in nadgradili. Primer uporabe poteka v naslednjih korakih:

Vstop v virtualno organizacijo (temelječo na infrastrukturi mrežnih storitev). Gradbeni inženir, ki ima natančno določene naloge v okviru virtualne organizacije, se prijavi v organizacijo z uporabo vmesnika, ki je čim bolj preprost za uporabo in prilagojen profilu osebe.

Iskanje virov¹, potrebnih pri delu. Vloga uporabnika v konkretni virtualni organizaciji omogoča dostop do geografsko porazdeljenih virov (dokumentov, storitev, opisov nalog, shramb, računalnikov). Izvršljivi programi in obstoječe mrežne aplikacije so tudi viri, ki jih je mogoče deliti in s katerimi se zaradi raznolikosti in kompleksnosti niso podrobno ukvarjali na projektu InteliGrid. Zato smo mrežne vire podrobno obravnavali v pričujočem delu.

Izdelava mrežne aplikacije. Uporabnik mrežnega sistema ima možnost za hiter razvoj zmogljivih in prožnih mrežnih aplikacij. Pri tem lahko uporablja obstoječe mrežne vire, ki so dostopni v okviru virtualne organizacije. Na nekaterih specializiranih področjih se pogosto razvijajo novi in boljši algoritmi, delotoki in načini obdelave obstoječih podatkov. Primer množice inženirskih programov, ki jih razvija neformalno povezana skupina razvijalcev, je OpenSees s področja potresnega inženirstva. Zato smo pri našem delu to raziskavo razširili tudi na inženirske aplikacije, ki bi bile vključene v mrežno okolje in tako postale pomemben vir, ki je deljiv v okviru virtualne organizacije. Ideja je, da bi tako kot razvijalci OpenSees, ki razvijajo svoje programe in jih objavljajo na spletu, tudi razvijalci inženirskih aplikacij oziroma podatkov svoje vire lahko prispevali v mrežno okolje in jih tudi uporabili pri izdelavi in izvajanju mrežnih aplikacij.

Izvajanje mrežne aplikacije. Uporabnik zažene mrežno aplikacijo preko vmesnika do mrežnega sistema. Uporabniški vmesniki do mrežnega sistema pa so lahko bodisi lahki (npr. mrežni portali) ali debeli (npr. urejevalniki in upravljalci delotokov).

Deljenje rezultatov. Po končani nalogi, analizi ali drugem delu uporabnik ponavadi producira vrsto novih virov (npr. dokumentov, risb, drugih podatkov, izvršljivih programov, mrežnih aplikacij), ki jih lahko deli v okviru virtualne organizacije. Te vire je treba ustrezno semantično opisati glede na obstoječo ontologijo mrežnih virov ali glede na domensko ontologijo. Kot smo že omenili, so lahko inženirski izvršljivi programi ravno tako vir v

¹ Predmet (angl. object) v terminologiji projekta InteliGrid.

okviru virtualne organizacije. Na obstoječih projektih jih niso podrobneje obravnavali, zato smo te manjkajoče segmente zajeli v pričujočem delu.

Sprožanje akcij. S tem ko nekdo v mrežno okolje vključi rezultate svojega dela, se lahko v okviru virtualne organizacije sprožijo nadaljnje akcije oziroma operacije, ki so odvisne od dogodka. V idealnem primeru bi se to dogajalo avtomatsko, glede na opise procesov. Projekt InteliGrid je v zvezi s tem definiral t. i. ontologijo poslovnih procesov. Na drugih področjih obstajajo tudi konceptualni modeli, npr. na področju podatkovnega rudarjenja je to CRISP-DM, ki definira vse faze, ki nastopajo pri procesih odkrivanja znanja iz podatkovnih zbirk (Knowledge Discovery in Databases, KDD).

Zapuščanje virtualne organizacije. Končni uporabnik zapusti virtualno organizacijo in s tem pospravi tudi vse vire, ki jih ne želi deliti.

Opisani primer uporabe je generičen, saj ga je mogoče uporabiti pri opravljanju vrste različnih nalog v gradbeniškem okolju oziroma v okviru gradbeniške virtualne organizacije. Konkretno probleme, ki bi jih na ta način lahko reševali, smo analizirali v prejšnjem poglavju o e-gradbeništvu. Definirali smo tudi nekaj konkretnih primerov uporabe, in sicer na področju tvorjenja dinamičnih virtualnih organizacij, ki so pomembne v vseh fazah graditve, računsko intenzivnih aplikacij, npr. računanje po metodi končnih elementov ali simulacije modelov stavb z namenom ocenitve potresnih tveganj, ter podatkovno in informacijsko intenzivne aplikacije, npr. upravljanje s projektnimi dokumentacijami ali simulacije na področju sanitarnega inženirstva.

V nadaljevanju natančneje obravnavamo pomembne uporabniške zahteve.

4.3 Uporabniške zahteve

Uporabniki se razlikujejo glede na svoje poznavanje gradbeniških in drugih aplikacij ter tehnologij mreže (na primer arhitekt, gradbeni inženir, menedžer). Obstaja tudi posebna vrsta uporabnikov, ki so zadolženi za vzdrževanje in nadaljnji razvoj sistema (aplikacijski in sistemski razvijalci, sistemski administratorji).

Tipičnega končnega uporabnika vodijo predvsem pojmi, vprašanja, naloge in zahteve, ki izhajajo s področja njegove aplikacije. Ni neobičajno, da končni uporabniki nimajo nobenega ali imajo zelo omejeno znanje o tehničnih podrobnostih gradbeniških in drugih aplikacij ter mrežnega sistema. Ključnega pomena pri tej vrsti uporabnikov so podatki, ki odsevajo različne pojme iz obravnavane domene ter narava same analize, ki se izvaja. Poleg večje storilnosti, zmogljivosti in učinkovitosti so zahteve takega uporabnika povezane z enostavnostjo uporabe, odzivnim časom, interaktivnostjo, prožnostjo, prilagodljivostjo ter pojmovno preprostostjo aplikacije, ki jo želi izvajati na mreži. Včasih se niti ne zaveda, da se aplikacija dejansko izvaja na mrežnem sistemu.

Na drugem koncu spektra so uporabniki, ki imajo precejšnje znanje o gradbeniških aplikacijah, tehnologijah mreže ali obojem hkrati. Takšni imajo lahko posebne zahteve glede definicije in konfiguracije samega procesa (izbiranje posebnih algoritmov, nastavljanje parametrov za izvajanje, definicija kompleksnih delotokov ter preference za uporabo točno določenih mrežnih virov pri izvajanju mrežnih aplikacij). Potresni inženir, na primer, lahko definira poseben delotok oziroma mrežno aplikacijo, ki bo preizkusila stavbo z uporabo akcelegramov različnih potresov. Uporabniške zahteve takih uporabnikov so predvsem odprtost in prožnost pri konfiguraciji in izvajanju aplikacij. Tudi pri njih so pomembne osnovne zahteve po večji učinkovitosti, zmogljivosti ter storilnosti mrežnega sistema.

Tretja kategorija uporabnikov mrežnih sistemov so aplikacijski in sistemski razvijalci ter sistemski administratorji. Ti razmeroma redko izvajajo dejanske analize. Njihove zahteve so predvsem povezane z vprašanji razširljivosti in nadgradljivosti sistema, vzdrževanja, upravljanja z virtualno organizacijo, systemske integracije in podobno.

Upoštevajoč generični primer uporabe, zgornji premislek ter konkretne primere uporabe mrežnega sistema smo sintetizirali naslednje ključne uporabniške zahteve.

Storilnost, učinkovitost, zmogljivost, nov način rabe. (1) Mrežna različica obstoječe aplikacije mora omogočati eno ali več od naslednjih koristi za uporabnika: (a) biti mora bolj

učinkovita, (b) imeti mora večjo storilnost oziroma zmogljivost, (c) ponujati mora nove možnosti oziroma nove načine uporabe aplikacije v primerjavi z njeno običajno verzijo, ki ne deluje na mreži. (2) Mrežni sistem mora biti nadgradljiv – povečevanje obsega delovanja mora omogočati preprosto dodajanje novih mrežnih virov in povečevanje števila uporabnikov ter njihovih zahtev brez izgube storilnosti.

Področje aplikacijske domene in opravil. (1) Sistem mora biti zgrajen tako, da omogoča široko množico obstoječih aplikacijskih domen, nalog in scenarijev (npr. aplikacije na področju potresnega inženirstva, sodelovanje pri izdelavi stolpnice in podobno); (2) Poleg tega mora biti dovolj prožen, da omogoča povsem nove aplikacije, ki so bile v preteklosti zunaj mrežnega okolja nepraktične ali nemogoče (npr. NEESGrid).

Enostavnost, prožnost uporabe. (1) Končni uporabniki sistema niso strokovnjaki na področju računalništva in se zanimajo izključno za uporabo aplikacije. Sistem bi morali uporabljati brez poznavanja tehnoloških podrobnosti, še posebej tistih, ki se nanašajo na uporabo mrežnih tehnologij. (2) Tehnološko zavednim končnim uporabnikom naj bi bilo omogočeno definirati, konfigurirati in parametrizirati podrobnosti samega procesa in aplikacije kot tudi različne aspekte mreže. (3) Uporabniki naj bi na preprost način poiskali obstoječe aplikacije ali storitve na mreži glede na vrsto različnih kriterijev.

Uporabniški vmesniki. Računalniške aplikacije na področju gradbeništva morajo biti prilagojene končnim uporabnikom. Programski vmesnik do glavnih komponent sistema mora kar se da skriti kompleksnost uporabljenih programov in naprav pred uporabniki. Zato morajo biti vsi uporabniški vmesniki zgrajeni tako, da so uporabniško prijazni. Idealni bi bili mrežni portali, ki nudijo dostop do

- računalniških infrastruktur,
- specializirane programske opreme,
- storitev, namenjenih medsebojnemu sodelovanju inženirjev pri gradnji objektov, načrtovanju različnih poskusov, nadzoru njihovega izvajanja ter analizi izhodnih podatkov (npr. storitev za planiranje gradnje itd.),

- storitev za vizualizacijo rezultatov različnih izračunov, postopkov itd., brkljanje in shranjevanje,
- visoke ravni varnosti, ki zavaruje dragocene podatke in opremo pred nepravilno uporabo (omogočen mora biti torej dober nadzor dostopa ter enkratno prijavljanje uporabnikov v virtualno organizacijo).

Koristni bi bili tudi mrežni portali za dostop do storitev za oddaljeno prisotnost (angl. telepresence) in oddaljeno opazovanje (angl. teleobservation), ki omogočajo vključenost gradbenega inženirja v porazdeljeni poskus, čeprav na mestu izvajanja ni fizično prisoten, ali do storitev za oddaljeni nadzor (angl. telecontrol) poteka gradnje v realnem času. Tovrstne zahteve presegajo okvir zastavljenega dela.

Spremljanje in nadzor (angl. monitoring). Uporabniki bi morali imeti možnost spremljati napredek pri izvajanju njihove mrežne aplikacije in biti sposobni ustrezno reagirati na določene spremembe pri delovanju (npr. prekinitvev tekoče aplikacije zaradi izjemnih dogodkov ali neugodnih vmesnih rezultatov).

Razširljivost. (1) Razvijalci aplikacij naj bi na preprost način prilagodili obstoječe aplikacije za uporabo v mrežnem okolju (angl. grid enabling). Pri tem procesu naj bi uporabili čim manj ali najraje sploh nobenih sprememb na obstoječih programih. (2) Sistemski razvijalci naj bi zmogli nadgraditi mrežni sistem, ne da bi pri tem posegali v njegove osnovne komponente.

Vzdrževanje in integracija. (1) Razvijalci aplikacij, sistemski razvijalci in administratorji naj bi enostavno integrirali nove aplikacije in druge nove komponente osnovnega sistema z drugimi tehnologijami (omrežja, spletne storitve, mrežne komponente, uporabniški vmesniki itd). (2) Vzdrževanje (npr. nadgrajevanje sistema, user management) osrednjega sistema in omogočenih mrežnih aplikacij naj bi bilo preprosto, tekoče in brez prekinitev.

4.4 Aplikacijske in systemske zahteve

Če želimo zadovoljiti uporabniške zahteve, predstavljene v prejšnjem poglavju, mora mrežni sistem zadovoljiti tudi vrsto dodatnih tehničnih zahtev, ki se nanašajo na podatke in programe ter na druge komponente mrežne infrastrukture. Poimenovali smo jih aplikacijske zahteve in zahteve mrežnega sistema. V nadaljevanju je predstavljen seznam najpomembnejših.

Deljenje virov in medobratovalnost. Mrežni sistem naj bi olajšal sodelovanje in deljenje ključnih virov mrežnih aplikacij, to so predvsem programi, podatki, mrežni diski in procesne enote.

Aplikacije. (1) Mrežni sistem naj bi omogočal izvajanje širokega spektra različnih programov, ki niso omejeni le na gradbeniško stroko. (2) Da bi lahko izvajali različne aplikacije na mrežnem sistemu, mora ta upoštevati zahteve, omejitve in nastavitve, ki jih definirajo uporabniki pri določeni aplikaciji. Zato mora biti vsaka aplikacija sintaktično ali semantično opisana in registrirana na mreži. To lahko dosežemo z uporabo konceptualnega modela mrežnih virov (opis v naslednjem poglavju).

Podpora parametričnim študijam. Veliko obstoječih algoritmov zahteva ponavljanje izvajanja istega procesa z različnimi parametri (za upravljanje z algoritmom) ali različne vhodne podatke (npr. datoteke, poizvedbe iz relacijskih podatkovnih baz). Tipični primeri so naloge optimizacije ali analize občutljivosti. Zato mora sistem omogočati mehanizme, ki podpirajo avtomatizirano inicializacijo programov glede na množico pravil, npr. algoritem A naj se zažene 10-krat za vse vrednosti parametra a od 2 do 20 s korakom 2 ali algoritem B naj se zažene 20-krat za vse vrednosti parametra a , ki so zapisane v datoteki B_vhodni_podatki.txt, ali algoritem C naj se zažene tolikokrat, kolikor je datotek v direktoriju C:\vhodne_datoteke\.

Posredovanje virov (angl. resource brokering) in razvrščanje poslov (angl. job scheduling). Mrežni sistem mora biti sposoben (1) priskrbeti vire, ki so trenutno na voljo glede na zahteve po izvajanju različnih poslov, (2) načrtovati izvajanje poslov z uporabo

izbranih virov ter (3) izvajati in (4) spremljati izvajanje poslov. Unikatne zahteve pri posredovanju virov in načrtovanju opravil na mrežnem sistemu zajemajo načrtovanje podatkovnih tokov, podporo nastavitvam različnih parametrov aplikacij, izbiro narave različnih poslov (področje uporabe, npr. simulacija potresa), samo tehniko modeliranja (npr. metoda končnih elementov ali metoda robnih elementov), druge semantične opise zgradbe (npr. glede na model IFC).

Popolnoma avtomatizirana agregacija virov. Da omogoči učinkovito izvajanje različnih poslov v mrežnem sistemu, mora posrednik virov avtomatično odkrivati vse obstoječe vire in izločiti tiste, ki ne ustrezajo zahtevam posamezne aplikacije.

Upravljanje z metapodatki. Zaradi velike količine široko porazdeljenih metapodatkov je treba zagotoviti možnosti za kreiranje, dostop, sledenje življenjski dobi ter destrukcijo takih virov.

Obsežno spremljanje in nadzor (angl. extensive monitoring). Semantična mreža mora vgrajevati tudi mehanizme za spremljanje in zbiranje informacij o statusu vseh poslov od začetka njihovega izvajanja (uspešen ali neuspešen začetek) in omogočiti uporabnikom, da te informacije spremljajo v času izvajanja. Dostopne informacije morajo vključevati tudi dnevnik napak (angl. error logs), naslov računalnika na Internetu, kjer se posel izvaja, itd.

Podatkovno orientirano posredovanje virov. Določeni programi, uporabni na področju gradbeništva, procesirajo ogromne količine porazdeljenih podatkov. Tipični primeri se nanašajo npr. na upravljanje s projektno dokumentacijo. Da bi omogočili učinkovito procesiranje teh podatkov, mora mrežni sistem minimalizirati prenos velikih količin podatkov prek omrežja. Dodatno mora podpirati tudi aplikacije, pri katerih ni mogoče prenašati podatkov po omrežju zaradi drugih razlogov, ne samo zaradi njihove količine, npr. zaradi varnostnih, zasebnostnih, zakonodajnih ali regulatornih razlogov. To pomeni, da mora mrežni sistem podpirati tudi takšne procese, kjer lahko program potuje k podatkom, v nasprotju z aplikacijami, kjer podatki potujejo do mesta, kjer so nameščeni programi.

Ničelna stopinja (angl. zero footprint). Mrežni sistem naj bi omogočal izvajanje neinteraktivnih (izvršljivih) programov. To so programi, ki se lahko zaženejo prek ukazne vrstice. Zato ne moremo zahtevati, da bi bili predhodno nameščeni na vse oddaljene računalnike. Namesto tega mora mrežni sistem sam zagotoviti mehanizme za prenos enega ali več izvršljivih računalniških programov, vključno z vsemi potrebnimi knjižnicami in podatki na oddaljene računalnike, kjer bodo izvajani. Poleg tega mora mrežni sistem, ko se izvajanje teh programov konča, vse izhodne datoteke kot tudi datoteke z dnevniki (dnevnik sprememb, dogodkov itd.) vrniti odjemalcu, stran oddaljenega računalnika pa počistiti, kot da se nič ni dogajalo.

Medobratovalnost. Mreža je definirana kot zbirka heterogenih in široko porazdeljenih virov, zato je treba nujno omogočiti prožen okvir za izvajanje poslov na velikem številu računalnikov, na primer na takih, ki imajo nameščene operacijske sisteme Linux in Windows. Še več, ta funkcionalnost mora biti realizirana na način, da razvijalcem mrežnih programov ni treba razvijati posebnih »ovitkov« (wrappers) ali drugih nizkonivojskih konstruktov, da svojim programom omogočijo izvajanje hkrati npr. v okoljih Linux in Windows. Če se uporabniški programi lahko izvajajo na večjem številu različnih platform (npr. Windows, Linux), mora mreža omogočati njihovo izvajanje na vseh takih računalnikih, dostopnih v mrežnem okolju. Vsak izvršljiv program ima določene minimalne zahteve po spominu, prostoru na disku ali pa po operacijskem sistemu. Tehnologije semantične mreže pa naj bi vključevale tudi mehanizme, ki bi lahko avtomatično izbirali optimalne računalnike oziroma računalniške gruče za izvajanje. Pri optimalnem odločanju bi bilo primerno upoštevati trenutno zasedenost oddaljenih gruč, število potrebnih gruč oziroma podatkovnih transferjev, geografsko razpršenost oziroma hitrost komunikacijskih omrežij itd.

Standardizacija. Da bi omogočili medobratovalnost in evolucijo mrežnega sistema (njegovega povezovanja z drugimi podobnimi sistemi), mora biti zasnovan glede na obstoječe standarde ter na standardih, ki trenutno nastajajo in sledijo odprtokodni tehnologiji. Pomembni obstoječi *de facto* standardi so npr. OGSA in WSRF. Na področju varnosti pa so pomembni standardi, ki temeljijo na sistemu kriptografije z uporabo javnih

ključev in digitalnim potrdilom, ki ustrezajo standardom X.509, WS-I Basic Security Profile, WS-Security ter WS-SecureConversation.

4.5 Netehnične zahteve

Vpeljava novih informacijskih tehnologij oziroma sistemov ne sme negativno vplivati na obstoječe delovne procese. V tem smislu obstaja vrsta netehničnih zahtev, ki jih mrežni sistem mora zadovoljiti. Glede na obravnavane primere uporabe smo poudarili nekaj pomembnejših zahtev, kot so enostavnost za uporabo in prijaznost do uporabnikov, prožnost npr. pri tvorjenju novih mrežnih aplikacij in hitrost odzivanja sistema.

Enostavnost za uporabo in prijaznost do uporabnikov. Osnovni namen, ki naj bi bil upoštevan pri načrtovanju in implementaciji sistema, temelječega na semantičnih in mrežnih tehnologijah, je poenostavitev in neobremenjevanje končnih uporabnikov z nepotrebniimi tehničnimi podrobnostmi. Gradbeni in drugi inženirji so strokovnjaki na svojem področju, kar ne pomeni, da so strokovnjaki tudi na področju mrežnih tehnologij. Zato morajo biti tovrstni detajli skriti pred uporabniki, spreminjanje določenih prednastavitev pa možno le na njihovo zahtevo.

Prožnost. Prožnost mrežnega sistema je pomembna predvsem pri različnih spremembah, ki se pogosto dogajajo na mreži. Na primer, če en uporabnik doda nekaj novih virov (podatkov, programov ali računalnikov), je zelo pomembno zagotoviti načine za njihovo preprosto vključevanje v nove kompleksne aplikacije (npr. v delotoke) s strani drugih uporabnikov.

Odzivni časi. V tem trenutku mreža ni sistem, ki bi bil primeren za aplikacije v realnem času. Na tem področju še vedno razvijajo nove prototipe in standarde. Primer projekta, ki se ukvarja z vprašanjem, kako zgraditi mrežni sistem, ki bi premagal trenutno obstoječe superračunalnike, je evropski projekt QosCosGrid. Pri takih sistemih pa je treba zagotoviti mehanizme, ki bi upoštevali vsakokratno zahtevo po zagotavljanju kakovosti storitev (npr. rezervacijo virov z določenimi lastnostmi, za določen čas in ceno).

4.6 Sinteza zahtev, ki se nanašajo na uporabo semantičnih tehnologij

Uporaba semantičnih tehnologij in ontologij lahko prispeva k reševanju vrste uporabniških, aplikacijskih in sistemskih zahtev. V nadaljevanju podajamo pregled področij, kjer bi lahko bile uporabne:

- **Modeliranje obstoječih aplikacij.** Konceptualno opisovanje obstoječih aplikacij lahko zagotovi hitrejšo predelavo in preprostejše vključevanje teh aplikacij v mrežno okolje. Posledično bi lahko lažje odkrivali in uporabljali obstoječe aplikacije.
- **Prožen razvoj novih, kompleksnih aplikacij,** ki so zgrajene modularno, npr. izhodne podatke iz faze predprocesiranja uporabimo kot vhodne podatke za modeliranje.
- **Računsko intenzivne aplikacije.** Izvajanje aplikacij v porazdeljenem okolju zahteva modeliranje računskih virov (računalnikov, shrambe), zato da lahko posrednik virov opravi ustrezno povezavo.
- **Podatkovno intenzivne aplikacije** – uporaba masivnih, porazdeljenih ali kompleksno strukturiranih podatkov. Pri kompleksno strukturiranih podatkih so uporabni tudi obstoječi domenski konceptualni modeli, kot so npr. ifcXML, eCognos, DublinCore in podobno.
- **Modeliranje dinamičnih virtualnih organizacij.** Pri tem se lahko opisi entitet (npr. človeških in drugih virov, pravice do dostopanja do virov itn.) vgradijo v mrežni sistem in se pozneje tudi upoštevajo npr. pri avtomatskem preprečevanju dostopa nepooblaščenim osebam. Posledično lahko dosežemo tudi večjo varnost virov.
- **Procesni vpogled.** Z uporabo ontologije poslovnih procesov, ki so jo zgradili na projektu InteliGrid (Gehre in sod., 2005; Stankovski in sod., 2005), bi lahko omogočili tudi vpogled v tekoče procese znotraj posameznih organizacij, ki se nanašajo na izdelavo natanko določenih produktov.

Za gradbenega inženirja je pomembno tudi, da so podprti specifični koncepti, ki jih uporablja pri vsakdanjem delu (npr. stena, stavba, graditev itd.), in ne samo informacijski

koncepti, ki se običajno povezujejo s tehnologijami semantične mreže (Turk, 2005). To posledično pomeni, da je pri razvoju semantične mreže treba upoštevati tudi domenske konceptualne modele oziroma ontologije.

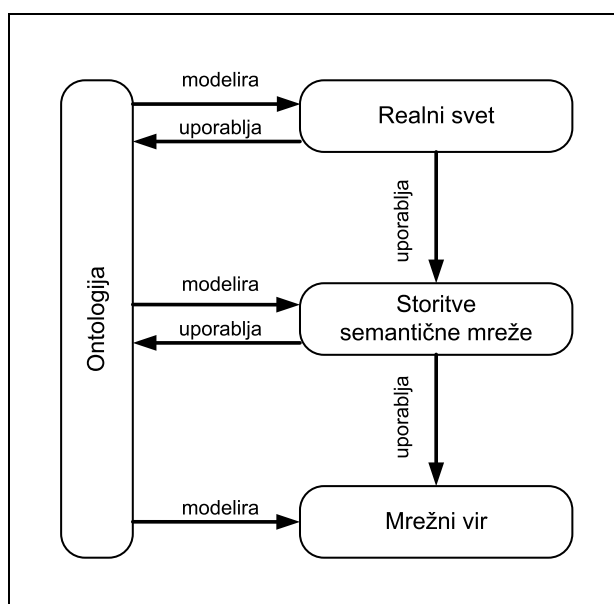
Zaradi velike količine široko porazdeljenih metapodatkov je treba zagotoviti tudi možnosti za njihovo kreiranje, dostop, sledenje življenjski dobi ter destrukcijo. Pri tem je treba upoštevati:

- **Porazdeljenost metapodatkov.** Metapodatke proizvaja in hrani vse večje število organizacij in posameznikov. Večina sistemov za upravljanje z metapodatki pa omogoča njihovo hranjenje in uporabo prek centraliziranih metapodatkovnih zbirk, tako da proizvajalci in uporabniki do njih dostopajo prek specializiranih vmesnikov.
- **Uniformni dostop.** Dostop in upravljanje z metapodatki naj bi bilo povsod omogočeno z eno samo generično storitvijo, ki izpostavlja enostavne skupne lastnosti, npr. povezavo metapodatkov z določenim virom, na katerega se ti metapodatki nanašajo, oziroma povezavo metapodatkov z entitetami znanja, na katere se nanašajo (npr. povezavo z ontologijami).
- **Evolucija metapodatkov.** Metapodatki so po svoji naravi dinamični viri, npr. beležke dokumentov pogosto spreminjajo ali nadomeščajo predhodne. Trenutne tehnologije slabo podpirajo propagacijo metapodatkov v porazdeljenih sistemih. Sistem za upravljanje z metapodatki, ki ohranja tudi podatke o njihovem stanju, naj bi omogočal odkrivanje izjem, npr. ko podatki niso več aktualni, in propagacijo spremenjenih metapodatkov do njihovih uporabnikov. Najnovejše raziskave na tem področju kažejo, da popolna avtomatizacija teh nalog verjetno ni mogoča (Missier in sod., 2007).
- **Kontrola dostopa.** Proizvajalec metapodatkov bi lahko imel tudi omejitve pri metapodatkih, ki jih proizvaja oziroma hrani. Na tem področju je bilo veliko narejeno z razvojem XACML.

5 Ontologija za sistem

Z ontologijo lahko formalno, eksplicitno specificiramo objekte, koncepte in druge entitete, ki obstajajo na katerikoli domeni, kot tudi relacije med njimi (glej Ogdenov trikotnik, slika 3-1). Ontologija med drugim lahko predstavlja model realnega gradbeniškega okolja, model storitev semantične mreže ali model virov, ki nastopajo v mrežnem okolju (slika 5-1).

Ontologija gradbeniškega okolja, lahko vključuje koncepte poslovnih procesov, produktov, ljudi ter njihovih vlog in pravic itd. Ontološki koncepti lahko specificirajo tudi namen in načine uporabe spletnih in mrežnih storitev (npr. ontologiji OWL-S in WSMO).



Slika 5-1: Razmerja med koncepti: ontologija, gradbeniško okolje (realni svet), storitev semantične mreže in mrežni vir

Fig. 5-1: Relationships among the concepts: ontology, civil engineering environment (real world), semantic grid service and grid resource

Možnosti za izboljšanje učinkovitosti obstoječih gradbeniških aplikacij temeljijo na izboljšanju izkoristka skupnih virov, novih načinov uporabe, boljši dostopnosti itd. Poleg

tega deljenje heterogenih virov na mreži lahko dejansko prispeva k razvoju popolnoma novih aplikacij, ki si jih v preteklosti nismo niti predstavljali. Potreba po deljenju računalniških virov na lokalnih in široko prostranih omrežjih (LAN, WAN) zahteva tudi, da prilagodimo obstoječe aplikacije za uporabo na mreži. Zato smo se pri našem delu odločili omejiti na specifikacijo tistih virov, ki zasedajo pomembno mesto v kompleksnem mrežnem okolju. Pri tem smo upoštevali tudi zahteve po njihovem deljenju in medsebojni obratovanosti.

5.1 Mrežni viri

Vir je karkoli, kar ima nekdo zaradi kakršnegakoli razloga za dovolj koristno, da to opiše. Naš namen pa je priporočiti strukturo za opisovanje virov, kar je eden od poglavitnih ciljev tega dela. Na začetku smo se zavestno odločili za odmik od običajnega načina definiranja virov na področju računalništva. Bežna analiza področja na veliko načinov ukazuje klasifikacije virov. Vire lahko klasificiramo kot digitalne in nedigitalne, podatke in metapodatke, tekstovne in multimedijske, statične in dinamične itd. Nedvomno je vsaka od teh razlik v določenem kontekstu koristna. Vsekakor pa ni mogoče na enostaven način delati teh razlik ali pa jih uporabljati na produktiven način, še manj pa je mogoče doseči univerzalni sporazum, da je ena razlika pomembnejša od druge, in je zato temeljna ali bistvena kategorizacija vira. Tako je razprava o mrežnih virih samo ena od razprav, ki poskušajo definirati, kaj lahko šteje kot osnovni ali temeljni razred virov (Chen, Finin in Joshi, 2004).

Glede na naravo gradbeniških aplikacij kot tudi aplikacij, ki niso povsem omejene na aplikacije gradbeniške stroke, **smo identificirali ključne vire**, skupne za organizacije in posameznike, ki nastopajo na mreži in si jih lahko delijo. Tako smo, glede na naše potrebe, definirali koncept mrežnega vira. Ontologijo mrežnih virov (Grid Resource Ontology, GRO, priloga) smo zapisali v jeziku OWL-DL. Pri delu smo uporabili orodje Protégé. Mrežni viri so lahko:

- **Podatki** za obdelavo, ki so dostopni v obliki elektronskih zbirk, npr. podatkovnih baz, podatkovnih tokov, tekstovnih dokumentov itn.

- **Izvršljivi (neinteraktivni) programi** s področja gradbeništva in širše, ki jih lahko uporabljamo za različne izračune, pri načrtovanju, projektiranju, vodenju projektov in podobno, brez povezave ali v povezavi z določenimi podatkovnimi viri. Definicija mrežnega sistema nas na noben način ne omejuje, da ne bi mogli obravnavati obstoječih izvršljivih programov kot mrežne vire.
- **Procesorji.** Enote za procesiranje, ki omogočajo (grobe, surove) računalniške zmogljivosti, potrebne za izvajanje programov.
- **Shrambe.** Naprave za fizično skladiščenje vhodnih in izhodnih podatkov različnih aplikacij, npr. med projektnim delom, modeliranjem, simulacijami in podobno.
- **Ljudje.** Strokovnjaki na različnih področjih, ki sodelujejo pri izdelavi kompleksnih produktov. Pripadajo določenim organizacijam in imajo natančno definirane pravice in obveznosti.
- **Naprave, izdelki.** Naprave in izdelki, ki nastopajo v realnem okolju, so tudi lahko viri, ki jih je treba pri določenih aplikacijah upoštevati. V mrežnem okolju lahko nastopajo tudi opisi takih objektov, kot so žerjavi, stene, hiše, stavbe itd., vendar jih zaradi njihove narave ni mogoče virtualizirati. V mrežo lahko vključimo samo njihove semantične opise.

Določene gradbeniške aplikacije pri modeliranju in simulacijah uporabljajo različne algoritme. Osrednji elementi takih aplikacij so podatki oziroma modeli, ki jih uporabljamo pri nadaljnjih izračunih in algoritmih. Pri celotnem procesu pa lahko uporabljamo večje število različnih algoritmov, ki so uporabni v različnih fazah razvoja enega kompleksnega gradbeniškega produkta (npr. stavbe).

Podana aplikacija je lahko na voljo v različnih implementacijah (algoritem, aplikacija, orodje, API). Ravno tako so tudi podatki za obdelavo lahko na razpolago v različnih oblikah, na primer kot strukturirani podatki v podatkovni bazi, datoteke (npr. dokumenti v obliki Excel, CSV) ali v obliki podatkovnih tokov, ki prihajajo iz različnih senzorjev.

Na tem mestu moramo pripomniti, da določenih virov, kot so npr. človeški viri, različni senzorji (npr. pri aplikacijah, ki se ukvarjajo s potresno varnostjo) idr., zaradi obsega in kompleksnosti v tej študiji nismo zajeli.

5.1.1 Podatki

Podatki, ki jih mrežna aplikacija potrebuje pri izvajanju, so lahko dostopni v najrazličnejših formatih, lahko so na različnih lokacijah na medmrežju (široko porazdeljeni podatki), lahko jih je količinsko veliko (masivne podatkovne zbirke) ali pa so visoko strukturirani (objektno orientirane podatkovne zbirke). Podatki vedno pripadajo eni ali večjemu številu različnih domen. Na primer, podatek, ki predstavlja opis določene stavbe, je lahko shranjen v datoteki. Z zornega kota mrežnega sistema je ta podatek datoteka, ki ima določen format, je shranjena na določeni lokaciji na mreži in jo je mogoče prenašati po mreži z uporabo različnih mrežnih protokolov (npr. RFT, GridFTP), medtem ko je z zornega kota gradbenega inženirja ta podatek model stavbe, ki ima določene strukturne elemente.

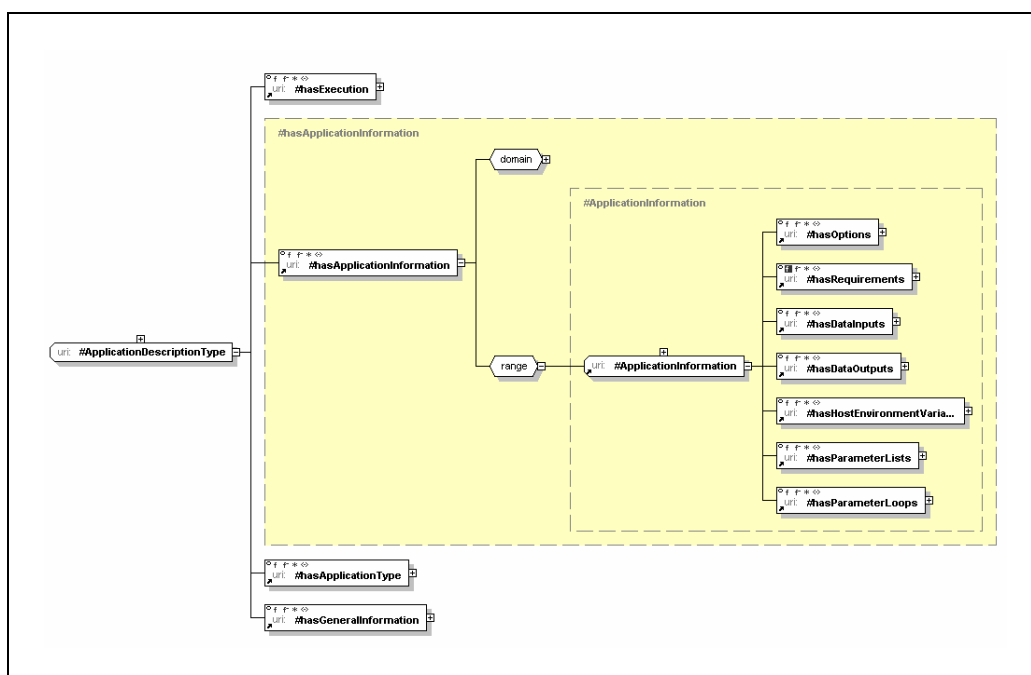
Za namene naše študije smo upoštevali predvsem zorni kot mrežnega sistema. Podatke smo obravnavali kot datoteke ali direktorije, ki vsebujejo večje število datotek (tabele iz relacijskih podatkovnih zbirk, ki jih lahko priskrbijo mrežne storitve OGSA-DAI, pa so tudi datoteke).

Pomembni dokumenti, ki se uporabljajo pri gradbenih projektih, so na primer prijava gradbišča, gradbeno dovoljenje, gradbeni dnevnik, knjiga obračunskih izmer, projektna dokumentacija itd. Največ stika z mrežno infrastrukturo pa bi lahko imeli projektanti, ki pogosto tvorijo različne datoteke (na primer v programih AutoCAD ali Word). Tehnologije, ki bi omogočale dostop do računalniške infrastrukture na gradbišču, danes še niso splošno razširjene. Pričakuje se, da se bo v prihodnosti tudi to zgodilo. Istočasno bo to povečalo pomen mrežnih sistemov.

Specializirane objektno orientirane podatkovne zbirke imajo že v svoji strukturi vgrajene semantične opise podatkov, ki jih hranijo. Primer takšne podatkovne baze je EDM podjetja EPM Technology.

5.1.2 Izvršljivi programi

Želeli smo doseči, da razvijalcem mrežnih aplikacij ne bo treba porabiti veliko razvojnega časa, da bi obstoječe izvršljive (neinteraktivne) programe in aplikacije prilagodili za izvajanje v mrežnem okolju. S tem ciljem smo definirali koncept izvršljivega računalniškega programa. Segment ontologije, ki se nanaša na ta koncept, je prikazan na sliki 5-2.



Slika 5-2: Koncept izvršljivih programov

Fig. 5-2: Concept of an executable

Denimo, da obstaja neki specializiran program, ki ga želi razvijalec mrežne aplikacije prilagoditi za izvajanje v mrežnem okolju. Na začetku lahko razvijalec ustvari opis lastnosti in zahteve programa, domene aplikacije itd. na uniformen način z uporabo ontološke definicije koncepta izvršljivega programa. V nadaljevanju lahko shrani izvršljivi program in pripadajoče knjižnice na določeni lokaciji v mrežnem okolju (na določen URI). Na koncu postopka registrira opis izvršljivega programa ter njegov URI pri informacijskih storitvah mreže. Tako lahko avtomatizirani agenti (npr. storitev posrednika mrežnih virov) in/ali ljudje poiščejo ta opis, pridejo do programa in ga uporabijo.

Definicijo izvršljivega programa smo zgradili zelo skrbno in tako omogočili podporo celi vrsti različnih mrežnih aplikacij. Z vsakim izvršljivim programom, ki ga naložimo na mrežo, se kreirajo metapodatki, ki ga opisujejo. Ta opis pa se lahko dopolnjuje tudi med izvajanjem (na polavtomatski način) in vključuje specifikacije končnega uporabnika, kot so vrednosti določenih parametrov, ki vplivajo na potek izvajanja, določitev vhodnih podatkov in izhodnih uniformnih identifikatorjev virov (URI).

V trenutku, ko je opis izvršljivega programa popolnoma specificiran, vsebuje naslednje podatke:

- Splošne informacije o izvršljivem programu, npr. narativen opis programa, prodajalec, verzija itd.
- Implementacijski vidiki aplikacije, kot je operacijski sistem, programski jezik, izvršljivi program in knjižnice, možnosti/opcije komandne vrstice in zahteve, kot so CPU-arhitektura, spomin in zahtevani prostor na disku. V času izvajanja tam, kjer je to potrebno, uporabnik lahko določi tudi zahtevani tip upravljalca poslov WS-GRAM (npr. Fork ali Condor), IP-naslove strežnikov, kjer naj se program izvaja, spremenljivke okolja (angl. environment variables) na oddaljenih računalnikih.
- Parametri, ki kontrolirajo implementirani algoritem. Tukaj so vključeni parametri, ki kontrolirajo iterativno izvajanje programa (npr. za vse vrednosti a od 1 do 100 s korakom 5), datotek in direktorijev (npr. da se program izvaja za vse datoteke znotraj enega direktorija). Informacije, ki definirajo vhodne in izhodne podatke, kot so izvir podatkov (lokacija), tip in protokol za prenos podatkov.

Na podlagi vseh teh informacij se lahko dinamično kreira tudi uporabniški vmesnik, ki je potemtakem različen za vsak izvršljiv program. Ideja je, da se opisi posameznih izvršljivih programov registrirajo pri informacijskih storitvah semantične mreže, na podlagi teh metapodatkov se nato dinamično vzpostavi uporabniški vmesnik, vsebujejo pa tudi vso potrebno informacijo o sistemskih zahtevah izvršljivega računalniškega programa. Slednja se uporabi v procesu določanja primernosti posameznih virov za izvajanje opravila. Skupaj z razvitimi informacijskimi storitvami opisna shema omogoča enostaven, uporabniku

prijazen in generičen način za vključevanje izvršljivih računalniških programov v mrežni sistem. Sklenemo lahko, da je koncept izvršljivega programa osnova za naslednje operacije:

- registracijo in iskanje izvršljivih programov, dostopnih v mrežnem okolju,
- dinamično konfiguracijo uporabniških vmesnikov glede na različne parametre in zahteve aplikacij ter
- sofisticirano upravljanje in posredovanje virov (prek posrednika mrežnih virov).

5.1.3 Ljudje (uporabniki mrežnega sistema in njihove pravice)

Uporabniki mrežnega sistema so različni strokovnjaki, npr. projektanti, izvajalci, nadzorniki, laboranti, inspektorji itd. V okviru virtualne organizacije lahko vse te osebe nastopajo kot fizične osebe ali kot zaposleni v različnih podjetjih. Glede na vlogo in naloge, ki jih imajo, so določene tudi njihove pravice. Za opisovanje oseb in njihovih pravic so tudi v preteklosti razvijali različne modele, kot je na primer model RBAC (spletni vir). Gre za plod dolgotrajnega razvoja na tem področju. Tovrstnih problemov nismo reševali, ontologijo mrežnih virov pa bi bilo treba razširiti s koncepti, ki bi podrobneje opisovali osebe in njihove vloge pri dostopanju do mrežnih virov.

5.2 Vključevanje domenskih konceptualnih modelov

Najnovejši standardni jezik za zapis ontologij OWL z uporabo mehanizma owl:imports omogoča prepletanje ontologij. Tako je omogočeno sklicevanje na ontološke koncepte iz drugih ontologij, dostopnih na medmrežju. Če želimo dodatno določiti pomen vhodne ali izhodne datoteke, programa ali implementiranega algoritma, podatkovnega toka, ki prihaja iz nekega senzorja, in podobno, se to lahko doseže s sklicevanjem na koncepte iz obstoječih domenskih ontologij, npr. IFC ali CRISP-DM. Ti ontologiji nista niti najmanj podobni in se uporabljata v povsem drugačne namene. Koncepte iz modela produktov IFC bi lahko uporabili za opisovanje in odkrivanje specializiranih gradbeniških programov in algoritmov, naloge, ki so jim namenjeni, sledenje njihovi uporabi skozi čas, optimalnejše posredovanje podatkov, ki so dodatno opisani, itd.

6 Arhitektura sistema

V tem poglavju opisujemo arhitekturo in načrt mrežnega sistema vključno z vsemi njegovimi komponentami. Pri snovanju arhitekture smo upoštevali SOA in OGSA, pri načrtovanju mrežnega sistema pa odprtokodne rešitve (GT4, OGSA-DAI, Triana, GridBus itd.). To poglavje je logično nadaljevanje poglavja o uporabniških in aplikacijskih zahtevah ter podrobno opisuje odločitve, ki smo jih morali sprejeti pri načrtovanju sistema, da bi zadovoljili te zahteve. Pri izdelavi systemskega načrta smo še posebej upoštevali ontologijo mrežnih virov in tako sledili enemu izmed zastavljenih ciljev naloge. Pri tem je šlo za iterativen proces izboljšave ontologije mrežnih virov glede na nameravane mrežne aplikacije. Upoštevali smo tudi raznolikost obstoječih programov in podatkovnih zbirk, ki smo jih želeli vključiti v mrežno okolje in izvajati v njem.

Če povzamemo ključne uporabniške, systemske in administracijske zahteve, naj bi mrežni sistem omogočal izdelavo inovativnih in zmogljivih mrežnih aplikacij, prožnost pri prilagajanju obstoječih izvršljivih programov za izvajanje na mreži, vzporedno izvrševanje poslov na optimalno izbranih računskih virih ter neoviran dostop do porazdeljenih in raznolikih podatkovnih virov (npr. relacijskih in nerelacijskih baz, datotečnih sistemov, tekstovnih zbirk). Izbira računskih virov mora potekati avtomatsko. Sistem mora torej ponuditi mehanizme, ki uporabnikom omogočajo lociranje, identifikacijo, dostop in integracijo porazdeljenih mrežnih virov, njihovo kombiniranje v kompleksne mrežne aplikacije ter izvajanje teh aplikacij (npr. gradbeniških aplikacij) na optimalno izbranih virih na mreži. Prožnost sistema in uporabniška prijaznost sta tudi zahtevani lastnosti takega sistema.

Glede na obseg različnih aplikacij (npr. programi za računanje, obdelavo podatkov, podatkovno rudarjenje itd.), ki smo jih želeli uporabljati v mrežnem okolju, pa smo morali upoštevati, da mora biti sistem čim bolj generičen oziroma ne omejen izključno na aplikacije, ki se pogosto uporabljajo pri raziskavah in delu v gradbeniški stroki.

Mrežno aplikacijo smo definirali kot delotok, ki vključuje enega ali več izvršljivih programov, vhodnih in izhodnih podatkov in storitev ter rešuje določene naloge v nekem vrstnem redu. Na primer, pri računanju elementov po metodi končnih elementov (Prelog, 1975; Dolenc, 2001) so to naslednje faze: priprava podatkov oziroma izdelava numeričnega modela, preračun po metodi končnih elementov oziroma analiza modela ter predstavitev in analiza rezultatov. Omogočanje obstoječih programov, ki podpirajo vse te faze za izvajanje v mrežnem okolju, je motivirano z izboljšano učinkovitostjo, zmogljivostjo in storilnostjo, z možnostmi za inovativne aplikacije, ki jih v preteklosti ni bilo mogoče izdelovati na preprost način, in z željo po izboljšanjem izkoristku obstoječih virov.

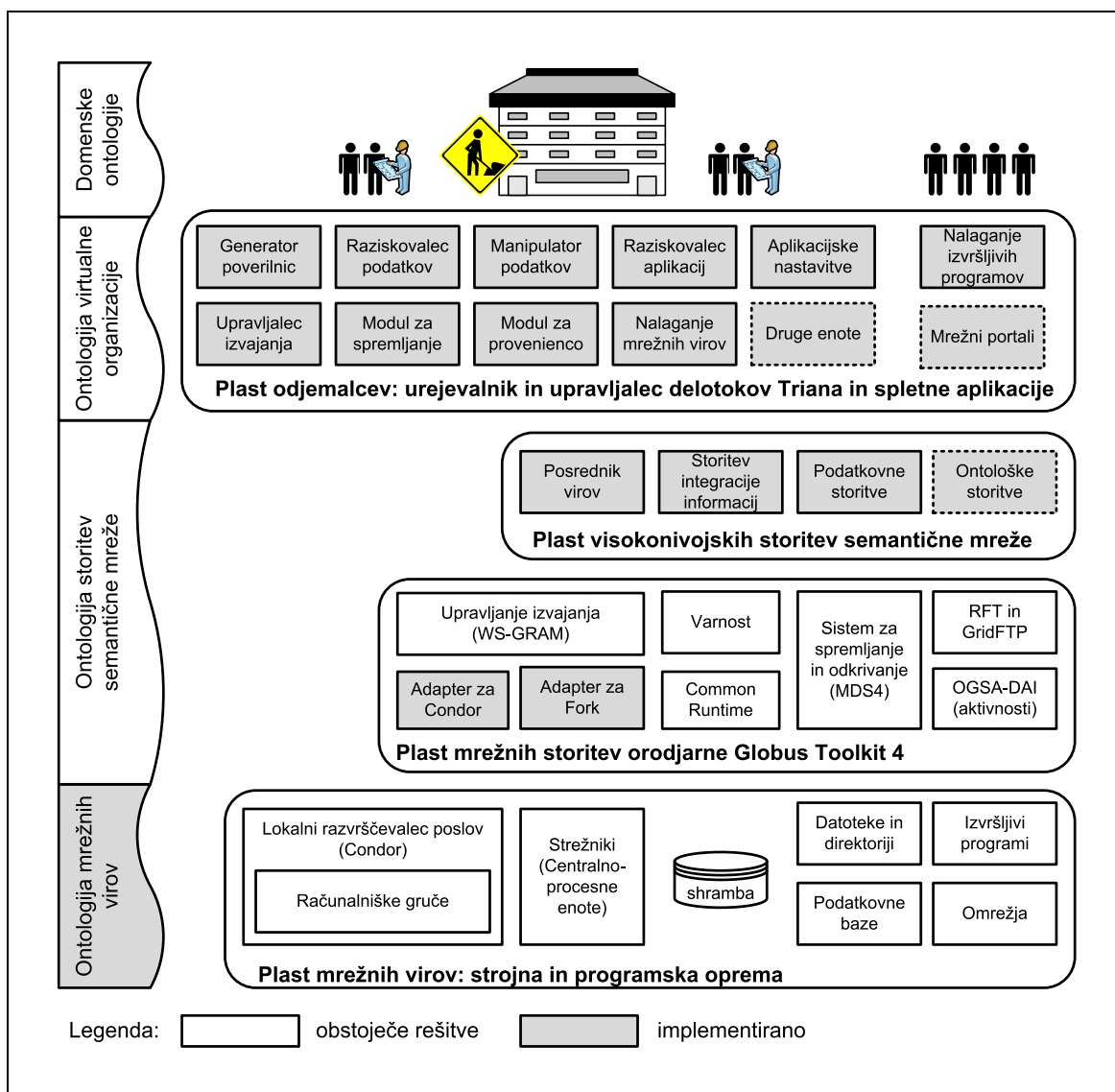
Ena od poglobitvenih zahtev pa je bila, da je vključevanje obstoječih izvršljivih računalniških programov v mrežno okolje enostavno, kar posledično olajša delo razvijalcem mrežnih aplikacij. Pomembno je tudi, da je kompleksnost mrežnega sistema skrita pred končnimi uporabniki. Ravno tako pa naj bi sistem omogočal tudi nove načine uporabe obstoječih programov.

Integracija programske opreme v mrežno okolje se običajno doseže na dva načina.

Enostavnejši je predelava izvršljivih računalniških programov v mrežne storitve, kot so to naredili Alonso in sod. (2006). Ta možnost ni primerna za računsko zahtevne aplikacije in veliko število uporabnikov. Tak način namreč ni nadgradljiv in kmalu privede do problemov z zmogljivostjo strežnika, ki storitev gosti. Zato smo se pri našem delu ukvarjali z izvajanjem izvršljivih, vendar neinteraktivnih računalniških programov, ki so bili na voljo v obliki samostojnih aplikacij, in ne v obliki storitev. Le na ta način je možna dinamična in optimalna razporeditev opravil na vire glede na trenutno stanje v mreži. Ta način odpravi tudi probleme z nadgradljivostjo sistema.

Pristop, ki smo ga uporabili, pa ni omejen samo na izvršljive programe. Izdelana ontologija mrežnih virov je uporabna tudi v primeru, ko so določene funkcije mrežnega sistema (npr. funkcija računanja pri metodi končnih elementov) dostopne v mrežnem sistemu v obliki konkretnih mrežnih storitev.

Razplaten načrt mrežnega sistema, ki smo ga izdelali, je predstavljen na sliki 6-1. Na splošno, komponente v višjih plasteh izkoriščajo komponente, ki so organizirane v nižjih plasteh. Koncepte iz ontologije mrežnih virov smo smiselno vgradili v sistem. Sistem, zgrajen na podlagi tega načrta, bo uporabnikom omogočal deljenje mrežnih virov.



Slika 6-1: Načrt semantične mreže
Fig. 6-1: Design of the semantic grid

Najnižja plast, ki je označena kot strojna in programska oprema, predstavlja plast virov, ki si jih bodo delili v okviru virtualne organizacije, to so predvsem podatki, izvršljivi programi

in podatki, dostopni na mreži, ter stroji, kot so naprave za hranjenje podatkov, enote za procesiranje in računalniške gruče.

Osrednji plasti predstavljata plast mrežnih storitev Globus Toolkit 4, ki priskrbi osrednje funkcionalnosti prilagojenega systemskega programja, in visokonivojske storitve, ki opravljajo ključne funkcije semantične mreže. Plast mrežnih storitev Globus Toolkit 4 vsebuje storitve za zagotavljanje varnosti (avtentikacija, avtorizacija, kodiranje), izvajanje opravil itd.

Visokonivojske storitve semantične mreže so: storitev posrednika virov, ki razvršča posle na trenutno proste vire; storitve integracije informacij (metapodatkov), ki hrani podatke o izvršljivih računalniških programih in podatkih, ki se trenutno nahajajo v mrežnem sistemu; podatkovne storitve, ki omogočajo dostop, preoblikovanje in integracijo različnih podatkovnih virov; ontološke storitve, ki omogočajo razumevanje mrežnega sistema in dostopnih virov. Končno, najvišja plast prikazuje različne komponente na strani odjemalcev, npr. gradbeniške in druge aplikacije, kot tudi virtualno organizacijo in realni svet, ki nastopa pri izdelavi kompleksnih produktov (hiš, mostov, stavb, letal, zdravil itd.).

Na najvišji plasti mrežnega sistema so predstavljeni uporabniški vmesniki. Predvideni uporabniški vmesniki so bodisi urejevalniki in upravljalniki delotokov (v konkretnem primeru smo uporabili Triano) ali uporabniški vmesniki, ki prevzamejo obliko mrežnih portalov. Programsko opremo Triana je bilo treba razširiti z novimi moduli, ki so predstavljene na sliki.

V nadaljevanju podrobno opisujemo vse plasti načrta semantične mreže kot tudi predvidene interakcije med njimi, predvsem pa se posvečamo tistim komponentam systemskega načrta, ki smo jih razvili sami ali pa v sodelovanju s partnerji na raziskovalnih projektih InteliGrid in DataMiningGrid.

6.1 Plast virov

Plast mrežnih virov zajema strojno in programsko opremo. Ključni cilj sistema je, da omogoča deljenje in medsebojno obratovalnost takih virov v kontekstu različnih gradbeniških aplikacij, ki smo jih opisali v drugem poglavju dela. Tipična osnovna strojna oprema (oziroma viri) so centralnoprocesne enote, primarni in sekundarni pomnilniki (npr. RAM, magnetni diski). To so ključni viri, ki se rabijo pri procesiranju in hranjenju podatkov. Pri določenih aplikacijah je količina obstoječih podatkov velika, pri drugih pa so strukture (npr. matrike), ki jih ustvarijo programi pri računanju, tako velike, da zahtevajo ogromno tovrstnih virov. Med programsko opremo pa sodijo podatki, metapodatki ter programje, ki omogoča obdelavo podatkov. Ti so shranjeni v različnih oblikah, na primer v datotekah ali relacijskih podatkovnih bazah. Računalniške gruče predstavljajo posebne vrste vire, ki omogočajo paralelno procesiranje in obstajajo praviloma v okviru ene same organizacije. Programje, ki omogoča tvorjenje računalniških gruč, je na primer sistem Condor.

6.2 Plast storitev semantične mreže

6.2.1 Posrednik virov

V našem mrežnem sistemu se posel nanaša na en neinteraktiven program (izvršljiv program in pripadajoče knjižnice), ki potrebuje podatke (podatki iz podatkovnih baz se pred uporabo lahko pretvorijo v datoteke) ter primerne računske mrežne vire za izvajanje. Multiposel (angl. multi-job) pa je definiran kot zbirka takih poslov. Primer multiposla je, ko si uporabnik želi izvajati program za izračun končnih elementov na 20 različnih datotekah. Za vsako vhodno datoteko je torej treba vnovič zagnati program. Druga možnost pa je, če obstaja določen parameter, ki pogojuje delovanje izvršljivega programa, na primer velikost nosilca. V tem primeru se lahko uporabnik odloči, da zažene program za vse vrednosti dolžine od 1 do 20 metrov s korakom 1, in tako ustvari 20 posameznih poslov, ki skupaj tvorijo multiposel (parametrična študija). Vse posle, ki jih vsebuje multiposel, pa je mogoče izvajati paralelno na mrežnem sistemu.

Posrednik virov (angl. Resource Broker) je storitev, zasnovana na podlagi programske opreme GridBus, ki temelji na odprti kodi (Buyya in Venugopal, 2004). Razširili smo jo in prilagodili konkretnim zahtevam, tako da nam omogoča optimalno razporejanje računskih poslov glede na trenutno stanje virov na mreži. Informacije o trenutnem stanju virov se pridobijo od informacijskih storitev. Posrednik virov posla jih preda izbranim strežnikom, na katerih teče storitev WS-GRAM. Izvedba enega multiposla lahko poteka na več strežnikih hkrati, v tem primeru se posamezni posli, ki sestavljajo multiposel, izvajajo vzporedno na več računalniških gručah hkrati. V primeru več zaporednih multiopravil se storitev posrednika virov zaporedno kliče večkrat.

Razvrščanje opravil na strežnike s storitvijo WS-GRAM poteka po politiki, ki jo implementira storitev posrednika virov. Ta poleg trenutne zasedenosti virov upošteva tudi velikost vhodnega podatkovnega nabora posla. V primeru velikega nabora vhodnih podatkov je namesto prenosa podatkov bolj smiselna izvedba posla na mestu, kjer se podatki nahajajo, ali v njihovi neposredni bližini. Tak pristop je pomemben tudi v primeru, ko podatkov zaradi varnostnih razlogov ne želimo prenašati zunaj domačega lokalnega omrežja.

6.2.2 Storitev integracije informacij

Informacijske storitve so utemeljene na sistemu MDS4, ki ga zagotavlja orodjarna GT4. Pri tem smo za modeliranje okolja uporabili ontologijo mrežnih virov, ki je bila predstavljena v prejšnjem poglavju. Sistem MDS4 smo nadgradili s posebno informacijsko storitvijo integracije informacij (angl. Information Integrator Service).

V mrežnem sistemu poleg ostalih virov nastopajo izvršljivi računalniški programi, ki tudi predstavljajo mrežne vire. Le-te je treba pred uporabo odkriti. Neposredna registracija informacije, ki opisuje te vrste virov v registru sistema MDS4, ni mogoča. Treba je bilo torej razviti nove mehanizme, namenjene odkrivanju obstoječih izvršljivih računalniških programov na semantični mreži.

Vsak izvršljiv računalniški program pa mora biti opisan v zadostni meri, da je omogočeno njegovo dinamično odkrivanje, razporejanje na ustrezne računske vire in izvajanje samega programa. Razvita storitev integracije informacij torej poskrbi za avtomatsko vpisovanje informacij o izvršljivih računalniških programih v porazdeljeno metapodatkovno zbirko. Ta informacija se shranjuje v obliki datotek XML, ki ustrezajo predpisani opisni shemi XML. Opisna shema XML pa temelji na ontoloških definicijah iz ontologije mrežnih virov. Torej, vhodni podatki storitve za integracijo informacij so datoteke XML, ki vsebujejo opise izvršljivih računalniških programov, ki jih želimo vključiti v mrežni sistem. Storitve je sposobna preoblikovanja te informacije v ustrezen format in registracije v sistem MDS4. Poskrbi tudi za iskanje in pretvorbo informacije v nasprotni smeri.

6.2.3 Podatkovne in metapodatkovne storitve mreže

Podatkovne storitve omogočajo neoviran dostop do porazdeljenih in raznolikih podatkovnih virov, transformacijo teh podatkov na samem podatkovnem viru in prenos rezultirajočih podatkov na dogovorjeno lokacijo. Podatkovne vire predstavljajo relacijske in nerelacijske podatkovne baze, datotečni sistemi in tekstovni repozitoriji, navsezadnje tudi ontologije in pripadajoče baze znanja. Uporabljene podatkovne storitve so bile oblikovane na tehnologijah OGSA-DAI in GridFTP. Namesto storitve GridFTP pa bi bilo mogoče uporabljati tudi storitev RFT (Madduri, Hood in Allcock, 2002). V okviru projektov InteliGrid in DataMiningGrid so bile razvite nekatere nove aktivnosti OGSA-DAI, ki so zadostile specifičnim zahtevam različnih demonstracijskih aplikacij. Za sam prenos podatkov so bile uporabljene dostavne aktivnosti OGSA-DAI, kar je primerno za prenose podatkov do velikosti 10 MB. Poleg tega tehnologije OGSA-DAI od verzije 2.0 naprej podpirajo tudi aktivnosti, ki kot prenosni protokol uporabljajo GridFTP. Ta možnost je primernejša za prenos večjih količin podatkov. Za vsako OGSA-DAI aktivnost pa se lahko razvije tudi ustrezna grafična enota za urejevalnik in upravljalec delotokov Triana. Posamezne aktivnosti je na ta način torej mogoče povezovati v delotoke.

6.2.4 Ontološke storitve

Temeljijo na tehnologijah, izdelanih v okviru projekta OntoGrid (Corcho in sod., 2006). Na tem projektu so izdelali štiri pomembne storitve, ki omogočajo delo z ontologijami, metapodatki, anotacijo ter avtomatsko sklepanje. Pri našem delu smo uporabili osnovno storitev, ki omogoča delo z ontologijami. Vanjo smo registrirali ontologijo mrežnih virov in nato smo lahko poizvedovali (z uporabo poizvedovalnega jezika Sparql) po različnih konceptih iz ontologije. Preverjanje konsistentnosti metapodatkovne zbirke je omogočeno z uporabo storitve za avtomatsko sklepanje, ki ne deluje na porazdeljen način. Storitve za avtomatsko sklepanje, ki bi delovale na porazdeljen način, pa v tem trenutku še vedno razvijajo.

6.3 Plast odjemalcev in uporabniških vmesnikov

Kot okolje za upravljanje in izvajanje delotokov smo izbrali programsko opremo Triana, ki bi jo bilo treba razširiti z ustreznimi moduli, ki predstavljajo grafične enote. Te enote so »miniaturni« odjemalci, ki komunicirajo z oddaljenimi mrežnimi storitvami, sprožajo izvajanje specifičnih procesov, ki sodelujejo pri izvedbi mrežnih opravil. Razširjena programska oprema Triana je generični uporabniški vmesnik do mrežnega sistema, prek katerega je mogoče izvajanje poljubnega delotoka oziroma poljubno kompleksne mrežne aplikacije.

Poleg dostopa do sistema prek programske opreme Triana obstaja tudi možnost razvoja uporabniških vmesnikov v obliki mrežnih portalov. Razvijamo jih lahko za posamezne specifične mrežne aplikacije. Tovrstni vmesniki niso generični, predstavljajo pa uporabnikom najprijaznejšo obliko dostopa do virov mrežnega sistema.

6.4 Virtualna organizacija in realni svet

Pri našem delu smo obravnavali predvsem možnosti za deljenje in dinamično integracijo virov ter tvorjenje virtualnih organizacij na področju gradbeništva. Inženirji gradbeništva lahko delajo na različnih področjih in v okviru različnih organizacij (samostojni podjetniki,

podjetja, državne institucije). Pri vsaki večji gradnji vedno sodelujejo inženirji gradbeništva, ki so zaposleni na različnih delovnih področjih, na primer pri nadzoru gradnje objektov, v operativi kot izvajalci gradbenih del oziroma organizatorji gradbišča, kot projektanti pri načrtovanju gradbenih objektov, v laboratoriju pri preučevanju in testiranju gradbenih materialov ali pri nadzoru investicij gradbenih del za državno upravo.

7 Prototip sistema

Ali nam izdelana ontologija mrežnih virov, razplastena arhitektura ter načrt semantične mreže omogočajo določene izboljšave, smo preverili tako, da smo zgradili prototip mrežnega sistema. Pri tem smo želeli omogočiti končnim uporabnikom, da uporabljajo mrežni sistem brez posebnega znanja o podrobnostih, ki se nanašajo na mrežne tehnologije. Sistem naj bi omogočal tudi večjo ustvarjalnost pri tvorjenju inovativnih mrežnih aplikacij. Za doseganje tega cilja je zahtevana višja stopnja abstrakcije od nižjenivojskih tehnologij, ki nam jo omogočata ontologija mrežnih virov ter tehnologija delotokov. Napredni uporabniki sistema, ki niso računalniški strokovnjaki, naj bi tako tudi imeli možnost nastopati kot razvijalci mrežnih aplikacij. V ta namen smo jim želeli omogočiti, da nalagajo in pozneje uporabljajo obstoječe izvršljive programe na mrežo z malo ali nič posegov v obstoječo aplikacijsko kodo. Medobratovalnost z drugimi mrežnimi sistemi smo želeli doseči z uporabo obstoječih in novo nastajajočih spletnih in mrežnih standardov.

Pri izdelavi prototipa smo uporabili obstoječe prilagojeno sistemsko programje ter druge odprtokodne rešitve. V sodelovanju z drugimi organizacijami smo razvili tudi geografsko široko porazdeljeno testno mrežno okolje, ki omogoča deljenje mrežnih virov v okviru virtualnih organizacij. Svoje vire so torej prispevale tudi sodelujoče organizacije. Pristop, ki smo ga izbrali, je našemu delu dal dodatno vrednost, saj smo lahko preizkušali predlagane rešitve na zelo napredni mrežni infrastrukturi, ki je temeljila na najnovejših standardih SOA, OGSA, WSRF itd. Na tako zastavljeni infrastrukturi smo zgradili in izvajali vrsto različnih kompleksnih mrežnih aplikacij s področij, kot so upravljanje s projektnimi dokumentacijami, združene digitalne knjižnice oziroma sanitarno inženirstvo.

7.1 Implementacija mrežnih storitev

Sistem, ki smo ga zgradili (slika 6-1), je utemeljen na prilagojenem sistemskem programu orodjarne GT4. Sistem Condor pa smo izbrali kot dodatno programsko opremo, ki omogoča

razvrščanje poslov na lokalnih gručah. Orodjarna GT4 je omogočila relativno enostaven razvoj novih storitev v skladu s specifikacijami WSRF. Tako kot smo domnevali, obstoječe nizkonivojske storitve orodjarne GT4 ne izpolnjujejo vseh tehničnih zahtev, zato smo razvili tudi nove visokonivojske mrežne storitve. Če je bilo le mogoče, smo to storili tako, da smo za izdelavo novih mrežnih storitev uporabili obstoječe odprtokodne rešitve, npr. v posrednik virov smo vgradili obstoječo tehnologijo GridBus. Razvili smo posrednik virov, metapodatkovne mrežne storitve ter storitev za integracijo informacij. Poleg teh smo v sistem vključili in uporabili ontološke storitve projekta OntoGrid.

Na podlagi programske opreme GridBus smo zgradili mrežno storitev – posrednik mrežnih virov (Kravtsov in sod., 2006), ki deluje na podlagi navodil za izvajanje mrežne aplikacije in opisov mrežnih virov. Glede na uporabniške zahteve in zahteve izvršljivih programov omogoča izbiro optimalnih računalniških gruč in strežnikov za izvajanje mrežne aplikacije. Nato pošlje navodilo za izvajanje mrežne aplikacije na vse izbrane računalnike, ki pripadajo virtualni organizaciji, poskrbi za prenos podatkov, izvršljivih programov in knjižnic na mesto izvajanja ter za nastavitve spremenljivk okolja na oddaljenih računalnikih. Posrednik virov omogoča tudi spremljanje in nadzor nad izvajanjem poslov. V trenutku, ko se izvajanje vzporednih poslov na oddaljenih računalnikih zaključi, posrednik virov prenese rezultate izvajanja na strežnik, ki ga je določil uporabnik, in pobriše zasedeni prostor na oddaljenih računalnikih. Pomembna izboljšava obstoječega programja GridBus je, da zdaj podpira standard WSRF.

Storitev za integracijo informacij smo zgradili z namenom, da podpira odkrivanje, karakterizacijo, spremljanje in nadzor nad različnimi mrežnimi viri in storitvami (Stankovski in sod., 2008a). Storitev integracije informacij preverja in hrani metapodatke o vseh obstoječih mrežnih virih (npr. opisi izvršljivih programov). Omogoča kreiranje registra virov z uporabo nižjenivojskega sistema MDS4, ki je vključen v GT4. Storitev za integracijo informacij omogoča tudi pregledovanje pomembnih podatkov o virih, na primer gruče, vključene v mrežni sistem, shramba in kapaciteta procesorjev, operacijski sistemi in trenutna zasedenost računalnikov. Ta storitev omogoča pregledovanje imen programov,

proizvajalcev, verzij, funkcionalne opise itd. Posrednik virov uporablja to storitev za planiranje, razporejanje in izvrševanje poslov.

Ontološke storitve, ki so vključene v testno mrežno okolje, temeljijo na predlogu arhitekture S-OGSA ter tehnologijah projekta OntoGrid, omogočajo pregledovanje konceptov in lastnosti iz ontologije mrežnih virov, ponujajo pa tudi možnost za shranjevanje instanc konceptov ter različne poizvedbe, ki uporabnikom lajšajo razumevanje kompleksnosti mrežne infrastrukture (Corcho in sod., 2006).

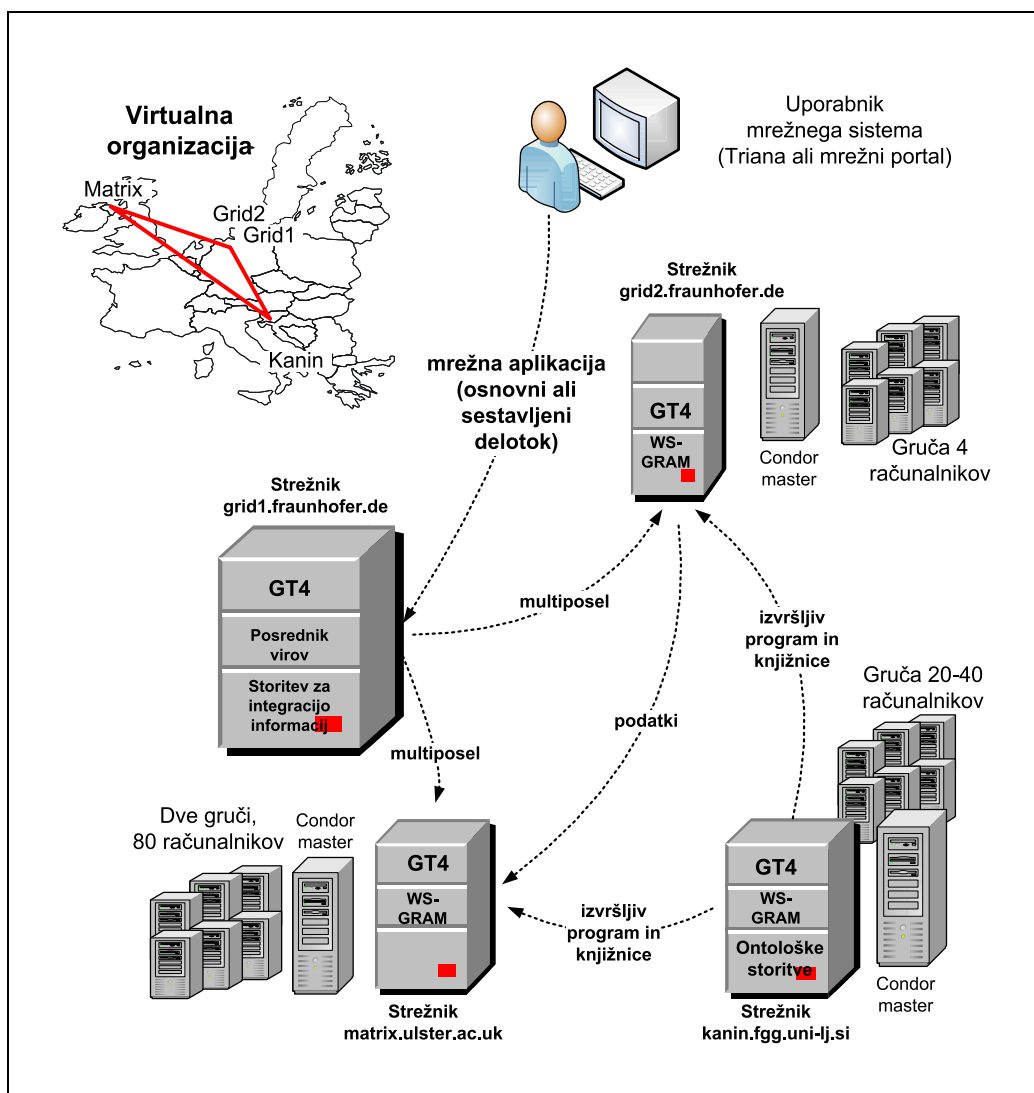
Podatkovne storitve vključujejo nekatere dodatke k storitvam OGSA-DAI, ki omogočajo dostop do relacijskih podatkovnih baz ter transformacije podatkov v formate, ki jih zahtevajo določeni izvršljivi programi. OGSA-DAI API omogoča hitro izdelavo odjemalcev za dostop do storitev OGSA-DAI. Tako smo lahko razvili zahtevane odjemalce in jih namestili v grafične enote v urejevalniku delotokov Triana. Nekateri odjemalci, ki smo jih razvili v okviru projekta DataMiningGrid, omogočajo integracijo porazdeljenih podatkovnih baz, izvajanje transformacij nad podatki (npr. iz enega formata v drugi), preproste izračune (npr. seštevanje) itd., ki pa jih pri tej raziskavi nismo uporabili. Največkrat smo uporabljali obstoječe odjemalce za dostop do storitve GridFTP oziroma njene najnovejše različice RFT, ki je zgrajena po standardu WSRF.

Ključna lastnost razvitih mrežnih storitev je, da omogočajo izvajanje **poljubnih** izvršljivih računalniških programov na optimalnih virih v mrežnem sistemu. Pri tem ni treba predelovati obstoječih izvršljivih programov v mrežne storitve, ki bi bile vključene v mrežni sistem.

7.2 Testno mrežno okolje

Slika 7-1 prikazuje testno mrežno okolje oziroma prototip mrežnega sistema, ki smo ga zgradili za evalvacijo integriranih sistemskih komponent. Okolje je sestavljeno iz treh računalniških gruč, ki temeljijo na programju Condor, ter nekaj samostojnih strežnikov. Na strežnike smo namestili obstoječe storitve prilagojenega systemskega programja GT4 vključno s sistemom MDS4, storitev WS-GRAM itd., ter novo izdelane visokonivojske

storitve: posrednik mrežnih virov, storitev za integracijo informacij, podatkovne storitve ter ontološke storitve. V sodelovanju s partnerji smo zgradili in izvedli večje število mrežnih aplikacij, ki so nam omogočile preverjanje delovanje mrežnega sistema.



Slika 7-1: Testno mrežno okolje

Fig. 7-1: Grid test bed

Tehnične značilnosti uporabljenih računalniških gruč in strežnikov ter njihove lokacije so predstavljene v preglednici 7-1. Na teh računalnikih so bili nameščeni tako operacijski sistemi Windows kot tudi Linux.

Preglednica 7-1: Osnovna konfiguracija testnega okolja

Table 7-1: Basic test bed configuration

Lokacija	Število procesorjev	Tipični procesor	Spomin
Univerza v Ulstru	50–70	Itanium II 900 MHz	2 GB
Univerza v Ulstru	4–10	Pentium 2–3 GHz	1 GB
Nemčija, inštitut Fraunhofer	4	Pentium 1.4 GHz	1 GB
Univerza v Ljubljani	20–40	Pentium 1.8 GHz	1 GB

7.2.1 Pravice, varnost in zasebnost v virtualni organizaciji

Z namenom preizkušanja varnostnih mehanizmov smo ustvarili dva overovitelja digitalnih potrdil, in sicer InteliGrid CA ter DataMiningGrid CA. Tako smo preizkusili eno ključnih zahtev na področju gradbeništva, to je možnosti za tvorjenje virtualnih organizacij. Mrežni sistem omogoča istočasno deljenje virov v okviru večjega števila virtualnih organizacij.

Varnostne storitve, ki so vključene v mrežni sistem, skrbijo za avtentikacijo uporabnikov, avtorizacijo dostopa in kodiranje prometa. Mehanizmi avtentikacije so utemeljeni na digitalnih poverilnicah standardov TLS in X.509 (spletna vira), ki jih izdajajo zaupanja vredne organizacije (npr. InteliGrid CA, DataMiningGrid CA). Sistem omogoča tudi enkratno prijavljanje, torej podpira delegacijo na osnovi posredniških certifikatov. Opisani mehanizmi so tisti, ki jih za zagotavljanje varnosti uporablja GT4.

7.3 Uporabniški vmesniki in odjemalci

Mrežno aplikacijo predstavlja več časovno zaporednih izvršljivih računalniških programov, vsak od teh pa se mora navadno izvesti z velikim številom različnih vhodnih parametrov ali različnih vhodnih podatkov. Opis mrežne aplikacije se poda v obliki delotoka. Generični uporabniški vmesnik do mrežnega sistema smo zasnovali na urejevalniku in upravljalcu delotokov Triana, ki omogoča enostavno, zelo prožno, grafično podprto kompozicijo in izvajanje delotokov. Poleg tega takšen uporabniški vmesnik skrije kompleksnost sistema pred končnim uporabnikom, npr. gradbenim inženirjem. Mrežne aplikacije namreč ni treba

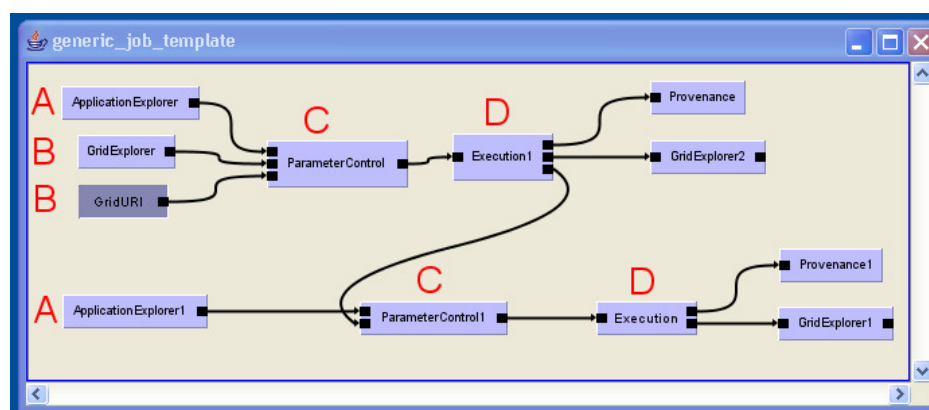
podajati v obliki opisov z abstraktnimi opisnimi jeziki, marveč se jih poda prek enostavnega grafičnega vmesnika, kar močno olajša uporabo mrežne storitvene infrastrukture.

Triana že v svoji osnovi vsebuje veliko število modulov, ki omogočajo sestavljanje in izvajanje kompleksnih delotokov, ki se izvajajo na medmrežju, ne pa tudi na mreži. Zato smo za Triano izdelali nove, dodatne module, ki omogočajo prožen dostop do mrežnih storitev ter integracijo in manipulacijo mrežnih virov. To so predvsem moduli za delo s podatki (datoteke, relacijske in XML-podatkovne baze), modul za odkrivanje in izbiro aplikacij na mreži, modul za določitev vhodnih podatkov in nastavitve parametrov mrežne aplikacije, modul za shranjevanje metapodatkov, ki se nanašajo na provenienco, itd.

Mrežna aplikacija je delotok, ki ga uporabnik sestavi s pomočjo uporabniškega vmesnika in se ob zagonu izvede na mreži. Vsak delotok je bodisi osnoven, če je namenjen izvajanju enega samega multiposla, bodisi sestavljen, če bodo rezultati enega osnovnega delotoka uporabljeni kot vhodni podatki za drugi osnovni delotok. Z drugimi besedami, sestavljeni delotok predstavlja zaporedje večjega števila multiposlov. Vsak multiposel pa pomeni vzporedno izvajanje enega samega izvršljivega računalniškega programa, vendar z različnimi vhodnimi podatki in/ali parametri. Med izvajanjem enega multiposla se praviloma izvedejo operacije dostopa in predprocesiranja porazdeljenih vhodnih podatkov, prenosa podatkov na vsa mesta vzporednega izvajanja izvršljivega programa ter združevanja dobljenih rezultatov na predpisani lokaciji.

Na sliki 7-2 je predstavljen sestavljeni delotok zaporednega izvajanja dveh multiposlov. Prikazani so različni moduli grafičnega vmesnika. Zgrajeni moduli upoštevajo tudi standard WSRF, kar omogoča komunikacijo z mrežnimi storitvami, ki tečejo v testnem okolju.

Funkcionalnosti, kot so razporejanje (in optimizacija) poslov, spremljanje izvajanja poslov, odpravljanje napak in druge so pred uporabnikom skrite in se izvedejo, ne da bi se uporabnik tega sploh zavedal.



Slika 7-2: Uporabniški vmesnik za sestavljanje delotokov
Fig. 7-2: User interface for workflow composition

Osnovni delotok, ki predstavlja en multiposel, je tipično sestavljen iz naslednjih modulov:

- **Raziskovalec aplikacij** (angl. Application Explorer) (modul A) omogoča izbiro izvršljivega računalniškega programa, ki bo osnova za multiposel.
- **Podatkovni moduli** (angl. Data Units) (moduli B) omogočajo izbiro in pripravo vhodnih podatkov, ki so dostopni na mreži.
- **Modul za nastavitev parametrov** (angl. Parameter Control) (modul C) uporabniku omogoča določanje vrednosti parametrov izbranega izvršljivega računalniškega programa.
- **Modul za izvajanje** (angl. Execution) (modul D) omogoča spremljanje in nadzor nad izvajanjem multiposla.
- **Modul za provenienco** (angl. Provenance) (modul E) zbira in shranjuje metapodatke o vsakem posameznem izvajanju mrežne aplikacije (za vsak osnovni delotok se uporablja po en modul).

7.4 Enostavno vključevanje obstoječih virov zlasti izvršljivih programov v mrežno okolje

V tem trenutku je eden pglavlnih problemov pri izgradnji mrežnih aplikacij problem vključevanja obstoječih izvršljivih programov v mrežno okolje. Mrežni viri, ki smo jih

obravnavali, so predvsem podatki, shranjeni v datotekah, direktorijih ali relacijskih podatkovnih zbirkah, izvršljivi (neinteraktivni) programi, računalniki, shramba itd. Konceptualni model mrežnih virov, ki smo ga zgradili, nam omogoča preprosto vključevanje (pa tudi odkrivanje in nadaljnjo uporabo) mrežnih virov.

Da to dokažemo, smo zgradili spletno aplikacijo, ki omogoča, da v mrežno okolje shranimo in nadalje tudi uporabljamo obstoječe izvršljive programe. Ko je program enkrat na mreži, ga lahko odkrijejo tudi posamezniki iz drugih organizacij in ga uporabljajo pri svojih mrežnih aplikacijah. Postopek poteka tako, da uporabnik najprej opiše svoj izvršljivi program in ga nato preprosto shrani na izbrano lokacijo na mreži.

Na začetku predpostavimo, da je inženir ravnokar izdelal izboljšavo obstoječega programa in ga zdaj želi vkomponirati v kompleksno mrežno aplikacijo. Na sliki 7-3 je predstavljen grafični vmesnik, ki omogoča zajem splošnih podatkov o izdelanem izvršljivem programu. V ta prototip smo vgradili minimalno možnosti za izbiro domenskih konceptov, ki se nanašajo na ta izvršljivi program, to je funkcionalno področje vgrajenega algoritma in procesna faza, kjer pride v poštev njegova uporaba. V prihodnje pa bi lahko na to mesto vgradili možnost izbire drugih domensko specifičnih konceptov iz drugih konceptualnih modelov.

Start	General Information	Execution Information	Input Data	Output Data	Requirements	Upload
Name:	<input type="text"/>	Group:	<input type="text"/>			
Version:	<input type="text"/>	CodeVersion:	<input type="text"/>	Build:	<input type="text"/>	
Vendor:	<input type="text"/>	Technique:	<input type="text"/>			
Functional Area	<input type="text" value="Classification"/>	crispDMPhase		Business Understanding		
Short Description:	<input type="text"/>					
Long Description:	<input type="text"/>					
Comment(*):	<input type="text"/>					
						Execution >>

Slika 7-3: Splošni podatki o izvršljivem programu

Fig. 7-3: General data about the executable

Slika 7-4: Metapodatki, povezani z izvajanjem

Fig. 7-4: Metadata concerning execution

V naslednjem koraku, ki je prikazan na sliki 7-4, zajamemo informacije, povezane z izvajanjem izvršljivega programa. Tukaj lahko definiramo vse vrste vhodnih podatkov, različne opcije, ki jih uporabniki programa podajo pred zagonom. Lahko se dodajo tudi nasveti v zvezi z uporabo različnih opcij, ki so uporabne pozneje, ob dinamičnem kreiranju uporabniških vmesnikov, ki omogočajo uporabo programa.

Slika 7-5: Opis vhodnih podatkov

Fig. 7-5: Description of input data

Vse vhodne podatke, ki jih pričakuje izvršljivi program, podamo v naslednjem koraku, ki je prikazan na sliki 7-5. Glede na konceptualni model mrežnih virov obstajajo le tri vrste vhodnih podatkov: direktoriji, datoteke ali datoteke, ki vsebujejo parametre.

The screenshot shows a web-based configuration interface with a navigation bar at the top containing tabs: Start, General Information, Execution Information, Input Data, Output Data, Requirements (highlighted in red), and Upload. The main content area is divided into two sections:

- Environment Variables(*):** This section features a table with columns for Name, Value, and Tooltip. Below the table are 'Delete' and 'New' buttons.
- Other Requirements(*):** This section includes four rows of configuration options:
 - Minimum Memory: [input field] MB [dropdown menu] Tooltip: [input field]
 - Minimum Disk Space: [input field] MB [dropdown menu] Tooltip: [input field]
 - Operating System: [dropdown menu with 'ALL' selected] Tooltip: [input field]
 - Architecture: [dropdown menu with 'ALL' selected] Tooltip: [input field]

At the bottom right of the form, there are two buttons: 'Save Requirements' and 'Upload >>'.

Slika 7-6: Spremenljivke okolja in druge systemske zahteve

Fig. 7-6: Environment variables and other system requirements

Nekateri izvršljivi programi niso neodvisni od platforme, ne glede na to pa mora mrežni sistem nemoteno delovati. Ker gre za heterogeni sistem, je pričakovati, da se v kompleksnem mrežnem okolju vedno najde določeno število računalnikov, diskov in drugih virov, ki ustrezajo zahtevam izvršljivega programa. Te zahteve je treba specificirati v naslednjem koraku (slika 7-6). Na primer, inženir lahko specificira vrednosti spremenljivk okolja, minimalen spomin, prostor na disku, operacijski sistem ali procesorsko arhitekturo, ki jo zahteva njegov izvršljivi program.

V zadnjem koraku, ki je prikazan na sliki 7-7, pa preprosto vključimo izvršljivi program tako, da ga skupaj z zahtevanimi knjižnicami prenesemo na določeno lokacijo na mreži. V tem trenutku se v storitvi za integracijo informacij registrira tudi instanca koncepta, ki opisuje ta izvršljiv program. Ta opis vsebuje tudi naslov (URI) lokacije programa na mreži. Potem ko je mrežni vir opisan, je ta opis registriran v okviru storitve za integracijo informacij. Od tega trenutka naprej ga vsi upravičeni uporabniki mrežnega sistema lahko poiščejo in tudi uporabijo v svojih mrežnih aplikacijah.

Start	General Information	Execution Information	Input Data	Output Data	Requirements	Upload
Uploading all the necessary files?						
Executable:		<input type="text"/>	<input type="button" value="Browse..."/>	Description(*):	<input type="text" value="null"/>	
Required Libraries(*):		<input type="text"/>	<input type="button" value="Browse..."/>	Description(*):	<input type="text"/>	
		<input type="text"/>	<input type="button" value="Browse..."/>	Description(*):	<input type="text"/>	
		<input type="text"/>	<input type="button" value="Browse..."/>	Description(*):	<input type="text"/>	
		<input type="text"/>	<input type="button" value="Browse..."/>	Description(*):	<input type="text"/>	
		<input type="text"/>	<input type="button" value="Browse..."/>	Description(*):	<input type="text"/>	
						<input type="button" value="Generate Description >>"/>

Slika 7-7: Obrazec za nalaganje izvršljivih programov

Fig. 7-7: Executable upload form

Zgrajena spletna aplikacija omogoča preprosto nalaganje obstoječih programov na kateremkoli mrežnem strežniku. Več kot 25 izvršljivih programov smo naložili na različne lokacije na mreži. Ti programi se lahko uporabljajo poljubno, npr. pri izdelavi aplikacije za združene digitalne knjižnice smo uporabili tudi program, ki omogoča avtomatsko učenje ontologij, ki ga je prispevala druga organizacija. Na enak način smo na mrežo naložili program, ki omogoča žetje metapodatkov iz oddaljenih digitalnih knjižnic.

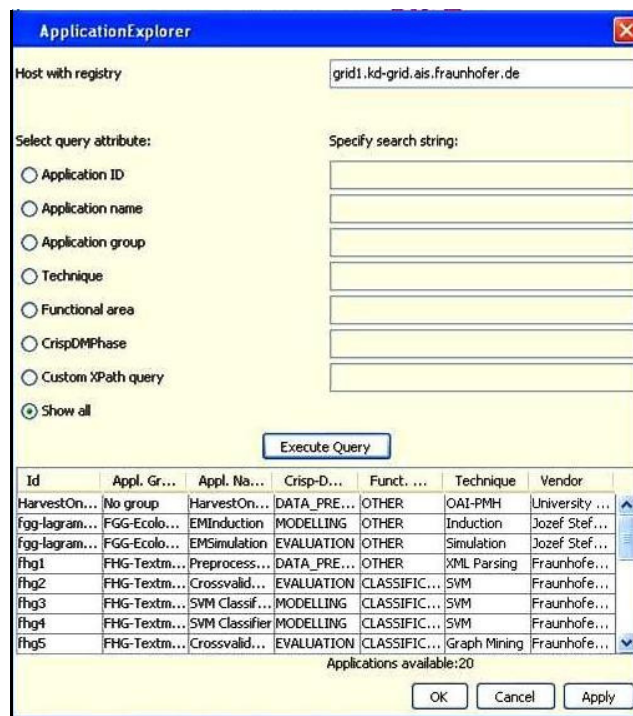
7.5 Odkrivanje in dinamična integracija mrežnih virov

Metapodatki, ki opisujejo mrežne vire, temeljijo na ontologiji mrežnih virov. Pri našem konceptualnem modelu je **izvršljivi program** eden ključnih konceptov. Instanca izvršljivega programa opisuje vse njegove invariantne lastnosti, na primer sistemsko arhitekturo, na kateri se lahko izvaja, njegovo lokacijo v mrežnem okolju, knjižnice, ki jih potrebuje za izvajanje, programske jezike, v katerih je napisan, verzijo itd. Uporabniki ne morejo spreminjati teh atributov, večinoma jih specificira razvijalec izvršljivega programa v trenutku, ko je objavljen na mreži. Instanca izvršljivega programa ravno tako vsebuje privzete vrednosti za vse opcije prihodnje mrežne aplikacije.

Medsebojno deljenje porazdeljenih virov omogočajo vgrajeni odjemalci, ki dostopajo do storitve za integracijo informacij in uporabnikom omogočajo izbiro in nadaljnjo uporabo zahtevanih virov. Na primer, napredni uporabnik, ki nastopa v vlogi razvijalca mrežne aplikacije, se lahko odloči poiskati na mreži določen izvršljivi program. Za iskanje uporabi

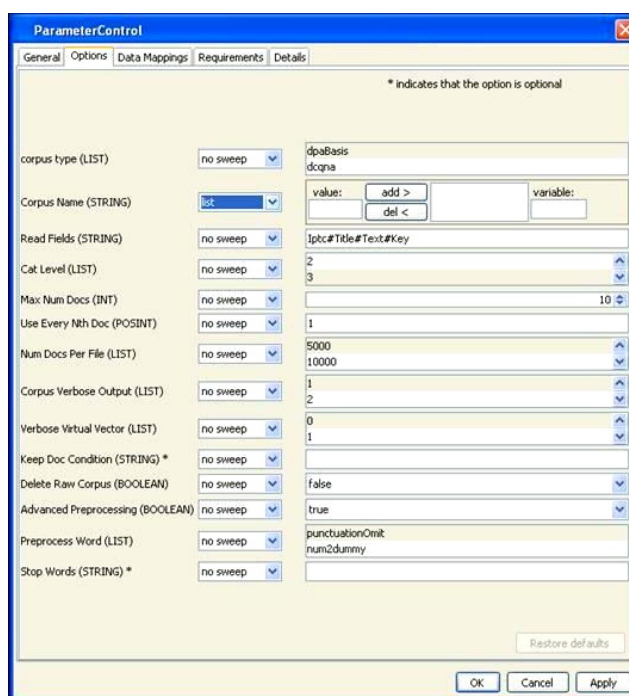
modul v Triani, ki je poimenovan kot raziskovalec aplikacij (modul A) (slika 7-8). Trenutno je omogočeno iskanje izvršljivih programov glede na področje uporabe in glede na lastnosti programa. Uporabnik vpiše iskalni kriterij, nato odjemalska komponenta kontaktira mrežno storitev za integracijo informacij, ki glede na kriterij ponudi vse primerne izvršljive programe, ki so dostopni na mreži.

Ko uporabnik izbere izvršljivi program, se glede na dostopne metapodatke avtomatsko generira obrazec, kjer lahko določi vrednosti vseh opcij, na primer vhodne podatke, parametre vgrajenih algoritmov ter dodatne zahteve programa po spominu, prostoru za shranjevanje podatkov itd. Zdaj je instanca **izvršljivega programa** popolnoma specificirana in predstavlja opis multiposla. Odjemalec ga pošlje posredniku virov, ki na podlagi opisa začne postopek izvajanja mrežne aplikacije (vzporednega izvajanja izvršljivega programa na mreži).



Slika 7-8: Raziskovalec aplikacij (modul A)

Fig. 7-8: Application Explorer (unit A)



Slika 7-9: Nastavitev opcij mrežne aplikacije (modul C)

Fig. 7-9: Grid application set-up options (unit C)

Posrednik virov uporablja metapodatke z namenom agregacije primernih virov, dostopnih na mreži. Še posebej koristni so naslednji metapodatki:

Statične zahteve po virih. To so metapodatki v zvezi s sistemsko arhitekturo in operacijskim sistemom. Programi, ki so napisani v jezikih, odvisnih od platforme, na primer C, se lahko tipično izvajajo na sistemskih arhitekturah in operacijskih sistemih, za katere so bili prevedeni (npr. PowerPC ali Intel Itanium z operacijskim sistemom Linux). Zato mora posrednik virov za izvajanje izbrati računalnike, ki omogočajo enako sistemsko arhitekturo in operacijski sistem, ki ga zahteva mrežna aplikacija. Na tem mestu bi v prihodnje lahko nadgradili koncept **izvršljivega programa** tako, da bo vseboval več lokacij (URI), kjer so shranjene različice programa, primerne za izvajanje na množici različnih operacijskih sistemov.

Nastavitev zahtev po spominu in shrambi. Znano je, da določeni programi zahtevajo zelo malo spomina in prostora na disku ob zagonu. Pri nekaterih pa se spomin in prostor na disku med izvajanjem programa povečujeta s količino podatkov, ki jih obdelujemo, ali z

velikostjo prostora možnih rešitev problema, ki ga raziskujemo. V tem smislu bi lahko končnim uporabnikom omogočili, da specificirajo te zahteve glede na količino podatkov, ki jih bodo obdelovali, ali glede na njihovo poznavanje vgrajenih algoritmov. Posrednik mrežnih virov bo upošteval zahteve, ki jih specificira uporabnik, in bo na mreži poiskal tiste računalnike, diske itd., ki bodo zadovoljevali te zahteve.

Nastavitev zahtev po določeni identiteti računskih strežnikov in gruč. V nekaterih primerih, ki so še posebno tipični za sodelovanje v okviru virtualne organizacije, si končni uporabniki na splošno želijo omejiti seznam računalnikov, na katerih se bo program izvajal. To lahko storijo glede na osebne preference oziroma glede na občutljivost podatkov. Na primer, če si gradbena firma ne želi konkurenčnemu podjetju razkriti supermodernega načrta nove stolpnice, kljub temu pa si želi z drugimi podjetji sodelovati na določenih segmentih procesa. Da bi zadovoljili to zahtevo, smo v ontologiji določili tudi koncepte, ki uporabniku omogočajo specifikacijo vseh možnih IP-naslovov. Takšen seznam omogoča, da posrednik virov poišče samo tiste dostopne računalnike, ki so v seznamu, in ignorira vse ostale ne glede na njihove zmogljivosti.

Celotno število poslov. Namesto da bi specificirali posamezne vrednosti za vsako opcijo in podatkovni vhod, ki ga zahteva gradbeniški program, obstaja možnost deklaracije določenih vrednosti (na primer, true, false) ali zank (na primer, od 0.5 do 10 s korakom 0.25). Temu pravimo multiposel. Posledično bo posrednik mrežnih virov najprej izbral tiste računalniške gruče, ki lahko izvajajo celotne sezname poslov istočasno, in tako zmanjšal potrebo po večkratnem prenosu podatkov na oddaljene gruče. Alternativa gručam pa so lahko tudi visoko zmogljivi večprocesorski računalniki. Na primer, če uporabnik specificira dve vhodni datoteki (a.txt, b.txt) za enak podatkovni vhod in dve zanki, ki se spreminjajo od 1 do 10 s korakom 1, kot parametre dveh opcij, bo posrednik mrežnih virov to prevedel v 200 (2x10x10) posameznih poslov. Če ne obstaja niti ena računalniška gruča, ki lahko istočasno izvede te posle, pa bo posrednik mrežnih virov množico poslov razbil in jih izvedel na več različnih gručah, ob tem pa bo najprej upošteval tiste, ki imajo največ nezasedenih računalnikov.

Dodatno posrednik mrežnih virov preverja informacije, dostopne v metapodatkovni instanci, ki predstavlja opis posla, ki postaja pomembna v trenutku, ko se začne izvajanje multiposla. Te informacije so na kratko razložene spodaj.

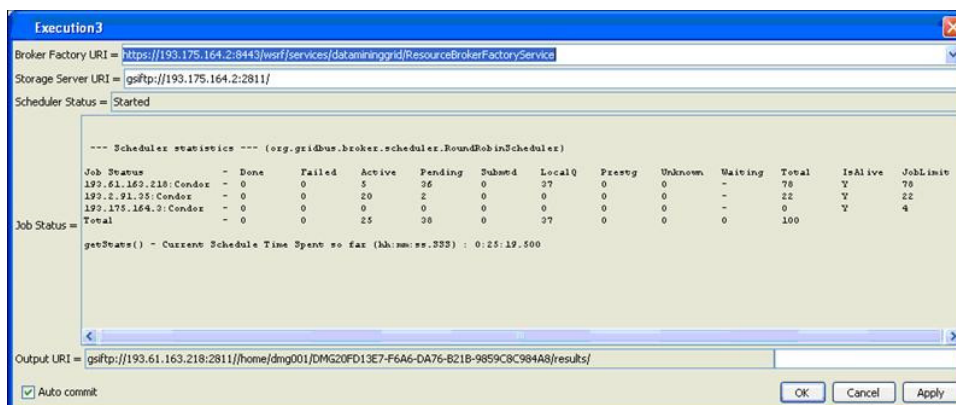
Instrukcije v zvezi s tem, kje se nahaja izvršljivi program, vključno z zahtevanimi knjižnicami, in načinom, kako se zažene. Ti podatki so zahtevani za izvedbo pripravnega prenosa. Z dostavo programov in podatkov na mesto izvajanja dinamično v času izvajanja je sistem zmožen izvajanja teh aplikacij na kateremkoli primernem računalniku, dostopnem na mreži. Za to pa ni potrebna vnaprejšnja instalacija samega programa.

Vsi vhodni in izhodni podatki, ki jih je treba prenesti na oddaljene računalnike pred izvajanjem.

Vse vrednosti opcij (parametri za izvajanje programa), ki jih je treba sporočiti izvršljivemu programu pred izvajanjem. Posrednik mrežnih virov je zgrajen tako, da izvaja izvršljive programe (batch-mode) iz komandne vrstice, in tako prenaša na program vrednosti opcij kot pare zastavica-vrednost (flag-value). Tukaj je vsaka zastavica fiksna in jo predstavlja posamezna opcija, ki jo izvršljivi program ima. Posamezne vrednosti pa se lahko spreminjajo v primeru multiposlov.

Metapodatki, ki so shranjeni v sistemu MDS4, se v času izvajanja aplikacije dinamično spreminjajo. Na sliki 7-10 je prikazan odjemalec, ki omogoča nadzor nad potekom izvajanja mrežne aplikacije, ki uporablja metapodatke, shranjene v sistemu MDS4.

Tako vidimo, da ključno vlogo v zgrajenem mrežnem sistemu igrajo metapodatki, ki mu dajejo zahtevano prožnost za vgraditev in nadaljnjo uporabo poljubnih izvršljivih programov v mrežnem okolju. To smo dosegli tako, da smo v mrežni sistem vgradili metapodatkovno infrastrukturo, ki je neodvisna od posameznih mrežnih aplikacij.



Slika 7-10: Metapodatki, povezani s potekom izvajanja (modul D)

Fig. 7-10: Execution monitoring metadata (unit D)

The screenshot shows a window titled "Provenance" with a table containing job execution details:

JobId	Status	GRAMId	SubmissionToGRA...	CompletionTime	JobDuration	FailureDescription	Variables
j0	done	GRAM01	2006-08-23T11:08...	2006-08-23T11:15...	0d:0h:7m:19s:0ms (439000ms)	No failure	\$R=1 \$OS=Linux \$B=2 \$jobname=j0
j1	done	GRAM01	2006-08-23T11:08...	2006-08-23T11:08...	0d:0h:0m:7s:990ms (7990ms)	No failure	\$R=2 \$OS=Linux \$B=3 \$jobname=j1
j2	done	GRAM01	2006-08-23T11:08...	2006-08-23T11:11...	0d:0h:3m:22s:23ms (202023ms)	No failure	\$R=3 \$OS=Windows \$B=4 \$jobname=j2
j3	done	GRAM01	2006-08-23T11:08...	2006-08-23T11:10...	0d:0h:2m:10s:20ms (130020ms)	No failure	\$R=4 \$OS=Windows \$B=5 \$jobname=j3
j4	done	GRAM02	2006-08-23T11:08...	2006-08-23T11:08...	0d:0h:0m:23s:979ms (23979ms)	No failure	\$R=5 \$OS=Linux \$B=7 \$jobname=j4
j5	done	GRAM03	2006-08-23T11:08...	2006-08-23T11:08...	0d:0h:0m:41s:0ms (41000ms)	No failure	\$R=6 \$OS=Solaris \$B=4 \$jobname=j5
j6	done	GRAM03	2006-08-23T11:08...	2006-08-23T11:10...	0d:0h:2m:12s:0ms (132000ms)	No failure	\$R=7 \$OS=Linux \$B=9 \$jobname=j6

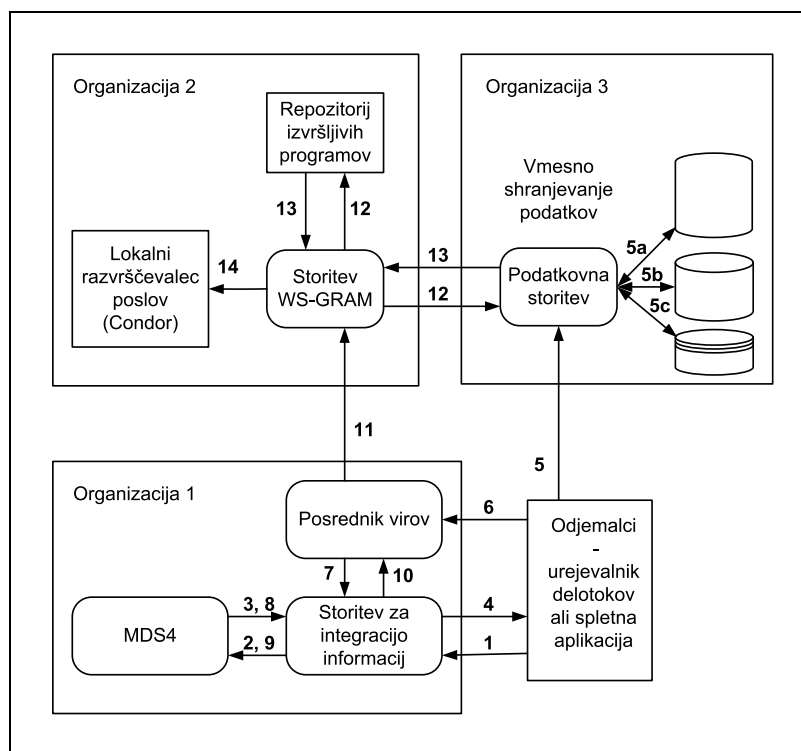
Slika 7-11: Metapodatki, povezani s provenienco (modul E)

Fig. 7-11: Provenance metadata (unit E)

V naslednjem poglavju si bomo ogledali tudi načine, ki jih ta sistem predvideva za razvoj novih, inovativnih mrežnih aplikacij.

7.6 Vpogled v delovanje sistema

Mrežni sistem, ki smo ga zgradili, je prožen, zmogljiv in nadgradljiv. Zaradi lažjega predstavljanja si lahko proces izvedbe osnovnega delotoka (enega samega multiposla) ogledamo po posameznih korakih.



Slika 7-12: Izvajanje multiposlov v mrežnem sistemu

Fig. 7-12: Multi-job execution in the grid system

Uporabnik uporabi modul A, ki služi izbiri izvršljivega računalniškega programa. Uporabnik ne more vedeti, kateri izvršljivi računalniški programi so trenutno na voljo na mreži, torej modul A o tem povpraša storitev integracije informacij, kar je na sliki 7-12 predstavljeno kot korak 1. Omenjena storitev poizvedbo usmeri na informacijski register, ki je implementiran s pomočjo sistema MDS4 (korak 2). Rezultati poizvedbe se vrnejo (koraka 3 in 4) in prikažejo uporabniku. Vrnjeni opisi izvršljivih računalniških programov, ki ustrezajo poizvedbi, vsebujejo tudi informacije, ki niso namenjene uporabnikom, temveč sistemu. Le-te opisujejo zahteve izvršljivega računalniškega programa v smislu zahtevnosti do virov, potrebnih programskih knjižnic, informacije o programskem jeziku, v katerem je program implementiran, itd. Poleg tega metapodatki vsebujejo tudi informacijo o možnih opcijah izvršljivega računalniškega programa. Ta informacija se uporabi pri dinamični izgradnji uporabniškega vmesnika modula C, ki uporabniku omogoča nastavitve vseh opcij in vhodnih podatkov.

Pri iskanju, predprocesiranju in združevanju vhodnih podatkov multiposla, ki so v porazdeljenem mrežnem okolju, uporabnik uporabi module tipa B. Dostop in prenos podatkov omogočata tehnologiji OGSA-DAI in GridFTP. Na sliki so te operacije predstavljene kot korak 5.

Po izbiri izvršljivega računalniškega programa in vhodnih podatkov uporabnik v modulu C določi ustrezne vhodne parametre izvršljivega računalniškega programa. Grafična podoba modula C se dinamično prilagaja glede na metapodatke o programu, ki povedo, katere vhodne parametre nudi. Tukaj je razvidno, da je konceptualni model mrežnih virov, ki smo ga zgradili, zelo koristen, saj omogoča prožno sestavljanje grafičnega vmesnika za najrazličnejše izvršljive programe. Modul C ponuja možnost nastavitve vzporednega izvajanja izbranega izvršljivega računalniškega programa. Uporabnik namreč lahko izbere cel nabor vrednosti za vsak vhodni parameter programa. Za vsako kombinacijo izbranih vrednosti se pri modulu D sestavi in izvede samostojen posel. Ker je teh poslov lahko več, je njihova množica multiposel.

Izbiri vhodnih parametrov izvršljivega računalniškega programa sledi izbira parametrov izvajanja aplikacije na mreži. Te se izberejo v modulu D, modul pa nato izdelava podrobni opis samostojnih poslov, ki ga preda storitvi posrednika virov (korak 6). Posrednik virov posle v multiposlu optimalno razvrsti na trenutno proste računalnike. Razvrščanje vrši na podlagi informacij, ki jih pridobi od storitve za integracijo informacij (koraki 7–10). Posli se posredujejo naprej storitvam WS-GRAM, ki jih izvedejo na lokalnih računalniških gručah (korak 11) z uporabo adapterja za Condor ali na strežniku z uporabo mehanizma Fork. Storitve WS-GRAM poskrbi tudi za prenos potrebnih vhodnih podatkov na izbrane računalnike. V ta namen kontaktira storitve GridFTP, ta pa prenese podatke in kodo izvršljivega računalniškega programa na želena mesta (koraka 12 in 13). Ko se vsi vzporedni posli, ki tvorijo multiposel, zaključijo, modul D uporabniku vrne naslov, na katerem so rezultati. Poleg tega so uporabniku na koncu vsakega multiposla na voljo podatki o poteku njihovega izvajanja (provenienca).

Zgoraj opisani koraki predstavljajo procese, ki potekajo ob izvedbi enega samega multiposla. Izhodni podatki takšnega multiposla so vhodni podatki multiposla, ki sledi. Tak primer je prikazan na sliki 7-2. V preglednici 7-2 pa je podan tudi osnovni algoritem, ki ga uporablja posrednik mrežnih virov (Kravtsov in sod., 2006). Prikazane so tudi njegove interakcije s storitvijo za integracijo informacij.

Preglednica 7-2: Algoritem za razvrščanje posrednika mrežnih virov

Table 7-2: Scheduling algorithm of the resource broker

```
Function Schedule(multi-job)
    // Množica R vseh dostopnih računskih virov (WS-GRAM)
    // sfiltriramo nekompatibilne računalnike
     $M = \{R_1, R_2, \dots, R_m\} \subseteq R$ 
    //  $|R_i|$  je število dostopnih procesorjev v prostem teku
    Uredimo M v padajočem vrstnem redu glede na  $|R_i|$ 
    // T je seznam gruč (WS-GRAM), ki jih uporabljamo pri izvajanju multiposla
    unsubmitted_jobs  $\leftarrow$  multi-job; T  $\leftarrow$  {}
    DO WHILE unsubmitted_jobs  $\neq$  {}
        // izberi WS-GRAM z največjo kapaciteto
         $R_h \in M, \forall j \neq h, |R_j| \leq |R_h|$ 
        Pošlji  $\min(|R_h|, |\text{unsubmitted\_jobs}|)$  poslov na  $R_h$ 
        // Odstrani  $R_h$  iz seznama M in ga shrani na začasni seznam T
         $M \leftarrow M \setminus R_h$ ;  $T \leftarrow T \cup R_h$ 
        Odstrani že razvrščene posle iz seznama unsubmitted_jobs
        IF M = {} AND unsubmitted_jobs  $\neq$  {} THEN
            M  $\leftarrow$  T; T  $\leftarrow$  {}
        END
    END // while loop
    RETURN success
END // Function Schedule
```

8 Evalvacija

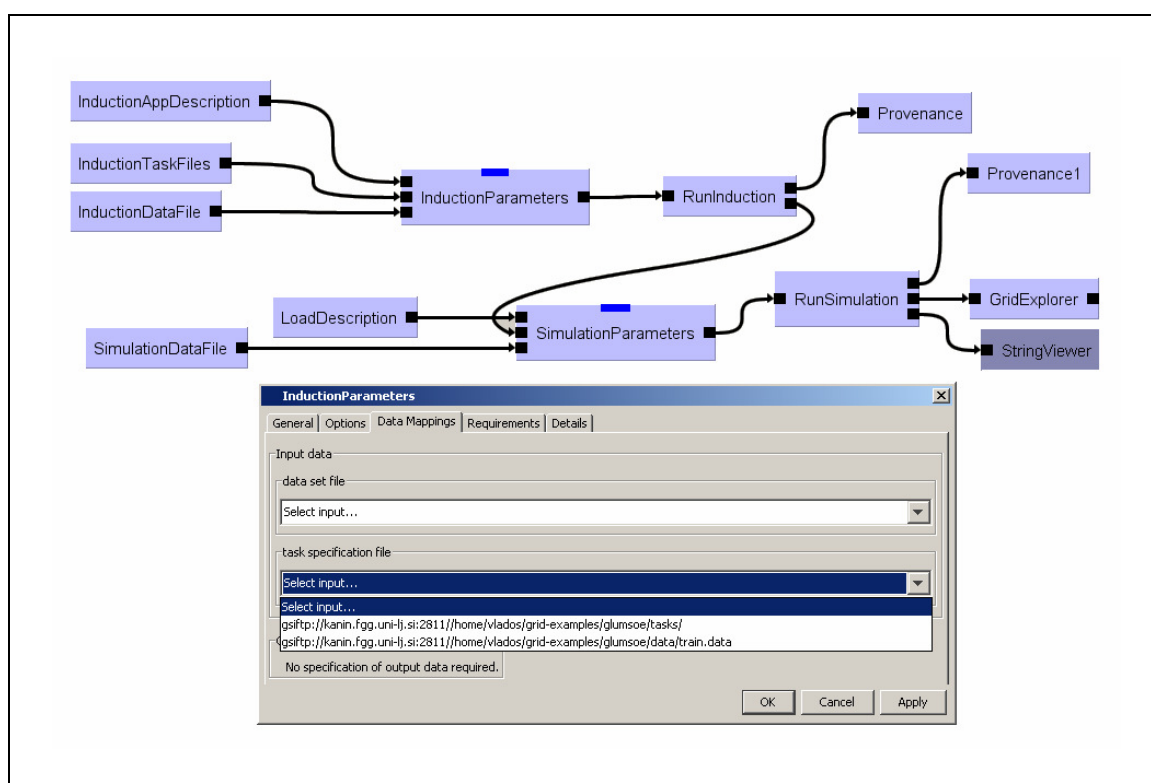
V tem poglavju evalviramo uporabnost konceptualnega modela mrežnih virov, visokonivojskih mrežnih storitev, ki smo jih zgradili, zgrajenih odjemalcev in uporabniških vmesnikov ter mrežnega sistema v celoti pri odgovoru na različne uporabniške, aplikacijske in sistemske zahteve. Podrobneje si bomo ogledali pozitivne značilnosti testnega okolja, ki so posledica uporabe semantičnih tehnologij.

8.1 Prožnost za razvoj novih, inovativnih mrežnih aplikacij

Testno mrežno okolje, ki smo ga razvili, podpira razvoj novih, inovativnih mrežnih aplikacij. Do funkcij mrežnega sistema lahko uporabniki dostopajo prek dveh različnih vrst vmesnikov, prek urejevalnika delotokov ali prek mrežnih portalov.

8.1.1 Izdelava kompleksnih mrežnih aplikacij

Ko uporabnik poseže po urejevalniku delotokov Triana, lahko z uporabo odjemalcev izbira poljubne, na mreži dostopne izvršljive programe in jih vgrajuje v svojo mrežno aplikacijo. Ta se mu kaže kot delotok, ki je sestavljen iz večjega števila osnovnih delotokov. Na ta način uporabniki, ki imajo določeno znanje o mrežnem sistemu, določijo kompleksne mrežne aplikacije in jih zaženejo preko urejevalnika in upravljalca delotokov. Na tem mestu podajamo primer mrežne aplikacije s področja sanitarnega inženirstva. Mrežna aplikacija je delotok, ki je sestavljen iz dveh osnovnih delotokov oziroma dveh zaporednih multiposlov. Prvi osnovni delotok se nanaša na proces indukcije ustreznih matematičnih modelov na podlagi različnih meritev. V naslednjem koraku se za vsak zgrajeni matematični model zažene tudi simulacija.

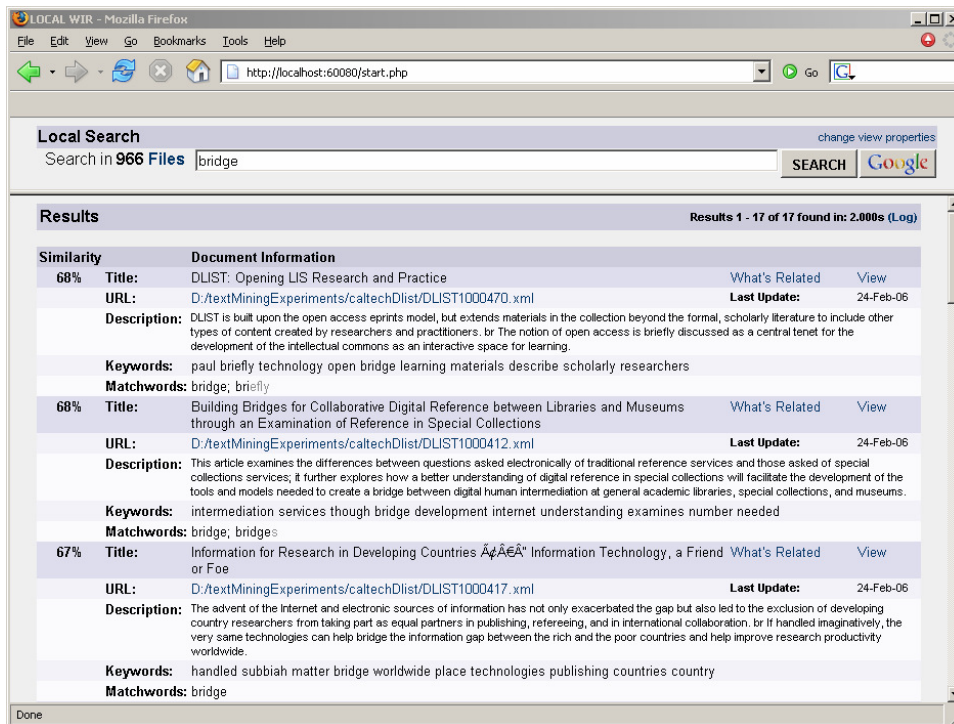


Slika 8-1: Mrežna aplikacija s področja sanitarnega inženirstva
Fig. 8-1: Grid application from the sanitary engineering domain

Zgradili smo in izvajali tudi mrežne aplikacije z drugih področij, kot so na primer bioinformatika, biomedicina ali avtomatsko učenje ontologij na področju avtomobilske industrije.

8.1.2 Izdelava mrežnih portalov

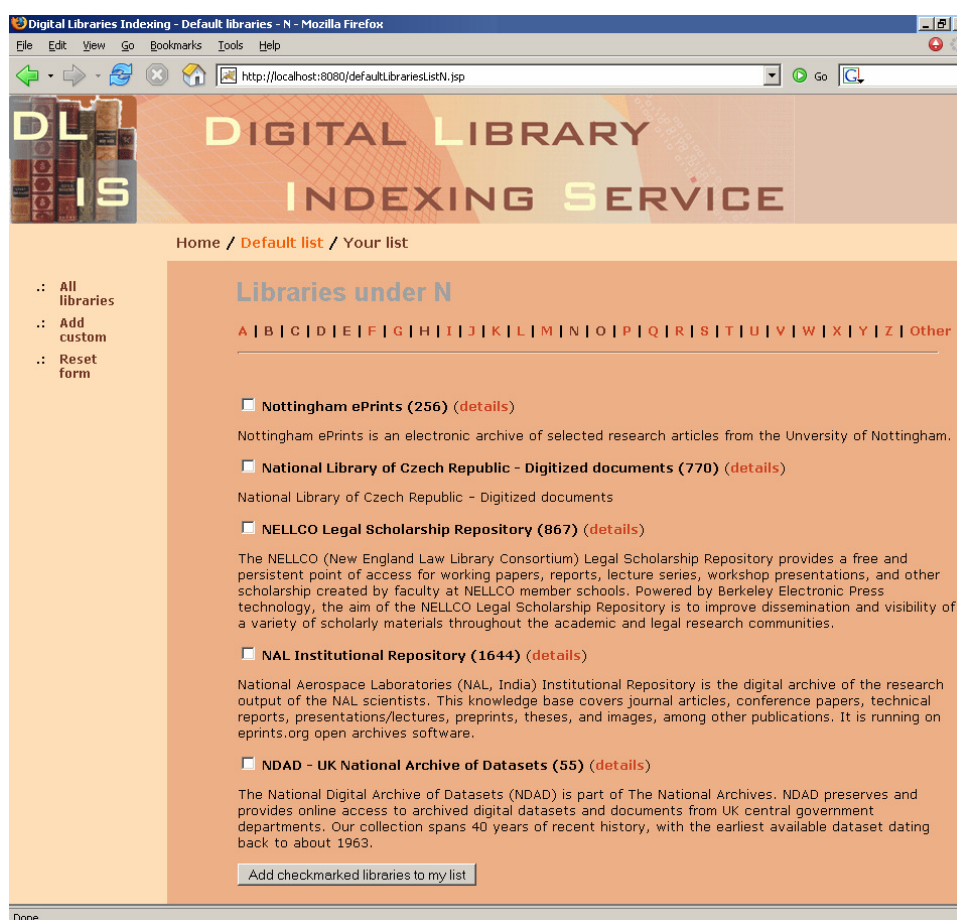
Manjšo prožnost, zakrivanje tehnoloških značilnosti mrežnega sistema ter večjo uporabniško prijaznost dosežemo, če se odločimo za izdelavo mrežnega portala (glej sliko 6-1). V zvezi s tem smo zgradili aplikacijo na področju upravljanja s projektnimi dokumentacijami, ki ima vnaprej definiran uporabniški vmesnik (slika spodaj). V tem primeru poteka dostop do funkcij mrežnega sistema prek portala, uporabnik pa lahko pri izvajanju mrežne aplikacije spreminja samo nekatere parametre multiposlov.



Slika: Semantično iskanje po dokumentih iz gradbeniških zbir (povzeto po: Trnkoczy, Turk in Stankovski, 2006)

Fig.: Semantic search in documents from civil engineering databases (according to: Trnkoczy, Turk, Stankovski, 2006)

Poleg dostopa do sistema prek programske opreme Triana obstaja tudi možnost razvoja uporabniških vmesnikov v obliki spletnih portalov. Tovrstni vmesniki niso generični, so pa uporabnikom najprijaznejša oblika dostopa do virov mrežnega sistema. Primere takšnih portalov na področju upravljanja s projektnimi dokumentacijami oziroma združevanja metapodatkov iz oddaljenih digitalnih knjižnic so predstavili Trnkoczy, Turk in Stankovski (2006).



Slika: Mrežni portal za združene digitalne knjižnice (povzeto po: Trnkoczy, Turk in Stankovski, 2006)

Fig.: Grid Portal for Federated Digital Libraries (according to: Trnkoczy, Turk, Stankovski, 2006)

Razvijamo jih lahko za posamezne specifične mrežne aplikacije. Zgradili smo primer mrežnega portala, ki omogoča izbiro in vzporedno pridobivanje metapodatkov iz geografsko porazdeljenih digitalnih knjižnic preko protokola OAI-PMH (slika zgoraj). Tovrstni vmesniki niso generični, predstavljajo pa uporabnikom najprijaznejšo obliko dostopa do virov mrežnega sistema.

8.2 Upoštevanje pravic uporabnikov

V mrežnem sistemu so zagotovljeni strogi varnostni mehanizmi, ki vključujejo nadzor dostopa do storitev in virov. Kontrola dostopa je implementirana konsistentno nad vsemi

storitvami, kodiranje in prijavljanje pa samo v primeru kritičnih aplikacij. Sistem, ki smo ga zgradili, uporablja le najbolj osnovne mehanizme za vzpostavljanje virtualnih organizacij, kar vključuje upravljanje uporabniških imen, skupin in njihovih vlog v sistemu ter kontrolo dostopa uporabnikov in skupin.

Vsak uporabnik potrebuje proxy potrdilo, ki predstavlja kopijo uporabnikovega originalnega X.509 potrdila s časovno omejitvijo. Upravljalet delotokov omogoča preprosto kreiranje in uporabo takih proxy potrdil preko generatorja poverilnic. Če ima uporabnik na voljo več digitalnih potrdil lahko izbira katerega izmed njih bo uporabil. Uporabnik lahko sam določi časovno omejitev potrdila (12 ur, 24 ur, 1 teden, 1 mesec). Generator poverilnic se uporablja po vsakem preteku veljavnosti proxy potrdila.

8.3 Povečanje zmogljivosti, hitrosti in nadgradljivosti sistema

Visoka računska zmogljivost, prepustnost ter nadgradljivost so znane lastnosti mrežnih sistemov (Grama, Gupta in Kumar, 1993; Kumar in Gupta, 1994). Istočasno so to tudi tipični razlogi, zaradi katerih inženirji in znanstveniki uporabljajo mrežne sisteme.

Tudi mrežni sistem, ki smo ga zgradili, je zmogljiv in nadgradljiv, kar smo preverili in potrdili pri izvajanju različnih mrežnih aplikacij (Stankovski in sod., 2008, a in b; Trnkoczy in Stankovski, 2008). Te lastnosti sistema smo preizkusili na tradicionalen način, tako da smo analizirali dostopne metapodatke o izvajanju posameznih poslov na različnih računalnikih v sistemu ter izračunali vrednosti parametrov: *pohitritev* (angl. speed-up) in *nadgradljivost* (angl. scale-up).

Naj bo N število računalnikov, ki smo jih uporabili pri izvajanju mrežne aplikacije. Naši izračuni kažejo, da je *pohitritev* sistema odvisna od števila N približno linearno (*pohitritev* $\sim 1/2 N$, ko je multiposel sestavljen iz večjih poslov, oziroma *pohitritev* $\sim 1/3 N$, ko je multiposel sestavljen iz manjših poslov) (glej preglednico 8-1).

Preglednica 8-1: Pohitritev sistema

Table 8-1: System speed-up

Število računalnikov [N]	Pohitritev (mali posli) [%]	Pohitritev (veliki posli) [%]
1	0,00	0,00
10	5,14	3,51
50	22,71	15,73
100	45,20	30,08

Nadgradljivost sistema pa je pomembna, če želimo, da ga uporablja večje število uporabnikov oziroma da podpira sočasno izvajanje večjega števila mrežnih aplikacij. Naši rezultati kažejo, da se s povečevanjem števila dostopnih računalnikov v mrežnem okolju ravno tako povečuje tudi število poslov, ki jih lahko sočasno izvaja mrežni sistem (glej preglednico 8-2). Pri tem so izgube zelo majhne. Če izvajamo večje posle, se odzivni čas sistema podaljša za približno 3,2 %, pri manjših poslih pa za 19,5 %, kar je razumljivo, saj posamezni posli trajajo zelo kratek čas.

Preglednica 8-2: Nadgradljivost sistema

Table 8-2: System scale-up

Število računalnikov [N]	Število poslov	Čas (veliki posli) [s]	Čas (mali posli) [s]
10	10	4715	695
50	50	4771	745
100	100	4867	831

8.4 Nižje- in višjenivojske funkcionalnosti izdelanega prototipa

Testno okolje, ki smo ga razvili, omogoča vrsto funkcionalnosti, med katerimi so najpomembnejše naslednje:

- Zmožnost izvajati vrsto različnih aplikacij na kateremkoli mrežnem strežniku ali računalniški gruči, ki je priklopljena na mrežo prek instalacije GT4.

- Metarazvrščanje poslov omogoča dinamično in avtomatsko alokacijo optimalnih računskih strežnikov v mrežnem okolju, kar se doseže prek uporabe storitve posrednika virov, storitve za integracijo informacij in sistema MDS4.
- Prenos programov, knjižnic in podatkov prek različnih administrativnih domen, kar se doseže prek mrežnih storitev GridFTP in RFT.

8.5 Povzetek lastnosti izdelanega prototipa

Na tem mestu lahko povzamemo, da mrežni sistem, ki smo ga zgradili in preizkusili, temelji na tehnologijah semantične mreže in ima naslednje lastnosti:

- omogoča enostavno vključevanje obstoječih virov (zlasti izvršljivih programov) v mrežno okolje (glej podpoglavje 7.4),
- omogoča izdelavo inovativnih, zmogljivih in kompleksnih mrežnih aplikacij oziroma nove načine rabe obstoječih virov (glej podpoglavje 8.1),
- podpira tvorjenje dinamičnih virtualnih organizacij in omogoča relativno varen dostop do mrežnih virov (glej podpoglavje 8.2),
- uporablja semantične tehnologije pri objavljanju, odkrivanju in dinamični integraciji mrežnih virov pri izvajanju mrežnih aplikacij (glej poglavje 5 in podpoglavje 7.5),
- ponuja možnost uporabe dveh vrst uporabniških vmesnikov (urejevalca in upravljalca delotokov ter specializiranih mrežnih portalov), kar poveča uporabniško prijaznost in prožnost sistema (glej podpoglavja 8.1.1 in 8.1.2),
- omogoča vzporedno izvajanje aplikacij na optimalno izbranih mrežnih virih (glej podpoglavje 7.6 in preglednico 7-2),
- je zmogljiv in nadgradljiv (glej podpoglavje 8.3),
- prožna arhitektura sistema, ki je zasnovana na SOA, je precej generična in podpira širok nabor različnih mrežnih aplikacij ter omogoča hitro zamenjavo nekaterih komponent v skladu z razvijajočimi se standardi, omogoča višjo raven abstrakcije oziroma neodvisnost testnega okolja od posameznih mrežnih aplikacij (glej poglavje 6),

- omogoča odkrivanje, izbiro ter dinamično integracijo mrežnih virov v času izvajanja mrežnih aplikacij (glej podpoglavje 7.5),
- omogoča večji izkoristek virov in bolj enakomerno obremenjenost obstoječih strežnikov in računalniških gruč zaradi možnosti njihovega deljenja čez meje organizacij, kar zagotavlja avtomatsko razvrščanje poslov na manj zasedene vire (glej podpoglavje 7.6),
- izogibanje eni sami točki odpovedi sistema (v primeru odpovedi virov se opravila premestijo na druge vire) (glej podpoglavje 7.6).

9 Diskusija

Problem razvoja primerne infrastrukture, ki bi podpirala sodelovanje med gradbeniškimi podjetji pri razvoju in izgradnji kompleksnih gradbeniških produktov, sodi med ključna odprta vprašanja na področju gradbene informatike (Turk, 2006 in 2007).

Naš cilj je bil generirati novo znanje o načinih izboljšave obstoječe mrežne infrastrukture, da bi postala primerna osnova za računalniško integrirano graditev. Na področju gradbeništva obstaja cela vrsta različnih računsko in informacijsko intenzivnih problemov, ki bi jih lahko ustrezno reševali z uporabo tehnologij mreže (Hartmann, 2006). Sistem, ki smo ga zgradili, podpira izdelavo takih mrežnih aplikacij, ki omogočajo sodelovanje pri projektne delu, visoko zmogljive računske operacije ter večji izkoristek semantično bogatih in geografsko široko porazdeljenih podatkov. Ugotovljene pomanjkljivosti obstoječih mrežnih infrastruktur, npr. pomanjkanje prožnosti pri izgradnji novih mrežnih aplikacij, smo reševali z uporabo semantičnih tehnologij. Izdelali smo ontologijo mrežnih virov ter testno mrežno okolje, ki vključuje posrednika virov, podatkovne, metapodatkovne, informacijske in ontološke mrežne storitve.

V tem poglavju povzemamo predlagane rešitve in izboljšave ter jih primerjamo z rezultati drugih študij. Zaključujemo s kritičnim pogledom na mrežne in semantične tehnologije in obravnavamo nekatera vprašanja, ki so zanimiva za nadaljnje delo.

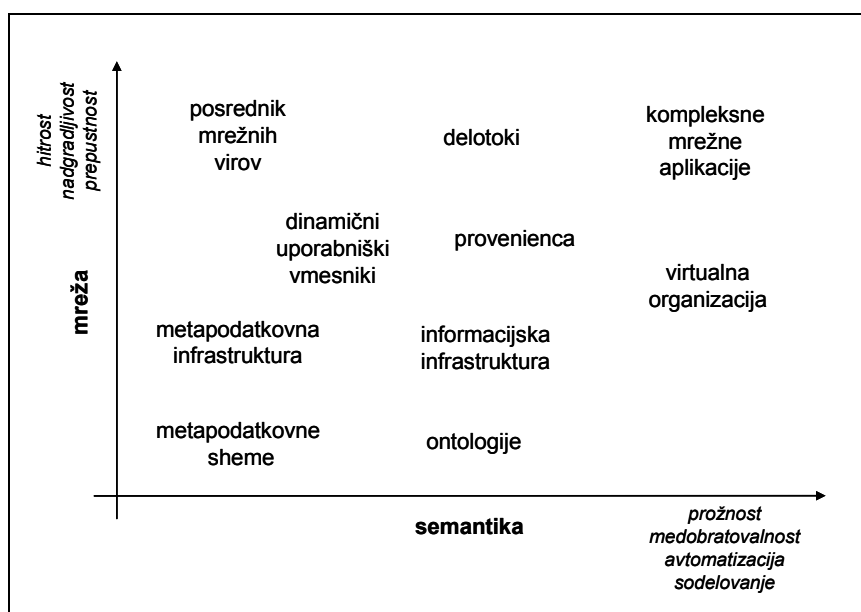
9.1 Prednosti in slabosti pristopa v primerjavi z drugimi študijami

V tej študiji predlagamo, da si lahko pri reševanju cele vrste inženirskih in znanstvenih problemov pomagamo s tehnologijami semantične mreže. **Semantično mrežo smo zgradili tako, da smo v obstoječi mrežni sistem vgradili metapodatkovno infrastrukturo, ki**

temelji na sporazumni ontologiji mrežnih virov in je neodvisna od posameznih mrežnih aplikacij.

Tako smo dosegli:

- enostaven način za vključevanje obstoječih izvršljivih programov za izvajanje na mreži,
- prožnost pri razvoju novih, kompleksnih mrežnih aplikacij, ki imajo obliko delotokov,
- odkrivanje in uporabo obstoječih izvršljivih programov na mreži,
- večjo stopnjo medobratovalnosti,
- dinamično izdelavo uporabniških vmesnikov,
- izvajanje računsko in podatkovno intenzivnih aplikacij,
- sledenje provenienci ter
- učinkovito posredovanje in boljši izkoristek virov na ravni mrežnega sistema.



Slika 9-1: Obravnavane semantične in mrežne tehnologije

Fig. 9-1: Studied semantic and grid technologies

Zgrajena metapodatkovna infrastruktura podpira zbiranje in izkoriščanje metapodatkov v vse našete namene. Posledično so tudi metapodatki prvorazredni mrežni viri. Nekateri

tehnološki vidiki semantičnih in mrežnih tehnologij o katerih diskutiramo v nadaljevanju, so predstavljeni na sliki 9-1. Medtem ko uporaba semantičnih tehnologij prispeva k večji prožnosti, medobratovalnosti, avtomatizaciji in sodelovanju, tehnologije mreže omogočajo večjo hitrost, nadgradljivost ter prepustnost aplikacij. Pri obravnavanih problemih na področju gradbeništva sta oba vidika semantične mreže zelo pomembna.

Ontologija mrežnih virov, ki smo jo zgradili, modelira nekatere pomembne vire, ki nastopajo v mrežnem okolju. Potem ko je vir opisan, se ga lahko preprosto vključi v mrežni sistem, njegov opis pa se registrira pri storitvi za integracijo informacij. V nadaljevanju, storitev posrednika virov uporablja metapodatkovno storitev za poizvedovanje, izbiro in dinamično integracijo virov. Tako je omogočeno tudi kombiniranje mrežnih virov v medobratovalno celoto. Zgrajeni sistem je nazorno pokazal, da ontologija mrežnih virov omogoča lažje povezovanje med arhitekturnimi plastmi semantične mreže, ravno tako pa tudi vključevanje, odkrivanje, rabo mrežnih virov ter njihovo dinamično integracijo in medobratovalnost.

9.1.1 Uporaba in razvoj mrežnih infrastruktur

Novo in inovativne računalniške aplikacije, večja dostopnost specializirane programske opreme in podatkov, hitrost in prožnost so ključ do večje konkurenčnosti gradbeniških podjetij. Povečanje konkurenčnosti na področju gradbeništva pa je pomemben cilj Evropske komisije (Salmelin, 2006). Tudi v Sloveniji so na tem področju potekale določene aktivnosti, npr. na projektu GridForum.si (Stankovski in sod., 2005). Kompleksni problemi, ki bi jih reševali z uporabo tehnologij mreže, pa niso tipični samo za gradbeniško panogo, temveč tudi za druga industrijska in znanstvena področja (Stankovski in sod., 2008, a in b).

Na področju uporabe in razvoja mrežnih infrastruktur pri reševanju gradbeniških problemov je bilo do danes po svetu narejeno zelo malo. Edini relevantni projekti in raziskave, ki smo jih zasledili v literaturi, so projekti NEESGrid, GEODISE in InteliGrid ter objavljena študija, ki uvaja mrežne storitve na področju strukturne dinamike (Alonso in sod., 2006).

Mrežni sistem, ki so ga zgradili na projektu GEODISE, je konceptualno najbolj podoben sistemu, ki smo ga zgradili in preizkusili. Tako kot naš tudi sistem GEODISE vključuje okolje za izdelavo delotokov. Delotok je predstavljen s primerkom ontološkega koncepta »naloga« (angl. task). Uporabnik prek grafičnega vmesnika izbere obstoječo nalogo, ki je shranjena v bazi znanja, in jo konfigurira. Delotok se avtomatsko prevede v validno Matlab *.m datoteko, ki se lahko nadalje uporabi za izvajanje v okolju MATLAB. Sistem GEODISE omogoča izvajanje aplikacij izključno v okolju MATLAB, medtem ko je naš sistem bistveno bolj generičen in neodvisen od platforme. To smo dosegli z ustreznim konceptualnim modeliranjem računalnikov in računalniških gruč, ki predstavljajo mrežne vire, vključno z nameščenimi operacijskimi sistemi (npr. Windows, Linux, Solaris itd.).

Podobnost pristopa h konceptualnemu modeliranju, ki smo ga uporabili pri našem delu, in pristopa InteliGrid ni naključna, saj je projekt potekal na KGI. Ključni cilj projekta InteliGrid je bil poiskati rešitev problema medobratovalnosti na kompleksni semantični mreži. V nasprotju s projektom InteliGrid, na katerem so obravnavali predvsem vprašanje modeliranja virtualnih organizacij na področju gradbeništva, poslovnih procesov in drugih visokonivojskih konceptov, smo mi pri konceptualnem modeliranju obravnavali problem medobratovalnosti programov, podatkov, računalnikov in drugih nizkonivojskih mrežnih virov. Na projektu InteliGrid so razvili napredno avtorizacijsko in tudi druge mrežne storitve, ki uporabniku omogočajo dostop do mrežnih virov glede na virtualno organizacijo, ki ji uporabnik pripada, ter njegovo vlogo in pooblastila. Pri tem igrata ključno vlogo politika vira, ki določa pogoje dostopanja do vira, ter politika virtualne organizacije, ki določa vlogo uporabnika v okviru virtualne organizacije. Pri gradnji avtorizacijske mrežne storitve je bil uporabljen močno uveljavljen konceptualni model RBAC. Na podlagi teh informacij sprejme avtorizacijski sistem odločitve, ki jo posreduje tudi drugim mrežnim storitvam sistema. V nasprotju s kompleksnim pristopom projekta InteliGrid smo mi pri doseganju večje varnosti in uveljavljanju pravic posameznikov, ki nastopajo v okviru virtualne organizacije, uporabili obstoječe rešitve, ki so vgrajene v prilagojeno sistemsko programje GT4.

Večina obstoječih gradbeniških aplikacij temelji na zbiranju, spreminjanju in izgradnji določenih podatkovnih zbirk oziroma gre za uporabo različnih (podatkovno analitičnih) algoritmov pri obdelavi podatkov pri različnih aplikacijskih domenah (Hartmann, 2006). S tega stališča je mrežna infrastruktura, ki so jo razvili na projektu NEESGrid, zelo neprožna, saj vključuje centralizirano podatkovno zbirko in druge specializirane mrežne storitve, ki so uporabne izključno v namene modeliranja in simulacij na področju potresnega inženirstva. V nasprotju s tem testno mrežno okolje, ki smo ga zgradili tukaj, ponuja možnosti izdelave aplikacij na več gradbeniških področjih, npr. pri reševanju računsko intenzivnih problemov, na področju upravljanja s projektnimi dokumentacijami, povezovanja geografsko porazdeljenih podatkovnih zbirk (npr. projektnih dokumentacij med različnimi podjetji, digitalnih knjižnic itd.).

Mrežno storitev, ki omogoča strukturno analizo, so predstavili Alonso in sod. (2006). V ta namen so predelali obstoječe programe za strukturno analizo. Predelava različnih specializiranih gradbeniških programov za izvajanje na mreži ni trivialen problem. Poleg tega, ta pristop lahko privede do problemov z zmogljivostjo strežnika, ki mrežno storitev gosti, npr. pri velikem številu uporabnikov, ki jo sočasno uporabljajo. Njihova mrežna storitev izkorišča lokalno dostopne računalnike, ne pa tudi geografsko široko porazdeljenih infrastruktur. Pri našem delu pa smo izkoristili metapodatkovno infrastrukturo in tako dosegli veliko bolj prožno, dinamično in optimalno razporeditev mutliposlov na oddaljene računalnike, čez meje organizacije, in to glede na trenutno obremenitev mrežnih računalnikov. Način, ki smo ga predstavili, odpravi probleme z zmogljivostjo, nadgradljivostjo, odpovedjo delov sistema, izboljša pa tudi izkoristek računalnikov in računalniških gruč.

9.1.2 Ontologija versus ontologije

Obstoj sporazumnih konceptualnih modelov (metapodatkovnih shem in ontologij) je pogoj za doseganje semantične medobratovalnosti na področju gradbeništva in drugih inženirskih področij. Gre za konceptualne modele, ki jih razvijajo z namenom deljenja informacij in znanja na določenih področjih, za katera so se zavezali vsi sodelujoči (ravno tako tudi računalniška orodja) v okviru virtualne organizacije. Sporazumne ontologije vzpostavljajo

skupen jezik, ki je osnova za vse interakcije na mreži (npr. interakcije mrežna storitev–mrežna storitev ali mrežna storitev–odjemalec–človek). Pri načrtovanju sistemov, ki temeljijo na konceptualnih modelih oziroma ontologijah, pa postaja jasno, da je strukturiran sistem večjega števila specializiranih ontologij bolj učinkovit glede na njihovo definicijo, vzdrževanje, prepletanje in procesiranje kot aplikacije, temelječe na eni sami veliki in kompleksni ontologiji, ki opisuje celotno inženirsko ali znanstveno področje. Na primer, problem nizke stopnje medobratovalnosti sistemov (Pazlar in Turk, 2008), ki temeljijo na konceptualnem modelu IFC, bi lahko pripisali njegovi kompleksnosti.

Eden pglavitnih ciljev dela je bil, da z ustreznim konceptualnim modeliranjem razvijemo hierarhično strukturo, ki omogoča opisovanje različnih entitet, ki nastopajo v mrežnem sistemu. Konceptualni model, ki smo ga zgradili, je poimenovan kot ontologija mrežnih virov (Grid Resource Ontology, GRO, priloga). Izdelana ontologija je odprte narave, ker definira kompaktni in minimalni nabor metapodatkov, ki jih potrebujejo viri, da so lahko na preprost način vključeni v mrežno okolje. Ontologija mrežnih virov torej predstavlja okvir, znotraj katerega je mogoče specificirati vse načine uporabe in obnašanja mrežnih virov.

Nekatere zgodnje poskuse v to smer so naredili projekti DataTAG, iVDGL, Globus in DataGrid. Kot rezultat tega dela so razvili shemo Grid Laboratory Uniform Environment (GLUE). GLUE je ontologija, ki zajema ključne aspekte obstoječih večjih mrežnih infrastruktur, ki pa ni bila nikoli uporabljena v praksi. Tudi Tangmunarunkit, Decker in Kesselman (2003) so predstavili teoretično študijo o tem, kako bi uporabili bazo znanja (ontologijo in metapodatke) pri načrtovanju in razvoju posrednika mrežnih virov. Takšen posrednik virov pa danes še ne obstaja. V jeziku OWL so izdelali ontologijo mreže tudi na projektu CoreGrid (Core Grid Ontology, CGO), ki prav tako ni preizkušena v praksi, saj se projekt ne ukvarja z razvojem testnih okolij (Xing, Dikaiakos in Sakellariou, 2006). Ontologija GRO, ki smo jo zgradili, je zapisana v standardnem jeziku OWL in je bila preizkušena v praksi. Aktivnosti v smeri definiranja ontologije mrežnih virov, ki bi podpirala razvoj medobratovalnih mrežnih sistemov, trenutno prevzema tudi delovna skupina STF 331: TC-GRID pri organizaciji ETSI.

Nizkonivojski poskus v to smer pa je razvoj jezika Job Submission Description Language (JSDL) v okviru organizacije OGF. Pri razvoju jezika JSDL pa ne upoštevajo uporabniških zahtev za izvajanje parametričnih študij, ki so tako pogoste na področju gradbeništva. Jezik Job Description Language (JDL), ki ga uporablja programje gLite, pa temelji na jeziku za oglaševanje poslov ClassAd, ki so ga definirali na projektu Condor.

Pri definiranju konceptov poslovanja, npr. modeliranja poslovnih procesov v okviru virtualnih organizacij na področju gradbeništva, je bilo do danes narejeno največ na projektu InteliGrid. Pri tem postaja jasno, da se popolna podpora integraciji poslovnih procesov lahko zagotovi samo, če so koncepti jasno definirani in dodelani do podrobnosti, ki omogočajo enoznačno opisovanje procesov oziroma njihovo specifikacijo. Tukaj gre za izjemno veliko zahtevo, ki še vedno predstavlja izziv pri načrtovanju sistemov, ki bi bili uporabni v praksi.

Pri našem delu smo uporabljali obstoječe standardizirane jezike za zapis konceptualnih modelov, to so predvsem jeziki XML, RDF ali OWL. Obravnavali smo tudi obstoječe domenske ontologije, npr. ontologijo spletnih storitev OWL-S, konceptualni model produktov ifcXML itd. Pri tem velja opozoriti na dejstvo, da kljub zrelosti nekaterih manj ekspresivnih vrst jezikov za tvorjenje ontologij, kot je RDFS, jih pri izgradnji realnih sistemov še vedno zelo malo uporabljajo, aplikacije, temelječe na jeziku OWL, pa so še v povojih (Missier in sod., 2007). Jeziki XML, RDF in OWL omogočajo napredne mehanizme za sklicevanje na obstoječe metapodatkovne sheme in ontologije, ki so dostopne bodisi lokalno ali na spletu. Na primer, v jeziku OWL je mogoče zapisati `<owl:imports rdf:resource="http://www.stankovski.net/gro/gro.owl" />` in tako v obstoječi ontologiji uporabiti drugo, ki se nahaja kjerkoli na spletu. Zato OWL omogoča zahtevano prožnost in prilagodljivost, ki bi lahko v prihodnje prispevala k razvoju zmogljivih sistemov, temelječih na hierarhično povezanih oziroma prepletenih ontologijah.

Domnevali smo, da lahko večjo prožnost, nadgradljivost in konceptualno preprostost sistema dosežemo z uporabo ontologije, ki bi modelirala vse plasti arhitekture semantične mreže. Medtem ko sta ontologiji OWL-S in WSMO namenjeni modeliranju višjenivojskih

plastí, smo se pri naši raziskavi še posebej omejili na najnižjo plast mrežnega sistema, to je plast mrežnih virov, saj smo v literaturi zasledili pomanjkanje tovrstnih raziskav.

Ugotavljamo tudi, da ročna izgradnja ontologij zahteva veliko znanja in truda, njihov primanjkljaj pa je v tem trenutku ozko grlo semantičnega spleta. Zato nekateri sodobni pristopi uporabljajo tudi metode strojnega učenja oziroma metode za avtomatsko učenje ontologij.

9.1.3 Infrastrukturni elementi semantične mreže in izkoristek virov

Že s preprostim konceptualnim modelom, ki smo ga predstavili, smo lahko izdelali in uspešno preizkusili zelo prožen mrežni sistem. Nastali prototip omogoča izdelavo kompleksnih mrežnih aplikacij, ki imajo obliko delotokov, ter nove načine rabe obstoječih izvršljivih programov. Zagotavlja tudi večji izkoristek obstoječih virov, za kar skrbi posrednik mrežnih virov, ki omogoča dinamično integracijo virov čez meje organizacij, uporabo obstoječih programov in aplikacij, upravljanje porazdeljenih podatkovnih zbirk, ki so shranjene v datotekah, direktorijih ali relacijskih podatkovnih zbirkah, itn.

V testno mrežno okolje smo vključili posrednika virov, metapodatkovne, informacijske ter ontološke mrežne storitve, ki v svoji zgradbi upoštevajo koncepte iz ontologije mrežnih virov. Medtem ko posrednik mrežnih virov, metapodatkovne in informacijske mrežne storitve temeljijo na tehnologijah, ki smo jih izdelali v okviru projektov InteliGrid in DataMiningGrid, smo pri izgradnji ontološke storitve uporabili tehnologije, razvite v okviru projekta OntoGrid. Naši rezultati kažejo, da smo s konceptualnim modeliranjem ter na podlagi štirih dodatnih storitev, ki so bile uporabljene pri izdelavi različnih kompleksnih mrežnih aplikacij, lahko zadovoljili več uporabniških zahtev. Na primer, dosegli smo večjo prožnost, zmogljivost, izogibanje eni sami točki odpovedi ter enakomerno obremenjenost virov, ki jo zagotavlja avtomatsko razvrščanje poslov na manj zasedene vire. V primeru odpovedi virov pa se posli premestijo na druge vire. Pri tem moramo nujno opomniti, da so posrednik virov, metapodatkovna in informacijska storitev direktno vključeni v procese avtomatskega izvajanja kompleksnih mrežnih aplikacij, medtem ko so ontološke storitve

izolirane in omogočajo poizvedbe ontoloških konceptov in relacij, torej prispevajo samo k razumevanju okolja in razmerij, ki vladajo v testnem mrežnem sistemu.

9.1.4 Tehnologije delotokov in uporabniški vmesniki

Mrežni sistem je potencialno velik, po vsem svetu porazdeljen sistem, ki lahko omogoča dostop do najbolj eksotičnih virov, kar posledično pomeni, da lahko na podlagi teh virov tvorimo nove zmogljive mrežne aplikacije, ki doslej sploh niso bile možne. Ob tem postaja jasno, da mreža potrebuje nove programske modele.

Tehnologije delotokov, ki smo jih obsežno uporabljali pri našem delu, nam omogočajo zahtevani premik k višji stopnji abstrakcije pri izdelavi mrežnih aplikacij (Taylor in sod., 2007). Delotoki so lahko abstraktni ali konkretni, kar so nazorno pokazali na projektu K-WfGrid. Njihov pomen je prepoznaven tudi v vrsti standardizacijskih aktivnosti, npr. pri standardizaciji jezikov BPEL in BPEL4WS, želje po standardizaciji vmesnikov do upravljalcev delotokov Triana, Taverna, Kepler itd.

Tehnologije delotokov in tehnologije za opisovanje poslovnih procesov so zelo koristne na področju gradbeništva. Scherer (2007) je predstavil sistem za modeliranje poslovnih procesov in upravljanje delotokov, ki pa je okvir za dejansko izvajanje kompleksnih opravil na storitveni oziroma mrežni infrastrukturi. Možnosti za integracijo gradbeniških procesov z uporabo tehnologij delotokov sta obravnavala tudi Cerovšek in Katranuschkov (2006).

Tehnologije delotokov nam omogočajo več vrst uporabniških vmesnikov do mrežnega sistema. Tako lahko bolj izkušeni uporabniki uporabljajo urejevalnik delotokov, manj izkušeni pa specializirane mrežne portale, v katerih so vgrajeni delotoki, ki za uporabnika niso vidni. Ta pristop zakriva nizkonivojske tehnološke posebnosti mrežnega sistema pred uporabniki in tako povečuje njegovo uporabnost, saj so vmesniki prilagojeni različnim tipom uporabnikov. Koristnost izdelave mrežnih portalov potrjujejo tudi rezultati Petrinje (2007), ki je izdelal mrežni portal za sledenje provenienci pri produktnih modelih, oziroma Trnkoczyja, Turka in Stankovskega (2006), ki so izdelali mrežni portal za združene digitalne knjižnice.

9.1.5 Prilagajanje obstoječe programske opreme za izvajanje na mreži

Z našim delom smo prispevali k reševanju problema integracije obstoječe programske opreme za izvajanje v kompleksnem mrežnem okolju. Ta cilj se lahko doseže na vsaj dva različna načina. Prvi je preoblikovanje izvršljivih računalniških programov v specifične mrežne storitve, kot so na primer naredili Alonso in sod. (2006) na področju strukturne dinamike, drugi način pa je prek uporabe metapodatkovne infrastrukture, ki je vgrajena v mrežni sistem.

Ugotavljamo, da podjetja, ki se ukvarjajo z računsko zahtevnim modeliranjem stavb, kot je na primer SoFiSTiK, lahko izkoristijo izvršljive programe, podatke, modele zgradb, računalnike, shrambo in druge vire, prosto dostopne na mreži, in tako dosežejo hitrejšo odzive na zahteve naročnikov.

V preteklih letih so prestrukturirali veliko obstoječih monolitskih aplikacij v tako imenovane aplikacijske zbirke (angl. application suite), ki imajo skupno notranjo predstavnost podatkov in uporabniški vmesnik (npr. GEOPAK Civil Engineering Suite podjetja Bentley). Ta pristop na daljši rok ne ponuja zahtevane prožnosti, ki jo omogočajo sistemi, temelječi na SOA in OGSA. Veliko specializiranih aplikacij na področju gradbeništva se ponavadi lahko izvaja na paketni način (angl. batch mode) in bi jih lahko parametrizirali ob začetku izvajanja. Z druge strani je znano, da velika večina specializiranih programov (oziroma algoritmov), ki jih razvijajo raziskovalci in inženirji, ni del aplikacijskih zbirk. Zato je omogočanje posamičnih aplikacij za izvajanje v mrežnem okolju veliko bolj zahtevno in predstavlja širši obseg zahtev za posrednika mrežnih virov. Testno mrežno okolje, ki smo ga zgradili, pa ponuja večjo prožnost in možnosti za sodelovanje med inženirji, deljenje specializiranih programov itd. kot obstoječi sistemi, npr. sistemi podjetij Bentley ali SoFiSTiK.

9.2 Analiza paradigme semantične mreže

Računalniška mreža ni sistem, ki bi bil omejen samo na določena področja uporabe. Tehnologije mreže niso alternativa medmrežnim tehnologijam, temveč njihova logična

nadgradnja. Zaradi virtualizacije, ki jo omogočajo spletne in mrežne storitve, je vsak podatek, program, računalnik, senzor ali kakršenkoli drug vir avtomatično dostopen kjerkoli in kadarkoli. Internet danes predstavlja izvir prostih in razpoložljivih virov, to pa velja tudi za mrežo, ki je njegova nadgradnja. Z druge strani v nasprotju z Internetom mreža ni sistem, ki bi omogočal neomejen dostop do virov. Zato nastajajo novi mrežni protokoli in storitve, ki se ukvarjajo s koordiniranim deljenjem, kontrolo izmenjave informacij ter rabo virov. Cilj je, da bodo v prihodnje lastniki mrežnih virov imeli možnost omejiti dostop do svojih virov ter ga tržno vrednotiti.

V našem delu smo pokazali, da tehnologije semantične mreže povišajo raven abstrakcije pri razvoju kompleksnih mrežnih aplikacij. Tak mrežni sistem omogoča veliko več kot samo povečanje hitrosti ter nadgradljivost obstoječih, računsko intenzivnih aplikacij. Semantična mreža temelji na storitveno orientirani arhitekturi in omogoča virtualizacijo virov, vključno s strojno opremo, aplikacijami in podatki. Tako je mogoče vire uporabljati kjerkoli in kadarkoli v okviru mrežnega sistema. Uporaba SOA pri izgradnji mrežnih sistemov pa ima tudi velik pomen, saj omogoča boljši izkoristek virov v mrežnem okolju zaradi njihove virtualizacije in dekompozicije aplikacij na storitvene sestavne dele (rahle povezanosti).

Semantična mreža je porazdeljeni sistem brez centralnega nadzora. Prilagojeno sistemsko programje omogoča delovanje celotnega sistema. Le-to mora nuditi enake možnosti, kot so tiste, ki jih operacijski sistemi ponujajo monolitnim aplikacijam: transparentnost, dodeljevanje virov (npr. procesorji, pomnilniški prostor, vhodno/izhodne naprave) različnim procesom, časovno razporejanje opravil, reševanje konfliktnih situacij (npr. sočasni dostopi do virov), optimizacija in nadzor uporabe virov, omogočanje dela uporabnikom, bdenje nad napakami itd. To pa so cilji, ki še vedno niso doseženi.

Pričakovati je, da bo mreža z uporabo semantičnih tehnologij v prihodnje omogočala reševanje bolj kompleksnih problemov, kot so tisti, ki jih v tem trenutku rešujejo superračunalniki. Nekateri evropski projekti so namenjeni ravno takim preizkusom (npr. QoSGrid). Tisoči ali celo milijoni procesorjev, ki bi postali dostopni v okviru virtualne organizacije, bi bili pomemben izvor računalniške moči, če bi jih lahko izrabili na

enostaven način. Je pa tudi res, da številni znanstveni problemi zahtevajo tesno sklopljene računalnike, ki imajo hitre komunikacijske možnosti. Na tem področju so tehnologije mreže še vedno v povojih.

9.3 Dejavniki pri vpeljavi mrežnih sistemov

Ključni izziv vpeljave mrežnih sistemov se skriva v prepoznavanju poslovnih prednosti, to je v izboljšanju učinkovitosti poslovanja, ki naj bi ga mrežni sistem omogočal skozi neposredno podporo poslovnim procesom (InteliGrid, spletni vir), večjo prilagodljivost in hitrejšo odzivnost na spremembe in dopolnitve. Mrežni sistemi organizacijam omogočajo razvoj novih storitev in proizvodov v smislu storitveno usmerjene ekonomije.

Ključni dejavnik tveganja pri vpeljavi mrežnih sistemov pa je pomanjkanje znanja in razumevanja poslovnih prednosti, ki jih ti sistemi ponujajo, premalo obstoječega znanja in izkušenj kot tudi pomanjkanje tehničnega znanja. Poslovne prednosti mrežnih sistemov pa je mogoče realizirati le na osnovi implementacije ustreznih tehnoloških prednosti, prek katerih lahko podjetja dosežejo hitrejšo in enostavnejšo prilagajanje aplikacij, boljšo integracijo, lažjo namestitvev in večjo stopnjo poslovne uporabe. Glavne tehnološke prednosti vpeljave mrežnih sistemov so predvsem: boljši izkoristek virov, inovativne in kompleksne aplikacije, ki rešujejo kompleksne inženirske probleme, večja prožnost in prilagodljivost razmeram na trgu itd.

Z uporabo tehnologij semantične mreže bi podjetja v poslovnem smislu lahko dosegla: večjo konkurenčnost, izboljšano učinkovitost poslovnih procesov, hitrejšo odzivnost na spremembe in boljšo prilagodljivost poslovnih procesov, znižanje cen vzdrževanja in nadgrajevanja virov, ki se zahtevajo za nemoteno poslovanje, ter boljšo usklajenost med informacijskimi tehnologijami in poslovnimi zahtevami. Ti cilji pa zaenkrat niso prioriteta slovenskih podjetij, ki glede na študijo Jurčiča (2007) kot pomembno prednost ocenjujejo večjo učinkovitost in manjše potrebe po vlaganjih v prihodnosti, medtem ko enostavnejšemu razvoju aplikacij in razgradnji v storitve ter boljši integraciji ne pripisujejo tolikšnega pomena.

Semantična mreža lahko močno prispeva k doseganju večje inovativnosti in konkurenčnosti, zato bi morali v prvi vrsti promovirati njene prednosti. Izdelava pilotnih projektov in izobraževanje sta ključni aktivnosti, ki sta potrebni za vpeljavo mrežnih sistemov v slovenska podjetja. Naše aktivnosti na projektu GridForum.si posredno razkrivajo dejstvo, da so slovenska podjetja pri vpeljavi mrežnih sistemov šele na začetku, kar pa je glede na to, kako dolgo tehnologije mreže obstajajo, pričakovano in primerljivo s tujino.

9.4 Standardizacija na področju tehnologij semantične mreže

Razvoj tehnologij semantične mreže zahteva oblikovanje novih aplikacijskih protokolov, storitev in programskih komponent, s pomočjo katerih je poenostavljena izgradnja zmogljivih mrežnih aplikacij. Pri tem je ključnega pomena standardizacija, saj je le tako mogoče doseči medobratovnost med mrežnimi sistemi. Delo v tej smeri je pripeljalo do razvoja referenčne arhitekture odprtih mrežnih storitev (OGSA), ki definira posamezne komponente tipičnega mrežnega sistema, njihove funkcionalnosti in odnose med posameznimi komponentami. Ugotavljamo pa, da na tem področju še vedno ne obstaja niti en sam poenoten in splošno priznan standard (Baker in sod., 2005). Obstaja vrsta standardov, specifikacij in priporočil različnih organizacij, kaj od tega bo sprejeto v industriji in odprtokodni skupnosti, pa danes ni popolnoma jasno. Večina specifikacij na tem področju je namreč zelo novih in nezrelih, veliko jih obstaja šele v obliki osnutkov, ki niso preizkušeni v praksi. Mrežne tehnologije so in bodo ostale utemeljene na tehnologijah spletnih storitev, čeprav ni povsem jasno, katere od mnogih dodatnih specifikacij spletnih storitev bodo prevladale. Tako na primer ni jasno, katere specifikacije za upravljanje in opis mrežnih poslov ali upravljanje podatkov, dostopnih na mreži, bodo ostale v veljavi. Očitno pa je, da gre za generične operacije, ki bi lahko bile standardizirane. Očitno je tudi, da se strokovnjaki strinjajo glede nujnosti uporabe storitev, ki imajo stanje.

Kljub veliki nedorečenosti na področju standardizacije mrežnih tehnologij pa pobude, kot sta OGSA in WSRF, nedvomno kažejo, da obstaja velik interes za definicijo enotnega standarda in da so še vedno potrebni določeni poskusi s to tehnologijo, da bi lahko definirali optimalne variante. Skupen trud pri snovanju teh specifikacij je namreč ugoden za razvoj

mrežnih standardov, in to kljub trenutni negotovosti, nekonsistenčnosti in slabi medobratovalnosti, s katerimi se soočamo razvijalci mrežnih aplikacij.

9.5 Odprta tehnološka vprašanja in nadaljnje delo

Semantična mreža je še vedno v nastajanju in ni pričakovati, da bodo določeni tehnološki problemi kmalu rešeni. Z nekaterimi od njih pa smo se ukvarjali tudi pri našem delu. Zaradi celovitosti smo pripravili njihov kratek pregled.

- **Ontologije.** Pri našem delu smo videli, da obstaja praktično neomejeno število načinov za uporabo kompleksnih konceptualnih modelov oziroma ontologij. Zato je dogovor za eno samo natančno določeno ontologijo, ki bi bila uporabna pri vseh mrežnih sistemih, velik problem, ki bo zahteval še ogromno truda.
- **Delotoki.** Danes obstaja le peščica sistemov, ki omogočajo urejanje in upravljanje delotokov. Med njimi so na primer Triana, Taverna, Kepler itd. Tudi Yahoo je nedavno predstavil sistem Pipes. Vsak od njih pa ima svoje formalizme za reprezentacijo delotokov, ki jih je treba poenotiti v okviru standardizacijskih organizacij.
- **Generična arhitektura mrežnega sistema.** Različne raziskovalne skupine po svetu predlagajo vedno nove rešitve za generične mrežne protokole. Arhitektura naj bi podpirala primere uporabe z različnih področij, ne samo z gradbeniškega. En tak predlog je npr. arhitektura S-OGSA, ki jo je predstavil projekt OntoGrid. Zato je treba vse pristope sintetizirati in vsakič znova izdelati preizkusne prototipe mrežnih sistemov.
- **Nadzor mrežnih virov.** Mreža je porazdeljen sistem, sestavljen iz najrazličnejših računalnikov, diskov, omrežij in drugih statičnih virov. V prihodnje bi bilo koristno določiti standardne vmesnike za nadzor in upravljanje teh virov.
- **Upravljanje s podatki.** Pokazali smo, da v mrežni sistem lahko vključimo posamezne datoteke, celotne direktorije kot tudi podatkovne baze (npr. relacijske podatkovne baze). Vmesniki OGSA-DAI so sicer zastavljeni zelo široko, vendar ne

zadoščajo za določene operacije in dostop do nekaterih podatkovnih zbirk, kot so na primer objektno orientirane baze.

- **Zasebnost in varnost.** V mrežnem sistemu nastopa veliko akterjev, ljudi, programov, podatkov itd., za katere je treba zagotoviti ustrezne protokole in storitve, ki bi tem entitetam zagotavljali zasebnost in varnost. Industrija gre danes v smer poenotenja pristopov v smislu enotne identitete.
- **Poslovni modeli in pogodbe na ravni storitev (Service Level Agreement, SLA).** Mreži je treba dodati možnosti, da upošteva različne poslovne modele in pogodbe na ravni storitev. Le-ti so trenutno še v povojih. Storitve naj bi samostojno sklepale posle na mreži glede na zahtevano kakovost storitev.
- **Virtualne organizacije.** Mreža kot sistem ne vsebuje konceptov in mehanizmov, ki omogočajo nastajanje in razvoj dinamičnih virtualnih organizacij na določenem inženirskem ali znanstvenem področju. Ob tem je treba zagotoviti mehanizme, ki omogočajo hitro vključevanje novih entitet v organizacijo in odhod drugih, sledenje provenienci, upoštevanje drugih posebnosti vpletenih entitet itd.
- **Mrežni portali in drugi uporabniški vmesniki.** Danes je dostop do mrežnih sistemov še vedno zelo zapleten. Namenjeni so elitnim uporabnikom, ki imajo poleg znanja na svojem področju tudi veliko znanja o zgradbi in možnostih za uporabo mrežnih sistemov. Pri tem je zelo zahteven problem izdelave uporabniških vmesnikov, ki bi temeljili na konceptualnih modelih oziroma ontologijah. Z našim delom smo prispevali v smeri dinamične izgradnje uporabniških vmesnikov glede na konceptualni model mrežnih virov.

10 Zaključek

Uporaba mrežnih sistemov na področju gradbeništva je smiselna in ima potencial, da omogoči reševanje kompleksnih inženirskih problemov. O uporabnosti mrežnih sistemov pričajo tudi nekateri večji mednarodni projekti na področju gradbene informatike, kot sta NEESGrid in InteliGrid, ki sta potekala v preteklih letih. Pri našem delu smo analizirali konkretne probleme, ki nastopajo pri vzpostavljanju dinamičnih virtualnih organizacij, zagotavljanju medobratovalnosti virov v mrežnem okolju ter izdelavi inovativnih in zmogljivih mrežnih aplikacij: delujoči prototip mrežnega sistema nam je omogočil izdelavo računsko intenzivnih aplikacij na področju sanitarnega inženirstva ter podatkovno in informacijsko intenzivnih aplikacij pri upravljanju s projektnimi dokumentacijami. Možne mrežne aplikacije, ki bi lahko potekale v zgrajenem testnem okolju, pa niso omejene samo na področje gradbeništva, temveč je testno okolje uporabno tudi širše pri drugih inženirskih panogah in na drugih znanstvenih področjih.

Pri izgradnji prototipa mrežnega sistema smo uporabljali semantične tehnologije in tako omogočili vrsto prednosti pred drugimi podobnimi sistemi. Ontologija mrežnih virov in posrednik virov, metapodatkovne storitve ter storitev za integracijo informacij so se izkazale za zelo koristne pri doseganju zmožnosti, ki jih ima tako zgrajeno testno mrežno okolje: omogočajo vključevanje obstoječih virov za uporabo v okviru virtualne organizacije, odkrivanje, izbiro in dinamično integracijo mrežnih virov v času izvajanja mrežnih aplikacij itd.

Pri našem delu smo podrobno obravnavali obstoječe vire (izvršljive programe, podatke, strežnike ter računalniške gruče), ki bi jih lahko organizacije in posamezniki vključevali, si jih delili in uporabljali v okviru virtualne organizacije. Teh vprašanj smo se lotili tako, da smo najprej definirali ontologijo mrežnih virov, kar je poglaviten, oprijemljiv in uporaben rezultat dela. Pri tem smo veliko pozornost posvetili modeliranju koncepta izvršljivega programa. Izdelan prototip mrežnega sistema je nazorno pokazal, da uporaba ontologije

mrežnih virov bistveno prispeva h konceptualni enostavnosti in prožnosti sistema.

Poglaviten cilj standardizacijske skupine STF 331: TC-GRID pri organizaciji ETSI, je tudi izdelati kompleten konceptualni model mrežnega okolja.

Ukvarjali smo se tudi z vprašanjem, kako vključiti obstoječe vire v mrežno okolje in jih nato kombinirati v kompleksne mrežne aplikacije. Prototip mrežnega sistema, ki smo ga zgradili in predstavili v tem delu, omogoča enostavno vključevanje obstoječih gradbeniških programov, podatkov (npr. o meritvah), računalnikov in računalniških gruč v mrežno okolje. Pri tem igrajo ključno vlogo metapodatki, ki opisujejo mrežne vire. Na primer, v trenutku, ko je izvršljivi program shranjen na nekem mrežnem strežniku in je njegov opis objavljen pri storitvi za integracijo informacij, postane avtomatsko na voljo tudi drugim inženirjem, ki sodelujejo v okviru virtualne organizacije. Z uporabo preprostega mrežnega portala, ki je predstavljen v tem delu, smo v testno mrežno okolje vključili več kot 25 obstoječih izvršljivih programov (Stankovski in sod., 2008a). Tako smo pokazali široko uporabnost ontologije mrežnih virov. Zaradi uporabe jezika OWL je izdelana ontologija nadalje razširljiva in omogoča vključevanje tudi domenskih konceptualnih modelov glede na področje mrežne aplikacije.

V zgrajenem testnem mrežnem okolju sodeluje več organizacij, ki si delijo različne vire (npr. podatke, izvršljive programe, računalniške gruče, strežnike). Nekateri mrežni viri, kot so podatki in izvršljivi programi, niso statični in jih z uporabo podatkovnih mrežnih storitev (npr. GridFTP) lahko dinamično premikamo po mreži. Na ta način smo omogočili tudi dinamično integracijo teh virov v trenutku izvajanja mrežnih aplikacij.

Pri arhitekturni zasnovi sistema smo upoštevali standarde OGSA, WSRF in druge trenutno nastajajoče standarde. Pri načrtovanju mrežnih storitev in njihovih interakcij pa smo uporabili tudi izdelan konceptualni model mrežnih virov. Rezultat je nekaj ključnih mrežnih storitev, kot so posrednik virov, metapodatkovna mrežna storitev ter storitev za integracijo informacij, ki temeljijo na sporazumni ontologiji mrežnih virov. Če je bilo le mogoče, smo pri izdelavi mrežnih storitev uporabili odprtokodne rešitve. Testno mrežno okolje je zgrajeno večinoma na podlagi mrežnih storitev, ki jih vključuje prilagojeno sistemsko

programje GT4 (npr. MDS4, GridFTP, RFT itd.), pri izdelavi posrednika mrežnih virov pa smo vključili obstoječo tehnologijo GridBus. Mrežne tehnologije, ki so vgrajene v GT4, omogočajo tudi večjo varnost pri tvorjenju virtualnih organizacij.

S stališča končnih uporabnikov je mreža zelo kompleksno okolje, v katerem so na voljo zelo različne mrežne storitve. Končni uporabnik pa izhaja iz svojih delovnih nalog, zato mora imeti možnost za kreativno delo in uporabo sistema, ne da bi se pri tem moral ukvarjati s posebnostmi mrežne infrastrukture. Ključna tehnološka rešitev, ki smo jo poiskali in uporabili pri našem delu, je tehnologija delotokov. Delotoki so lahko osnovni ali sestavljeni in dejansko predstavljajo mrežne aplikacije. Na zahteve po večji uporabniški prijaznosti pa smo odgovorili tako, da smo v arhitekturni zasnovi sistema predvideli možnosti za dve vrsti odjemalskih aplikacij. Eden od načinov uporabe mrežnega sistema je prek urejevalnika in upravljalca delotokov Triana, drugi pa prek specializiranih mrežnih portalov. Pestrost možnosti pri izdelavi odjemalcev pa je omogočena ravno zaradi konceptualne preprostosti mrežnega sistema, ki temelji na ontologiji mrežnih virov.

Urejevalnik in upravljalca delotokov je zelo prožna rešitev in omogoča večjo stopnjo kreativnosti pri odkrivanju obstoječih mrežnih virov in njihovem vključevanju v mrežno aplikacijo. Na podlagi izvršljivih programov, ki smo jih vključili v mrežno okolje, smo tvorili enostavne in sestavljene delotoke (mrežne aplikacije) in jih paralelno izvajali (Stankovski in sod., 2008, a in b).

Specializirani mrežni portali so v primerjavi z urejevalnikom in upravljalcem delotokov veliko bolj uporabniško prijazni, saj pred uporabniki zakrivajo kompleksnost mrežnega sistema. Primer mrežnega portala, ki smo ga izdelali, predstavlja združeno digitalno knjižnico, ki izrablja podatkovno in računsko intenzivne algoritme pri obdelavi podatkov (Trnkoczy, Turk in Stankovski, 2006; Trnkoczy in Stankovski, 2008). Pokazali smo tudi koristnost uporabe obstoječih izvršljivih programov na mreži, npr. programi za avtomatsko učenje ontologij (Stankovski in sod., 2008b), ki jih lahko prispevajo različne organizacije, ki sodelujejo v okviru virtualne organizacije.

V tej raziskavi smo se pri konceptualnem modeliranju omejili na nekatere ključne vire, ki obstajajo v kompleksnem mrežnem okolju, tako da smo upoštevali raznolikost podatkov, programov ter tudi dostopne računske in hrambene zmogljivosti. Nove študije v prihodnosti bi lahko upoštevale tudi senzorje in druge naprave, ki se pogosto uporabljajo na področju gradbeništva, na primer pri potresnem inženirstvu, sanitarnem inženirstvu ali celo pri izdelavi pametnih hiš (Trnkoczy in Stankovski, 2006).

Deljenje in izkoristek obstoječih virov v okviru virtualne organizacije ter možnost za izdelavo novih, inovativnih mrežnih aplikacij ima potencial, da podjetju omogoči večjo učinkovitost, prožnost in nadaljnje prilagajanje fenomenu enkratnosti, ki je tipičen za področje gradbeništva. Z izdelavo enostavnega konceptualnega modela mrežnih virov in njegovega smiselnega vključevanja v arhitekturo in načrt mrežnega sistema smo dosegli veliko pozitivnih učinkov. Delujoči prototip mrežnega sistema pa nam je omogočil, da smo dokazali hipotezo tega dela.

11 Povzetek

Medmrežne tehnologije so lahko osnova za povezovanje procesov v vseh fazah graditve in vodijo v t. i. računalniško integrirano graditev. Cilji informacijsko-komunikacijskih infrastruktur so omogočiti dinamično sodelovanje med podjetji, povezati procese, izboljšati storitve in izkoristek obstoječih programov, podatkov, informacij in znanj. Kljub dolgotrajnemu raziskovanju področja ti cilji še vedno niso doseženi in naloga je prispevek k reševanju tega problema s pomočjo računalniške mreže.

Računalniška mreža ali preprosto **mreža** (angl. grid) je storitvena infrastruktura, ki nastaja z namenom, da omogoči neomejeno in lahkotno deljenje široko porazdeljenih, heterogenih virov, npr. strojne opreme (računalnikov, pomnilnikov), programske opreme (računalniških programov, podatkov), senzorjev in človeških virov, ter na ta način prispeva k reševanju kompleksnih inženirskih in znanstvenih problemov. Za gradbeništvo so zaradi razpršenosti podjetij in dinamičnosti dela tehnologije mreže zelo obetavne. Gradbeništvo ima glede mrežnih infrastruktur tudi nekaj specifičnih zahtev, ki izhajajo iz neponovljivosti (oz. enkratnosti) izdelkov in procesov, v katerih nastajajo.

O uporabnosti mrežnih sistemov v gradbeništvu pričajo večji mednarodni projekti, kot sta NEESGrid in InteliGrid, ki sta potekala v preteklih letih. Uporabnost računalniških tehnologij za tvorjenje virtualnih organizacij na področju gradbeništva so večkrat obravnavali tudi v preteklosti, na projektih, kot sta bila ToCEE in ISTforCE, vendar v tistem času tehnologije mreže še niso bile na voljo. Pri našem delu smo analizirali konkretne probleme, ki nastopajo pri vzpostavljanju dinamičnih virtualnih organizacij, zagotavljanju medobratovalnosti virov v mrežnem okolju ter izdelavi inovativnih in zmogljivih mrežnih aplikacij

Hipoteza dela je, da je to, kar mreži trenutno manjka, da bi postala primerna osnova za računalniško integrirano graditev, večja uporaba semantičnih tehnologij in tudi mehanizmi,

kot so posredniki virov, podatkovne, metapodatkovne, informacijske in ontološke storitve. Namen omenjenih storitev je, da omogočajo avtomatizirano odločanje o izbiri virov na podlagi dostopnih opisov (metapodatkov), kot tudi dejansko izvajanje kompleksnih operacij (opravil, poslov) na mreži. Skupek tehnologij mreže, konceptualnih modelov oziroma ontologij ter metapodatkovnih, informacijskih in ontoloških storitev imenujemo tehnologije semantične mreže.

Delujoči prototip mrežnega sistema nam je omogočil izdelavo računsko intenzivnih aplikacij na področju sanitarnega inženirstva ter podatkovno in informacijsko intenzivnih aplikacij pri upravljanju s projektno dokumentacijo. Možne mrežne aplikacije, ki bi lahko potekale v zgrajenem testnem okolju, pa niso omejene samo na področje gradbeništva, temveč je razvito testno okolje uporabno tudi širše pri drugih inženirskih panogah in na drugih področjih znanosti.

Sistematično zbirali, analizirali in dokumentirali smo ključne uporabniške, aplikacijske in systemske zahteve, na katere mora odgovoriti mrežni sistem. Pri tem smo veliko pozornosti posvečali zahtevam za izdelavo novih, inovativnih mrežnih aplikacij, zmogljivosti in nadgradljivosti, prožnosti, uporabniški prijaznosti ter varnosti mrežnega sistema. Še posebno smo upoštevali zahteve po enostavnosti in konceptualni preprostosti mrežnega sistema, ki bi povišala raven abstrakcije pri izdelavi in izvajanju (inženirskih) mrežnih aplikacij oziroma omogočila, da bi bil mrežni sistem neodvisen od posameznih aplikacij. Zahteve, ki so specifične za področje gradbeništva, smo obravnavali skupaj z drugimi zahtevami, ki so tipične pri reševanju kompleksnih industrijskih in znanstvenih problemov.

Pri našem delu smo podrobno obravnavali obstoječe vire (izvršljive programe, podatke, strežnike ter računalniške gruče), ki bi jih lahko organizacije in posamezniki vključevali, si jih delili in uporabljali v okviru virtualne organizacije. Teh vprašanj smo se lotili tako, da smo najprej definirali ontologijo mrežnih virov, kar je poglavitno, oprijemljivo in uporaben rezultat tega dela. Pri tem smo veliko pozornost posvetili modeliranju koncepta izvršljivega programa. Izdelan prototip mrežnega sistema je nazorno pokazal, da uporaba ontologije mrežnih virov bistveno prispeva h konceptualni enostavnosti in prožnosti sistema.

Ontologija mrežnih virov in posrednik virov, metapodatkovne storitve ter storitev za integracijo informacij so se izkazale za zelo koristne pri doseganju zmožnosti, ki jih ima tako zgrajeno testno mrežno okolje: omogočajo vključevanje obstoječih virov za uporabo v okviru virtualne organizacije, odkrivanje, izbiro in dinamično integracijo mrežnih virov v času izvajanja mrežnih aplikacij itd.

Ukvarjali smo se z vprašanjem, kako vključiti obstoječe vire v mrežno okolje in jih nato kombinirati v kompleksne mrežne aplikacije. Prototip mrežnega sistema, ki smo ga zgradili in predstavili v tem delu, omogoča enostavno vključevanje obstoječih specializiranih gradbeniških programov, podatkov (npr. o meritvah), računalnikov in računalniških gruč v mrežno okolje. Pri tem igrajo ključno vlogo metapodatki, ki opisujejo mrežne vire. Na primer, v trenutku, ko je izvršljivi program shranjen na nekem mrežnem strežniku in je njegov opis objavljen pri storitvi za integracijo informacij, postane avtomatsko na voljo tudi drugim inženirjem, ki sodelujejo v okviru virtualne organizacije. Z uporabo preprostega mrežnega portala, ki je predstavljen v tem delu, smo v testno mrežno okolje vključili več kot 25 obstoječih izvršljivih programov (Stankovski in sod., 2008a). Tako smo pokazali široko uporabnost ontologije mrežnih virov. Zaradi uporabe jezika OWL je izdelana ontologija nadalje razširljiva in omogoča vključevanje tudi domenskih konceptualnih modelov glede na področje mrežne aplikacije.

V zgrajenem testnem mrežnem okolju sodeluje več organizacij, ki si delijo različne vire (npr. podatke, izvršljive programe, računalniške gruče, strežnike). Nekateri mrežni viri, kot so podatki in izvršljivi programi, niso statični in jih z uporabo podatkovnih mrežnih storitev (npr. GridFTP) lahko dinamično premikamo po mreži. Na ta način smo omogočili tudi dinamično integracijo teh virov v trenutku izvajanja mrežnih aplikacij.

Pri arhitekturni zasnovi sistema smo upoštevali standarde OGSA, WSRF in druge trenutno nastajajoče standarde. Pri načrtovanju mrežnih storitev in njihovih interakcij pa smo uporabili tudi izdelan konceptualni model mrežnih virov. Rezultat je nekaj ključnih mrežnih storitev, kot so posrednik virov, metapodatkovna mrežna storitev ter storitev za integracijo informacij, ki temeljijo na sporazumni ontologiji mrežnih virov. Če je bilo le mogoče, smo

pri izdelavi mrežnih storitev uporabili odprtokodne rešitve. Testno mrežno okolje je zgrajeno večinoma na podlagi mrežnih storitev, ki jih vključuje prilagojeno sistemsko programje GT4 (npr. MDS4, GridFTP, RFT itd.), pri izdelavi posrednika mrežnih virov pa smo vključili obstoječo tehnologijo GridBus. Mrežne tehnologije, ki so vgrajene v GT4, omogočajo tudi večjo varnost pri tvorjenju virtualnih organizacij.

S stališča končnih uporabnikov je mreža zelo kompleksno okolje, v katerem so na voljo zelo različne mrežne storitve. Končni uporabnik pa izhaja iz svojih delovnih nalog, zato mora imeti možnost za kreativno delo in uporabo sistema, ne da bi se pri tem moral ukvarjati s posebnostmi mrežne infrastrukture. Ključna tehnološka rešitev, ki smo jo poiskali in uporabili pri našem delu, je tehnologija delotokov. Delotoki so lahko osnovni ali sestavljeni in dejansko predstavljajo mrežne aplikacije. Na zahteve po večji uporabniški prijaznosti pa smo odgovorili tako, da smo v arhitekturni zasnovi sistema predvideli možnosti za dve vrsti odjemalskih aplikacij. Eden od načinov uporabe mrežnega sistema je prek urejevalnika in upravljalca delotokov Triana, drugi pa prek specializiranih mrežnih portalov. Pestrost možnosti pri izdelavi odjemalcev pa je omogočena ravno zaradi konceptualne preprostosti mrežnega sistema, ki temelji na ontologiji mrežnih virov.

Urejevalnik in upravljalca delotokov je zelo prožna rešitev in omogoča večjo stopnjo kreativnosti pri odkrivanju obstoječih mrežnih virov in njihovem vključevanju v mrežno aplikacijo. Na podlagi izvršljivih programov, ki smo jih vključili v mrežno okolje, smo tvorili enostavne in sestavljene delotoke (mrežne aplikacije) in jih paralelno izvajali (Stankovski in sod. 2008a, 2008b).

Specializirani mrežni portali so v primerjavi z urejevalnikom in upravljalcem delotokov veliko bolj uporabniško prijazni, saj pred uporabniki zakrivajo kompleksnost mrežnega sistema. Primer mrežnega portala, ki smo ga izdelali, predstavlja združeno digitalno knjižnico, ki izrablja podatkovno in računsko intenzivne algoritme pri obdelavi podatkov (Trnkoczy, Turk in Stankovski, 2006; Trnkoczy in Stankovski, 2008). Pokazali smo tudi koristnost uporabe obstoječih izvršljivih programov na mreži, npr. programi za avtomatsko

učenje ontologij (Stankovski in sod., 2008b), ki jih lahko prispevajo različne organizacije, ki sodelujejo v okviru virtualne organizacije.

V tej raziskavi smo se pri konceptualnem modeliranju omejili na nekatere ključne vire, ki obstajajo v kompleksnem mrežnem okolju, tako da smo upoštevali raznolikost podatkov, programov ter tudi dostopne računske in hrambene zmogljivosti. Nove študije v prihodnosti bi lahko upoštevale tudi senzorje in druge naprave, ki se pogosto uporabljajo na področju gradbeništva, na primer pri potresnem inženirstvu, sanitarnem inženirstvu ali celo pri izdelavi pametnih hiš (Trnkoczy in Stankovski, 2006). Prehod medmrežja na protokol IPv6 bo v prihodnje omogočil, da bodo vse naprave v enem okolju lahko dostopne v obliki spletnih storitev, kar bo povečalo pestrost dostopnih virov ter omogočilo veliko novih kompleksnih mrežnih aplikacij.

Na tem mestu lahko sklenemo, da smo pri izgradnji systemskega prototipa uporabljali semantične tehnologije in tako omogočili vrsto prednosti pred drugimi podobnimi sistemi.

Testno mrežno okolje, ki smo ga zgradili:

- omogoča izdelavo inovativnih mrežnih aplikacij oziroma nove načine rabe obstoječih virov,
- omogoča višjo raven abstrakcije oziroma neodvisnost testnega okolja od posameznih mrežnih aplikacij,
- podpira tvorjenje dinamičnih virtualnih organizacij,
- omogoča preprosto vključevanje obstoječih virov (npr. izvršljivih programov) v mrežno okolje,
- omogoča odkrivanje, izbiro ter dinamično integracijo mrežnih virov v času izvajanja mrežnih aplikacij,
- omogoča relativno varen dostop do mrežnih virov,
- ponuja možnost uporabe dveh vrst uporabniških vmesnikov (urejevalca in upravljalca delotokov ter specializiranih mrežnih portalov), kar poveča uporabniško prijaznost in prožnost sistema,
- je zmogljivo in nadgradljivo.

Deljenje in izkoristek obstoječih virov v okviru virtualne organizacije ter možnost za izdelavo novih inovativnih mrežnih aplikacij lahko podjetju omogoči večjo učinkovitost, prožnost in nadaljnje prilagajanje fenomenu enkratnosti, ki je tipičen za področje gradbeništva.

Veliko pozitivnih rezultatov smo dosegli že z izdelavo enostavnega konceptualnega modela mrežnih virov ter njegovim smiselnim vključevanjem v arhitekturo in načrt mrežnega sistema. Delujoči prototip mrežnega sistema potrjuje osnovni hipotezi tega dela.

Nadaljnje delo. Če primerjamo delotoke in posamezne storitve, ki jih delotoki predstavljajo, so delotoki bistveno bolj učinkoviti pri deljenju znanja, izvajanju procesov in komunikaciji. Zato se zdi smiselno, da bi raziskave osredotočili na probleme odkrivanja in deljenja znanja s pomočjo kompleksnih delotokov oziroma na njihovo avtomatizirano sestavljanje glede na specifične uporabniške zahteve.

12 Summary

Internet technologies are the basis for the integration of processes in all phases of construction, which leads to computer-integrated construction. Key goals of information and communication infrastructures are to enable dynamic cooperation between companies, integrate processes, improve services and reuse existing programs, data, information and knowledge. These goals have not yet been met despite long-term research in this area and the present work is a contribution towards the solution of this problem through the use of grid computing.

Grid computing, or shortly **grid**, is a service infrastructure being developed to facilitate infinite and seamless sharing of widely distributed, heterogeneous resources, such as hardware (computers, storage devices), software (computer programs, data), sensors and human resources. Hence, it is expected that grid technology will contribute to finding the solution of complex engineering and scientific problems. In the domain of civil engineering, many companies are geographically distributed and due to dynamic working conditions, grid technology is very promising. Moreover, a number of specific requirements arise from this one-of-a-kind phenomenon, that is, the uniqueness of the civil engineering products and processes.

The use of grid systems in the field of construction has been demonstrated by some big international projects, such as NEESGrid and IntelliGrid, which were recently completed. The usefulness of information technologies for the formation of virtual organisations in the construction sector has also been evaluated in various projects in the past, such as ISTforCE, but at that time, grid technologies were not available. We now analyse the practical problems related to the formation of dynamic virtual organisations, interoperability of resources in the network and the creation of innovative and powerful applications.

The hypothesis of this research is that grid systems can become a viable platform for computer integrated construction if semantic technologies are used for their development i.e. ontologies and mechanisms, such as resource brokers, data, metadata, information and ontology services. The purpose of these services is to enable automated decision-making about the selection of resources on the basis of available descriptions (metadata), as well as the actual implementation of complex operations (tasks, jobs) on the grid. The combination of grid technologies, conceptual models i.e. ontologies, and metadata, information and ontology services are called semantic grids.

a functional prototype of a grid system helped us to develop intensive applications in the field of sanitary engineering and data- and information-intensive applications for the management of project documentation. Potential grid applications that could run in the developed grid test bed are not limited to the construction field, but can also be extended to other industries in the engineering and scientific fields.

The key user, application and system requirements were systematically collected, analysed and documented. A lot of attention was paid to the definition of requirements for the development of new, innovative grid applications, scalability, flexibility, user friendliness and security of the grid system. Particularly important is the requirement for conceptual simplicity of the grid system, which increases the level of abstraction in the development and execution of (engineering) grid applications, enabling the grid system to be independent from the individual grid applications. In this study, we considered requirements related to civil engineering along with other requirements that are typical for the resolution of complex industrial and scientific problems.

Existing resources (programs, data, servers and computing clusters) that the companies and individuals can grid enable, share and use in the context of virtual organisations were analysed in detail. These open issues were tackled by first defining a Grid Resource Ontology, which is a major, tangible and useful result of this work. Special attention was paid when modeling the concept of an executable. The developed prototype clearly demonstrates the significant contribution of the Grid Resource Ontology to achieving

enhanced conceptual simplicity and flexibility of the grid system. Grid Resource Ontology, the resource broker, metadata, and information integration services also proved to be very useful in achieving the capabilities of the prototype of the grid system: they facilitate integration of existing resources for use within virtual organisations, discovery, selection, and dynamic integration of grid resources at the time of execution of grid applications, and so on.

We dealt with the question of how to integrate existing resources in a grid environment and then combine these resources into complex grid applications. The prototype of a grid system, which was built and presented in this study, allows for easy integration of existing specialised programs, data (i.e. measurements), computers and computing clusters in the grid environment. Metadata that describe the grid resources play a key role in the prototype. For example, at the moment, when an executable is stored on a grid server and its description is published in the information integrator service, it becomes automatically available to other engineers involved in the context of the virtual organisation. By using a simple grid portal, which is also presented in this study, more than 25 existing programs (executables) were included in the grid test bed environment (Stankovski et al., 2008). This demonstrates the wide applicability of the developed Grid Resource Ontology. Due to the use of Web Ontology Language, the developed Grid Resource Ontology could further be extended to conceptual models in other domains depending on the area of the grid applications.

The developed grid test bed involves several organisations that share a variety of resources (i.e. data, executables, computer clusters and other grid servers). Some network resources, such as data and executables are not static, so they can be moved dynamically in the network with the use of data grid services (i.e. GridFTP). In this way, dynamic integration of these resources is made possible at the time of execution of the grid applications.

The architectural design of the grid system took into account OGSA, WSRF, and other currently emerging standards. When designing the grid services and their interactions, we also made use of the developed conceptual model of grid resources. This resulted in a few

key grid services, such as resource broker, metadata and information integrator services, which are based on the Grid Resource Ontology. Whenever it was possible, open source solutions were used for development. The test bed is built on the basis of grid services that are part of the GT4 grid middleware (i.e. MDS4, GridFTP, RFT, etc.). For the development of the resource broker, the existing GridBus open source technology was used. Grid technologies, which are incorporated into GT4, allow for achieving greater security within the virtual organisations environment.

From the end users' perspective, the grid is a complex environment in which different grid services are available. The end user should be able to use the system creatively without the need to deal with low level details of the underlying grid infrastructure. A key technological solution that we have found and used in our work is workflow technology.

Workflows can be simple or complex, and, in fact, they represent grid applications. The requirement to improve user-friendliness was satisfied by incorporating two types of client applications in the architecture and design of the grid system. The first possibility is to run grid applications from the Triana workflow editor and manager; the second possibility is to use specialised grid portals. The variety of possibilities for the development of client-side applications is particularly due to the conceptual simplicity of the grid system, which is based on Grid Resource Ontology.

The workflow editor and manager is a very flexible solution and allows for a greater degree of creativity in the discovery of existing grid resources and their integration into complex grid applications. On the basis of executables, which are included in the grid environment, it is possible to form simple and complex workflows and run jobs in parallel (Stankovski et al. 2008, 2008b).

Specialised grid portals are more user-friendly than the editor and manager of workflows, since the complexity of the underlying grid environment is hidden from the end user. For example, a grid portal that exploits data and compute-intensive data processing algorithms was developed for a federated digital library (Trnkoczy, Turk and Stankovski, 2006; Trnkoczy and Stankovski, 2008). We have also demonstrated the benefits of using existing

executables on a grid, for example, ontology learning executables (Stankovski et al., 2008b). These can be contributed by the companies and individuals that participate in the virtual organisation.

The conceptual modeling in this study is limited to some key resources that exist in a complex grid environment. The diversity of data and programs, and the available computation and storage capabilities, is reflected in the developed Grid Resource Ontology. Future studies may also take into account devices like sensors, which are often used in the construction sector, i.e. earthquake engineering, sanitary engineering or even in the manufacturing of smart houses (Trnkoczy and Stankovski, 2006). The Internet transition to the IPv6 protocol will make it possible in the future to interface with such devices via Web services. This will increase the diversity of available resources and facilitate many new complex grid applications.

It may therefore be concluded that semantic technologies were used in the development of the prototype, resulting in a number of advantages over other similar grid systems. The developed prototype and grid test bed:

- supports the development of innovative grid applications and new ways of using existing resources,
- allows for a higher level of abstraction i.e. independence of the grid environment from the individual network applications,
- supports the creation of dynamic virtual organisations,
- allows for seamless integration of existing resources (i.e. executables),
- supports the discovery, selection, and dynamic integration of grid resources during the execution of grid applications,
- allows for relatively secure access to network resources,
- offers the possibility to use two types of user interfaces (workflow editor and manager and specialised grid portals), which increases user-friendliness and flexibility of the grid system,
- is powerful and scalable.

Civil engineering companies can improve their efficiency and flexibility, and further adapt to the one-of-a-kind phenomenon by sharing and using existing resources in the context of virtual organisations and, with the resulting improved possibilities, develop new, innovative grid applications.

The developed simple conceptual model of grid resources and its meaningful integration into the architecture and design of the grid system made it possible to achieve many positive effects. The working prototype of the grid system proves the hypothesis of this study.

Further work. Workflows represent significantly more effective means for sharing the knowledge, implementing processes and improving communication than the individual services they represent. Therefore, it seems reasonable to explore how knowledge could be shared by means of complex workflows, and how workflows could be automatically composed according to specific user requirements.

Viri

Uporabljeni viri

- Adamku, G., Stuckenschmidt, H. 2005. Implementation and Evaluation of a Distributed RDF Storage and Retrieval System. V: Proc. 2005 IEEE/WIC/ACM International Conf. Web Intelligence (WI 05). IEEE CS Press: str. 393–396.
- Adeva, J.J.G., Calvo, R. 2006. Text Mining with Pimiento. IEEE Internet Computing 10, 4: 27–35.
- Allen, G., Angulo, D., Goodale, T., Kielmann, T., Merzky, A., Nabrzysky, J. in sod. 2002. GridLab: Enabling Applications on the Grid. V: Grid Computing – GRID 2002: Third International Workshop. Baltimore, 18. november 2002. Lecture Notes in Computer Science 2536: str. 39–45.
- Alonso, J.M., Hernandez, V., Lopez, R., Molto, G. 2006. Experiences using grid services in structural dynamics. V: Martinez, M. (ur.), Scherer, R. (ur.) Proc. 6th European Conference on Product and Process Modelling. ECPPM 2006. e-Work and eBusiness in Architecture, Engineering and Construction. Valencia, 13.–15. september 2006. London, Taylor & Francis Group plc: str. 359–366.
- Antonioletti, M., Atkinson, M., Baxter, R., Borley, A., Chue Hong, N.P., Collins, B. in sod. 2005. The design and implementation of Grid database services in OGSA–DAI. Concurrency and Computation: Practice and Experience 17, 2–4: 357–376.
- Aristoteles. 1992. Metafizika. Zagreb, HSN.
- Baker, M., Apon, A., Ferner, C., Brown, J. 2005. Emerging Grid Standards. Computer, IEEE Computer Society 38, 4: 43–50.
- Barton, T., Basney, J., Freeman, T., Scavo, T., Siebenlist, F., Welch, V., Ananthakrishnan, R., Baker, B., Goode, M., Keahey, K. 2006. Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, GridShib, and MyProxy. V: Making PKI Easy to Use. Gaithersburg, 4.–6. april 2006. Proc. 5th Annual PKI R&D Workshop.
- Beetz, J., van Leeuwen, J.P., de Vries, B. 2005. An Ontology Web Language Notation of the Industry Foundation Classes. V: Scherer, R.J. (ur.), Katranuschkov, P. (ur.) in Schapke, S.–E. (ur.). Proc. 22nd CIB W78 Conference on Information

- Technology in Construction. Dresden, Technical University of Dresden. CIB publication 304: str. 193–198.
- Berners–Lee, T., Fischetti, M. 1999. Weaving the Web. San Francisco, Harper.
- Berners–Lee, T., Hendler, J., Lassila, O. 2001. The Semantic Web. *Scientific American* 284, 5: 34–43.
- Björk, B.-C. 1989. Basic structure of proposed building product model. *Computer Aided Design* 21, 2: 71–78.
- Bourdeau, M., Giraud–Carrier, F., Rezgui, Y., Zarli A. 2001. Knowledge Management in the Construction Industry: the e–COGNOS approach. V: e.2001 – eBusiness and eWork 2001. Venice, Italy, 17.–19. oktober 2001.
- Broekstra, J., Kampman, A., van Harmelen, F. 2002. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *Proc. International Semantic Web Conf. (ISWC 02)*. LNCS 2342, Springer-Verlag: str. 54–68.
- Brooke, J.M., Parkin, M.S. 2008. Enabling scientific collaboration on the Grid. *Future Generation Computer Systems*, (v tisku).
- Bubak, M., Gubała, T., Kapalka, M., Malawski, M., Rycerz, K. 2005. Workflow composer and service registry for grid applications. *Future Generation Computer Systems* 21, 1: 79–86.
- Buyya, R., Venugopal, S. 2004. The Gridbus toolkit for service oriented grid and utility computing: an overview and status report. *Proc. 1st IEEE International Workshop on Grid Economics and Business Models, 2004 (GECON 2004)*: str. 19–66.
- Cai, M., Frank, M. 2004. RDFPeers: A Scalable Distributed RDF Repository Based on A Structured P2P Network. V: *Proc. 13th International World Wide Web Conf. (WWW 04)*. ACM Press: str. 650–657.
- Camarinha–Matos, L.M., Afsarmanesh, H. 2005. Collaborative networks: A new scientific discipline. *Journal of Intelligent Manufacturing* 16,4–5: 439–452.
- Camarinha–Matos, L.M., Afsarmanesh, H., Lima, C. 1999. Hierarchical Coordination in virtual enterprise infrastructures. *Journal of Intelligent and Robotic Systems* 26,3–4: 267–287.
- Cannataro, M., Talia, D. 2004. Semantics and Knowledge Grids: Building the Next Generation Grid. *IEEE Intelligent Systems* 19, 1: 56–63.
- Canon, S., Chan, S., Olson, D., Tull, C., Welch, V. 2003. Using CAS to Manage Role–Based VO Sub–Groups. V: *Proc. Computing in High Energy Physics (CHEP'03)*.

- Cerovšek, T. 2003. Porazdeljena računalniško integrirana gradnja pri pogojih necelovitosti. Doktorska disertacija. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 308 f.
- Cerovšek, T., Katranuschkov, P. 2006. Active process reuse model for collaboration, Special Issue Process Modelling, Process Management and Collaboration. *ITcon* 11: 467–488.
- Chen, H., Finin, T., Joshi, A. 2003. An Ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review* 18: 197–207.
- Chong, E. in sod. 2005. An Efficient SQL-based RDF Querying Scheme. Proc. 31st International Conf. Very Large Data Bases (VLDB 05). VLDB Endowment: str. 1216–1227.
- Churches, D., Gombas, G., Harrison, A., Maassen, J., Robinson, C., Shields, M., Taylor, I., Wang, I. 2006. Programming scientific and distributed workflow with Triana services. *Concurrency and Computation: Practice and Experience* 18, 10: 1021–1037.
- Cooke, A., Gray, A., Ma, L.S., Nutt, W., Magowan, J., Oevers, M., Taylor, P., Byrom, R., Field, L., Hicks, S., Leake, J., Soni, M., Wilson, A., Cordenonsi, R., Cornwall, L., Djaoui, A., Fisher, S., Podhorszki, N., Coghlan, B., Kenny, S., O'Callaghan, D. 2003. R-GMA: An information integration system for grid monitoring. *Lecture Notes in Computer Science* 2888: 462–481.
- Corcho, O. in sod. 2006. An Overview of S-OGSA: A Reference Architecture for the Semantic Grid. *Elsevier Journal of Web Semantics* 4, 2: 102–115.
- Corcho, O., M. Fernandez-Lopez, and A. Gomez-Perez. 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering* 46, 1: 41–64.
- Czajkowski, K., Kesselman, C., Fitzgerald, S., Foster, I. 2001. Grid Information Services for Distributed Resource Sharing. V: Proc 10th IEEE International Symposium on High-Performance Distributed Computing. IEEE Press: str. 181–194.
- De Roure, D., Jennings, N.R., Shadbolt, N.R. 2003. The Semantic Grid: A future e-Science infrastructure. V: Berman, F. (ur.), Fox, G. (ur.) in Hey, A.J.G. (ur.). *Grid computing – making the global infrastructure a reality*. John Wiley & Sons Ltd.: str. 437–470.
- Decker, S., Fensel, D., van Harmelen, F., Horrocks, I., Melnik, S. Klein, M. Broekstra J. 2000. Knowledge Representation on the Web. V: Proc. International Workshop on Description Logics (DL2000). Aachen, Germany.

- Dolenc, M. 2001. Programske komponente za analizo konstrukcij po metodi končnih elementov. Doktorska disertacija 137. Ljubljana, Univerza v Ljubljani: 205 f.
- Dolenc, M., Katranuschkov, P., Gehre, A., Kurowski, K., Turk, Ž. 2007. InteliGrid platform for virtual organisations interoperability. *ITcon* 12: 459–477.
- Dolenc, M., Klinc, R., Turk, Ž., Katranuschkov, P., Kurowski, K. 2008. Semantic Grid Platform in Support of Engineering Virtual Organisations. *Informatica* 32: 39–49.
- Dolenc, M., Stankovski, V., Turk, Ž. 2005. Grid Technology in Civil Engineering. V: Topping, B.H.V. (ur.). *Innovation in Civil and Structural Engineering Computing*. Saxe–Coburg publications on computational engineering. Stirling, Saxe-Coburg: str. 75–96.
- Eastman, C., Chase, S., Assal, H. 1993. System Architecture for Computer Integration of Design and Construction Knowledge. *Automation in Construction* 2, 2: 95–107.
- Eastman, C., Teicholz, P., Sacks, R., Liston, K. 2008. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. New Jersey, John Wiley & Sons, Inc.
- Ekholm, A. 1996. A Conceptual Framework for Classification of Construction Works. *ITcon* 1: 25–50.
- Fajfar, P. 2000. A nonlinear analysis method for performance based seismic design. *Earthquake Spectra* 16, 3: 573–592.
- Fensel, D. 2000. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Berlin, Springer-Verlag.
- Ferneley, E., Wetherill, M., Rezgui, Y. 2002. Toward the Construction Knowledge Economy: the e-COGNOS project. V: Proc. ECIS 2002. Gdansk, Poland, 6.–8. junij 2002: str. 1508–1516.
- Ferraiolo, D.F., Kuhn, D.R. 1992. Role Based Access Control. Proc. 15th National Computer Security Conference: str. 554–563.
- Foster, I. 2005. Globus Toolkit Version 4: Software for Service-Oriented Systems. V: Jin, H. (ur.), Reed, D. (ur.) in Jiang, W. (ur.). IFIP International Federation for Information Processing, NPC 2005. *Lecture Notes in Computer Science* 3779: str. 2–13.
- Foster, I., Kesselman, C. 1997. Globus: A Metacomputing Infrastructure Toolkit. *International J. Supercomputing Applications* 11, 2: 115–128.
- Foster, I., Kesselman, C. 1998. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, 1. ed.

- Foster, I., Kesselman, C., Nick, J.M., Tuecke, S. 2002. Grid Services for Distributed System Integration. *Computer* 35, 6: 37–46.
- Foster, I., Kesselman, C., Tuecke, S. 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications* 15, 3: 200–222.
- Fox, G. 2003. Data and metadata on the semantic Grid. *Computing in Science & Engineering* 5, 5: 76–78.
- Frey, J., Tannenbaum, T., Livny, M., Foster, I., Tuecke, S. 2002. Condor-G: A Computation Management Agent for Multi-Institutional Grids. *Cluster Computing* 5, 3: 237–246.
- Gehre, A., Katranuschkov, P., Scherer, R.J. 2004. Agent-enabled Peer-To-Peer Infrastructure for Cross-Company Teamwork. V: Dikbas, A. (ur.), Scherer, R.J. (ur.) *Proc. eWork and eBusiness in Architecture, Engineering and Construction ECPPM 2004*. Istanbul, 8.–10. septembra 2004. Balkema, str. 445–452.
- Gehre, A., Katranuschkov, P., Stankovski, V., Scherer, R.J. 2005. Towards Semantic Interoperability in Virtual Organisations. V: Scherer, R.J. (ur.), Katranuschkov, P. (ur.), Schapke, S.-E. (ur.) *Proc. cib-w78 2005 22nd Conference on Information Technology in Construction*. CIB Publication, ISBN 3–86005–478–3: str. 307–314.
- Gemmill, J., Robinson, J.P., Scavo, T., Bangalore, P. 2008. Cross-domain authorization for federated virtual organizations using the myvocs collaboration environment. *Concurrency and Computation: Practice and Experience* (v tisku).
- Grama, A., Gupta, A., Kumar, V. 1993. Isoefficiency Function: A Scalability Metric for Parallel Algorithms and Architecture. V: *Special Issue on Parallel and Distributed Systems: From Theory to Practice*. *IEEE Parallel and Distributed Technology* 1, 3: 12–21.
- Gruber, T. 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies* 43,5–6: 907–928.
- Hannus, M. 1992. Information models for performance driven computer integrated construction. V: Vanier, D.J. (ur.), Thomas, J.R. (ur.) *Joint CIB Workshops on Computers and Information in Construction*. Montreal, 12. –14. maja 1992. Ottawa, NRC-CNRC & CIB Proceedings, Publication 165: str. 258–270.
- Hartmann, D. 2006. Advanced computer methodologies for solving future challenges in engineering and construction. V: Martinez, M. (ur.), Scherer, R. (ur.) *Proc. 6th European Conference on Product and Process Modelling ECPPM 2006. e-Work and eBusiness in Architecture, Engineering and Construction*. Valencia, 13.–15. september 2006. London, Taylor & Francis Group plc: str. 359–366.

- Howard, R., Björk, B.-C. 2007. Building Information Models – Expert's Views on BIM/IFC Developments. w78 maribor, str. 47–54.
- Johnston, W.E. 2004. Semantic Services for Grid-Based, Large-Scale Science. *IEEE Intelligent Systems* 19, 1: 34–39.
- Jurič, M.B. 2007. Storitveno usmerjena arhitektura. *SRC info* 49: 14–18.
- Jurič, M.B., Heričko, M., Rozman, I. 2007. Raziskava o uporabi storitvenih arhitektur v Sloveniji. *Uporab. inform. (Ljubl.)* 15, 1: 4–10.
- Karvonen, I., van den Berg, R., Bernus, P., Fukuda, Y., Hannus, M., Hartel, I., Vesterager, J. 2003. Global Engineering and Manufacturing in Enterprise Networks (GLOBEMEN). *VTT Symp. 2003*: str. 224–395.
- Katranuschkov, P., Scherer, R.J., Turk, Ž. 2001. Intelligent services and tools for concurrent engineering? An approach towards the next generation of collaboration platforms. *Electron. j. inf. tech. constr.* 6: 111–128.
- Keahey, K., Welch, V. 2002. Fine-Grain Authorization for Resource Management in the Grid Environment. *Proc. Grid2002 Workshop*.
- Keahey, K., Welch, V., Lang, S., Liu, B., Meder, S. 2003. Fine-Grain Authorization Policies in the GRID: Design and Implementation. *Proc. 1st International Workshop on Middleware for Grid Computing*.
- Kravtsov, V., Stankovski, V., Niessen, T., Schuster, A. 2006. Service-based Resource Brokering for Grid-based Data Mining. *Proc of Int'l Conference on Grid Computing and Applications GCA 2006*. Las Vegas, CSREA: str. 163–169.
- Kumar, V., Gupta, A. 1994. Analyzing Scalability of Parallel Algorithms and Architectures. *Journal of Parallel and Distributed Computing (special issue on scalability)* 22, 3: 379–391.
- Laaroussi, A., Fies, B., Vankeisbelckt, R., Hans, J. 2007. Ontology-aided FMEA for construction robots.
- Lima, C., Wetherill, M., Fies, B., Zarli, A., Rezgui, Y. 2002a. The e-CKMI: an Ontology-enabled platform to support KM in construction. V: *eWork 2002 Conference*. Praga, oktober 2002.
- Lima, C., Wetherill, M., Zarli, A., Bourdeau, M., Rezgui, Y. 2002b. The e-CKMI: the e-COGNOS approach to support Knowledge Management in the Building and Construction Industry. V: *Proc. ECPPM 2002 eWork and eBusiness in AEC*. Portorož, Slovenija, september 2002.
- Litzkow, M., Livny, M. 1990. Experience with the Condor distributed batch system. *Proc. IEEE Workshop on Experimental Distributed Systems*: str. 97–100.

- Lorch, M., Cowles, B. 2004. Conceptual Grid Authorization Framework and Classification. GGF Working Group on Authorization Frameworks and Mechanisms.
- Madduri, R.K., Hood, C.S., Allcock, W.E. 2002. Reliable File Transfer in Grid Environments. Proc. 27th Annual IEEE International Conference on Local Computer Networks (LCN'02).
- Maly, K., Zubair, M., Li, X. 2005. A High Performance Implementation of an OAI-based Federation Service. Proc. 11th International Conference on Parallel and Distributed Systems 1: str. 769–774.
- Marks, E.A., Bell, M. 2006. Service Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology. John Wiley & Sons Ltd.: str. 384.
- Medved, R. 2004. Oblikovanje lokalnih podjetniških grozdov v Sloveniji. Magistrsko delo. Ljubljana, Univerza v Ljubljani: 85 f.
- Missier, P., Alper, P., Corcho, O., Dunlop, I., Goble, C. 2007. Requirements and Services for Metadata Management. IEEE Internet Computing 11, 5: 17–25.
- Niblett, P., Graham, S. 2005. Events and service-oriented architecture: The OASIS Web Services Notification specifications. IBM Systems Journal, Service-Oriented Architecture 44, 4.
- Ogden, C.K, Richards, I.A. 1994. The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. Orlando, Florida, Harcourt Brace Jovanovich Publishers, (prva edicija 1923).
- Pazlar, T., Turk, Ž. 2008. Interoperability in practice: geometric data exchange using the IFC standard. ITcon 13: 362–380.
- Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S. 2001. A Community Authorization Service for Group Collaboration. Proc. IEEE 3rd International Workshop on Policies for Distributed Systems and Networks.
- Petrinja, E. 2007. Upravljanje z izvornimi podatki v navideznih organizacijah inženirskega načrtovanja. Doktorska disertacija. Ljubljana, Univerza v Ljubljani, Ekonomska fakulteta: 253 f.
- Petrinja, E., Stankovski, V., Turk, Ž. 2007. A provenance data management system for improving the product modelling process. Automation in Construction 16, 4: 485–497.
- Plaszczak, P., Wellner Jr., R. 2006. Grid Computing: The Savvy Manager's Guide. Moragan Kaufmann, Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris, San Diego, San Francisco, Singapore, Sydney, Tokyo.

- Platon. 1976. Država. Ljubljana, DZS.
- Porter, M.E. 2000. Location, Competition and Economic Development: Local Clusters in a Global Economy. *Economic Development Quarterly* 14, 1: 15–34.
- Prelog, E. 1975. Metoda končnih elementov. Ljubljana, Univerza v Ljubljani, FAGG.
- Privošnik, M., Marolt, M., Kavčič, A., Divjak, S., Turk, Ž., Stankovski, V., Dolenc, M. 2006. AgentGrid : načrtovanje večagentnih sistemov s pomočjo okolja GRID. Ljubljana: Fakulteta za računalništvo in informatiko: 22 str.
- Salmelin, B. 2006. European competitiveness, new business and work paradigms supported by the EU research. V: Martinez, M. (ur.), Scherer, R. (ur.) Proc. 6th European Conference on Product and Process Modelling. ECPPM 2006. e-Work and eBusiness in Architecture, Engineering and Construction. Valencia, 13.–15. september 2006. London, Taylor & Francis Group plc: str. 3–7.
- Sandhu, R., Coyne, E.J., Feinstein, H.L., Youman, C.E. 1996. Role-Based Access Control Models. *IEEE Computer* 29, 2: 38–47.
- Scavo, T., Welch, V. 2008. A grid authorization model for science gateways. *Concurrency and Computation: Practice and Experience* (v tisku).
- Scherer, R.J. 2007. Product Model Based Collaboration. V: Rebolj, D. (ur.) Proc. Cib–w78 24th Conference on Information Technology in Construction. CIB Publication, 26.–29. 6. 2007, Maribor, Slovenia: str. 11–20.
- Schevers, H., Drogemuller, R. 2005. Converting the Industry Foundation Classes to the Web Ontology Language. V: Proc. 1st International Conference on Semantics, Knowledge and Grid, IEEE Computer Society: str. 73.
- Shadbolt, N., Berners–Lee, T., Hall, W. 2006. The Semantic Web Revisited. *IEEE Intelligent Systems* 21, 3: 96–101.
- Shearer, C. 2000. The CRISP–DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing* 5, 4: 13–22.
- Siebenlist, F., Nagaratnam, N., Welch, V., Neuman, B.C. 2004. Security for Virtual Organizations: Federating Trust and Policy Domains. V: *The Grid: Blueprint for a New Computing Infrastructure* (druga edicija).
- Slak, S. 2002. Grozdi ne zrastejo sami od sebe. Ljubljana, 27. 6. 2002, Dnevnik: str. 10.
- Sotomayor, B., Childers, L. 2006. Globus Toolkig 4: Programming Java Services. Moragan Kaufmann, Amsterdam, Botson, Heidelberg, London, New York, Oxford, Paris, San Diego, San Francisco, Singapore, Sydney, Tokyo.
- Staab, S. (ur.), Studer, R. (ur.) 2004. Handbook on Ontologies. Berlin, Heidelberg, New York, Springer-Verlag.

- Stankovski, V. in sod. 2008b. Digging Deep into the Data Mine with DataMiningGrid. *IEEE Internet Computing* 12, 6: 69–76.
- Stankovski, V., Debeljak, M., Bratko, I., Adamič, M. 1998. Modelling the population dynamics of red deer (*Cervus elaphus* L.) with regard to forest development. *Ecological modelling* 108, 1–3: 145–153.
- Stankovski, V., Dubitzky, W. 2007. Special Section: Data mining in grid computing environments. *Future Generation Computer Systems* 23, 1: 31–33.
- Stankovski, V., Gehre, A., Katranuschkov, P., Balaton, E., Turk, Ž. 2005. *Ontology Services for the Virtual Enterprise*. V: Cunningham, P. (ur.) in Cunningham, M. (ur.). *Innovation and the knowledge economy: issues, applications, case studies. Information and communication technologies and the knowledge economy*. Amsterdam, IOS Press: str. 254–260.
- Stankovski, V., May, M., Franke, J., Schuster, A., McCourt, D., Dubitzky, W. 2004. *A Service Centric Perspective for Data Mining in Complex Problem Solving Environments*. V: Arabnia, H. R. (ur.), Ni, J. (ur.). *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'04) II*: str. 780–787.
- Stankovski, V., Petrinja, E., Dolenc, M., Privošnik, M., Marolt, M., Kavčič, A., Divjak, S., Turk, Ž. 2005. *GridForum.si: Grid infrastruktura za virtualne organizacije. Zaključno poročilo o rezultatih opravljenega raziskovalnega dela na projektu v okviru ciljnega raziskovalnega programa (CRP) »Konkurenčnost Slovenije 2001–2006« za obdobje od 1. 10. 2003 do 30. 09. 2005*. Ljubljana, UL, Fakulteta za gradbeništvo in geodezijo in UL, Fakulteta za računalništvo in informatiko: 19 str.
- Stankovski, V., Swain, M., Kravtsov, V., Niessen, T., Wegener, D., Kindermann, J., Dubitzky, W. 2008a. Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. *Future Generation Computer Systems* 24, 4: 259–279.
- Tangmunarunkit, H., Decker, S., Kesselman, C. 2003. *Ontology-Based Resource Matching in the Grid: The Grid Meets the Semantic Web*. *Proc. 2nd International Semantic Web Conference (ISWC)*, Sanibel Island, Florida, USA.
- Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M. (uredniki). 2007. *Workflows for e-Science Scientific Workflows for Grids*. XXII, ISBN: 978–1–84628–519–6: 530 str., 181 slik.
- Trnkoczy, J., Stankovski, V. 2008. Improving the Performance of Federated Digital Library Services. *Future Generation Computer Systems* 24, 8: 824–832.

- Trnkoczy, J., Turk, Ž., Stankovski, V. 2006. A grid-based architecture for personalized federation of digital libraries. *Library Collections, Acquisitions, and Technical Services* 30, 3–4: 139–153.
- Turk, Ž. 1992. Okolje za računalniško integrirano projektiranje gradbenih konstrukcij. Doktorska disertacija. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 176, 27 f.,
- Turk, Ž. 2001. Phenomenological foundations of conceptual product modelling in architecture, engineering and construction. *Artif. intell. eng.* 15: 83–92.
- Turk, Ž. 2006. Construction informatics: definition and ontology. *Advanced engineering informatics* 20, 2: 187–199.
- Turk, Ž. 2007. Construction Informatics in European Research: Topics and Agendas. *J. Comp. in Civ. Engrg.* 21, 3: 211–219.
- Turk, Ž., Dolenc, M., Stankovski, V., Petrinja, E. 2005. Semantic Grid – Interoperability Solution for Construction VO? V: Selvaraj, H. (ur.), Srimani, P.K. (ur.). *ITCC 2005: International Conference on Information Technology: Coding and Computing*, 4.–6. april 2005, Las Vegas, Nevada. Los Alamitos, California: IEEE Computer Society: str. 795–796.
- Turk, Ž., Duhovnik, J. 1995. Slovenia and computer representation of design standards and building codes. *International journal of construction information technology* 3, 1: 55–71.
- Turk, Ž., Stankovski, V., Dolenc, M., Cerovšek, T. 2004a. Semantic extensions for the virtual enterprise's grid infrastructure. V: Arabnia, H.R. (ur.). *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications : PDPTA '04*. 21.–24. junij 2004. Las Vegas, Nevada, USA. Sunnyvale, California: CSREA: str. 788–793.
- Turk, Ž., Stankovski, V., Dolenc, M., Cerovšek, T. 2004b. Semantic grid infrastructure for AEC virtual enterprise – a research agenda. V: *International conference on construction information technology (INCITE 2004): World IT for design & construction*. Langkawi, Malaysia, 18.–21. februar 2004. ISBN 983–2724–17–1: str. 409–416.
- Vamvatsikos, D., Cornell, A. 2006. Direct estimation of the seismic demand and capacity of oscillators with multi-linear static pushovers through IDA. *Earthquake engineering and structural dynamics* 35: 1097–1117.
- Venugopal, S., Buyya, R., Winton, L. 2004. A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids. V: *Proc. 2nd workshop on Middleware for grid computing*. Toronto, Ontario, Canada: str. 75–80.

- Welch, V., Foster, I., Scavo, T., Siebenlist, F., Catlett, C., Gemmill, G., Skow, D. 2007. Scaling teragrid access: A testbed for identity management and attribute-based authorization. V: Proc. TeraGrid Conference 2007.
- Welch, V., in sod. 2003. Security for Grid Services. V: Proc. 12th IEEE International Symposium on High Performance Distributed Computing. IEEE Press.
- Xing, W., Dikaiakos, M.D., Sakellariou, R. 2006. A Core Grid Ontology for the Semantic Grid. V: Sixth IEEE International Symposium on Cluster Computing and the Grid CCGRID 06, May 2006 Page(s). vol. 1, 16–19: str. 178–184.
- Yurchyshyna, A., Zucker, C.F., Le Thanh, N., Lima, C., Zarli, A. 2007. Towards an ontology-based approach for conformance checking modeling in construction. V: Rebolj, D. (ur.) Proc. Cib-w78 24th Conference on Information Technology in Construction. CIB Publication, 26.–29. junij 2007, Maribor, Slovenija: str. 195–202.
- Zhugue, H. 2004. Semantics, Resource and Grid. Future Generation Computer Systems 20, 1: 1–5.

Spletni viri

- ABAQUS. <http://www.abaqus.com/> (12. 6. 2008).
- Adida, B., Birbeck, M. RDFa Primer 1.0: Embedding RDF in XHTML. W3C working draft, maj 2006. <http://www.w3.org/TR/xhtml-rdfa-primer/> (12. 6. 2008).
- AgentGrid. Načrtovanje večagentnih sistemov s pomočjo okolja grid. <http://www.agentgrid.org/> (17. 4. 2008).
- AIM. Instant Messenger. <http://dashboard.aim.com/aim> (12. 6. 2008).
- ANSYS. <http://www.ansys.com/> (12. 6. 2008).
- ARPANET. Advanced Research Projects Agency Network. <http://en.wikipedia.org/wiki/ARPANET> (12. 6. 2008).
- AUTODESK. <http://usa.autodesk.com/> (17. 4. 2008).
- Axelsson, J. in sod. XHTML 2.0. W3C working draft, julij 2006. <http://www.w3.org/TR/2006/WD-xhtml2-20060726Overview.html> (12. 6. 2008).
- Bergman, M.E., Feser, J.E. Industrial and Regional Clusters: Concepts and Comparative Applications. <http://www.rri.wvu.edu/WebBook/Bergman-Feser/contents.htm> (12. 6. 2008).
- Boca. Boca RDF shramba. <http://ibm-slrp.sourceforge.net> (17. 4. 2008).

- BPEL4WS. Web Services Business Process Execution Language. OASIS Standard WS-BPEL 2.0. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html> (12. 6. 2008).
- Cactus. An Open Source problem solving environment. <http://www.cactuscode.org/> (12. 6. 2008).
- Condor. Condor Version 6.8.2 Manual.
<http://www.cs.wisc.edu/condor/manual/v6.8/ref.html> (17. 4. 2008).
- CORBA. Common Object Request Broker Architecture.
<http://www.omg.org/gettingstarted/corbafaq.htm> (12. 6. 2008).
- CoreGRID. The European Research Network on Foundations, Software Infrastructures and Applications for Large Scale Distributed, Grid and Peer-to-Peer Technologies. <http://www.coregrid.net/> (12. 6. 2008).
- CRISP-DM. <http://www.crisp-dm.org/> (17. 4. 2008).
- CSF. Community Scheduling Framework. Programska oprema.
<http://sourceforge.net/projects/gcsf/> in
http://www.globus.org/Grid_software/computation/csf.php (17. 4. 2008).
- Data Design System. <http://www.dds-cad.co.uk/> (17. 4. 2008).
- DataGrid. VOMS Architecture version 1.1. 2003. http://grid-auth.infn.it/docs/VOMS-v1_1.pdf (29.6.2007).
- DataMiningGrid. 2008. Data Mining Tools and Services for Grid Computing Environments. <http://www.datamininggrid.org/> (17. 4. 2008).
- DC. Dublin Core Metadata Glossary.
<http://library.csun.edu/mwoodley/dublincoreglossary.html> (17. 4. 2008).
- DCOM. Distributed Component Object Model.
<http://www.microsoft.com/com/default.mspx> (12. 6. 2008)
- DEISA. Distributed European Infrastructure for Supercomputing Applications.
<https://www.deisa.org/> (17. 4. 2008).
- DMTF. Distributed Management Task Force. <http://www.dmtf.org/home> (12. 6. 2008).
- EAD. Encoded Archival Description. <http://www.loc.gov/ead/> (17. 4. 2008).
- EGEE. <http://www.eu-egee.org/> (17. 4. 2008).
- EPM Technology. <http://www.epmtech.jotne.com/> (17. 4. 2008).
- ETSI firms up on GRID. 2006. ETSI news release (6. 6. 2006).
http://www.etsi.org/pressroom/Previous/2006/2006_07_grid.htm (17. 4. 2008).

- FaCT. Fast Classification of Terminologies.
<http://www.cs.man.ac.uk/~horrocks/FaCT/> (12. 6. 2008).
- FaCT++. The FaCT reasoner implemented in C++.
<http://owl.man.ac.uk/factplusplus/> (12. 6. 2008).
- FIRE. The European FIRE: Future Internet Research and Experimentation Activities.
http://www.panlab.net/fileadmin/documents/FIRE/FIRE_report_14_June_2007_distribution_info_day.pdf (12. 6. 2008).
- Flickr. <http://www.flickr.com> (12. 6. 2008).
- Foster, I. 2002. What is the Grid? A three point checklist. Grid Today.
<http://www.gridtoday.com/02/0722/100136.html> (17. 4. 2008).
- Foster, I., Kishimoto, H., Savva, A., Berry, D., Djaoui, A., Grimshaw, A., Horn, B., Maciel, F., Siebenlist F., Subramaniam R., Treadwell J., Von Reich J. GWD-I (draft-ggf-ogsa-spec-1.5-008). The Open Grid Services Architecture, Version 1.5. http://www.ogf.org/Public_Comment_Docs/Documents/Apr-2006/draft-ggf-ogsa-spec-1.5-008.pdf (17. 4. 2008).
- Foster, I., Liming, L. 2004. The Role of Standards In The Grid. Software Tech News, Volume 7, Number 1 – Grid Computing.
<http://www.softwaretechnews.com/stn7-1/grid-standards.html> (17. 4. 2008).
- Foster, I., Maguire, T., Snelling, D. GWD-R (draft-ggf-ogsa-wsrf-basic-profile-043). OGSA WSRF Basic Profile 1.0.
http://www.ogf.org/Public_Comment_Docs/Documents/Oct-2005/draft-ggf-ogsa-wsrf-basic-profile-v43.pdf (17. 4. 2008).
- Future for European Grids. GRIDs and Service Oriented Knowledge Utilities, Vision and Research Directions 2010 and Beyond. Next Generation GRIDs Expert Group Report 3. Cracow Grid Workshop 2006.
<http://www.cyfronet.pl/cgw06/agenda.html> (17. 4. 2008).
- Gehrig, D. 2004. Guide to the NEESGrid Reference Implementation. Tehnično poročilo. NEESGrid. http://NEESGrid.ncsa.uiuc.edu/documents/TR_2004_04.pdf (17. 4. 2008).
- GEODISE. Grid Enabled Optimisation and Design Search for Engineering.
<http://www.geodise.org/> (12. 6. 2008).
- gLite. Lightweight Middleware for Grid Computing. <http://glite.web.cern.ch/glite/> (12. 6. 2008).
- Globus Toolkit. <http://www.globus.org/toolkit/> (17. 4. 2008).
- GLUE. European Efforts Usable Ontology of Grid Resource Descriptions.
<http://www.semanticgrid.org/GGF/ggf13/ggf13-semantic.ppt> (12. 6. 2008).

- GRAPHISOFT. <http://www.graphisoft.com/> (17. 4. 2008).
- GridBus. Grid Service Broker – a grid scheduler for computational and data grids.
<http://www.Gridbus.org/broker/> (17. 4. 2008).
- GridForum.si. Grid infrastruktura za virtualne organizacije. <http://www.gridforum.si/>
(17. 4. 2008).
- Gridipedia. The European Grid Marketplace. <http://www.gridipedia.eu/> (12. 6. 2008).
- GridLab. <http://www.gridlab.org/> (17. 4. 2008).
- GridPortlets. <http://www.gridsphere.org/> (17. 4. 2008).
- GRMS. GridLab Resource Management System.
<http://www.Gridlab.org/WorkPackages/wp-9/> (17. 4. 2008).
- GRO. Grid Resource Ontology. <http://www.stankovski.net/gro/gro.owl> (17. 4. 2008).
- GSI. The Globus Security Team, Globus Toolkit Version 4 Grid Security
Infrastructure: A Standards Perspective.
<http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf> (17. 4. 2008).
- GT4. Globus Toolkit 4. <http://www.globus.org/toolkit/docs/4.0/> (17. 4. 2008).
- Herman, I. Semantic Web Adoption and Applications.
<http://www.w3.org/People/Ivan/CorePresentations/Applications/Applications.pdf>
(12. 6. 2008).
- HPC-Europa. <http://www.hpc-europa.org/> (12. 6. 2008).
- IAI. International Alliance for Interoperability. <http://www.iai-international.org> (17. 4. 2008).
- ICH. Interoperability Clearinghouse Glossary of Terms.
<http://www.ichnet.org/glossary.htm> (17. 4. 2008).
- ICQ. I seek you. <http://www.icq.com/> (12. 6. 2008).
- IETF. The Internet Engineering Task Force. <http://www.ietf.org/> (12. 6. 2008).
- IFC. Industry Foundation Classes. <http://www.fiotech.org/projects/idim/ifcs.htm> (17. 4. 2008).
- iLane. www.intellimec.com/news/ (17. 4. 2008).
- InteliGrid. Interoperability of Virtual Organisations on a Complex Semantic Grid.
<http://www.inteliGrid.com/> (17. 4. 2008).
- ISTforCE. Intelligent Services and Tools for Concurrent Engineering. <http://cib.bau.tu-dresden.de/istforce/> (17. 4. 2008) in <http://istforce.eu-project.info/> (17. 4. 2008).

Jena. A semantic web framework for Java. <http://jena.sourceforge.net/> (12. 6. 2008).

KAON2. Infrastructure for managing OWL-DL, SWRL, and F-Logic ontologies.
<http://kaon2.semanticweb.org/> (12. 6. 2008).

Kepler. Kepler Project. <http://kepler-project.org/> (12. 6. 2008).

Kowari Metastore. An Open Source, massively scalable, transaction-safe, purpose-built database for the storage, retrieval and analysis of metadata.
<http://www.kowari.org/> (12. 6. 2008).

K-WfGrid. Knowledge-based Workflow System for Grid Applications.
<http://www.kwfgrid.eu/> (12. 6. 2008).

LCG-2. <http://lcg.web.cern.ch/LCG/> (17. 4. 2008).

LHC. Large Hadron Collider Beauty Computing Grid – Technical Design Report.
http://lcg.web.cern.ch/LCG/tdr/LCG_TDR_v1_04.pdf (17. 4. 2008) in <http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/> (17. 4. 2008) in <http://lhcb-public.web.cern.ch/> (17. 4. 2008).

MARC. <http://www.loc.gov/marc/> (17. 4. 2008).

MDS4. Monitoring and discovery in a web services framework: Functionality and performance of Globus Toolkit MDS4. Technical Report. Mathematics and Computer Science Division, Argonne National Laboratory. <http://www-unix.mcs.anl.gov/~schopf/Pubs/mds4.hpdc06.pdf> (17. 4. 2008).

MPI. The Message Passing Interface (MPI) standard. <http://www-unix.mcs.anl.gov/mpi/> (12. 6. 2008).

MSC.NASTRAN. <http://www.mssoftware.com/> (12. 6. 2008).

NEES. <http://www.nees.org/> (17. 4. 2008).

NEESit. NEES Cyberinfrastructure Center. <http://it.nees.org/> (17. 4. 2008).

Next Generation Grids 2. Requirements and Options for European Grids Research 2005–2010 and Beyond, Expert Group Report.
ftp://ftp.cordis.europa.eu/pub/ist/docs/ngg2_eg_final.pdf (17. 4. 2008).

NG4J. Named Graphs for Jena. <http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/> (17. 4. 2008).

Nimrod/G. <http://www.csse.monash.edu.au/~davida/nimrod/> (17. 4. 2008).

Nimrod/G. Tools for Distributed Parametric Modelling.
<http://www.csse.monash.edu.au/~davida/nimrod/nimrodg.htm> (12. 6. 2008).

NorduGrid. <http://www.nordugrid.org/> (12. 6. 2008).

- OAI-PMH. The Open Archives Initiative Protocol for Metadata Harvesting. Protocol Version 2.0 of 14. 6. 2002.
<http://www.openarchives.org/OAI/openarchivesprotocol.html> (17. 4. 2008).
- OASIS. <http://www.oasis-open.org/> (17. 4. 2008).
- OGF. Odprti forum za mrežo = Open Grid Forum. <http://www.gridforum.org/> (17. 4. 2008) in <http://www.ogf.org> (17. 4. 2008).
- OGSA. Open Grid Services Architecture. <http://www.globus.org/ogsa/> (17. 4. 2008).
- OGSA-DAI. <http://www.ogsadai.org.uk/> (17. 4. 2008).
- OGSI. Open Grid Services Infrastructure Version 1.0. http://www-unix.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf (17. 4. 2008).
- OMII-Europe. <http://omii-europe.org/OMII-Europe/> (12. 6. 2008).
- OntoGrid. <http://www.ontogrid.net> (12. 6. 2008).
- Ontoprise. <http://www.ontoprise.de/> (12. 6. 2008).
- OpenSees. <http://opensees.berkeley.edu/index.php> (17. 4. 2008).
- OWL. The Web Ontology Language. <http://www.w3.org/TR/owl-features/> (17. 4. 2008).
- OWL-FEATURES. <http://www.w3.org/TR/owl-features/> (17. 4. 2008).
- OWL-GUIDE. <http://www.w3.org/TR/owl-guide/> (17. 4. 2008).
- OWL-S. Semantic Markup for Web Services. <http://www.w3.org/Submission/OWL-S/> (12. 6. 2008).
- Pellet. Pellet OWL-DL Reasoner. <http://pellet.owldl.com/> (12. 6. 2008).
- Pipes. <http://pipes.yahoo.com/pipes/> (17. 4. 2008).
- Poročilo DJRA1.1. EGEE Middleware Architecture.
<https://edms.cern.ch/file/476451/1.0/architecture.pdf> (17. 4. 2008).
- Poročilo DNA4.1. EGEE Application Deployment Plan.
<https://edms.cern.ch/document/722131/1> (17. 4. 2008).
- Protégé. The Protégé ontology editor. <http://protege.stanford.edu/> (12. 6. 2008).
- ProvenanceGrid. Enabling and Supporting Provenance in Grids for Complex Problems.
<http://www.gridprovenance.org> (12.6.2008).
- QosCosGrid. Quasi-Opportunistic Supercomputing for Complex Systems in Grid Environments. <http://www.qoscosgrid.eu/> (12. 6. 2008).
- RacerPro. RacerPro – an OWL reasoner and inference server for the Semantic Web.
<http://www.racer-systems.com/> (12. 6. 2008).

RBAC. Role Based Access Control (RBAC) and Role Based Security.

<http://csrc.nist.gov/groups/SNS/rbac/> (12. 6. 2008).

RDF. Resource Description Framework. <http://www.w3.org/RDF/> (17. 4. 2008).

RMI. Remote Method Invocation.

<http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp> (12. 6. 2008)

Rossiter, N. The challenge of Interoperability: The Grid, Semantic Web and Categories.

<http://computing.unn.ac.uk/staff/cgnr1/> (17. 4. 2008).

RPC. Remote Procedure Call. <http://www.opengroup.org/onlinepubs/9629399/> (12. 6. 2008).

SAML. Technical Overview of the OASIS Security Assertion Markup Language V1.1.

<http://www.oasis-open.org> (17.4.2008).

SciX. Open, self organising repository for scientific information exchange.

<http://www.scix.net> (17. 4. 2008).

Sesame2. <http://www.openrdf.org> (17. 4. 2008).

Sibboleth. <http://symlabs.com/> (17. 4. 2008).

SiGNet. Pilotna postavitev tehnologije Grid v Sloveniji. [http://www-](http://www-f9.ijs.si/egee/signet.html)

[f9.ijs.si/egee/signet.html](http://www-f9.ijs.si/egee/signet.html) (12. 6. 2008).

SIMDAT. Grids for Industrial Product Development.

<http://www.scai.fraunhofer.de/simdat.html> (12. 6. 2008).

SIMULIA. <http://www.simulia.com/> (17. 4. 2008).

SNIA. Storage Networking Industry Association. <http://www.snia.org/home> (12. 6. 2008).

SOA Institute. <http://www.soainstitute.org/> (17. 4. 2008).

SOAP. Simple Object Access Protocol ver. 1.2. <http://www.w3.org/TR/soap/> (12. 6. 2008).

SOFISTIK. <http://www.sofistik.com/> (17. 4. 2008).

Sparql. Protokol Sparql. <http://www.w3.org/TR/rdf-sparql-protocol/> (17. 4. 2008).

STEP. Standard for the Exchange of Product model data. ISO 10303.

<http://www.iso.org/iso/> (12. 6. 2008).

Taverna. Taverna Workbench. <http://taverna.sourceforge.net/> (12. 6. 2008).

TLS. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346.

<http://tools.ietf.org/html/rfc4346> (17. 4. 2008).

Triana. <http://www.trianacode.org/> (17. 4. 2008).

- UDDI. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec (17. 4. 2008).
- UniGrids. Uniform Interface to Grid Services. <http://www.unigrids.org/> (12. 6. 2008).
- W3C. The World Wide Web Consortium. <http://www.w3.org/> (12. 6. 2008).
- Wikipedia. <http://www.wikipedia.org> (12. 6. 2008).
- WS-Addressing. Web Services Addressing 1.0 – Core, W3C Recommendation 9. 5 2006. <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/> (17. 4. 2008).
- WSDL. Web Services Description Language ver. 1.1. <http://www.w3.org/TR/wsdl/> (12. 6. 2008).
- WS-Eventing. Web Services Eventing. W3C Member Submission. <http://www.w3.org/Submission/WS-Eventing/> (17. 4. 2008).
- WS-I. Web Services Interoperability Organization. <http://www.ws-i.org/> (12. 6. 2008).
- WSMF. The Web Service Modeling Framework. <http://www.swsi.org/resources/wsmf-paper.pdf> (12. 6. 2008).
- WSMO. Web Service Modeling Ontology. <http://www.wsmo.org/> (12. 6. 2008).
- WSN. Družina specifikacij OASIS WSN. http://www.oasis-open.org/committees/documents.php?wg_abbrev=wsn (17. 4. 2008).
- WSRF. OASIS Web Services Resource Framework (WSRF) TC. <http://www.oasis-open.org/committees/> (17. 4. 2008).
- WSRF. Web Services Resource Framework. OASIS. Primer v1.2. <http://docs.oasis-open.org/wsr/wsrf-primer-1.2-primer-cd-01.pdf> (29. 6. 2007).
- X.509. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820. <http://www.ietf.org/rfc/rfc3820.txt> (17. 4. 2008).
- XACML. Extensible Access Control Markup Language. <http://www.oasis-open.org/committees/xacml>
- XML. eXtensible Markup Language. <http://www.w3.org/XML/> (12. 6. 2008).
- XPath. XML Path Language (XPath) Version 1.0. <http://www.w3.org/TR/xpath> (12. 6. 2008).
- XQuery. W3C XML Query (XQuery). <http://www.w3.org/XML/Query/> (17. 4. 2008).
- XtreemOS. Building and Promoting a Linux-based Operating System to Support Virtual Organizations for Next Generation Grids. <http://www.xtreemos.org/> (12. 6. 2008).
- YouTube. <http://www.youtube.com> (12. 6. 2008).

Prilogi

Priloga A Ontologija mrežnih virov

Ontologija mrežnih virov (Grid Resource Ontology, GRO) je dostopna tudi na spletnem naslovu: <http://www.stankovski.net/gro/gro.owl>.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns="http://www.stankovski.net/gro#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.stankovski.net/gro">
  <owl:Ontology rdf:about="">
    <rdfs:comment xml:lang="en">Grid Resource Ontology. Developed by Vlado Stankovski. (c)
2004-2008.</rdfs:comment>
  </owl:Ontology>
  <owl:Class rdf:ID="Grid">
    <owl:disjointWith>
      <owl:Class rdf:ID="GridResource"/>
    </owl:disjointWith>
    <owl:disjointWith>
      <owl:Class rdf:ID="RealWorld"/>
    </owl:disjointWith>
  </owl:Class>
  <owl:Class rdf:ID="EarthquakeEngineering">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="CivilEngineeringDomain"/>
    </rdfs:subClassOf>
```

```
</owl:Class>
<owl:Class rdf:ID="ArchitectureRequirement">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="ApplicationDescription"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="architecture"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="Language"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="OperatingSystemRequirement"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="GeneralInformation"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="ParameterLoop">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Application"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="Requirement"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Option"/>
  </owl:disjointWith>
</owl:Class>
```

```
<owl:Class rdf:about="#GridResource">
  <owl:disjointWith>
    <owl:Class rdf:about="#RealWorld"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Grid"/>
</owl:Class>
<owl:Class rdf:ID="SingleJobProvenance">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Provenance"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="SingleGramProvenance"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#Option">
  <owl:disjointWith>
    <owl:Class rdf:about="#Requirement"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#ParameterLoop"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Application"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Environment">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="variable"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
```

```

    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:ID="value"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:ID="toolTip"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#Grid"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>The environment variables to be set at the execution machine before
execution.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="DirectlyExecutableLanguage">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Language"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="SandboxLanguage"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#Application">
  <rdfs:subClassOf rdf:resource="#GridResource"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasApplicationDescription"/>

```

```
</owl:onProperty>
  <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith>
  <owl:Class rdf:ID="Person"/>
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="Device"/>
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="Data"/>
</owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="FEASoftware">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#CivilEngineeringDomain"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="InformationIntegratorService">
  <owl:disjointWith>
    <owl:Class rdf:ID="WS-GRAM"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="ResourceBroker"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="OntologyService"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="MetadataService"/>
  </owl:disjointWith>
<rdfs:subClassOf>
  <owl:Class rdf:ID="SemanticGrid"/>
</rdfs:subClassOf>
```



```
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#SingleGramProvenance">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Provenance"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#SingleJobProvenance"/>
</owl:Class>
<owl:Class rdf:about="#SandboxLanguage">
  <owl:disjointWith rdf:resource="#DirectlyExecutableLanguage"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Language"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="EcosystemModelling">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#CivilEngineeringDomain"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Person">
  <rdfs:subClassOf rdf:resource="#GridResource"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Device"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#Data"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#Application"/>
</owl:Class>
<owl:Class rdf:ID="GramIPRequirement">
  <owl:disjointWith>
    <owl:Class rdf:ID="MinMemoryRequirement"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="MinDiskSpaceRequirement"/>
```

```
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="GramJobManagerRequirement"/>
</owl:disjointWith>
<rdfs:subClassOf>
  <owl:Class rdf:about="#Requirement"/>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:ID="label"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:about="#value"/>
    </owl:onProperty>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:ID="unit"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

```
<owl:Class rdf:about="#MinDiskSpaceRequirement">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Requirement"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#label"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#value"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#unit"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:equivalentProperty rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    ></owl:equivalentProperty>
  <owl:disjointWith>
    <owl:Class rdf:about="#MinMemoryRequirement"/>
  </owl:disjointWith>
</owl:Class>
```

```
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:about="#GramJobManagerRequirement"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#GramIPRequirement"/>
</owl:Class>
<owl:Class rdf:about="#ApplicationDescription">
  <rdfs:subClassOf rdf:resource="#Application"/>
</owl:Class>
<owl:Class rdf:about="#RealWorld">
  <owl:disjointWith rdf:resource="#GridResource"/>
  <owl:disjointWith rdf:resource="#Grid"/>
</owl:Class>
<owl:Class rdf:about="#Language">
  <rdfs:subClassOf rdf:resource="#ApplicationDescription"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#GeneralInformation"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#ArchitectureRequirement"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#OperatingSystemRequirement"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="RemoteFileDir">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="StorageDevice"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Provenance">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Data"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="OptionData"/>
  </owl:disjointWith>
```

```
<owl:disjointWith>
  <owl:Class rdf:ID="DataOutput"/>
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:ID="DataInput"/>
</owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="OAI-PMH-Related">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="DigitalLibrary"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >An OAI-PMH compliant digital library harvester.</rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#CivilEngineeringDomain">
  <rdfs:subClassOf rdf:resource="#RealWorld"/>
  <owl:disjointWith>
    <owl:Class rdf:ID="DataMiningDomain"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="BusinessProcessDomain"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#StorageDevice">
  <owl:disjointWith>
    <owl:Class rdf:ID="ProcessingDevice"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Device"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#OperatingSystemRequirement">
  <rdfs:subClassOf rdf:resource="#ApplicationDescription"/>
  <rdfs:subClassOf>
    <owl:Restriction>
```

```
<owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:cardinality>
<owl:onProperty>
  <owl:DatatypeProperty rdf:ID="os"/>
</owl:onProperty>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
    >1</owl:cardinality>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:about="#label"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#Language"/>
<owl:disjointWith>
  <owl:Class rdf:about="#GeneralInformation"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#ArchitectureRequirement"/>
</owl:Class>
<owl:Class rdf:ID="RemoteProtocol">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="remoteProtocol"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Grid"/>
</owl:Class>
<owl:Class rdf:about="#DataOutput">
```

```
<owl:disjointWith rdf:resource="#Provenance"/>
<owl:disjointWith>
  <owl:Class rdf:about="#OptionData"/>
</owl:disjointWith>
<owl:disjointWith>
  <owl:Class rdf:about="#DataInput"/>
</owl:disjointWith>
<rdfs:subClassOf>
  <owl:Class rdf:about="#Data"/>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#MetadataService">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#SemanticGrid"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#WS-GRAM"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#ResourceBroker"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#OntologyService"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#InformationIntegratorService"/>
</owl:Class>
<owl:Class rdf:about="#BusinessProcessDomain">
  <rdfs:subClassOf rdf:resource="#RealWorld"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#DataMiningDomain"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#CivilEngineeringDomain"/>
</owl:Class>
<owl:Class rdf:about="#Data">
  <owl:disjointWith rdf:resource="#Person"/>
```

```
<owl:disjointWith>
  <owl:Class rdf:about="#Device"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#Application"/>
<rdfs:subClassOf rdf:resource="#GridResource"/>
</owl:Class>
<owl:Class rdf:ID="MeshGenerator">
  <rdfs:subClassOf rdf:resource="#FEASoftware"/>
</owl:Class>
<owl:Class rdf:about="#DigitalLibrary">
  <rdfs:subClassOf rdf:resource="#CivilEngineeringDomain"/>
</owl:Class>
<owl:Class rdf:about="#OntologyService">
  <owl:disjointWith>
    <owl:Class rdf:about="#WS-GRAM"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:about="#ResourceBroker"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#MetadataService"/>
  <owl:disjointWith rdf:resource="#InformationIntegratorService"/>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#SemanticGrid"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ValueList">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="listItem"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
```



```
<rdfs:subClassOf rdf:resource="#Data"/>
</owl:Class>
<owl:Class rdf:about="#GramJobManagerRequirement">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Requirement"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="jobManagerList"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#label"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:about="#MinMemoryRequirement"/>
  </owl:disjointWith>
  <owl:disjointWith rdf:resource="#MinDiskSpaceRequirement"/>
  <owl:disjointWith rdf:resource="#GramIPRequirement"/>
</owl:Class>
<owl:Class rdf:about="#WS-GRAM">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#SemanticGrid"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
```

```
<owl:Class rdf:about="#ResourceBroker"/>
</owl:disjointWith>
<owl:disjointWith rdf:resource="#OntologyService"/>
<owl:disjointWith rdf:resource="#MetadataService"/>
<owl:disjointWith rdf:resource="#InformationIntegratorService"/>
</owl:Class>
<owl:Class rdf:about="#ProcessingDevice">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Device"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#StorageDevice"/>
</owl:Class>
<owl:Class rdf:ID="Fork">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="JobManager"/>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="Condor"/>
  </owl:disjointWith>
</owl:Class>
<owl:Class rdf:about="#Condor">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#JobManager"/>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Fork"/>
</owl:Class>
<owl:Class rdf:about="#GeneralInformation">
  <rdfs:subClassOf rdf:resource="#ApplicationDescription"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="longDescription"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
</owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#Language"/>
<owl:disjointWith rdf:resource="#OperatingSystemRequirement"/>
<owl:disjointWith rdf:resource="#ArchitectureRequirement"/>
</owl:Class>
<owl:Class rdf:about="#JobManager">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#SemanticGrid"/>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  ></rdfs:comment>
</owl:Class>
<owl:Class rdf:about="#SemanticGrid">
  <rdfs:subClassOf rdf:resource="#Grid"/>
</owl:Class>
<owl:Class rdf:about="#MinMemoryRequirement">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Requirement"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#label"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#value"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
```

```
>1</owl:cardinality>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    <owl:onProperty>
      <owl:DatatypeProperty rdf:about="#unit"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#MinDiskSpaceRequirement"/>
<owl:disjointWith rdf:resource="#GramJobManagerRequirement"/>
<owl:disjointWith rdf:resource="#GramIPRequirement"/>
</owl:Class>
<owl:Class rdf:about="#ResourceBroker">
  <owl:disjointWith rdf:resource="#WS-GRAM"/>
  <owl:disjointWith rdf:resource="#OntologyService"/>
  <owl:disjointWith rdf:resource="#MetadataService"/>
  <owl:disjointWith rdf:resource="#InformationIntegratorService"/>
  <rdfs:subClassOf rdf:resource="#SemanticGrid"/>
</owl:Class>
<owl:Class rdf:about="#Device">
  <owl:disjointWith rdf:resource="#Person"/>
  <owl:disjointWith rdf:resource="#Data"/>
  <owl:disjointWith rdf:resource="#Application"/>
  <rdfs:subClassOf rdf:resource="#GridResource"/>
</owl:Class>
<owl:Class rdf:about="#OptionData">
  <owl:disjointWith rdf:resource="#Provenance"/>
  <owl:disjointWith rdf:resource="#DataOutput"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#DataInput"/>
  </owl:disjointWith>
```

```
<rdfs:subClassOf rdf:resource="#Data"/>
</owl:Class>
<owl:Class rdf:about="#DataMiningDomain">
  <owl:disjointWith rdf:resource="#CivilEngineeringDomain"/>
  <owl:disjointWith rdf:resource="#BusinessProcessDomain"/>
  <rdfs:subClassOf rdf:resource="#RealWorld"/>
</owl:Class>
<owl:Class rdf:about="#Requirement">
  <owl:disjointWith rdf:resource="#ParameterLoop"/>
  <owl:disjointWith rdf:resource="#Option"/>
  <rdfs:subClassOf rdf:resource="#Application"/>
</owl:Class>
<owl:Class rdf:about="#DataInput">
  <owl:disjointWith rdf:resource="#Provenance"/>
  <owl:disjointWith rdf:resource="#OptionData"/>
  <owl:disjointWith rdf:resource="#DataOutput"/>
  <rdfs:subClassOf rdf:resource="#Data"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="jobsStatus">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Provenance meta data is attached to each single job.</rdfs:comment>
  <rdfs:domain rdf:resource="#Provenance"/>
  <rdfs:range rdf:resource="#SingleJobProvenance"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasApplicationDescription">
  <rdfs:range rdf:resource="#ApplicationDescription"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Each grid-enabled application has an ApplicationDescription.</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isGridEnabled">
  <rdfs:domain rdf:resource="#GridResource"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Some resources can be grid enabled.</rdfs:comment>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="gramsStatus">
```

```
<rdfs:domain rdf:resource="#Provenance"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Status of the WS-GRAM service.</rdfs:comment>
<rdfs:range rdf:resource="#SingleGramProvenance"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="provenance">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Provenance metadata.</rdfs:comment>
<rdfs:domain rdf:resource="#ApplicationDescription"/>
<rdfs:range rdf:resource="#Provenance"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#jobManagerList">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Currently, the job manager can be Fork or Condor.</rdfs:comment>
<rdfs:range rdf:resource="#JobManager"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="optionData">
<rdfs:range>
<owl:DataRange>
<owl:oneOf rdf:parseType="Resource">
<rdf:rest rdf:parseType="Resource">
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>posint</rdf:first>
<rdf:rest rdf:parseType="Resource">
<rdf:rest rdf:parseType="Resource">
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>double</rdf:first>
<rdf:rest rdf:parseType="Resource">
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>boolean</rdf:first>
<rdf:rest rdf:parseType="Resource">
<rdf:rest rdf:parseType="Resource">
<rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>list</rdf:first>
```

```
</rdf:rest>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>string</rdf:first>
</rdf:rest>
</rdf:rest>
</rdf:rest>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>float</rdf:first>
</rdf:rest>
</rdf:rest>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>int</rdf:first>
</owl:oneOf>
</owl:DataRange>
</rdfs:range>
<rdfs:domain rdf:resource="#OptionData"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Options for parameter sweeps.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#toolTip">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Tool tip for the various user interfaces.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="shortDescription">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>At most three sentences long description of the executable.</rdfs:comment>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="parameterName">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Name of application parameter.</rdfs:comment>
<rdfs:domain rdf:resource="#ParameterLoop"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:ID="execution">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >javaExecution</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
              >unixExecution</rdf:first>
            <rdf:rest rdf:parseType="Resource">
              <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
              <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                >pythonExecution</rdf:first>
            </rdf:rest>
          </rdf:rest>
        </owl:oneOf>
      </owl:DataRange>
    </rdfs:range>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Execution mode. Currently supported: Java, C, Unix, Python.</rdfs:comment>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:about="#value">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >String value.</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="build">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Version of the build, e.g. 5222.</rdfs:comment>
  </owl:DatatypeProperty>
```



```

<owl:DatatypeProperty rdf:about="#label">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Datatype property "label" is used in a number of services.</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#os">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >ALL</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >WINNT51</rdf:first>
          <rdf:rest rdf:parseType="Resource">
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >LINUX</rdf:first>
          </rdf:rest>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Operating system needed for execution. ALL - if binary is in Java, WINNT51 for Windows
  applications and LINUX for linux binaries.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#architecture">
  <rdfs:domain rdf:resource="#ProcessingDevice"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Architecture, needed for execution of the binary.</rdfs:comment>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"

```

```
>ALL</rdf:first>
<rdf:rest rdf:parseType="Resource">
  <rdf:rest rdf:parseType="Resource">
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >PPC</rdf:first>
    <rdf:rest rdf:parseType="Resource">
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >ITANIUM</rdf:first>
    </rdf:rest>
  </rdf:rest>
</rdf:rest>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>INTEL</rdf:first>
</rdf:rest>
</owl:oneOf>
</owl:DataRange>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="codeVersion">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Version of the source code, e.g. 2.4.11.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#variable">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  ></rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="toValue">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Parameter sweep definition. E.g., in "from x=1 to A", toValue equals to A.</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="step">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
```

```

>Parameter sweep definition. E.g., in "from x=1 to A step B", step equals to
B.</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="uploadDate">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#date"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Date when the software (e.g. application or data) was grid-enabled.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="swVersion">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Version of the software, e.g. 2.2.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="id">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#unit">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:rest rdf:parseType="Resource">
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >GB</rdf:first>
          <rdf:rest rdf:parseType="Resource">
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >TB</rdf:first>
          </rdf:rest>
        </rdf:rest>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>

```

```
<rdfs:domain rdf:resource="#StorageDevice"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Needed disk space is measured in units: MB, GB and TB.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#remoteProtocol">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Two remote, data access protocols are supported: gsiftp and http.</rdfs:comment>
<rdfs:range>
<owl:DataRange>
<owl:oneOf rdf:parseType="Resource">
<rdf:rest rdf:parseType="Resource">
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>http</rdf:first>
<rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:rest>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>gsiftp</rdf:first>
</owl:oneOf>
</owl:DataRange>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="fromValue">
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Parameter sweep definition. E.g., in "from x=A to B", fromValue equals to
A.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="protocol">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Data access protocol.</rdfs:comment>
<rdfs:range>
<owl:DataRange>
<owl:oneOf rdf:parseType="Resource">
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>gsiftp</rdf:first>
```

```
<rdf:rest rdf:parseType="Resource">
  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >http</rdf:first>
</rdf:rest>
</owl:oneOf>
</owl:DataRange>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="vendor">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Software vendor.</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#listItem">
  <rdfs:domain rdf:resource="#ValueList"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Item in a list.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="applicationID">
  <rdfs:subPropertyOf rdf:resource="#id"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Globally unique application ID. It is set automatically.</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="jobID">
  <rdfs:subPropertyOf rdf:resource="#id"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="io">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Input/output data type.</rdfs:comment>
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:rest rdf:parseType="Resource">
```

```
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Directory</rdf:first>
<rdf:rest rdf:parseType="Resource">
  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Parameterfile</rdf:first>
</rdf:rest>
</rdf:rest>
<rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>Datafile</rdf:first>
</owl:oneOf>
</owl:DataRange>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#longDescription">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Long description of the algorithm.</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="submissionTime">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
  <rdfs:domain rdf:resource="#Provenance"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Provenance data.</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="resultsLocation">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Provenance data.</rdfs:comment>
  <rdfs:domain rdf:resource="#Provenance"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="requiredLibrary">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
```

```
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>From grid perspective, required library(-ies) is a remote file or directory.</rdfs:comment>
<rdfs:range rdf:resource="#RemoteFileDir"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="schedulerStatus">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Provenance"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Provenance data.</rdfs:comment>
</owl:FunctionalProperty>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.3.1, Build 430) http://protege.stanford.edu -->
```

Priloga B Prevod citata na začetni strani

Rupreht

Magister doctissime! Verjemite mi,
da je pred vami eden izmed vaših
najbolj predanih učencev.
Odkrito mi povejte, kaj je magija:
resnica ali zabloda,
je znanost ali ni?

Agripa

Resnična magija je znanost vseh znanosti,
pojasnitev vseh misterijev, ki se razodevajo magom
skozi veke
v vseh deželah in narodih.

*Dialog iz opere Ognjeni angel Sergeja Prokofjeva na podlagi
istoimenske novele Valerija Briusova.*