

RĪGAS TEHNISKĀ UNIVERSITĀTE

V. Žuravļovs

**Objektu individuālajās īpašībās sakņotas
decentralizētas datu vadības sistēmas
konceptijas izstrāde**

PROMOCIJAS DARBS

2012

RĪGAS TEHNISKĀ UNIVERSITĀTE
Datorzinātnes un informācijas tehnoloģijas fakultāte
Lietišķo datorsistēmu institūts

Vadims ŽURAVĻOVS

Datorsistēmu doktora programmas doktorants

**OBJEKTU INDIVIDUĀLAJĀS ĪPAŠĪBĀS SAKŅOTAS
DECENTRALIZĒTAS DATU VADĪBAS SISTĒMAS
KONCEPCIJAS IZSTRĀDE**

Promocijas darbs

Zinātniskā vadītāja
Dr. sc. ing., asociētā profesore
E. LATIŠEVA

Rīga 2012



Šis darbs izstrādāts ar Eiropas Sociālā fonda atbalstu projektā «Atbalsts RTU doktora studiju īstenošanai».

Anotācija

Parasti informācijas sistēmām, kas strādā ar fiziskiem objektiem, ir jārisina jautājumi par fizisko objektu individuālo īpašību glabāšanu un pārvaldību. Informācijas sistēmas, kuras koncentrējas uz objektu individuālajām īpašībām sauc uz objektu individuālajās īpašībās sakņotām sistēmām. Maziem un vidējiem uzņēmumiem, konkurējot ar lieliem uzņēmumiem un starptautiskām korporācijām, ir nepieciešamas datu vadības sistēmas, kas nodrošina tādas priekšrocības kā: pakāpeniskas ieviešanas iespējas un ieviešanas izmaksu sadale, autonoma darbība utt. Klasiskā datu vadības sistēmu teorija apgalvo, ka viens no efektīvākajiem datu glabāšanas un pārvaldības veidiem ir centralizētas datu vadības sistēmas. Šim datu vadības veidam ir savi riski, ierobežojumi un problēmas, kā rezultātā tam atbilstošās sistēmas nevar nodrošināt pilnvērtīgu nepieciešamo konkurētspēju. Papildu centralizētam datu vadības sistēmām eksistē decentralizētās datu vadības sistēmas, tomēr tām ir līdzīgi riski, ierobežojumi un problēmas. Viens no iespējamajiem veidiem, kā labot šo situāciju, ir izstrādāt tādu datu vadības sistēmas koncepciju, kas būtu piemērota tieši objektu individuālajās īpašībās sakņotu sistēmu vajadzībām.

Promocijas darbā ir izstrādāta objektu individuālajās īpašībās sakņotas decentralizētu datu vadības sistēmu koncepcija. Šī koncepcija ir veidota speciāli objektu individuālajās īpašībās sakņotām sistēmām, lai, salīdzinājumā ar klasiskajām datu vadības sistēmām, būtu iespējams izstrādāt informācijas sistēmas ar mazākiem riskiem, ierobežojumiem un problēmām. Darba sākumā ir izvirzīti kritēriji objektu individuālajās īpašībās sakņotām sistēmām. Kritēriji ir izvirzīti pamatojoties uz objektu individuālajās īpašībās sakņotu sistēmu vajadzībām un uz mūsdienu informācijas sistēmu attīstības pamattendencēm. Ir veikts klasiskās datu vadības sistēmas koncepcijas pētījums, kas ietver sevī novērtējuma atbilstību izvirzītajiem kritērijiem. Pamatojoties uz veikto pētījumu, ir secināts, ka neeksistē tāda klasiska datu vadības sistēmas koncepcija, kas pilnvērtīgi atbilst izvirzītajiem kritērijiem. Tādēļ tika izstrādāts teorētiskais pamatojums koncepcijai, kas ietver sevī darbības objektu tipus un objektu mijiedarbības scenārijus. Tika klasificētas objektu īpašības, definēti atribūti un noteikumu grupas. Tika veikta koncepcijas pārbaude atbilstoši iepriekš izvirzītajiem kritērijiem. Koncepcijas praktiskai realizācijai, datu glabāšanas un pārvaldības nolūkiem tiek piedāvāti datu apmaiņas formātu paplašinājumi. Dotais teorētiskais pamatojums ir izmantots programmatūras kompleksa izstrādāšanai, kas ietver sevī datu glabāšanas un pārvaldības formātu realizācijas (eksistējošo formātu paplašinājumi) un programmatūru, kas nodrošina uz koncepcijas bāzētu projektu izveidošanu un implementēšanu. Izmantojot izstrādāto

programmatūras kompleksu, ir parādītas dažādu veidu datu vadības uzdevumu praktiskās realizācijas.

Par promocijas darba galvenajiem rezultātiem ir nolasīti 14 referāti 12 starptautiskās konferencēs, kā arī tie ir atspoguļoti 11 zinātniskajos rakstos.

Darbs sastāv no ievada, 4 nodaļām un secinājumiem. Tajā ir 196 lappuses, 38 attēli un 14 tabulas pamatteksta, 264 nosaukumu literatūras saraksts un 7 pielikumi.

Annotation

Generally, information systems dealing with physical objects have to solve issues regarding storing and managing individual properties of physical objects. Information systems with focus on individual properties of objects are referred to as information systems based on individual object properties. Small and medium-sized companies, while competing with big enterprises and international corporations, need data management systems that can provide benefits such as gradual implementation options, and distribution of integration costs, autonomous operation, etc. The traditional data management system theory states that centralized data managing systems are among the most effective data storing and managing methods. This data managing method has its risks, limitations, and problems, thus, systems that are compatible with it cannot provide maximum competitiveness that is needed. In addition to centralized data management systems, there are also decentralized data management systems, but they have similar risks, limitations, and problems. This situation can be remedied by developing a data management system conception that corresponds to necessities of the systems based on object individual properties.

In this thesis, a conception is developed for decentralized data management systems based on object individual properties. This conception has been designed specifically for systems based on individual object properties in order to – unlike the traditional data management systems – provide possibilities to develop information systems with lower risks, limitations, and problems. In the beginning, criteria are set for systems based on individual object properties. The criteria are set, based on necessities of systems based on object individual properties and on basic tendencies for the development of state-of-the-art information systems. An examination of traditional data management system conception was performed that included rating conformity to the set criteria. Based on the performed study, it is concluded that none of the traditional data management system conceptions fully meets the set criteria. Thus, theoretical basis was developed for a conception that includes types of operation objects and object interaction scenarios. Object properties were categorized and groups of attributes and rules were defined. Conception examination was performed according to previously set criteria. In order to practically implement this conception, data exchange format extensions were proposed for data storing and managing purposes. The theoretical basis is used for developing software complex that includes implementations of data storing and managing formats (the existing format extensions) and software which provides creation and implementation of projects based on this conception. Using the developed software complex, practical implementations of various data managing tasks are shown.

The main results of this thesis are reported in 14 reports in 12 international conferences and published in 11 publications.

The thesis includes introduction, 4 chapters and conclusions. It contains 196 pages, 38 figures and 14 tables in the main text, 264 titles large bibliography and 7 appendixes.

Pateicība darba veicinātājiem

Darba autors izsaka vislielāko pateicību savam zinātniskajam vadītājam RTU Informācijas programmēšanas katedras profesorei Eleonorai Latiševai par desmit gadu sadarbību, pacietību, ticību, audzināšanu, vērtīgiem komentāriem un būtiskiem ierosinājumiem pētījumu veikšanas laikā.

Autors vēlētos īpaši pateikties RTU Informācijas programmēšanas katedras lektoram dr. Gintam Jēkabsonam par atbalstu, sniegtam idejām un ieteikumiem.

Darba autors izsaka pateicību arī Eiropas Sociālajam fondam un Rīgas Tehniskajai universitātei par sniegto materiālo atbalstu promocijas darba izstrādes laikā.

Liels paldies manai sievai Jeļenai, bērniem Aleksandrai un Rodionam par viņus atbalstu un pacietību. Paldies vecākiem un draugiem, kas atbalstīja ceļā uz doktora zinātniskā grāda iegūšanu.

Saturs

Ievads.....	16
1. Datu vadības sistēmu koncepciju analīze no datu glabāšanas un pārvaldības viedokļa.....	24
1.1. Objektu individuālajās īpašībās sakņotas sistēmas (OIĪSS).....	24
1.1.1. Informācijas sistēmas uzbūves tendences	24
1.1.2. OIĪSS apraksts	27
1.1.3. Kritēriju definēšana objektu individuālajās īpašībās sakņotām sistēmām.....	27
1.1.4. OIĪSS piemēri.....	29
1.2. Datu vadības sistēmas koncepcijas.....	35
1.2.1. Monolītas datu vadības sistēmas koncepcijas	36
1.2.2. Decentralizētas datu vadības sistēmas koncepcijas	42
1.3. DVS atbilstība objektu individuālajās īpašībās sakņotu sistēmu kritērijiem.....	47
1.3.1. Monolītās datu vadības sistēmas	47
1.3.2. Decentralizētas datu vadības sistēmas.....	49
1.4. Kopsavilkums un secinājumi.....	50
2. OIĪSS izmantoto tehnoloģiju izpēte	53
2.1. Piemēru apskats tehnoloģiju identificēšanai	53
2.2. Tehnoloģijas	55
2.2.1. Tehnoloģiju salīdzināšanas parametri	55
2.2.2. Tehnoloģiju apskats.....	58
2.2.3. Tehnoloģiju parametru salīdzinājums	72
2.3. Datu apmaiņas formāti.....	74
2.4. Kopsavilkums un secinājumi.....	77
3. Koncepcijas apraksts	80
3.1. Koncepcijas vispārīgais apraksts.....	80
3.2. Dokumenta iekšējā struktūra	81
3.3. Resursu mijiedarbības scenāriji	83
3.4. Atribūtu un noteikumu grupas.....	87
3.5. Koncepcijas pārbaude definēto kritēriju atbilstībai.....	89
3.6. Kopsavilkums un secinājumi.....	90
4. DARSIR koncepcijas ieviešana.....	93
4.1. Datu apmaiņas un glabāšanas formātu izvēle.....	93
4.2. Atribūtu un noteikumu grupas ieviešana.....	94
4.3. Tehnoloģijas izvēle.....	107
4.3.1. Ietekmes kritēriji.....	107
4.3.2. Tehnoloģiju izvēles aprēķinu metode.....	114
4.3.3. Ietekmes koeficientu vienvērtīgās nozīmes aprēķins	117
4.4. Radio Frekvences Identifikācijas tehnoloģijas apskats.....	118
4.4.1. RFID tehnoloģijas standarti.....	118
4.4.2. EPCglobal rekomendētais risinājums.....	119

4.4.3.	Risinājums ražošanai	122
4.5.	Drošības jautājumu apskats	123
4.5.1.	RFID tehnoloģijai specifiskā uzbrukumu taksonomija	123
4.5.2.	Informācijas drošības problēmu piemērotība DARSIR koncepcijai	126
4.5.3.	RFID kontrole informācijas drošības risku samazināšanai	127
4.6.	DARSIR koncepcijas praktiskie pētījumi	130
4.7.	Kopsavilkums un secinājumi	131
	Galvenie rezultāti un secinājumi	134
	Literatūra	138
	Pielikumi.....	158
1.	pielikums. DARSIR koncepcijas datu apmaiņas tehniskā realizācija	159
2.	pielikums. Atribūtu grupas realizācija.....	161
3.	pielikums. Resursu noteikumu tehniskā realizācija.....	171
4.	pielikums. Vizuālā interaktīvā simulācija (VIS)	176
5.	pielikums. Koncepcijas pārbaude ar tipisku uzdevumu	180
6.	pielikums. Koncepcijas pārbaude ar netipisku uzdevumu.....	186
7.	pielikums. Jaunas funkcijas ieviešana	196

Pamattteksta attēlu saraksts

1.1. attēls	MDVS, DDVS un OIĪSS raksturojumu pārklājums	26. lpp
1.2. attēls	Daudz datubāzu sistēmas taksonomija (adaptēta no [Sim1995])	42. lpp
1.3. attēls	Aģents mijiedarbojas ar vidi (adaptēta no [Jen1998])	43. lpp
2.1. attēls	Trīs kategoriju svītrkodu piemēri	58. lpp
2.2. attēls	Lineāru svītrkodu piemēri	58. lpp
3.1. attēls	Viena ADR mijiedarbības scenārijs	83. lpp
3.2. attēls	Vienlaicīga divu ADR mijiedarbības scenārijs	84. lpp
3.3. attēls	Dažādu ADR secīgas mijiedarbības scenārijs	85. lpp
4.1. attēls	UPC-A standarta struktūra	93. lpp
4.2. attēls	GS1 US: UPC standarta shēmas	94. lpp
4.3. attēls	EPC Type I struktūra (adaptēts no [Bro2001a])	94. lpp
4.4. attēls	EPCglobal standarts (adaptēts no [EPC2010a])	119. lpp
4.5. attēls	EPCglobal starpprogrammatūras arhitektūra (adaptēts no [Bau2005])	120. lpp
4.6. attēls	“Lightweight RFID bus framework” starpprogrammatūras arhitektūra (adaptēts no [Jun2004])	121. lpp
P1.1. attēls	“Apple” resursa piemērs	158. lpp
P2.1. attēls	RI prezentācijas piemērs	160. lpp
P2.2. attēls	Resursa individuālas īpašības prezentācijas piemērs	161. lpp
P2.3. attēls	Vitamīnu kompleksa „Vitabiotics Feroglobin B12” marķējums, kur atspoguļoti dati par uzturvērtībām	161. lpp
P2.4. attēls	Vitamīnu kompleksa „Vitabiotics Feroglobin B12” marķējuma prezentācijas piemērs	162. lpp
P2.5. attēls	Saistības atribūta piemērs	163. lpp
P2.6. attēls	Laika atribūta piemērs	163. lpp
P2.7. attēls	Vēstures atribūta piemērs	164. lpp
P2.8. attēls	Pozicionēšanas GPS koordinātu piemērs	165. lpp
P2.9. attēls	Pozicionēšanas iekštelpu koordinātu piemērs	165. lpp
P2.10. attēls	Piekļuves tiesību izmantošanas piemērs	166. lpp
P2.11. attēls	Klašu identifikatora izmantošanas piemērs	167. lpp
P2.12. attēls	Piemērs resursa klasificēšanai ar klašu datu pievienošanu	167. lpp
P2.13. attēls	Konfigurācijas atribūtu piemērs	168. lpp
P2.14. attēls	Prezentācijas atribūtu piemērs	169. lpp

P3.1. attēls	Noteikuma JSONR piemērs	171. lpp
P3.2. attēls	Noteikuma RMPL piemērs	171. lpp
P3.3. attēls	Notikuma realizācijas piemērs	172. lpp
P3.4. attēls	Nosacījuma realizācijas piemērs	173. lpp
P3.5. attēls	Ciklu realizācijas piemērs	173. lpp
P5.1. attēls	Viedās mazgāšanas mašīnas risinājums	183. lpp
P6.1. attēls	API darbības shēma no [Fre2011a], starp darbinieku un darba dēvēju	186. lpp
P6.2. attēls	Apstiprinātā likme un vidējās izmaksas atkarībā no mēģinājumu skaita (Simulācija I)	193. lpp
P6.3. attēls	Apstiprinātā likme un vidējās izmaksas atkarībā no mēģinājumu skaita (Simulācija II)	194. lpp

Pamatteksta tabulu saraksts

1. tabula	Saīsinājumu saraksts	13. lpp
1.1. tabula	IS attīstības pamattendences (adaptēts no [Sim1995])	24. lpp
1.2. tabula	Izvirzītie kritēriji objektu individuālajās īpašībās sakņotām sistēmām	27. lpp
1.3. tabula	MDVS kritēriju atbilstības salīdzinājuma tabula	47. lpp
1.4. tabula	DDVS kritēriju atbilstību salīdzinājuma tabula	49. lpp
2.1. tabula	Tehnoloģiju grupas pēc izmantošanas mērķiem	55. lpp
2.2. tabula	Tehnoloģiju salīdzināšanas parametri	56. lpp
2.3. tabula	Tehnoloģiju parametri salīdzinājuma tabula	72. lpp
3.1. tabula	DARSIR koncepcijas atribūtu un noteikumu grupas	87. lpp
4.1. tabula	Septiņas fundamentālās bāzes mērvienības (adaptēts no [Tay2008])	95. lpp
4.2. tabula	Papildu četras bāzes bez dimensiju mērvienības (adaptēts no [Bro2001b])	96. lpp
4.3. tabula	Fizisko parametru un ārējā vides tehnoloģiju IK	115. lpp
4.4. tabula	Tehnoloģiju izmaksas IK	115. lpp
4.5. tabula	Datu drošības un konfidencialitātes IK	116. lpp

Saīsinājumu saraksts

Tiek pieņemts, ka vispārīgie termini, kas tiek izmantoti darbā, ir definēti kā latviešu nosaukuma saīsinājumi. Izņēmums ir termini, kuriem saīsinājums angļu valodā tiek parasti izmantots kā nosaukums, tāpēc saīsinājums dots no angļu valodas nosaukuma.

Saīsinājumi ir apvienoti 1. tabulā:

1. tabula

Saīsinājumu saraksts

Saīsinājums	Pilnais nosaukums latviešu valodā	Pilnais nosaukums angļu valodā
ADR	aktīvais dinamiskais resurss	active dynamic resource
API	aplikācijas programmēšanas interfeiss	application programming interface
AR	aktīvais resurss	active resource
ASR	aktīvais statiskais resurss	active static resource
CRC	cikliska atlaišanu pārbaude	cyclic redundancy check
CVP	centralizētais vadības punkts	centralized control point
DARSIR*	saglabāti resursā dati un noteikumi	data and rules saved in resource
DB	datubāze	database
DBVS	datubāzes vadības sistēma	database management system
DDB	dalītā datubāze	distributed database
DDBVS	dalīta datubāzes vadības sistēma	distributed database management system
DDVS	decentralizēta datu vadības sistēma	decentralized data management system
DM	datu modelis	data model
DVS	datu vadības sistēma	data management system
EPK	elektroniskā produkta kods	electronic product code
GD	galvenā datne	master file
GPRS	vispārējais pakešu radio pakalpojums	general packet radio service
GPS	globālā vietnoteices sistēma	global positioning system
GSM	globālo mobilo sakaru sistēma	global system for mobile communications
HTML	hiperteksta iezīmēšanas valoda	hypertext markup language
HTTP	hiperteksta pārraides protokols	hypertext transfer protocol

IDM	izplātās datnes modelis	flat file model
IDS	integrētais datu krājums	integrated data store
IK	ietekmes kritērijs	impact criteria
IMS	informācijas vadības sistēma	information management system
IS	informācijas sistēma	information system
JSON	<i>JavaScript</i> objektu notācija	<i>JavaScript</i> object notation
JSONR	<i>JavaScript</i> objektu notācija resursam	<i>JavaScript</i> object notation for resource
MAC	mediju piekļuves kods	media access code
MAS	daudzaģentu sistēmās	multi agent system
MDVS	monolīta datu vadības sistēma	monolithic database management systems
NFC	tuva darbības lauka komunikācija	near field communication
NPII	nepārtrauktība pareizība izolēšana izturība	atomicity correctness isolation durability
OIĪSS	objektu individuālajās īpašībās sakņotas sistēma	data management system based on individual object properties
OOM	objektu orientētais modelis	object oriented model
PB	protokolu buferis	protocol buffers
PML	fizikālā iezīmēšanas valoda	physical markup language
PR	pasīvais resurss	passive resource
RFID	radio viļņu identifikācija	radio frequency identification
RI	resursa identifikators	resource identification
RII	resursa individuālās īpašības	resource individual properties
RPML	resursu fizikālā iezīmēšanas valoda	resource physical mark-up language
SQL	strukturēta vaicājumvaloda	structured query language
TD	transakcijas datne	transaction file
URL	informācijas resursa unificētais lokators	uniform resource locator
UTC	universālais koordinētais laiks	coordinated universal time
WSN	bezvadu sensoru tīkls	wireless sensor network
WWW	globālais tīmeklis	world wide web

*Promocijas darba izstrādātajai koncepcijai ir doti saīsinājumi angļu valodā, jo tieši šajā variantā tas ierakstīts starptautiskajos ziņojumos.

Ievads

Mūsdienīgās informācijas sistēmās (ang. *Information System, IS*) *fiziskais objekts* ir viena no pamatsastāvdaļām tādās nozarēs kā loģistika [Suh2005], apkopes servisi [Lai2011], viedās ēkas [Mon2007], reālās vides monitorings [Mon2007] utt. Klasiskā pieeja IS darbā ar fiziskiem objektiem ir datu glabāšanas un pārvaldības centralizācija. Klasiskā teorija apgalvo [Lah2005, Hei2005, Bau2005, Sar2010], ka viens no efektīvākajiem datu glabāšanas un pārvaldības veidiem ir monolītas datu vadības sistēmas. Šim datu vadības veidam ir savi riski, ierobežojumi un problēmas [Dat2004]:

- *infrastrukturā nodrošināšanas problēma*, nepieciešams organizēt datu nosūtīšanu no jebkuras IS vietas uz centrālo datu glabātuvi;
- *datu privātuma un drošības riski* – centralizēta datu glabāšana apvieno visus datus vienkopus, kas pastiprina nesankcionētas piekļuves un datu bojāšanas riskus;
- *reālā laika savienojuma ierobežojums* – monolīta datu vadības sistēma datu apstrādes laikā prasa savienošanos starp centralizētu glabātuvi un lietojumprogrammu. Savienojuma problēmas noved pie problēmām lietojumprogrammu darbībā.

Monolītas datu vadības sistēmas ir veidotas sākot no XX gadsimta 70. gadu beigām [Rot1977, Ber1993, And1998, Ber1999]. Vēlāk sāk attīstīties pieeja, kas balstās uz datu glabāšanas un pārvaldības decentralizāciju – *decentralizētas datu vadības sistēmas*. Piemēram, *daudzaģentu sistēmas* [Gra1996, Woo2001] un *sensoru tīkli* [Soh2007] sāka strauji izplatīties tikai XX gadsimta 90. gados.

Decentralizētām datu vadības sistēmām bieži ir tie paši riski, ierobežojumi un problēmas, kas ir tipiski monolītām datu vadības sistēmām. Piemēram, *sensoru tīkliem* ir aktuāla *infrastrukturā nodrošināšanas problēma* un *reālā laika savienojuma ierobežojums* [Dar2010].

Izstrādājot IS, jānodrošina lēmumu pieņemšanas laikā visi dati un darbības, kas ietekmē lēmumu pieņemšanu. Viena no mūsdienīgajām IS attīstības pamattendencēm ir reālās pasaules darbības principu izmantošana [Sim1995]. Lai nodrošinātu lēmumu pieņemšanu uz vietas ir iespējams saņemt informāciju arī no ārējās vides nevis no datubāzes. Šī ideja ir jau daļēji pētīta darbos [Jen1998, Feb1999], bet, ja informāciju par objektu jau glabā pašā objektā, tas nodrošina gan ātrāku apstrādi, gan minimizē liekās – ar datu iegūšanu un datu apkalpošanu saistītās – darbības, ņemot vērā arī citus procesus sistēmā.

Promocijas darbā ir izstrādāta pieeja risku, ierobežojumu un problēmu novēršanai *objektu individuālajās īpašībās sakņotām sistēmām (OIĪSS)* – tā ir datu vadības sistēmas

konceptija, kas paredzēta specifiski tieši OIĪSS gadījumam. Ir izvirzīti OIĪSS kritēriji, kas samazina tipiskus riskus, ierobežojumus un problēmas, un izveidota koncepcija, kas atbilst šiem kritērijiem. Tāda veida koncepciju iespējams izveidot, ja glabā objektu individuālās īpašības (atribūtus un uzvedības noteikumus) kopā ar objektu. Eksistē tehnoloģijas, kas jau šodien nodrošina fizisko objektu informācijas glabāšanu pašā objektā, tas ir *radio viļņu identifikācija* (ang. *Radio Frequency Identification, RFID*) [Hei2005, Bau2005, Hun2007], *tuva darbības lauka komunikācija* (ang. *Near Field Communication, NFC*) [Ker2011], *svītrkodu* [Lah2005, Ste2005] utt.

Datu apjoms, ko ir iespējams glabāt fiziskos objektos, pēdējo desmit gadu laikā ir palielinājies no dažiem baitiem līdz pat megabaitiem un turpina augt [Lah2005]. No tehnoloģisko iespēju viedokļa jau ir pietiekams atmiņas apjoms ne tikai lai glabātu informāciju decentralizēti pašos fiziskajos objektos, bet arī lai glabātu aprakstus, kā pārvaldīt šos objektus.

Promocijas darbā ir izstrādātas objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcijas teorētiskās pamattēzes, kas uz datu vadības sistēmas teorijas [Dat2004] bāzes, ļauj nodrošināt šādas priekšrocības:

- koncepcija ir bāzēta uz objektorientētu pieeju, kas nodrošina fizisko objektu daudzveidības un unikalitātes atbalstīšanu;
- visa informācija par fizisko objektu tiek glabāta pašā objektā, tātad:
 - lēmumu pieņemšanas brīdī visi nepieciešamie dati ir pieejami uz vietas, kas ļauj atteikties no centralizētas datu glabāšanas un pārvaldības infrastruktūras nodrošinājuma;
 - paaugstinās datu privātuma līmenis, datu īpašnieks pats var kontrolēt datu pieejamību, jo fizisko objektu datu piekļuvei jānodrošina objektu atrašana datu apstrādes vietā.

Tēmas aktualitāte

Pasaules informācijas un komunikāciju tehnoloģijas tirgus strauji paplašinās. Tirgū ienākot jauniem dalībniekiem, to savstarpējā konkurence arvien pieaug [KOM2004a]. Mūsdienās ekonomiskās reālijas liek uzņēmumiem aktīvāk izmantot informācijas un komunikāciju tehnoloģijas. Pieaugošā konkurence piespiež kompānijas meklēt jaunus paņēmienus, lai paaugstinātu biznesa efektivitāti, samazinātu produkta pašizmaksu, šādā veidā uzlabojot savu resursu izmantošanu.

Mūsdienās iegultās sistēmas aizņem 98% no visām datoru sistēmām [Ten2000]. Komisijas paziņojumā padomei, Eiropas parlamentam, Eiropas ekonomikas un sociālo lietu

komitejai un reģionu komitejai [KOM2007] ir norādīts, ka Septītajā ietvarprogrammā (2007–2013) informācijas un komunikāciju tehnoloģijas 2007. – 2008. gada darba programmā norādīti četri problēmjautājumi: veselības aprūpe, viedo transportlīdzekļu un mobilitātes sistēmas, mikrosistēmas un nanosistēmas, organiskā elektronika un nākotnes tīkli. Tas liek koncentrēties uz risinājumiem, kas izmanto iegultās un mobilitātes sistēmas iespējas.

Informācijas un komunikāciju tehnoloģiju lietošana uzņēmumos ir atzīta par vienu no veiksmes faktoriem Eiropas konkurētspējas uzlabošanai. Taču jaunu biznesa procesu un jaunu biznesa modeļu efektīva ieviešana, lai izmantotu informācijas un komunikāciju tehnoloģijas potenciālu, joprojām ir izaicinājums, jo īpaši miljoniem Eiropas mazo un vidējo uzņēmumu. [KOM2004b] Maziem un vidējiem uzņēmumiem, konkurējot ar lieliem uzņēmumiem un starptautiskām korporācijām, ir nepieciešamas datu vadības sistēmas, kas nodrošina tādas priekšrocības kā: *nelielu risinājumu implementēšana, pakāpeniska ieviešanas iespējas un ieviešanas izmaksu sadale, autonoma darbība* utt. Šīs priekšrocības ir iespējams nodrošināt, izmantojot mūsdienīgas iegultās sistēmas kopā ar decentralizētu datu vadības sistēmu pieeju.

Mūsdienās Eiropas Savienībā, tajā skaitā Latvijā, ir izmantoti aparatūras un programmatūras risinājumi, kuru pamatā jau šodien ir iespējams pētīt un izveidot jaunas datu vadības sistēmas koncepcijas, kurās implementēt jaunas pakāpes IS. Viena no tāda tipa tehnoloģijām ir *radio frekvenču identifikācija*. [EPC2006] ir parādīts, ka šī tehnoloģija tiešām atvērš vārtus uz jaunu informācijas sabiedrības attīstības posmu, bieži vien to dēvē par „lietu internetu”, kurā ir savienoti ne vien datori un sakaru galiekārtas, bet arī ikviens priekšmets, kas ikdienā atrodas mums apkārt – apģērbs, patēriņa preces u.c. Tieši šis aspekts bija iemesls, kādēļ 2006. decembra Eiropadome lūdza Eiropas Komisiju 2008. gada pavasara Padomē pārskatīt nākamās interneta paaudzes un tīklu problēmas.

Eiropas Savienībā nozīmīgs princips ir personas datu aizsardzība. Līguma par Eiropas Savienību 6. pantā teikts, ka Eiropas Savienība ir dibināta uz brīvības, demokrātijas, cilvēktiesību un pamatbrīvību ievērošanas principiem; 30. pantā noteikts, ka personas datu savākšanai, uzglabāšanai, apstrādei, analīzei un informācijas apmaiņai sadarbībā ar policiju jānotiek saskaņā ar noteikumiem par personas datu aizsardzību [KOM2005]. Tas liek pievērst īpašu uzmanību datu konfidencialitātes, integritātes un pieejamības nodrošināšanas jautājumiem.

Promocijas darba mērķis

Promocijas darba mērķis ir izstrādāt objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepciju un pārbaudīt tās praktiskās realizācijas iespējas.

Darba uzdevumi

Promocijas darba mērķa sasniegšanai tika risināti šādi darba uzdevumi:

- ņemot vērā mūsdienu informācijas sistēmu attīstības pamattendences un *objektu individuālajās īpašībās sakņotu sistēmu (OIĪSS)* specifiskās vajadzības, jāidentificē kritēriji OIĪSS un jāveic vispazīstamāko datu vadības sistēmu koncepciju analīze atbilstoši izvirzītajiem kritērijiem;
- noteikt tehnoloģijas, kādas ir iespējams izmantot OIĪSS, izvirzīt tehnoloģiju salīdzināšanas parametrus un veikt atbilstošu tehnoloģiju analīzi;
- jāveic OIĪSS koncepcijas izstrāde un jāpārbauda koncepcijas atbilstība OIĪSS izvirzītajiem kritērijiem;
- noteikt kritērijus, kas ietekmē OIĪSS piemērotas tehnoloģijas izvēli, piedāvāt ietekmes kritēriju salīdzināšanas metodi un uz tā bāzes identificēt vispiemērotākās tehnoloģijas izstrādātajai koncepcijai;
- jāveic izstrādātās koncepcijas datu privātuma un drošības jautājumu pētīšana;
- jāveic izstrādātās koncepcijas programmatūras elementu realizācija un jāpārbauda koncepcijas praktiskās realizācijas iespējas.

Pētīšanas metodes

Promocijas darba teorētiskie pētījumi pamatojas uz pieejamās zinātniskās literatūras avotu analīzi; izvirzīto uzdevumu risināšanai tiek izmantota sistēmu teorija, algebras un kopu teorijas elementi, reālo sistēmu funkcionēšanas simulēšanas paņēmieni un programmatūras inženierijas metodes.

Darba zinātniskais jaunieguvums ir šāds:

- ir identificēti 8 kritēriji, kuriem jāatbilst *objektu individuālajās īpašībās sakņotai sistēmai (OIĪSS)*;
- ir izpētītas 9 monolītas un 5 decentralizētas klasiskās datu vadības sistēmas un novērtēta to atbilstība izvirzītajiem OIĪSS kritērijiem. Ir noskaidrots, ka neviena no apskatītajām datu vadības sistēmām neatbilst visiem izvirzītajiem OIĪSS kritērijiem, radot nepieciešamību pēc objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcijas izstrādes;
- ir identificētas 16 tehnoloģijas un tehnoloģiju modifikācijas, kas parasti tiek izmantotas OIĪSS risinājumos, un piedāvātas trīs tehnoloģiju grupas, kas klasificē tehnoloģijas pēc OIĪSS izmantošanas nolūkiem. Ir izvirzīti 8 tehnoloģiju salīdzinājuma parametri, pamatojoties uz dažādu tehnoloģiju kopējiem parametriem, kuri ietekmē OIĪSS risinājuma tehnoloģijas izvēli;

- ir izstrādāta objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcija, kas atbilst izvirzītajiem OIĪSS kritērijiem; ir klasificēti piedāvātās koncepcijas 3 objektu pamattipi, 3 objektu mijiedarbības scenāriji un 10 objektu īpašību tipi. Piedāvātajai koncepcijai ir šādas priekšrocības: lēmumu pieņemšana ir realizēta uz vietas fiziskajā objektā; tiek atbalstīta fizisko objektu daudzveidība un unikalitāte; ir paaugstināts datu privātuma līmenis; ir izdarīta atteikšanās no centralizētas datu glabāšanas un pārvaldības infrastruktūras nodrošinājuma.
- ir identificētas trīs galvenās tehnoloģiju ietekmes kritēriju grupas un klasificēti 20 ietekmes kritēriji. Ietekmes kritēriju pamatā ir izveidota *IKKATN* (saīsinājums no *Ietekmes Kritēriju Kombinācijas Aprēķins Tehnoloģiju Novērtējumam*) metode, lai palīdzētu izvēlēties konkrētam OIĪSS risinājumam piemērotāko tehnoloģiju;
- ir izpētīts datu aizsardzības nodrošināšanas jautājums OIĪSS koncepcijas pielietošanas kontekstā un izvirzītas rekomendācijas informācijas drošības risku samazināšanai.

Pētījumu praktiskā nozīmība

Darba praktiskā nozīmība ir programmatūras elementu izstrāde, kurus var pielietot OIĪSS risinājumos. *RPML* (*paplašināmās iezīmēšanas valodas (XML) paplašinājums*) un *JSONR* (*JavaScript objektu notācijas (JSON) paplašinājums*) datu glabāšanas un pārvaldības formātus iespējams izmantot programmatūrā, kas paredzēta darbam ar fiziskiem objektiem (piemēram, loģistikā, ražošanā utt.). Programmatūru, kas izveidota un aprakstīta promocijas darbā var izmantot OIĪSS risinājumu uzbūvei, implementēšanai un modificēšanai. Pētījumu rezultāts galvenokārt piemērots maziem un vidējiem uzņēmumiem, kuriem nepieciešamas tādas konkurences priekšrocības kā nelielu risinājumu implementēšana, pakāpeniskas ieviešanas iespējas un ieviešanas izmaksu sadale, autonoma darbība utt.

Darba aprobācija

Darba aprobācija notikusi, iepazīstinot ar pētījuma rezultātiem zinātnieku auditoriju, tika nolasīti 14 referāti (no tiem 7 ārvalstīs, t.i., Ķīnā, Spānijā, Bulgārijā un Lietuvā, un 7 Latvijā) 12 starptautiskās zinātniskās konferencēs:

1. Informatics in the Scientific Knowledge 2012, ISK 2012, Varna, Bulgārija, 2012. gadā. 27. – 29. jūnijā.
2. Agents and Multiagents Systems for Enterprise Integration 2012, ZOCO 2012, Salamanca, Spānija, 2012. gada 29. martā.
3. International Conference on Computer and Management 2012, CAMAN 2012, Vuhan, Ķīna, 2012. gada 9. – 11. martā.

4. RTU 52. Starptautiskā zinātniskā konference, Rīga, Latvija, 2011. gada 13. oktobrī.
5. 6th Electrical and control technologies, Kauņa, Lietuva, 2011. gada 5. – 6. maijā.
6. RTU 51. Starptautiskā zinātniskā konference, Rīga, Latvija, 2010. gada 12. – 16. oktobrī.
7. Applied Information and Communication technologies. The 4-th International Scientific Conference, Jelgava, Latvija, 2010. gada 22. – 23. aprīlī.
8. RTU 49. Starptautiskā zinātniskā konference, Rīga, Latvija, 2008. gada 12. – 14. oktobrī.
9. Informatics in the Scientific Knowledge 2008, ISK 2008, Varna, Bulgārija, 2008. gadā. 26. – 28. jūnijā.
10. Modeling of business, industrial and transport systems 2008, MBIT 2008, Riga, Latvia, 2008. gada 7. – 10. maijā.
11. RTU 48. Starptautiskā zinātniskā konference, Rīga, Latvija, 2007. gada 11. – 13. oktobrī.
12. RTU 47. Starptautiskā zinātniskā konference, Rīga, Latvija, 2006. gada 12. – 14. oktobrī.

Publikācijas

Pētījumu rezultāti ir atspoguļoti 11 publikācijās, kuras tapušas gan strādājot promocijas darba autoram patstāvīgi, gan sadarbībā ar līdzautoriem:

1. Zuravlyov V., Latisheva E., Lavendels J. Concept of Data Management Systems Based on Individual Object Properties // Proceedings of papers 'Informatics in the scientific knowledge 2012', Bulgaria, Varna, 27–29 June, 2012, pp. 269–279.
2. Zuravlyov V., Latisheva E. Research of Data Storage and Management Technology Applied to DARSIR Concept // Proceedings of papers 'Informatics in the scientific knowledge 2012', Bulgaria, Varna, 27–29 June, 2012, pp. 280–294.
3. Zuravlyov V., Matrosov A., Rutko D., Behavior Pattern Simulation of Freelance Marketplace, Workshop on Agents and Multi-agent systems for Enterprise Integration, ZOCO 2012, Salamanca, Espana, March, 2012.
4. Zuravlyov V., Matrosov A., Rutko D., Freelance resource management system optimization, The 2nd International Conference on Computer and Management, CAMAN 2012, Wuhan, China, March, 2012.
5. Zuravlyov V., Matrosov A. Multi-agent system built using RFID technology // Proceedings of the 6th international conference on electrical and control technologies, ISSN 1822-5934, Lithuania, Kaunas, 5–6 May, 2011. – pp. 15–20.

6. Zuravlyov V., Kryukov D., Kairish V. Security problems related to RFID-based concept usage and methods to counter them // Proceedings of the 6th international conference on electrical and control technologies, ISSN 1822-5934, Lithuania, Kaunas, 5–6 May, 2011. pp. 76–81.
7. Zuravlyov V., Latisheva E., Karish V. The Theoretical Basis and Practical Usage of the Simulation for Creating IT solutions for Data Management Systems // Proceedings of the international conference on Applied Computer Systems, ISSN 1407-7493, Latvia, Riga, 2011., pp. 138–144.
8. Zuravlyov V., Latisheva E. Various Aspects of RFID Technology and Their Use in Monitoring and Management of Traffic // Scientific Journal of RTU. 5. series., Datorzinātne. – 34. vol., Latvia, Riga, 2008, pp. 238–245.
9. Zuravlyov V. Various Aspects of 'Smart Laundry' Task Implementation, Using 'RFID' Technology // Proceedings of papers 'Informatics in the scientific knowledge 2008', Varna, Bulgaria, 26–28 June, 2008, pp. 150–156.
10. Zuravlyov V. Various Aspects of Simulation's Usage for Creation of Real Solutions with „DARSIR” Conception as Example // Proceedings of Conference 'Modeling of business, industrial and transport systems 2008; MBITS 2008', ISBN 978-9984-818-04-7, Latvia, Riga, 7–10 May, 2008, pp. 132–135.
11. Zuravlyov V. Main Principles of a New Concept of Designing Data Management Systems // Scientific Journal of RTU. 5. series., Datorzinātne. – 30. vol., Latvia, Riga, 2007, pp. 38–46.

Darba autora raksts „*Behavior Pattern Simulation of Freelance Marketplace*”, kuru viņš rakstīja kopā ar Dmitriju Rutko un Antonu Matrosovu, ir ieguvis balvu „*PAAMS '12 Award of Scientific Excellence*” starptautiskajā konferencē „*10th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 12)*“, Salamankā, Spānijā.

Darba struktūra

Promocijas darbs sastāv no ievada, 4 nodaļām, secinājumiem, bibliogrāfijas un 7 pielikumiem.

Ievadā ir pamatota veikto pētījumu aktualitāte, formulēts darba mērķis un uzdevumi, uzskaitītas promocijas darba izstrādē lietotās zinātniskās metodes, aprakstīta pētījumu zinātniskā novitāte un iegūto rezultātu praktiskā nozīmība, kā arī ir raksturota darba aprobācija.

Darba 1. nodaļā, ir izvirzīti kritēriji objektu individuālajās īpašībās sakņotām sistēmām un izpētītas klasiskās monolītās un decentralizētās datu vadības sistēmas koncepcijas, lai

noskaidrotu to atbilstību šiem kritērijiem. Ir identificēti vispazīstamāko datu vadības sistēmu koncepciju ierobežojumi un priekšrocības, kas tālāk promocijas darbā tiek ņemti vērā objektu individuālajās īpašībās sakņotas datu vadības sistēmas koncepcijas uzbūvē.

Darba 2. nodaļā ir izvirzīti tehnoloģiju salīdzināšanas parametri ar mērķi veikt objektu individuālajās īpašībās sakņotas sistēmas izmantoto tehnoloģiju analīzi un piedāvāt parametru salīdzināšanas bāzi. Nodaļā iegūtie rezultāti ir izmantoti objektu individuālajās īpašībās sakņotām informācijas sistēmām nepieciešamo tehnoloģiju izvēles metožu pielietošanai.

Darba 3. nodaļā ir aprakstīta objektu individuālajās īpašībās sakņotas datu vadības sistēmas koncepcija. Klasificēti vispārīgā līmenī koncepcijas darbības objektu tipi, objektu mijiedarbības scenāriji, objektu īpašības un noteikumu grupas, kā arī veikta koncepcijas pārbaude 1. nodaļā izvirzītajiem individuālajās īpašībās sakņotas sistēmas kritērijiem.

Darba 4. nodaļā ir veikti detalizēti objektu atribūtu un noteikumu grupu pētījumi, ar mērķi sagatavot teorētisko bāzi to realizēšanai programmatūrā. Ir izvirzīti svarīgākie koncepcijas tehnoloģiju izvēles kritēriji, balstoties uz kuriem ir izveidota tehnoloģiju izvēles metode. Ar šīs metodes palīdzību ir identificētas vispiemērotākās tehnoloģijas, pamatojoties uz IS būtiskākajiem darba apstākļiem. Tehnoloģija, kura ieguva vislabāko rezultātu pēc metodes pielietošanas tiek izmantota drošības jautājumu detalizētai analīzei.

Promocijas darba rezultāti un secinājumi ir apkopoti darba noslēguma nodaļā.

Koncepcijas praktiskie pētījumi ir atspoguļoti pielikumos: DARSIR koncepcijas datu apmaiņas tehniskā realizācija, resursu atribūtu grupas realizācija, resursu noteikumu tehniskā realizācija, vizuālā interaktīvā simulācija, koncepcijas pārbaude ar tipisku uzdevumu, koncepcijas pārbaude ar netipisku uzdevumu un jaunas funkcijas ieviešana.

1. Datu vadības sistēmu koncepciju analīze no datu glabāšanas un pārvaldības viedokļa

Mūsdienās, ja nepieciešams apskatīt informācijas apstrādi *informācijas sistēmu* (ang. *Information System, IS*) kontekstā, pirmkārt tiek apskatītas *datu vadības sistēmas* (ang. *Data Management System, DVS*). DVS var sadalīt divās lielās kategorijās: *monolītās datu vadības sistēmās* (ang. *Monolithic Data Management Systems, MDVS*) un *decentralizētās datu vadības sistēmās* (ang. *Decentralized Data Management Systems, DDVS*). Šo divu kategoriju sistēmām būtiski atšķiras uzbūves principi un īpašības.

Šai nodaļai ir trīs pamatmērķi. Pirmais mērķis ir definēt kritērijus objektu individuālajās īpašībās sakņotām decentralizētām datu vadības sistēmām. Otrais mērķis ir izpētīt vispazīstamākās DVS koncepcijas un novērtēt to atbilstību izvirzītajiem kritērijiem. Trešais mērķis ir no apskatītajām datu vadības sistēmu koncepcijām apkopot labās prakses principus un metodes, kas ir izturējušas laika pārbaudi. Apkopotie rezultāti tiks izmantoti objektu individuālajās īpašībās sakņotu sistēmu koncepcijas izstrādei.

1.1. apakšnodaļā ir izvirzīti kritēriji *objektu individuālajās īpašībās sakņotām sistēmām* (ang. *Data Management System Based on Individual Object Properties, OIĪSS*). 1.2. apakšnodaļā galvenā uzmanība veltīta MDVS un DDVS apskatam, nodaļas izvirzīto mērķu sasniegšanai. 1.3. apakšnodaļā ir dota apskatīto DVS koncepciju analīze un aplūkota to atbilstība 1.1. apakšnodaļas izvirzītajiem kritērijiem. Nodaļas beigās ir dots kopsavilkums un secinājumi.

1.1. Objektu individuālajās īpašībās sakņotas sistēmas (OIĪSS)

Šajā apakšnodaļā ir izvirzīti kritēriji objektu individuālajās īpašībās sakņotām sistēmām. Vispirms ir apskatītas mūsdienīgas IS galvenās attīstības tendences un dots apraksts objektu individuālajās īpašībās sakņotām sistēmām. Uz šīs informācijas pamata ir definēti kritēriji objektu individuālajās īpašībās sakņotām sistēmām. Nobeigumā ir apskatīti objektu individuālajās īpašībās sakņotu sistēmu tipiskie piemēri, lai pārbaudītu izvirzīto kritēriju darbību reālos piemēros.

1.1.1. Informācijas sistēmas uzbūves tendences

Lai definētu kritērijus objektu individuālajās īpašībās sakņotām sistēmām, jādefinē IS uzbūves principi. IS uzbūves principus reglamentē dažādi ar datorsistēmām saistīti virzieni: datu vadības sistēmas, programmēšanas valodas, tehnoloģijas un IS projektu vadība. Katram

virzienam ir milzīgs skaits iespējamo variantu. IS attīstības tendenču analīze ļauj izvēlēties gala lietotājam vispiemērotāko variantu.

DVS pētījuma laikā ir analizēti sekojoši informācijas avoti: datu vadības sistēmas teorija un ieskats nākotnē [Dat2004, Cod1990, Con2009, Ben2004, Sim1995, Bar1996, Bon1981, Bro1989, Dat2001b, Inm1998, Mey1998]; datu vadības sistēmas darbības principi [Alt1980, And1998, Atk1987, Bay1980, Ber1980, Ber1996, Blo2002, Bro2002, Wid1996]; datu vadības sistēmas manifesti [Atk1990, Dat2000a]; atsevišķas datu vadības sistēmas realizācijas [Gre2007, Adl2001, Bjo1973, Car1999, Dat1993, Row1987] utt.

Analizējot programmēšanas valodas un ar tām saistītās sfēras, ir apskatīta programmēšanas valodas teorija [Boo1991, Eck2003, Orl2004], programmēšanas valodas un vides realizācijas [AOS2011, Bur2005, Tel2011], datubāzes programmēšanas valodas [ISO1999, Dat1997, Cel1995, Ber1995, Cha1974, Cha1977, Atk1987], datu apmaiņas protokoli [Ber1992, Bay2010, Cro2006, Goo2012c], iezīmēšanas valodas [Cel1995, Ber1993, Ber1999, Aut2002, Ant2005, Cag2000] utt.

Apskatot IS projektu vadības nozari, ir parādījusies tendence uz spējo (ang. *agile*) metožu lietošanu [Pic2010, Hig2009, Ken2000, Ken2010].

Tiek apkopotas iepriekš informācijas avotos apskatīto IS attīstības tendences. Rezultāts pamatā sakrīt ar [Sim1995] pētījumu. Pētījumā tiek apgalvots, ka eksistē četras IS attīstības tendences, kas ir apkopotas 1.1. tabulā.

1.1. tabula

IS attīstības pamattendences (adaptēts no [Sim1995])

Nr.	Nosaukums	Apraksts
1	<i>Skaitļošanas resursu pārvaldības sadale un decentralizācija</i>	Ar katru gadu datorsistēmas kļūst mazākas un jaudīgākas. Izmantojot datu komunikācijas līdzekļus (lokālos un bezvadu tīklus), ir iespējams apmainīties ar informāciju starp ierīcēm. Gandrīz visus nepieciešamos darbus lietotāji var realizēt lokāli, nepieslēdzoties lieliem serveriem.
2	<i>IS komponentu neviendabīgums</i>	Dati tiek glabāti dažādos formātos, datu apstrāde tiek nodrošināta ar dažādu veidu programmatūru un aparatūru.
3	<i>Standartu izplatīšana</i>	2. tendences seku samazinājumu nodrošina ar standartu ieviešanu. Lietotājiem ir nepieciešama brīvība aparatūru un programmatūras izvēles procesā. Tas nodrošina brīvu konkurenci un vienkāršu IS izplatīšanu.
4	<i>Reālās pasaules modelēšana</i>	Viens no būtiskākajiem trūkumiem IS ieviešanā ir atšķirības starp IS un reālās pasaules procesiem. Viens no piemēriem ir

	<i>informācijas sistēmās</i>	uz <i>objektiem orientētu modeļu</i> (ang. <i>Object Oriented Model, OOM</i>) izmantošana. OOM ir principi, kuri bija piemēroti programmēšanai un datu vadībai. OOM ir vēl ciešāks sakars starp uzdevumu projektēšanu un realizēšanu.
--	------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1. tendence nosaka, ka notiek skaitļošanas resursu sadalīšana. Piemēram, ja dati tiek vadīti centralizēti, tad nepieciešama infrastruktūra (piemēram, bezvadu tīkls, internets utt.) ar kuras palīdzību notiek datu pārsūtīšana uz vadības centru. Tomēr, ja dati ir decentralizēti, tad nepieciešams nodrošināt pietiekamus skaitļošanas resursus uz vietas, piemēram, ja visas skaitļošanas funkcijas sadalītas starp visiem IS objektiem.

2. tendence nosaka, ka IS vienlaicīgi eksistē dažādi komponenti (bieži dažādu izstrādātāju veidoti) un ir nepieciešams nodrošināt mijiedarbību starp komponentiem. Konkrēta IS komponentu kopa nosaka datu glabāšanas un pārvaldības principa izvēli. Piemēram, centralizētā pārvaldības principa izmantošanas gadījumā, kad dati tiek glabāti teksta failā, tipisks uzlabojums ir relācijas datubāzes ieviešana (piemēram, skat. [Dat2004]). Tas ir gadījumā, kad jāmaina datu vadības sistēma uz decentralizētu. Tad var rasties būtiskas problēmas, kas neļauj veikt sistēmas pārbūvi, tad ir nepieciešams projektēt un izveidot jaunu sistēmu.

3. tendence nosaka standartizāciju. Monolītiem un decentralizētiem datu vadības principiem eksistē savstarpēji nesavienojami standarti. Piemēram, *daudzaģentu sistēmās* (ang. *Multi Agent System, MAS*) (detalizētāk skat. [Woo2001]) datu vadību nodrošina, izmantojot programmēšanas valodas, ar MAS programmatūras bibliotēkām (piemēram, Java [AOS2011, Tel2011], Python [Goo2011] utt.). Bet relācijas datubāzēm labāk izmantot datubāzes programmēšanas valodu (piemēram, *strukturēta vaicājumvaloda* (ang. *Structured Query Language, SQL*)).

4. tendence piedāvā reālās pasaules modelēšanu. Ir modelēšanas procesi, kurus uzbūve pieprasa monolītus datu vadības principus. Piemēram, ja ir noliktava, kur nepieciešams automatizēt preču uzskaiti. To parasti realizē ar relācijas datubāzes palīdzību (piemēram, [Dat2004]). Bet, ja nepieciešams modelēt skudru pūznīšu darbu, to iespējams izdarīt ar MAS (skat. [Vid2010]), kas savukārt nodrošina decentralizētus datu vadības principus.

Viena no galvenajām īpašībām, kas ir svarīga DVS koncepcijas izvēles gadījumā, ir monolītais vai decentralizētais datu glabāšanas un pārvaldības princips. Nākamajās nodaļās ir dots apskats no datu glabāšanas un pārvaldības principu raksturojuma skatu punkta.

1.1.2. OIĪSS apraksts

Objektu individuālajās īpašībās sakņotas sistēmas operē ar fiziskiem objektiem. Par *objektu* tiek uzskatīts reālās pasaules fiziskais objekts ar unikālām individuālajām īpašībām. *Īpašības* - tās ir objekta fiziskās īpašības (atribūti) un darbības ar tiem (noteikumi).

Fiziskā objekta struktūra ir aprakstīta, lietojot kopu teorijas pierakstu. Ja F ir fiziskais objekts:

$$F = \{A, N\},$$

kur A ir objekta atribūtu kopa, kas satur atribūtus no 1 līdz i :

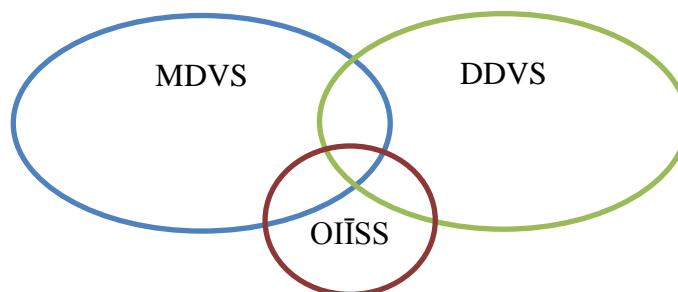
$$A = \{A_1, A_2, \dots, A_i\}$$

un N ir objekta noteikumu kopa, kas satur noteikumus no 1 līdz i :

$$N = \{N_1, N_2, \dots, N_i\}.$$

1.1.3. Kritēriju definēšana objektu individuālajās īpašībās sakņotām sistēmām

Ja tiek apskatītas objektu individuālajās īpašībās sakņotas sistēmas no datu pārvaldības viedokļa, tad daļa no raksturojumiem pārklājas starp monolītām un decentralizētām datu vadības sistēmām. Tiek prognozēts, ka daļa no nepieciešamajiem raksturojumiem nav pilnvērtīgi realizējami ar klasisko datu vadības sistēmu palīdzību (grafisko attēlojumu skat. 1.1. attēlā).



1.1. attēls. MDVS, DDVS un OIĪSS raksturojumu pārklājums

Pirmais promocijas darba uzdevums ir izvirzīt kritērijus OIĪSS klāstam, lai pārbaudītu MDVS un DDVS atbilstību šiem kritērijiem. No IS attīstības 1., 2. un 3. pamattendences (skat. 1.1. tabulā) izriet (i) kritērijs par *skaitļošanas resursu pārvaldības sadali un decentralizāciju*, kā arī (ii) kritērijs par *standartu izmantošanu*. Savukārt IS attīstības 4. pamattendence nosaka, ka jāvērsas pie reālās pasaules modelēšanas IS'ās.

OIĪSS tā ir sistēma, kas operē ar fiziskiem objektiem. Par *objektu* tiek uzskatīts reālās pasaules fiziskais objekts ar unikālām individuālajām īpašībām. Viens objekts atšķiras no cita objekta ar savām individuālajām īpašībām. Tās var būt objekta fiziskās īpašības un/vai operācijas, kuras jāpiemēro objektam. No tā visa izriet (iii) kritērijs par *fizisko objektu*

daudzveidības atbalstīšanu. Cilvēks fizisko objektu individuālo īpašību noskaidrošanai analizē objektu. Analīzei viņš izmanto pieredzi un uz vietas pieejamo informāciju. Definēts (iv) kritērijs *par lokālu datu piekļuves nodrošināšanu*. Ja visa nepieciešamā informācija ir pieejama, tad iespējams pieņemt lēmumu uz vietas. Tiek nodrošināts (v) kritērijs *par autonomu darbību*.

Lielām IS parasti izmanto MDVS (detalizētāk skat. [Sim1995]). Pamatojoties uz (v) kritēriju *par autonomu darbību*, var pieņemt, ka IS sistēma ir sadalīta mazos neatkarīgos risinājumos. Tiek ieviests (vi) kritērijs, ka *nelielu risinājumu ieviešana jābūt ekonomiski izdevīgai*. Vairākos informācijas avotos (detalizētāk skat. [Pic2010, Hig2009, Ken2000, Ken2010]) ir rekomendēts IS būvēt nelielās iterācijās, izmantojot spējos izstrādes principus. Šī rekomendācija definē/nosaka (vii) kritēriju *par pakāpeniskām ieviešanas izmaksām*.

Straujās informācijas un komunikāciju tehnoloģiju izaugsmes rezultātā pēdējo desmit gadu laikā strauji pieauga biznesa atkarība no uzticamas un drošas informācijas un komunikāciju tehnoloģiju funkcionēšanas. Būtu grūti atrast biznesa sfēru, kas nav atkarīga no informācijas tehnoloģijām un nebūtu atkarīga no informācijas procesiem. Informācijas sistēmas kļuva valdošās globālajā sabiedrībā un biznesā. Atkarība no šīm sistēmām un informācijas, kuru tās pārvalda, ir neapstrīdama. Trīs informācijas drošības pamati – konfidencialitāte, integritāte un pieejamība kļūst par galvenajiem faktoriem konkurētspējīgu priekšrocību un misijai kritisko faktoru nodrošinājumā, lai padarītu efektīvu un veiksmīgu biznesa procesu funkcionēšanu. [Kno2004] Balstoties uz to, ir izvirzīts (viii) kritērijs *par datu konfidencialitāti, integritāti un pieejamību*.

Divi no definētajiem kritērijiem ir vispārīgi kritēriji, tas ir (ii) *standartu izmantošana* un (viii) *datu konfidencialitātes, integritātes un pieejamības nodrošināšana*. Tie ir kritēriji, kas ir svarīgi jebkurai mūsdienīgai IS. Pārējie kritēriji ir specifiski OIĪSS risinājumiem.

Visi kritēriji ir apkopoti 1.2. tabulā.

1.2. tabula

Izvirzītie kritēriji objektu individuālajās īpašībās sakņotām sistēmām

Nr.	Nosaukums	Apraksts
I.	<i>Skaitļošanas resursu pārvaldības sadale un decentralizācija</i>	Atteikšanās no centralizētas vadības iespējama tad, ja risinājuma realizēšanai ir pietiekamas lokālās datorsistēmas skaitļošanas jaudas.
II.	<i>Standartu izmantošana</i>	Lai IS lietotājiem nodrošinātu aparatūras un programmatūras izvēles brīvību, ir jānodrošina mūsdienīgu standartu atbalsts. Īpaši nepieciešama brīvība aparatūras izvēles procesā. Jauna risinājuma ieviešanas gadījumā eksistējošās aparatūras

		risinājumu izmantošana samazina ieviešanas izmaksas.
III.	<i>Fizisko objektu daudzveidības atbalstīšana</i>	Sistēmai jābūt spējīgai operēt ar lielu objektu skaitu. Katram objektam ir savas individuālās īpašības. Vienam objektam var būt unikālas individuālās īpašības, kas nav piemērotas citiem objektiem.
IV.	<i>Lokālas datu piekļuves nodrošināšana</i>	Informācijas saņemšana no avotiem, kas neatteicas uz pašu objektu, aizņem papildus resursus (datu pārraide līdz krātuvei, datu ņemšana no krātuves, datu apmaiņas interfeisu nodrošināšana, datu krātuves vietu organizēšana, datu apmaiņas nodrošināšana caur dažādām krātuvēm utt.). Ja datus iespējams glabāt universālā vietā (piemēram, pašā fiziskajā objektā), tad iespējams būtiski samazināt datu nodrošināšanas izmaksas.
V.	<i>Autonoma darbība</i>	Ja katram DVS darbības elementam ir pieejami lokāli dati, kas nepieciešami lēmumu pieņemšanai, tad ir iespējams teikt, ka ir nodrošināta autonoma darbība.
VI.	<i>Nelielu risinājumu ieviešanas ekonomiskais izdevīgums</i>	Gadījumā, kad fizisko objektu skaits ir neliels, jābūt ekonomiskam pamatojumam izmantot tieši šo datu pārvaldības principu. Ja nepieciešams nodrošināt nelielu skaitu objektu mijiedarbību, ekonomiski neizdevīgi izmantot lielus DVS risinājumus.
VII.	<i>Pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale</i>	(vi) kritērijs par nelieliem risinājumiem savukārt prasa zemas ieviešanas izmaksas. IS ieviešana parasti ir vajadzīga, lai automatizētu manuālo darbu, un lielas ieviešanas izmaksas var samazināt IS ieviešanas ekonomisko izdevīgumu.
VIII.	<i>Datu konfidencialitātes, integritātes un pieejamības nodrošināšana</i>	Informācijas drošība ir svarīga visos informācijas procesos, fiziskos un elektroniskos, neatkarīgi no tā, vai tajos iesaistīti cilvēki un tehnoloģijas vai attiecības ar tirgus partneriem, klientiem vai trešajām personām. Tas attiecas uz visiem informācijas un tās aizsardzības aspektiem visos tās dzīves cikla punktos organizācijā.

1.1.4. OIĪSS piemēri

Šis apakšnodaļas ietvaros ir apskatīti IS piemēri, kas atbilst izvirzītajiem OIĪSS kritērijiem (skat. 1.2. tabulā). OIĪSS piemēri ir:

- *Autonomie kooperatīvie loģistikas procesi;*
- *Tīrīšanas, remonta un apkopes servisi;*
- *Viedās ēkas;*
- *Monitoringa, informēšanas un regulēšanas sistēmas;*
- *Uzņēmuma iekšējās politikas īstenošanas kontrole;*
- *Darba procesa analīze;*
- *Utt.*

Detalizēti ir apskatīti četri no minētajiem OIĪSS piemēriem. Katram no aprakstītajiem piemēriem ir dots problēmas apraksts ar atsauksmēm uz ārējiem avotiem. Sākotnēji ir aprakstīti tipiski MDVS uzbūves principu risinājumi. Papildus tiek piedāvāts, kā uzbūvēt jaunas IS, pamatojoties uz OIĪSS kritērijiem.

Autonomie kooperatīvie loģistikas procesi

Loģistika ir kravu pārvietošanas no viena punkta uz otru ar iepriekš definētām klientu vai kompāniju prasībām. Loģistika sevī ietver tādus darbības elementus kā transports, inventārs un noliktavas. Tās pamatfunkcija ir kravu saglabāšana un transportēšana, ievērojot privātuma un drošības jautājumus.

Pēdējos desmit gados pakāpeniski ir palielinājusies loģistikas struktūras sarežģītība un tās dinamika [Suh2005]. Eksistē vairāki iemesli paaugstinātām prasībām pret IS struktūras sarežģītību, piemēram, vairāku uzņēmumu ražošanas un loģistikas tīklu integrācija. Šis efekts tiek pastiprināts ar iekšēju un dinamisku sarežģītību (piemēram, lielāks skaits produktu variantu). Tapāt klienta dinamiska uzvedība pastiprina šo situāciju [Lar1999].

Efektīvai plānošanai un vadībai ir nepieciešama plaša un uzticama informācija [Cha2005]. Tomēr bāzes algoritmi, balstoties uz aprēķina iespējamo jaudu un sakarā ar lielo informācijas apjomu, drīz tiks izsmelti. Tas ir jāņem vērā. Ir iespējams prognozēt, ka nākotnē centralizētās plānošanas un kontroles metodes nebūs spējīgas apstrādāt visu piegādāto informāciju. Risinājums šai dilemmai ir visu nepieciešamo informāciju decentralizēta glabāšana pašas loģistikas objektā, kā arī iespēja pieņemt lēmumus uz vietas. Šī mērķa sasniegšanai loģistikas objektiem jābūt intelektuāliem [Mil2008].

Šīs koncepcijas galvenā ideja ir attīstīt decentralizētas un hierarhiskas plānošanas metodes, aizvietot eksistējošās centralizētās un hierarhiskās plānošanas pieejas. Tajā paredzēts, ka ir prognozēta iespēja, ka sistēmas sadarbības elementiem ir spēja un iespēja pieņemt lēmumus pastāvīgi [Bem2005]. Šīs koncepcijas uzskats zināmā mērā ir palielināt visu loģistikas objektu intelektu, lai uzglabātu un apstrādātu attiecīgos datus, pieslēgtos un

sadarbotos ar citiem loģistikas objektiem, kā arī iegūt informāciju par apkārtējo vidi un mijiedarboties ar to [Mil2008].

Katram loģistikas dalībniekam ir sava iekšēja, autonoma darbību organizācija un tiek izmantots savs IT risinājums. Ir gadījumi, kad viena firma pārpērk konkurentu, un tad vienā kompānijā tiek izmantoti dažādi IT risinājumi vienlaicīgi. Ir kompānijas, kurās nav ieviesta pilna IT automatizācija – tas bieži notiek tad, kad kā loģistikas dalībnieks ir trešās pasaules valstis. Ir gadījumi, kad uz īsu laika periodu jākomunicē ar netipiskiem loģistikas dalībniekiem, piemēram, tas novērojams avāriju loģistikā [She2007]. Visos gadījumos nepieciešams lēts, neatkarīgs, drošs un viegli ieviešams IT risinājums.

Tāda tipa IT risinājums var būtiski samazināt loģistikas izmaksas, ja tiek novērsti riski [Tsa2012]. Piemēram, viegli bojājamos produktu transportēšanas gadījumos, ja ir jātransportē produkts, kuram ir zināms un ar temperatūras sensoriem pārbaudīts temperatūras režīms. Ja ar produktu notiek incidents un produkts ir sabojāts, nav jēgas produktu transportēt līdz klientam. Ja krava satur iebūvētas instrukcijas sakarā ar šo noteikumu izpildīšanu (piemēram, ražotāja informēšana un kravu utilizācija), tad to iespējams izpildīt uzreiz un nav nepieciešams transportēt kravu līdz pasūtītājam (samaksājot par pilnu transportēšanu), saņemt ziņojumu no pasūtītāja par kravas sabojāšanos un tikai tad risināt problēmas par kravas transportēšanu.

Šī uzdevuma tipiska realizācija ir MDVS izmantošana. Ja darba laikā notiek datu izmaiņas, tad parasti jābūt iespējai šos datus nosūtīt uz centrālo krātuvi. Ir izņēmumi, kad dati tiek akumulēti un nosūtīti uz atsevišķiem punktiem. Loģistikas procesu gadījumā tā var būt noliktava, veikals, muiža utt. IT risinājumā tas nozīmē, ka jānodrošina infrastruktūra, kad visos posmos, kur nepieciešama lēmumu pieņemšana, būtu pieeja no *centralizēta vadības punkta* (ang. *Centralized Control Point, CVP*). Tas savukārt paaugstina IT risinājuma izmaksas vai arī jāsamazina lēmumu pieņemšanas iespējas.

Loģistikā tiek izmantota *radio viļņu identifikācijas* (ang. *Radio Frequency Identification, RFID*) tehnoloģija, kas paaugstina datu glabāšanas iespējas. Tagad iespējams informāciju glabāt RFID birkās un pārvietot to kopā ar precī vai konteineru, kas vienlaicīgi pārvieto daudzas preces. Ir risinājumi, kad RFID birku apvieno kopā ar sensoru (piemēram, temperatūras sensoru), tad konteineru pārvietošanas laikā iespējams saglabāt vides apstākļu parametrus (piemēram, temperatūras parametrus). Tā kā lēmumu pieņemšana tomēr notiek tikai vietās, kur ir savienojums ar CVP, tad iespējami gadījumi, kad laicīgi reaģēt uz vides apstākļiem nav iespējams. Ir nepieciešami risinājumi, kad lēmumu pieņemšana būtu iespējama uzreiz, kad rodas kritiska situācija. Piemēram, saldētu dārzenu pārvietošanas gadījumā, paaugstinoties temperatūrai konteinerā, tiek veikta derīguma termiņu analīze un uz vietas pieņemts lēmums par preču transportēšanu uz tuvāko noliktavu nevis uzreiz uz veikalu.

Vienā konteinerā var atrasties tūkstošiem preču, tās iespējams klasificēt pēc ražotāju preču veidiem, bet preces dzīves ciklā tiek veiktas modifikācijas (krāsošana, jaunu elementu pievienošana utt.), kas piešķir objektam individuālas īpašības. Tipiskas MDVS izmantošanas gadījumā nepieciešams nodrošināt lielu datu pārraidi objektu individuālo īpašību izmaiņu gadījumos un lēmumu pieņemšanas brīdī. Eksistē gadījumi, kad glabāt visus datus par objektu pašā objektā ir alternatīvs risinājums, kas ir izdevīgāk nekā izmantot citas DVS.

Tīrīšanas, remonta un apkopes servisi

Mūsdienās lielās konkurences dēļ pakalpojuma sniedzēji ir spiesti samazināt izmaksas. Viens no virzieniem, kā iespējams to realizēt, ir efektīvāk vadīt pašu procesu, analizējot [Lai2011] un atrodot efektīvāko vadības koncepciju. Cits virziens ir automatizēt darbības, līdz ar to sniedzot iespēju to veikt bez cilvēkresursu iesaistes. Automatizācija daļēji iespējama tīrīšanas, remonta un apkopes servisos. Ar tādu servisu tiek saprasts pakalpojums, kad servisa sniedzējs saņem precī (vai saņem pieeju precei) un maina tās kvalitātes parametrus, piemēram, tīra, remontē utt.

Viens no piemēriem ir automatizēts veļas mazgāšanas serviss. Tās ir intelektuāli automatizētas veļas mazgātavas servisa prasības apģērba tīrīšanai, kas sastāv no mazgāšanas, žāvēšanas, gludināšanas un profesionālas auduma kopšanas. Galvenais šīs sistēmas uzdevums ir minimizēt cilvēkresursu kļūdas, kas varētu sabojāt apģērbus. Tehniskās prasības nodrošina maksimāli atļauto auduma apstrādi un kopšanu atbilstoši ekspluatācijas nosacījumiem.

Pakalpojumu sniedzējiem parasti ir dažādi klienti. Klienti var saņemt dažādus pakalpojumus, ļaujot vienam un tam pašam servisam mainīt pakalpojuma sniegšanu. Parasti dažādi pakalpojumu sniedzēji izmanto dažādus IT risinājumus. Tas nozīmē, ka katram ir savs unikālais IT risinājums vai tas pats risinājums, bet cita veida instalācija. Pakalpojumu sniedzēji neapmainās ar informāciju, bet pats klients ir ieinteresēts, lai procesa informācija būtu pieejama visiem pakalpojumu sniedzējiem. Bet tajā pašā laikā pakalpojuma nodrošināšanā viens no svarīgākajiem parametriem ir klienta konfidencialitāte – klients nav ieinteresēts, lai procesa informācija būtu visiem publiski pieejama.

Aprakstīto uzdevumu tipiskais risinājums ir MDVS izmantošana. Tāds risinājums atklāj daudz problēmu ar datu individuālo īpašību pareizu detektēšanu un pareizu apkalpošanas servisa parametru izvēli. Parasti tāda tipa risinājumi paredz, ka daudzi darbi notiek manuālā režīmā. Servisa darbinieks pats noskaidro visu nepieciešamo informāciju un pats definē servisa apkalpošanas režīmu. Tā kā tas pārsvarā ir manuāls darbs, tad cilvēku pieļautās kļūdas ir viena no būtiskākajām problēmām.

Iespējams ieviest risinājumus, kas samazina cilvēka faktoru un maksimāli automatizē servisu darbu. Piemēram, veļas mazgātavas servisa gadījumā visas preces ir identificētas

(parasti ar svītrkodu vai RFID birku), visi nepieciešamie dati tiek glabāti MDVS, un servisa režīmi tiek izvēlēti pamatojoties uz MDVS glabātiem datiem. Lielākā daļa no servisu nodrošināšanas kompānijām neuzskata, ka tāda tipa risinājumi ir ekonomiski izdevīgi. Pārāk liels risks ir par objekta individuālajos īpašību izmaiņām, kad objekts atrodas citā servisa kompānijā. Mūsdienās netiek nodrošinātas kopējās datu krātuves, kurās iespējams glabāt tāda tipa informāciju. Bieži paši lietotāji datu konfidencialitātes dēļ, negrib savu personīgo informācijas glabāšanu atļaut citam personām (parasti uzņēmumiem). Tādu situāciju iespējams mainīt, ja nodrošina iespēju pašam lietotājam atbildēt par datu drošību un konfidencialitāti. Viens no variantiem ir datu glabāšana pašā objektā, piemēram, tādiem nolūkiem izmantojot RFID birku. Lietotājs var aizvērt datus (piemēram, ar šifrēšanas iespējām) un atvērt datus (piemēram, ar šifra atslēgu) tiem, kuriem ir atļauta datu pieejamība.

Viedās ēkas

Viedā ēka (ang. *smart house*) ir automatizēts mūsdienīgu ēku veids, kas nodrošina ērtu cilvēku sadarbību ar augsto tehnoloģiju ierīcēm. Parasti ar “viedo māju” saprot ēku pārvaldīšanas sistēmu, kas spēj atpazīt konkrētas situācijas, kas notiek ēkā un iepriekš definētā veidā reaģēt uz attiecīgajām situācijām. Viena apakšsistēma var kontrolēt citu apakšsistēmu uzvedību ar iepriekš izstrādātu algoritmu palīdzību. Galvenā sistēmas nozīme ir nodrošināt ēku atsevišķu apakšsistēmu integrāciju vienā pārvaldības sistēmā.

Dažādos pētījumos [Cha2008, Cha2009, Mon2007, Tan2012] tiek apskatīti mūsdienīgu viedo māju veidi un tiek definētas konkrētas tendences un problēmas, kuras ir nepieciešams atrisināt.

Tiek piedāvāti koncepcijas apraksti un realizācijas piedāvājumi [Mon2007]. Nepieciešams apvienot vairākas ražotāju izmantotās tehnoloģijas, augsto tehnoloģiju ierīces vienā pārvaldības sistēmā. Nepieciešams universālais mehānisms šo darbību organizēšanai.

Tiek apskatītas sadarbības iespējas ar citām viedajām mājām. Viens no piemēriem [Tam2012] ir atjaunojamo enerģijas resursu (piemēram, vēja vai saules) kopēja izmantošana. Viena no mūsdienu tendencēm ir e-veselība principu izmantošana, kad pacientu (gados veco cilvēku vai cilvēku ar speciālām vajadzībām) aprūpe tiek pārnesta no slimnīcas uz cilvēka dzīvesvietu [Cha2008, Cha2009]. Tāda tipa risinājumi rada decentralizētu IT risinājumu uzbūvi, kad unikālā viedā māja ar instalāciju palīdzību var komunicēt savā starpā un citiem IT risinājumiem.

Mūsdienās tipisks viedās ēkas risinājums (piemēram, skat. [Cha2008, Cha2009, Mon2007, Tan2012]) ir centralizēta vadības sistēma. Tas ir ekonomiski izdevīgi, ja nepieciešams nodrošināt instalāciju, kas saistīta ar daudziem iesaistītajiem objektiem. Ir gadījumi, kad tiek ieviestas nelielas instalācijas. Piemēram, tiek ņemta tālvadības aparatūra,

tiek ieprogrammēta universāla pults un caur to nodrošināta aparatūras vadība. Ja nepieciešams pieņemt lēmumu (piemēram, noteiktā laikā ieslēgt mājas apkuri), tad jāizveido programmatūra vai jāinstalē lielas centralizētas vadības sistēmas. Nepieciešams vienkāršs un lēts mehānisms, pāriešana no viedās ēkas daļējām instalācijām uz centralizētu vadību.

Parasti viedās ēkas IS atslēgas elements ir lietotājs. Viņu identifikācija tiek nodrošināta ar identifikācijas metodēm: koda ievadīšana, sejas atpazīšana, identifikācijas kartes izmantošana utt. Kopā ar identifikatoru iespējams glabāt arī lietotāja individuālās īpašības (vēlamā telpas temperatūra, alerģijas, radnieku informācija kritiskiem gadījumiem, asinsgrupa utt.). Objekta individuālās īpašības dod papildus informāciju viedajai ēkai lēmumu pieņemšanas gadījumā. Tāda tipa dati ir konfidenciāli, tāpēc glabāt tāda tipa datus centralizēti ir liels risks, bet ir jānodrošina datu pieejamība no jebkura vietas, kur nepieciešams pieņemt lēmumu. Lietotājam jābūt iespējām pašam regulēt kāda tipa datu konfidencialitātes jūtīgumu, kas ir publiski pieejami. Viens no iespējamajiem variantiem ir datu glabāšana lietotāju identifikācijas kartē, tādu iespēju nodrošina RFID birka.

Monitoringa, informēšanas un regulēšanas sistēmas

Tipisks piemērs tāda veida sistēmām ir automātiskās durvju slēdzenes. Ja nepieciešams atvērt slēdzeni, tad tas izdarāms vienīgi autorizētām personām, pēc autorizēšanās iespējams saglabāt vēsturi par personām, kuras un kad ir autorizējušās. Bieži tāda tipa uzdevumi neprasa monolītu sistēmu, kad visi dati tiek regulēti centralizēti – pietiek ar visas informācijas (autorizēto personu saraksta un a gadījumu vēsture) lokālu glabāšanu.

Galvenais elements monitoringa un regulēšanas sistēmās ir fiziskais objekts. Aprakstītās sistēmas nodrošina trīs pamatfunkcijas ar fiziskiem objektiem, kurus iespējams kombinēt pēc nepieciešamības:

1. *Monitorings* – fiziskie objekti tiek definēti kā datu nesēji, notiek informācijas uzkrāšana un saglabāšana. Visas iekšpusē iebūvētās funkcijas neietekmē pašus fiziskos objektus. Tipiskas funkcijas ir logošana, trešās puses sistēmu/objektu informēšana utt.
2. *Informēšana* – sistēmas pārvaldības bloks analizē pašreizējo situāciju un pēc iebūvētiem noteikumiem informē fiziskos objektus. Savukārt fiziskie objekti var saņemt šo informāciju un reaģēt uz to, piemēram, uzsākot konkrētas darbības.
3. *Regulēšana* – parasti šī funkcija apvieno monitoringa un informēšanas funkciju ar iespēju pārvaldīt fiziskus objektus. Piemēram, iepriekš aprakstītais piemērs par automātisko durvju slēdzeņu izmantošanu ir tieši tāda situācija. Sistēma monitorē apkārtējo vidi, gaida informāciju par lietotāju, kurš grib atvērt slēdzeni. Kad lietotājs autorizējas, sistēmā notiek slēdzeņu atvēršana vai lietotāja informēšana par atteikumu. Visa informācija tiek logota.

Parasti MDVS principu izvēle ir tipisks risinājums tāda veida sistēmām, bet, ja nepieciešams nodrošināt nelielu funkcionalitāti, tas kļūst ekonomiski neizdevīgi. Piemēram, ja nepieciešams nodrošināt tikai vienu durvju automātisku regulēšanu, nav ekonomiska pamatojuma iebūvēt centralizētu vadības sistēmu. Pēc vienu durvju instalācijas, ja nepieciešams nodrošināt visu ēkas durvju pieslēgšanu, to nepieciešams izdarīt bez pirmo durvju instalācijas demontāžas.

Eksistē vairāki risinājumi, kuru pamatā ir sensoru tīkli (skat. [Dar2010, Soh2007]). Ja nepieciešams monitorings un informēšana, sensori to var izdarīt vienkārši un lēti. Lokālu uzdevumu risinājumam arī *sensoru tīkls* ir ekonomiski neizdevīgs, šis risinājums vairāk ir piemērots (t.i. ekonomiski izdevīgāks) lielām instalācijām.

Pieņemsim, ka mums ir ceļu satiksmes regulēšanas uzdevums, kas atteicas tikai uz vienu krustojumu. Īsā laika periodā te ir daudz dažāda veida objektu, visi ceļu satiksmes objekti jāregulē reālā laikā. Mūsdienās vislabākajā gadījumā tā ir video kamera, kas fiksē pārkāpumus (luksofora sarkanās gaismas ignorēšana, ātruma režīma pārkāpšana utt.), rezultāts automātiski tiek nosūtīts uz CVP vai glabāts uz vietas un saņemts manuāli. Ieviešot visiem ceļu satiksmes dalībniekiem universālu identifikatoru ar objektu individuālo īpašību glabāšanu pašā objektā (piemēram, RFID birkā), kas savukārt dod iespēju izmantot sarežģītus autonomu lēmumu pieņemšanas scenārijus. Piemēram, Latvijas Republikā atļautais vieglā transporta ātruma ierobežojums pilsētās ir 50 km/h, bet to iespējams ceļu satiksmes dalībniekam, kam bijuši pārkāpumi, samazināt uz 40 km/h. Ātruma mērīšanas sensors, saņemot informāciju par mašīnas ātruma režīmu, izmanto objekta unikālās īpašības (piemēram, ātruma ierobežojums 40 km/h) un pieņem lēmumu, pamatojoties uz lokāli pieejamo informāciju.

1.2. Datu vadības sistēmas koncepcijas

Šī promocijas darba robežās tiek uzskatīts, ka *datu vadības sistēmas koncepcija* ir sistematizēti un apkopoti uzskati par šo sistēmu un procesiem tajā, ka tas ir viens no galvenajiem soļiem IS izstrādē vai modernizācijā. Citiem vārdiem sakot, koncepcija ir ģenerālais plāns, vadošā ideja, ceļu sistēma, kas apraksta kārtējā uzdevuma risinājumu. Darbu kontekstā *datu vadības sistēma* un *datu vadības sistēmas koncepcija* ir sinonīmi.

Visbiežāk datu vadības sistēmas aprakstam pielieto *datu modeļus* (ang. *Data Model, DM*). Adaptējot no [Dat2004], DM tā ir abstrakta, pieticīga, loģiska definīcija par objektiem, operatoriem un citiem elementiem, t. i. vispār tā ir datu piekļuves abstrakta mašīna, ar kuru mijiedarbojas lietotājs. Objekti nodrošina datu struktūras modelēšanu un operatoru datu uzvedību. DM var definēt abstraktai programmēšanas valodai, šīs konstrukcijas iespējams izmantot plaša uzdevumu klāsta realizēšanai, bet paši par sevi DM nav tieši saistīti ne ar vienu

konkrētu uzdevumu realizāciju. Balstoties uz [Lza2011] ir definēts, ka DM tie ir datu loģiskās struktūras organizācija datubāzē. Datubāzes pārvaldības sistēma parasti uztur tikai vienu datu modeli.

Terminu DM pirmoreiz ieviesa *Edgars E. Codd* savā ziņojumā [Cod1980]. Tajā ir doti fakti, ka *relācijas modelis* – tas ir pirmais aprakstītais DM. Pirmie teorētiskie modeļi bija realizēti, pamatojoties uz reāli strādājošām sistēmām. Relācijas modelis, tas ir pirmais DM, kas bija izveidots vispirms kā teorētiskais modelis. Un tikai uz relācijas modeļa pamata tika veikta programmatūras realizācija. Ne visas DVS ir aprakstītas ar modeļiem, darbā eksistē arī realizācijas apraksti.

1.2.1. Monolītas datu vadības sistēmas koncepcijas

No MDVS kopas DM daudzuma izšķir četrus modeļus: hierarhiskais, tīkla, relāciju un uz objektu orientēts modelis. Mūsdienās visizplātītākie DM izpētes virzieni ir relāciju un uz objektu orientēts modelis. Parasti citus DM apraksta, izmantojot šos četrus bāzes modeļus. Visas pārējās DVS iespējams būvēt, izmantojot četrus pamatmodeļus, bet parasti tiek izmantoti relāciju un uz objektu orientētais modeļi. Tādas DVS, kurām ir noteikti DM, tiek aprakstītas kā DM. Ir tādi, kuri skaitās relāciju modeļa paplašinājumi, tiem nav definēts stingrs DM, un tāpēc tie ir aprakstīti kā *datubāzes*.

1.2.1.1. Izplātās datnes modelis

Viena no pirmajām MDVS bija programmatūra, kas nodrošina *izplātās datnes modeli* (ang. *flat file model*) [Sim1995]. Šī DM dati tiek glabāti izplātās datnēs, kur dati tiek glabāti teksta vai bināra veidā. Ir divi glabāšanas veidi, kas glabā vienā rindā: vienu vērtību vai viena aprakstīta objekta vērtības kopu. Šādos ierakstos, viens vienīgs lauks var būt atdalīts ar norobežotājiem, piemēram, ar komatiem, vai dažādām citām zīmēm, kā arī lauks var būt ar fiksēto garumu. Pēdējā gadījumā var būt vajadzīgs papildinājums, lai sasniegtu šo garumu. Var būt vajadzīga arī papildus formatēšana, lai izvairītos no norobežotāju kolīzijām. Šajā modelī nav strukturālo relāciju starp ierakstiem.

Datnes apstrādes modelis tiek nodrošināts ar *galveno datni* (ang. *Master File, GD*) un *transakcijas datni* (ang. *Transaction File, TD*). GD glabā esošo datu krājumu, bet ar TD tiek nodrošināts datu izmaiņas mehānisms. Piemēram, GD glabā informāciju par pašreizējo noliktavas statusu (preces un atlikumi), TD – tā ir jaunas preces piegāde. Tad programmatūra nodrošina jaunas GD izveidošanu, kad esošā GD tiek papildināta ar TD datiem. Izplātās datnes piemērs ir `/etc/passwd` un `/etc/group` UNIX operētājsistēmās [Rob2005a, Rob2005b, Lov2005].

Tiešā veidā šajā MDVS izmantošana ir nepamatota, šis modelis bija izmantots tehnisko līdzekļu ierobežojumu dēļ. Jau eksistē daudz efektīvāki DM, kas aprakstīti tālāk darbā.

1.2.1.2. Datubāzu ēra

Ir pieņemts uzskatīt, ka termins *datubāze* (ang. *Database, DB*) pirmoreiz ir noformulēts un publicēts 60. gadu sākumā. Mūsdienās parasti divus terminus *datubāze* un *datubāzu vadības sistēmas* (ang. *Database Management System, DBVS*) izmanto kā sinonīmus. Bet tomēr ir atšķirības. [Dat2004] tika definēts, ka *datubāze* – tā ir dažu permanentu (glabāti pastāvīgi) datu kopa, kurus izmanto patērētājs (piemēram, uzņēmums) ar lietojumprogrammatūras sistēmu starpniecību. Citu definīciju dod [Lza2011], DB - tas ir savstarpēji saistītu informacionālu objektu tematisks kopums, kas ar speciālas pārvaldības sistēmas starpniecību organizēts tā, lai nodrošinātu ērtu informācijas izguvi, veiktu tās atlasīšanu un kārtošanu. Informācija datu bāzē parasti ir sadalīta ierakstos (tabulās), no kuriem katram var būt viens vai vairāki lauki (kolonnas).

DBVS, vai *datubāzes menedžeris* (ang. *database manager*), vai *datubāzes serveris* (ang. *database server*) [Dat2004] – tās ir programmatūras, kas atrodas starp datubāzi un patērētāja lietojumprogrammatūras sistēmām. [Lza2011] definē, ka tā ir lietojumprogrammatūra, kas organizē datus datubāzē, nodrošinot to uzglabāšanu, izguvi, drošību un integritāti. Datubāzes pārvaldības sistēma parasti veic izdrukājamo pārskatu formalizēšanu, kā arī eksportu uz un importu no citām lietojumprogrammām, izmantojot datņu standartformātus. Faktiski, terminu DBVS izmanto, kad nepieciešams nosaukt konkrētu produkta realizāciju, piemēram, Oracle database [Gre2007], Microsoft SQL Server [Hot2008], DB2 Universal Database [Dat1993] utt.

Lai pilnīgi saprastu situāciju, nepieciešams pieminēt, ka DBVS tas ir galvenais, bet ne vienīgais programmatūras sistēmas komponents. Eksistē arī utilīti, aplikāciju izstrādāšanas rīki, projektēšanas līdzekļi utt. Bet šīs nodaļas ietvaros galvenā uzmanība ir pievērsta DBVS, tāpēc ka parasti šie rīki universāli. Piemēram, mūsdienās aplikāciju izstrādāšanas rīks Eclipse (skat. detalizētāk [Bur2005]) dod iespēju izveidot IS, kas ir visu klasisko MDVS pamatā.

1.2.1.3. Hierarhiskais modelis

Viens no klasiskajiem DB piemēriem ir *hierarhiskais modelis*, kas var būt attēlots kā koks, kur vienā objektā iekļauti citi objekti. Objekti, kas atrodas kopā vienā objektā, atrodas vienā līmenī. Objekts no pirmā līmeņa iekļauj objektus no otrā līmeņa, savukārt otrā līmeņa objekts satur objektus no trešā līmeņa utt. Entītijai tipi ir saistīti cits ar citu, izmantojot 1:N kartēšanu, parasti tos sauc par viens-pret-daudziem relācijām.

Šī struktūra ļauj dublēt informāciju, izmantojot priekšteča / pēcnācēja attiecības: katram no vecākiem var būt daudz bērnu, bet katram bērnam ir tikai viens no vecākiem. Īpašā ierakstā visi atribūti ir uzskaitīti zem entitijas tipa. Šim modelim ir vienīgais pirmais elements. Šis ierobežojums ir galvenais trūkums, jo reālo pasauli nav iespējams dabiskā veidā aprakstīt ar koka struktūru, kur ir vienīgais pirmais elements.

Vispazīstamākā hierarhiskā modeļa komerciāli izdevīgā realizācija ir *IMS* (ang. *Information Management System*), kuru izstrādāja *IBM*.

1.2.1.4. Tīkla modelis

Tīkla modelis ir hierarhiskā modeļa tiešais konkurents [Sim1995], vēl šo modeli sauc par hierarhiskā modeļa paplašinājumu. Ja hierarhiskais modelis strukturē datus kā ierakstu koku, kur katram ierakstam ir viens vecāku ieraksts un daži bērnu ieraksti, bet tīkla modelis ļauj katram ierakstam veidot dažus vecāku ierakstus un bērnu ierakstus, tādējādi formējot režģa struktūru. Šajā tīkla struktūrā katrs elements var būt saistīts ar jebkuru citu elementu.

Tīkla modelis sastāv no ierakstu un sakaru salikuma starp šiem ierakstiem. Sakaru veidošanai netiek iedalīti speciali ierobežojumi. Hierarhiskajā modelī vienam ierakstam ir viens sencis; tīkla modelim iespējams glabāt jebkuru senču skaitu. Par nozīmīgu modeļa plusu var uzskatīt efektīvas realizācijas iespēju pēc atmiņas un operativitātes patēriņa rādītājiem. Salīdzinājumā ar hierarhisko modeli, tīkla modelis paver plašas iespējas patvaļīgo sakaru realizācijās. Tīkla modeļa izmantošanu apgrūstina datu glabāšanas shēmas pastāvība un īpaši sarežģīta projektēšana, kā arī sarežģīta datu apstrādes mehānismu izstrāde. Daudzu sakaru patvaļīgas uzstādīšanu dēļ ir novājināta sakaru integritātes kontrole.

70. gados tika radīts komerciālu popularitāti ieguvušais produkts *IDS* (ang. *Integrated Data Store*), kuru izstrādāja General Electric.

1.2.1.5. Relāciju modelis

Mūsdienās vispopulārākais ir DM, parasti ar terminu datubāze saprot relācijas datubāzi (sīkāk skat. [Mai1983]). Daudzi zinātnieki, kuri strādā datubāzu sfērā (piemēram, *Date C.J.* [Dat2004] grāmatā) uzskata, ka relācijas datubāze ir vislielākais ieguvums datubāzu jomā pēdējo 40 gadu laikā. Relāciju modeli pirmoreiz ierosināja *Edgars E. Codd* 1970. gadā [Cod1970] pētījumā, bet pašas idejas jau bija publicētas [Cod1969]. Pirmoreiz relāciju modelis tika aprakstīts kā datu modelis [Cod1980]. Balstoties uz [Cod1990] tika piedāvāts paplašināt relāciju modeli līdz otrajai versijai „The Relational Model Version 2”. Pirmā versija definēja kopējo datubāzes problemātiku, otrā versijā ir abstraktais plāns kopējai sistēmai. Detalizētāk par *Edgars E. Codd* ieguldījumu relācijas modelī pētīts [Dat2001b].

Relāciju modeļa galvenā ideja ir aprakstīt datubāzi kā predikātu kolekciju pār galīgo predikatīvo mainīgo *komplektu* (ang. *set*), kas apraksta robežas iespējamajām mainīgo

vērtībām un kombinācijām. Datubāzes saturs jebkurā laika momentā ir galīgs (loģiskais) datubāzes modelis, t. i. relāciju komplekts, viena uz predikatīvo mainīgo, tā, lai visi predikāti būtu apmierināti. Pieprasījums informācijai no datubāzes (datubāzes pieprasījums) arī ir predikatīvs.

Relāciju modeļa mērķis ir nodrošināt deklarātīvo metodi datu un pieprasījumu specifikācijai: mēs zinām, kādu informāciju satur datubāze un kādu informāciju mēs gribam saņemt no tās, un ļaujam datubāzes vadības sistēmas programmatūrai rūpēties par datu struktūru aprakstiem. Relācijas modelis nodrošina datu satura papildināšanu bez aplikācijas koda izmaiņām.

Tas ir datu modelis, kurā dati organizēti, izmantojot savstarpējas mijiedarbības, kas parasti atspoguļotas tabulas formā. Tāda tipa datubāzēs starp divām vai vairākiem tabulām var tikt veidotas sakarības, kuras definē, ka katram galvenās tabulas ierakstam ir sakarība ar vienu vai vairākiem ierakstiem sekojošās tabulās. Šīs sakarības var būt iedalītas: viens–pret–daudziem, viens–pret–vienu, daudzi–pret–daudziem.

Pirmoreiz pilnvērtīgi relācijas datubāzēs tika ieviests transakcijas mehānisms. Transakcija [Dat2004] – tā ir loģiska darba vienība, kas sākas ar sākšanas operāciju un beidzas ar transakcijas apstiprināšanas operāciju vai transakcijas atteikšanas operāciju. Transakcijai ir četras galvenās īpašības, tās ir – *nepārtrauktība, pareizība, izolēšana un izturība* (ang. *Atomicity Correctness Isolation Durability, NPID*). Detalizētāk skatīties [Har1983].

Vienu no galvenajām lomām relatīvās datubāzes pārvaldības sistēmās spēlē *strukturēta vaicājumuvaloda* (ang. *Structured Query Language, SQL*) programmēšanas valoda [Cel1995, ISO1999, Dat1997], kura tika speciāli izstrādāta, lai veiktu datubāzes vaicājumus. Pirmajai SQL versijai bija nosaukums SEQUEL (definēts balstoties uz [Cha1974]), valoda tika uzlabota un specificēta kā SEQUEL/2 (definēts [Cha1977]). SQL ir relācijas datubāzes formāts de-facto, bet diemžēl arī šodien lielākā daļa no DBVS pilnvērtīgi neizmanto aprakstīto standartu (piemēram, skat. pētījumos [Fis1997, Dat2004]).

1.2.1.6. Objektorientēts modelis

Viens no pirmajiem mēģinājumiem definēt *objektorientētu modeli* tika izdarīts [Atk1990] manifestā, ziņojumā par objektorientētām programmēšanas datubāzes valodām [Atk1987]. *Objektorientēts modelis* ļauj datubāzes objektiem izskatīties kā programmēšanas valodas objektiem vienā vai dažās programmēšanas valodās. Objektu datubāzes vadības sistēma paplašina programmēšanas valodas ar caurspīdīgiem noturīgiem datiem, laiksakritības vadību, datu atkopšanu, asociatīviem vaicājumiem un citam spējām. Viena no galvenajām idejām ir paaugstināt abstrakcijas līmeni un pāriet no baitiem un bitiem, uz objektiem un operācijām ar objektiem.

Šajā modelī katra lietotāja noteiktā struktūra ir objekts, kurš tieši tiek vadīts ar datubāzes palīdzību. Relācijas modelī saites tiek vadītas ar izveidotām atslēgām, savukārt šajā modelī lietotājs pats definē saites un DBVS automātiski ģenerē vadības metodes – dinamiski veidojot, dzēšot vai krustojot saites. Norādes šajā gadījumā ir tiešas, nav nepieciešams apskatīt un salīdzināt vai arī meklēt indeksu, kas var stipri ietekmēt veikspēju. Tāpēc šo objektu koncepciju labāk izmantot, kad vajadzīga augsti veikspējīga datu apstrāde, kuriem ir sarežģīta struktūra, kad tiek apstrādāta datubāze ar lielu sarežģītu saišu daudzumu (krusteniskas norādes, norādes, kuras savieno vairākus objektus ar vairākām „*daudz – pret – daudziem*” attiecībām). Atšķirībā no relāciju modeļa, objektu modelis pilnībā atbalsta objektorientētas programmēšanas valodas. Lietojumprogrammas funkcionē ar objektiem, kuri ir saglabāti datubāzē, kura izmanto standartu objektorientētu valodas semantiku un operācijas.

Objektu modelis ir piemērotāks sadalītas skaitļošanas organizēšanai. Vairāki lietojumi darbojas sadalītajās vidēs, kurās objekti ir sadalīti dažādās stacijās un serveros, kur jebkurš lietotājs var saņemt piekļuvi pie jebkura objekta.

Galvenie pētījumi šajā jomā notika no 80. gadu beigām līdz 90. gadu vidum. Šai laika periodā vairākiem zinātniekiem bija viedoklis, ka tirgū objektorientētas datubāzes aizstās relācijas modelis. Mūsdienās šāds viedoklis ir nelielai pētnieku grupai, tiek uzskatīts, ka šī joma ir diezgan specifiska (piemēram, skat. [Mey1998]). Tomēr ir sfēras, kurās objektorientētais modelis tiek efektīvi izmantots (skat. detalizētāk [Bar1996]), piemēram, [Dat2004] min šādus piemērus: automatizētas projektēšanas sistēmas, automatizēta ražošanas vadība, automatizētas programmatūras izstrādāšanas sistēmas, ģeoinformācijas sistēmas, zinātne, medicīna, dokumentu apstrāde un glabāšana utt.

Dažas pazīstamākās objektu datubāzes vadības sistēmas ir IBM Lotus Notes/Domino, Jasmine, ObjectStore.

1.2.1.7. Objektu-relāciju modelis

90. gadu beigās parādījās pirmās *objektu-relācijas DBVS*, tās parasti sauc par *universāliem serveriem* (ang. *universal servers*). Tādu tipu produktu radīšanas iemesls bija objektorientētā un relāciju modeļu apvienošana.

Objektu-relācijas modelis satuvina datubāzu un lietojumprogrammu izstrādi, nodrošinot to, ka datubāze izmanto to pašu tipu sistēmu kā lietojumprogrammām. Tajā laikā objektu datubāzes mēģina ieviest galvenās objektorientētās programmēšanas idejas, tādas kā polimorfisms un iekapsulēšana. Tādējādi ar datubāzi var strādāt kā ar relāciju datubāzi un papildus izmantot objektu orientētās datubāzes priekšrocības bez to lielās sarežģītības. Tās atbalsta datu modeļa paplašināšanu ar paša izstrādātajiem datu-tipiem un metodēm.

Mūsdienās relāciju DBVS ir tendence iebūvēt objektu-relācijas modeli. Tāda DBVS ir PostgreSQL (detalizētāk skat. [Sto1986, Row1987, Sto1987, Sto1991]) un Starburst (detalizētāk skat. [Haa1989, Lin1987]), bet šo struktūru atbalsta arī DB2 Universal Database [Adl2001, Car1999] un citas populāras DBVS.

1.2.1.8. Aktīvās datubāzes

Iepriekš aprakstītajos modeļos dati tiek atdalīti no datu apstrādes mehānismiem. Šeit ir izmantota pretēja pieeja. Visas manipulācijas, kuras ir iespējams realizēt ar datiem servera pusē, tiek realizētas ar *trigera* palīdzību. *Trigeris* – tās ir gatavas procedūras (iepriekš nokompilētas), kas glabājas datubāzē un automātiski tiek palaistas, kad notiek iepriekš definētie notikumi. Trigeri var raksturot ar 3 sastāvdaļām [Dat2004]: notikums, nosacījums un darbība. Notikumu un nosacījumu kopumu sauc par *trigera notikumu*, bet visu triju pamata komponentu kopumu sauc par *trigeri*. Tipiska trigera realizācija dod iespēju trigeri palaist *pirms notikuma* (ang. *before*), *pēc notikuma* (ang. *after*) vai *notikuma darbības vietā* (ang. *instead of*). Detalizētāk skatīties [Ros1994, Ros1998, Wid1996].

Trigeri parasti tiek izmantoti sekojošiem nolūkiem [Dat2004]: informēšana (piemēram, DB lietotājs tiek informēts par izņēmuma palaišanu), kļūdas labošana (piemēram, mainīgu vērtību uzskaitē vai mainīga vērtību izmaiņas kontrole), audits (piemēram, informācijas reģistrēšana par iepriekš definētām DB izmaiņām, kurš un kad tās veica), jaudu aprēķins (piemēram, reģistrēt laiku, kad notika darbība; vai iepriekš definēto noteikumu trasēšana DB) un kompensēto darbību nodrošināšana (piemēram, viena ierakstu dzēšanas gadījumā tiek dzēsti visi saistītie ieraksti citās tabulās).

Šim DB virzienam ir perspektīvi pētījumi [Bro1989], kuros pētnieki cenšas apvienot mākslīgo intelektu ar datubāzes tehnoloģijām.

1.2.1.9. Datu noliktavas un citas specifiskas datu vadības sistēmas

40. gadu beigās un 50. gadu sākumā parādījās nepieciešamība pēc lēmumu pieņemšanas atbalsta sistēmām. Šādu sistēmu galvenā ideja ir vēsturiskās operatīvās informācijas analīze nodrošināšana, lai uz biznesa procesu izpētes pamata nodrošinātu uzņēmējdarbības regulēšanu. 70. gadu sākumā sākās pētījumi (piemēram, [Mor1971]), kas piesaistīja šim procesam datorsistēmas. Vispirms šādi risinājumi tika izveidoti uz *datu krātuves* (ang. *data store*) bāzes. 80. gadu sākumā pēc dažādiem analītiskiem pētījumiem [Kee1978, Alt1980, Ben1981, Bon1981] tieši tāda tipa nolūkiem sāka izmantot relāciju datubāzes. No pētījuma [Inm1988] izriet, ka pirmoreiz tika ieviests termins, ka tāda tipa sistēmas sauc par *datu noliktavām* (ang. *data warehouse*). Viens no pirmajiem pētījumiem ir [Del1988, Inm1988], bet detalizēti apraksti parādījās vēlāk (skat. [Inm1992, Inm1994]). Viena

no definīcijām [Lza2011], ka tās ir dažādu uzņēmumu uzņēmējdarbības sistēmas, kurās ir nozīmīga savākto datu centrālā glabātuve.

Datu noliktavas sāka kļūt populāras 90. gadu sākumā. Un šajā laikā tika noskaidrots, ka lietotāji veido pieprasījumus tikai ierobežotai datu daļai, bet lielākā daļa no informācijas netiek izmantota. Un šiem ierobežotajiem datiem bija nepieciešama ātra piekļuve. Tiek izveidots jauns datu noliktavas veids – *datu vitrīnas*.

Datu vitrīnas – tā ir ierobežota daļa no kopējas datubāzes, kas ir vajadzīga operatīvo pieprasījumu organizēšanai. Dati saņemti reālā laika pieprasījumos, kas optimizēti ātrākai piekļuvei. Datu vitrīna ir datu noliktavas apakškopa, kas nodrošina atsevišķa departamenta vai atsevišķas biznesa funkcijas prasības. Datu vitrīna satur datu noliktavas datu apakškopu, parasti kopsavilkuma informācijas veidā, kas attiecas uz atsevišķo departamentu vai biznesa funkciju. Tā kā datu noliktava kļūst arvien lielāka, tad iespēja nodrošināt vairākas uzņēmuma vajadzības noved pie dažādiem kompromisiem. Datu vitrīnu popularitāti izskaidro fakts, ka korporatīvo datu noliktavu realizācija un izmantošana sagādā grūtības.

Statistikas datubāze [Dat2004] – tā ir datubāze, kur atļauts palaist vaicājumus tikai agregātiem datiem (summētas, ar vidējo vērtību aprēķinu, utt.), tas nozīmē, ka nav iespējami vaicājumi ar elementāriem datiem. Piemēram, statistikas datubāzei atļauts palaist vaicājumu „*Kāds ir vidējais atalgojums darbiniekiem?*”, bet tomēr palaist vaicājumu „*Kāds ir atalgojums darbiniekam Mēri?*” ir aizliegts. Tāda tipa datubāzes var definēt kā datubāzes ar ieviestu ierobežotu datu piekļuves politiku.

1.2.2. Decentralizētas datu vadības sistēmas koncepcijas

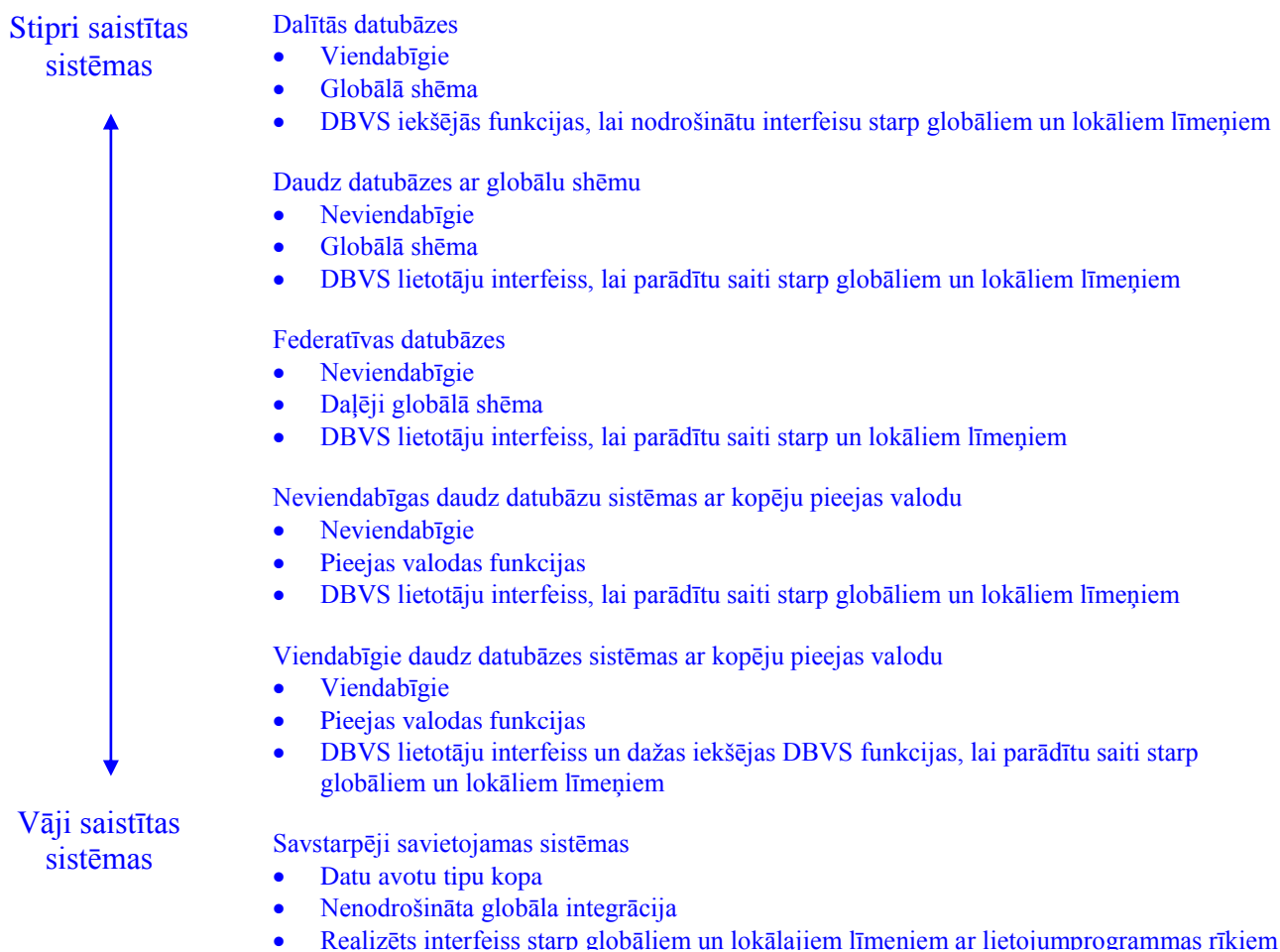
No DDVS kopas šajā apakšnodaļā tiek apskatīti pieci galvenie DVS veidi: *dalītās datubāzes, daudzagentu sistēmas, hipervides datubāzes, vienranga datubāzes un sensoru tīkli*. Dalītās datubāzes apvieno sevī kā MDVS, tā arī DDVS pamatraksturojumus un nodrošināta centralizētu datu vadību, bet tomēr tiek izmantoti arī decentralizēti datu glabāšanas un vadības principi. Šim datubāzes veidam galvenais mērķis (skat. divpadsmit pamatmērķi, kuri ir aprakstīti [Dat1990]) ir autonomu datorsistēmu mijiedarbība, tiek pieņemts šo DVS veidu uzskatīt par DDVS.

1.2.2.1. Dalītās datubāzes

Dalītās datubāzes (ang. *Distributed Database, DDB*), definējums ir adaptēts no [Dat2004]), tas ir, kad viena lietojumprogramma var „*pārredzami*” apstrādāt datus, kas ir sadalīti starp daudzām dažādu datubāzes kopām, kuru vadību nodrošina dažādas DBVS, kas strādā ar dažāda veida datorsistēmām un ar dažādām uzstādītajām operētājsistēmām un ir savienotas ar komunikāciju tīkliem. Termins „*pārredzami*” nozīmē, ka lietojumprogramma no

loģikas viedokļa apstrādā datus tādā veidā, ka datu pārvaldība pilnīgi tiek nodrošināta ar vienu DBVS, kura strādā vienā datorsistēmā.

DDB sastāv no daudzām datubāzēm kopā, kopējā daudz datubāzu sistēmas taksonomija ir dota 1.2. attēlā. Vertikālajā asī parādīts integrācijas līmenis, jo augstāk, jo labāk nodrošināta integrācija.



1.2. attēls. Daudz datubāzu sistēmas taksonomija (adaptēta no [Sim1995])

DDB sastāv no mezgliem, kur katrs mezgls ir pilnvērtīga DBVS. Mezgli mijiedarbojas savā starpā tādā veidā, ka jebkura mezgla lietotājs var saņemt datus no jebkuras DDB sastāvdaļām. Katrā mezglā ir papildus programmatūras bloks, kas papildina esošo DBVS un nodrošina tās funkcionalitāti. DBVS, kurā iebūvēts papildus DDB programmatūras bloks, sauc par *dalītām datubāzes pārvaldības sistēmām* (ang. *Distributes Database Management Systems, DDBVS*).

DDB mērķis ir sadalīt informācijas avotus pa autonomiem blokiem. Parasti tas ir fiziski ģeogrāfiskais sadalījums. Bieži katra atsevišķa organizācija izmanto savu DBVS, un DDBVS nodrošina neviendabīgu DBVS mijiedarbību. Tiek pieņemts, ka lielākā daļa no pieprasījumiem pēc datiem tiek veikta lokālajā DBVS un nav nepieciešams nodrošināt

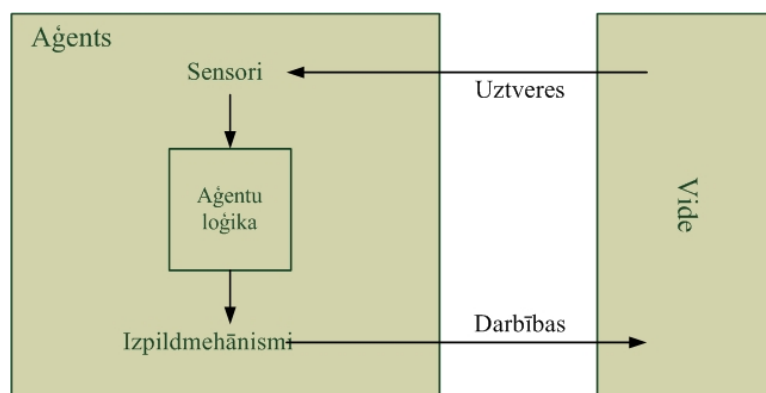
MDVS, ātrdarbības (ja ir centrālā vadība, tad jebkura veida pieprasījums jāsūta uz MDVS) un izmaksu dēļ (parasti jau ir lokālās DBVS, un vienas DBVS datu apvienošana ar MDVS prasa milzīgus resursus).

DDB sistēmu aktīvi sāka pētīt 70. gados. Viens no pirmajiem pārskatiem ir [Rot1977]. Viens no pirmajiem komerciālajiem produktiem, kas pilnvērtīgi izmanto DDB principus, bija *SDD-1* (ang. *System for Distributed Databases 1*) [Rot1980]. Tiek secināts [Dat2004], ka lielākajā daļā no mūsdienīgajiem komerciālajiem produktiem, kuriem ir SQL atbalsts, ir iebūvēts DDB modulis, bet ar dažādām DDB integrācijas iespējām.

Mūsdienīgās DDB izmantošanas tendences parāda, ka, pamatojoties uz lielāko tiesu lētām un mobilām datorsistēmām, kā arī ar bezvadu komunikācijas tehnoloģijas izplatīšanos, nodrošina jauna tipa DDB radīšanu. Tāda tipa sistēmas sastāv no tūkstošiem mezglu, bet šie mezgli ir mobili, bieži nav pieejami tie, kuru operatīvie raksturojumi atšķiras no lielu serveru sistēmu raksturojumiem (piemēram, skat. [Ren1997, And1998, Gra1996]).

1.2.2.2. Daudzaģentu sistēmas

Daudzaģentu sistēmas (ang. *Multi Agent System, MAS*) – tās ir sistēmas, kas tiek veidotas no daudziem mijiedarbojamiem skaitļošanas elementiem, ko sauc par *aģentiem*. Aģenti ir datorsistēmas, kas ievietotas kaut kādā vidē un nodrošina divas galvenās iespējas. Pirmā iespēja, tā ir kaut kādā mērā nodrošināt autonomu darbību, citiem vārdiem sakot, nodrošināt lēmumu pieņemšanu balstoties uz informāciju, ko saņem no ārējās vides, lai nodrošinātu izvirzītos mērķus. Otrā iespēja ir tāda, ka, šie elementi var mijiedarboties cits ar citu un mainīt apkārtējo vidi, nodrošinot ne tikai informācijas apmaiņu, bet arī veikt kādas sociālas aktivitātes analogiski tam, kā cilvēki to dara savā ikdienas dzīvē. Citiem vārdiem sakot, koordinācija, sadarbība, sarunas un citas darbības. Tipiska mijiedarbības shēma, kas adaptēta no [Jen1998], ir dota 1.3. attēlā.



1.3. attēls. Aģents mijiedarbojas ar vidi (adaptēta no [Jen1998])

Aģenti ir autonomi un var būt heterogēni pēc savas izcelsmes. MAS raksturojumi ir šādi [Jen1998]: katram aģentam ir nepilnīga informācija vai iespējas problēmu risināšanai,

tāpēc katram aģentam ir ierobežots viedoklis; nav globālas sistēmas kontroles; dati ir decentralizēti, un datu apstrāde ir asinhrona.

MAS ir relatīvi jauna apakšnozare datorsistēmās, kas tika pētītas sākot no 80-ajiem gadiem, un šī tēma ir ieguvusi plašu atzinību sākot no 90. gadu vidus. No tā laika starptautiskā interese šajā nozarē ir strauji palielinājās [Woo2001].

MAS pētījumu mērķis ir atrast tādas metodes, kas ļaus mums izveidot sarežģītas sistēmas, kas sastāv no autonomiem aģentiem, kas darbojas uz lokālu zināšanu bāzes un kurām piemīt ne tikai ierobežotas iespējas, bet tās tomēr spēj veikt pārdomātas globālas darbības. Mēs gribam zināt, kā var visas iespējamās sistēmas aģentu darbības sadalīt starp individuālo aģentu darbību kopām. MAS mērķis ir apgrieztās inženierijas fenomēns, kas apraksta skudru kolonijas, ekonomiku un imūnsistēmu. MAS pieeja problēmai ir izmantot sevi pierādījušas ierīces no *spēļu teorijas* (ang. *game theory*), ekonomikas un bioloģijas. Šo pieeju papildina ar idejām un algoritmiem, mākslīgā intelekta pētījumiem, *tiešo plānošanu* (ang. *namely planning*), *pamatojuma metodēm* (ang. *reasoning methods*), *meklēšanas metodēm* (ang. *search methods*) un mašīnāpmācību [Vid2010].

Parasti MAS pēta atsaucē uz aģentiem, kas realizēti programmatūrā. Bet, aģenti *daudzaģentu sistēmā* var būt roboti, cilvēki, kā arī cilvēku komandas. MAS var būt arī kombinētās cilvēku-aģentu komandas [Feb1999].

1.2.2.3. Hipervides datubāzes

Globālais tīmeklis (ang. *World Wide Web, WWW*) [Lza2011] ir globāla hiperteksta sistēma, kas izmanto tīklu Internet kā informācijas transportēšanas mehānismu. Šo hiperteksta sistēmu veido informācija, kas sadalīta atsevišķās vienībās, kuras saistītas ar īpašām asociatīvām saitēm — hipersaitēm, tādējādi veidojot tīmekli.

Mūsdienās globālais tīmeklis ir milzīga *hipervides datubāze*. Un Internets tas ir arī milzīgs tīkls, kuru izmantojot, šī datubāze ir iedarbināta. WWW kā tādu izveidoja *Tims Berners-Li* (ang. *Tim Berners-Lee*) (fakts no [Ber1999]) laikā no 1989. līdz 1990. gadam. Tas ir veids, kā ar vienkāršām metodēm risināt dažādus sarežģītus uzdevumus ar tālvadības piekļuves palīdzību. WWW pamatā ir hiperteksts, kuru piedāvāja *Teds Nelsons* (ang. *Ted Nelson*) 1965. gadā (skat. ziņojumu [Nel1965]). *Hiperteksts* [Dat2004] – tas ir informācijas strukturēšanas veids, kas ļauj ievietot teksta dokumentā informāciju par citu dokumentu un datņu (vai citu dokumentu un datņu komponentu) atrašanās vietu ar iekšējo saišu palīdzību.

Eksistē trīs galvenie hipervides datubāzes nodrošināšanas līdzekļi [Dat2004]: *informācijas resursa unificētais lokators* (ang. *Uniform Resource Locator, URL*), *hiperteksta iezīmēšanas valoda* (ang. *Hypertext Markup Language, HTML*) (skat. [Ber1993]) un *hiperteksta pārraides protokols* (ang. *Hypertext Transfer Protocol, HTTP*) (skat. [Ber1992]).

Hipervides datubāze ir decentralizēta datubāze, kuru iespējams sadalīt mazos tīklos, parasti to sauc par *Intranetu*. Lietotāji ar pārlūkprogrammu mijiedarbojas ar šo hipervides datubāzi, katram mezglam ir savs serveris un vismaz viens URL. Katrs serveris parasti satur daudzus dokumentus, kas savienoti savu starpā ar saitēm, kuras sauc par hipersaitēm. Dokumentos iespējams ievietot saites uz dažādu tipa informāciju: teksts, video, skaņa, attēli utt. Noklikšķinot peli uz kādas no hipersaitēm, ar šī protokola starpniecību tiek atvērts attiecīgais dokuments neatkarīgi no tā, kur šis dokuments tīklā ir izvietots.

Katra servera realizācija ir slēpta no pārlūkprogrammas, ar kuru mijiedarbojas parastais lietotājs. Serveris var glabāt visu dokumentus datnes veidā, vai arī serveris strādā, izmantojot relāciju DBVS.

1.2.2.4. Vienranga datubāzes

Terminu P2P piedāvāja *Parbauels Johansons* (ang. *Parbawell Yohanson*), kad 1984. gadā piedalījās IBM projektā „*Advanced Peer to Peer Networking*”. *Vienranga datubāze* (ang. *Peer-to-Peer Database, P2P*) tas ir dalīta vienranga dalībnieku lietojumprogrammas arhitektūra, kad visi uzdevumi un darba noslodzes ir sadalīti starp dalībniekiem. Tāda tīkla dalībniekus sauc par *mezgliem* (ang. *peer*). P2P popularizācija sākās ar datņu apmaiņas sistēmas izplatīšanos. Viena no pirmajām tāda tipa sistēmām bija Napster.

Eksistē dažādas P2P modifikācijas, tipiska realizācija ir dalītā, bet eksistē arī centralizētas variācijas. Piemēram, ar P2P veidu klasificēšanu iespējams iepazīties [Lua2005] pētījumā. Detalizētāku aprakstu par P2P koncepciju iespējams apskatīt [Sch2002, Buf2008, Sch2005] informācijas avotos.

P2P mezgli ļauj savu datorsistēmas resursa daļu izmantot P2P darbu nodrošināšanai; tādus resursus kā procesoru jauda, disku vieta, tīkla joslas platums. Pats P2P mezgls var regulēt to, cik lielu resursu iespējams iedalīt kopējām vajadzībām. Katrs P2P mezgls izpilda divas funkcijas: serveris (piegādātājs) un klients (patērētājs). Centralizēta vadība netiek izmantota, visi P2P mezgli ir vienranga dalībnieki.

1.2.2.5. Sensoru tīkls

Mūsdienās izmanto terminu sensoru tīkls, bet uzskata, ka tas ir *bezvadu sensoru tīkls* (ang. *Wireless Sensor Network, WSN*). Galvenais darbības elements ir *sensors*.

WSN – tie ir savstarpēji saistīti ģeogrāfiski sadalīti autonomi sensori, kuri tiek izmantoti lai saņemtu informāciju no ārējās vides par temperatūru, skaņu, vibrāciju, kustību utt. Katrs autonomais sensors ir nodrošināts ar bezvadu datu saņemšanas un nodošanas aparāturu, caur kuru nodrošina datu apmaiņu ar blakus sensoriem. WSN, izmantojot sensorus savienojumu, nodrošina sensoru tīklu no dažiem metriem līdz vairākiem kilometriem.

Detalizētāk par WSN un par izmantotajiem datu pārraides protokoliem iespējams lasīt [Dar2010, Soh2007] informācijas avotos.

WSN tēma kļuva populāra 90. gados, var uzskatīt, ka šī tēma diezgan jauna. Tas ir saistīts ar to, ka tikai no tā laika aparatūras risinājumu izmaksas ir krietni samazinājušās. Pirmie pētījumi bija saistīti ar militāriem mērķiem. Mūsdienās WSN izplatījās vairākās sfērās. Terminam *vides monitorings* tiek izmantots atsevišķs termins *ārējās vides sensoru tīkls* (ang. *Environment Sensor Networks, ESN*), *ārējās vides sensoru tīkls* arī nodrošina okeāna monitoringu (piemēram, [Vas2005]), vulkāniem (piemēram, zemestrīces detektēšana), mežiem (piemēram, meža ugunsgrēku atklāšana), pilsētās (piemēram, gaisa piesārņojuma monitorings) utt. *Ražošanas monitorings* tiek izmantots ražošanā, piemēram, atjaunojamās enerģijas ražošanā detektē ūdenslīmeni vai saules enerģijas stiprumu.

WSN galvenie raksturojumi ir dalītā DVS, kur katrs sensors ir autonoma sistēma. WSN bez pārbūves var pieslēgt jaunus sensorus un atslēgt vecos, viena sensora darbības pārtraukšana parasti būtiski neietekmē kopējo sistēmas darbību. Nav nepieciešama papildu infrastruktūra, sensoriem ir iebūvēts datu pieņemšanas/nodošanas bloks, ar kuru tiek nodrošināta komunikācija.

1.3. DVS atbilstība objektu individuālajās īpašībās sakņotu sistēmu kritērijiem

Iepriekšējās apakšnodaļās ir apskatītas vispazīstamākās DVS koncepcijas. Šajā apakšnodaļā ir apkopota informācija, lai pārbaudītu apskatītās DVS atbilstību objektu individuālajās īpašībās sakņotas sistēmas kritērijiem. Vienai DVS ir iespējamas daudzas realizācijas variācijas. Šajā nodaļā tiek apskatītas tipiskās realizācijas. Šīs nodaļas mērķis ir pārbaudīt DVS atbilstību objektu individuālajās īpašībās sakņotās sistēmas kritērijiem, kas ir apkopoti 1.2. tabulā. Ja neviena no apskatītajām DVS pilnībā neatbilst izvirzītajiem kritērijiem, tad rodas nepieciešamība pēc DVS koncepcijas izstrādes.

1.3.1. Monolītās datu vadības sistēmas

Pamatoties uz 1.2.1. nodaļas MDVS apskatu ir pārbaudīta MDVS atbilstība objektu individuālajās īpašībās sakņotām sistēmām. Pārbaude nodrošina atbilstību izvirzītajiem kritērijiem, kas ir aprakstīti 1.2. tabulā.

Mūsdienās trīs no apskatītajām MDVS (*izplātās datnes modelis, hierarhiskais modelis un tīkla modelis*) jau ir zaudējušas savu aktualitāti (fakts no [Sim1995]). Par šīm MDVS analīzes rezultāti ir pievienoti datu tabulai 1.3., bet tos komentēt nav nepieciešams.

Ir kritēriji, kurus nav iespējams nodrošināt ar MDVS, tie ir (i) *skaitļošanas resursu pārvaldības sadale un decentralizācija* un (iv) *lokālas datu piekļuves nodrošināšana*. Klasiskā teorija uzskata, ka MDVS pārvalda datus centralizēti, kas ir pretēji (i) kritērijiem. Un datu piekļuvei ir nepieciešams pieslēgties centralizētai datu krātuvei, kas neatbilst (iv) kritērijiem.

Mūsdienās visām populārajām MDVS ir nodrošināti tādi kritēriji kā (ii) *standartu izmantošana* un (viii) *datu konfidencialitāte, integritāte un pieejamības nodrošināšana*.

Kritēriju par (iii) *fizisko objektu daudzveidību atbalstīšanu* pilnībā nodrošina *objektorientētais modelis*. *Objektu relāciju modelis* nodrošina šo kritēriju daļēji. Visas pārējās MDVS pilnībā nodrošina atbilstību šim kritērijam, ja tiek nodrošināta iespēja pielietot šo MDVS ar *objektorientētu modeli*.

Kritērijs (v) *par autonomu darbību* MDVS kontekstā ir pretrunīgs. Autonomai darbībai ir nepieciešams nodrošināt datu pieejamību lokāli, kas netiek nodrošināts. Tomēr pašus DVS uzbūves principus iespējams izmantot autonomai darbībai, tas ir atkarīgs no konkrētas realizācijas.

Divi no kritērijiem ir cieši saistīti viens ar otru, tas ir (vi) *mazu risinājumu ieviešanas ekonomiskais izdevīgums* un (vii) *pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale*. Centralizētas datu glabāšanas un pārvaldības nodrošināšana prasa lielas sākotnējās investīcijas. Tas ir viens no iemesliem, kāpēc (vi) kritērijs parasti netiek nodrošināts. Ir iespējami izņēmumi, tās ir MDVS, kas ir domātas mobilajām ierīcēm un mazām datubāzēm. (vii) kritēriju ieviešana prasa kopēju risinājuma sākuma datu glabāšanas un pārvaldības uzbūvi. Viena no priekšrocībām ir samazināts lietojumprogrammas izstrādāšanas cikls ar nosacījumu, ka netiek mainīta datu glabāšanas struktūra.

MDVS atbilstība objektu individuālajās īpašībās sakņotām sistēmām ir apkopotas 1.3. tabulā. Tabulā ir izmantotas vērtības, kritēriju atbilstībai: 0 – neatbilst, 0/1 – daļēji atbilst un 1 – atbilst.

1.3. tabula

MDVS kritēriju atbilstības salīdzinājuma tabula

(I - skaitļošanas resursu pārvaldības sadale un decentralizācija; II - standartu izmantošana; III - fizisko objektu daudzveidības atbalstīšana; IV - lokālas datu piekļuves nodrošināšana; V - autonoma darbība; VI - nelielu risinājumu ieviešanas ekonomiskais izdevīgums; VII - pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale; VIII - datu konfidencialitātes, integritātes un pieejamības nodrošināšana)

MDVS \ Kritērijs	I	II	III	IV	V	VI	VII	VIII
Izplātās datnes modelis	0	0	0	0/1	0	0/1	0	0
Hierarhiskais modelis	0	0/1	0	0	0	0/1	0/1	0
Tīkla modelis	0	0/1	0	0	0	0/1	0/1	0
Relāciju modelis	0	1	0	0	0/1	0/1	0/1	1

Objektu orientēts modelis	0	1	1	0	0/1	0/1	0/1	1
Objektu relāciju modelis	0	1	0/1	0	0/1	0/1	0/1	1
Aktīvas datubāzes	0	1	0/1	0	0/1	0/1	0/1	1
Datu noliktavas un citi specifiskie datu vadības sistēmas	0	1	0/1	0	0/1	0/1	0/1	1

1.3.2. Decentralizētas datu vadības sistēmas

Pamatoties uz 1.2.2. nodaļas DDVS apskatu, ir pārbaudīta DDVS atbilstība objektu individuālajās īpašībās sakņotām sistēmām. Pārbaude tika veikta, lai nodrošinātu 1.2. tabulā definēto kritēriju atbilstību.

Pirmais kritērijs par *(i) skaitļošanas resursu pārvaldības sadali un decentralizāciju* ir nodrošināts, pamatojoties uz decentralizētu datu vadības principu kā tādu. Tomēr *dalītas datubāzes* un *sensoru tīkli* atbilst kritērijiem daļēji. *Dalītās datubāzes* tiek raksturotas ar MDVS un DDVS principiem vienlaicīgi. *Sensoru tīklos* katrs atsevišķais mezgls saņem uzdevumu no blakus mezgliem. Ir iespējams, ka rezultāti ir apvienoti un apstrādāti tikai vienā mezglā.

Mūsdienās visām populārajām DDVS ir nodrošināts *(ii) kritērijs par standartu izmantošanu*. Cits kritērijs *(iii) par fizisko objektu daudzveidību atbalstīšanu* pilnvērtīgi var nodrošināt *daudzaģentu sistēmas*. *Dalītām datubāzēm* un *hipervides datubāzēm* tas ir iespējams, ja šīs DDVS tiek realizētas ar *objektorientētu pieeju*.

Divi kritēriji *(iv) lokālas datu piekļuves nodrošināšana* un *(v) autonoma darbība* ir cieši saistīti. Tie pilnībā atbilst kritērijiem *vairākaģentu sistēmās*. *Dalītām datubāzēm* to var nodrošināt, ja datu apmaiņa notiek, izmantojot mezglus. Visiem mezgliem jābūt realizētiem ar DDVS principu palīdzību. Tādām DDVS kā *hipervides datubāzes* un *vienranga datubāzes* atbilstība šiem kritērijiem ir atkarīga no konkrētas realizācijas. *Sensoru tīklam* dati kā tādi netiek glabāti vispār. Tie tiek pieprasīti no pamatmezgla un tiek nodrošināta datu pārraide līdz šim mezglam. Ja nav saites ar pamatmezglu, viss tīkls strādā bez mērķa.

Divus sekojošos kritērijus iespējams apskatīt kopā. Tie ir *(vi) nelielu risinājumu ieviešanas ekonomiskais izdevīgums* un *(vii) pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale*. Tādām DDVS kā *dalītas datubāzes*, *daudzaģentu sistēmas* un *hipervides datubāzes* ir nepieciešams pilns izstrādes cikls līdz gatava risinājuma ieviešanas. Tas savukārt apgrūtina realizēt mazus risinājumus, vai arī lieli risinājumi jāievieš pakāpeniski. *Vienranga datubāzēm* un *sensoru tīkliem* galvenais programmatūras un aparatūras nodrošinājums jau eksistē, un pielietošana ir vienkāršāka. Piemēram, *sensoru tīklam* pamatelementi ir sensori, kurus iespējams nodrošināt vēlāk un papildināt eksistējošu risinājumu.

Kritērijs (viii) *datu konfidencialitātes, integritātes un pieejamības nodrošināšana* ir sarežģīti realizējams priekš DDVS. Vienīgais kas atbilst šim kritērijam ir *dalītas datubāzes*, MDVS un DDVS principu apvienošanas dēļ. Visām pārējām DDVS ir ierobežojumi datu drošību mehānismu izmantošanai, datu integritātes kontrolei utt.

DDVS atbilstība objektu individuālajās īpašībās sakņotām sistēmām ir apkopotas 1.4. tabulā. Tabulā ir izmantotas vērtības kas atbilst izvirzītajiem, kritērijiem: 0 – neatbilst, 0/1 – daļēji atbilst un 1 – atbilst.

1.4. tabula

DDVS kritēriju atbilstību salīdzinājuma tabula

(I – skaitļošanas resursu pārvaldības sadale un decentralizācija; II – standartu izmantošana; III – fizisko objektu daudzveidības atbalstīšana; IV – lokālas datu piekļuves nodrošināšana; V – autonoma darbība; VI – nelielu risinājumu ieviešanas ekonomiskais izdevīgums; VII – pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale; VIII – datu konfidencialitātes, integritātes un pieejamības nodrošināšana)

DDVS \ Kritērijs	I	II	III	IV	V	VI	VII	VIII
Dalītas datubāzes	0/1	1	0/1	0/1	0/1	0/1	0/1	1
Daudzaģentu sistēmas	1	1	1	1	1	0/1	0/1	0/1
Hipervides datubāzes	1	1	0/1	0/1	0/1	0/1	0/1	0/1
Vienranga datubāzes	1	1	0	0/1	0/1	1	1	0/1
Sensoru tīkls	0/1	1	0	0	0	1	1	0/1

1.4. Kopsavilkums un secinājumi

Šajā promocijas darba nodaļā ir identificēti kritēriji objektu individuālajās īpašībās sakņotām sistēmām un veikta vispazīstamāko MDVS un DDVS koncepciju analīze to atbilstībai izvirzītajiem kritērijiem. Kopējais nodaļas mērķis ir atrast DVS koncepciju, kas atbilst izvirzītajiem kritērijiem un noteikt, vai no šīs analīzes izriet, ka nepieciešama jauna DVS koncepcijas izstrāde.

Nodaļā paveiktais:

- ir identificētas IS galvenās attīstības tendences, analizējot DVS nodrošināšanas mehānismus;
- ir pētītas IS galvenās attīstības tendences un objektu individuālajās īpašībās sakņotas sistēmas, identificējot kritērijus objektu individuālajās īpašībās sakņotām sistēmām;
- ir pārbaudītas deviņas MDVS un piecas DDVS koncepcijas izvirzīto kritēriju atbilstībai;
- ir identificēti labākie praktiskie risinājumi, kurus iespējams izmantot DVS koncepcijas izstrādē.

Sasniegtie rezultāti ir šādi:

- ir definētas četras IS attīstības pamattendences: *skaitļošanas resursu pārvaldības sadale un decentralizācija, IS komponentu nevienmērīgums, standartu izplatīšana, reālās pasaules modelēšana informācijas sistēmās;*
- ir izvirzīti astoņas galvenie kritēriji objektu individuālajās īpašībās sakņotām sistēmām: (i) *skaitļošanas resursu pārvaldības sadale un decentralizācija, (ii) standartu izmantošana, (iii) fizisko objektu daudzveidības atbalstīšana, (iv) lokālas datu piekļuves nodrošināšana, (v) autonoma darbība, (vi) nelielu risinājumu ieviešanas ekonomiskais izdevīgums, (vii) pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale, (viii) datu konfidencialitātes, integritātes un pieejamības nodrošināšana;*
- ir pārbaudītas deviņas monolītas un piecas decentralizētas datu vadības sistēmas koncepcijas definēto OIĪSS kritēriju atbilstībai;
- ir izvirzīti trīs labākie praktiskie risinājumi, kuri tiks izmantoti OIĪSS koncepcijas realizācijai: *objektorientēta pieeja, daudzagentu sistēmas datu sadalīšanas principi un aktīvu datubāzu triggeru darbības principi.*

Galvenie secinājumi ir šādi:

- Pamatojoties uz 1.2 tabulā apkopotajiem kritērijiem objektu individuālajās īpašībās sakņotām sistēmām:
 - Pēc MDVS apskata ir definēts, ka pilnībā ir iespējams sasniegt trīs kritērijus: (ii) *standartu izmantošana, (iii) fizisko objektu daudzveidību atbalstīšana un (viii) datu konfidencialitātes, integritātes un pieejamības nodrošināšana.* MDVS nevar nodrošināt atbilstību diviem kritērijiem uzbūves principu dēļ, t.i. (i) *skaitļošanas resursu pārvaldības sadali un decentralizāciju un (iv) lokālas datu piekļuves nodrošināšanu.*
 - Pēc DDVS analīzes noteikts, ka pilnībā ir iespējams sasniegt vienu kritēriju (ii) *standartu izmantošana.* Visi pārējie kritēriji ir sasniegti daļēji. DDVS parasti izpilda specializētus uzdevumus un nodrošina atbilstību kritērijiem, risinot savas sfēras uzdevumus. Piemēram, uz *vienranga datubāzes* pamata ir izveidota programmatūra datu apmaiņai starp dažādiem lietotājiem ar tīkla starpniecību.
- Kritērijs (iii) *fizisko objektu daudzveidību atbalstīšana* pamatā ir nodrošināts ar *objektorientētu pieeju.* Piemēram, to parasti nodrošina, izmantojot *objektorientētu modeli.*

- Ir noskaidrots, ka no apskatītajām klasiskajām MDVS un DDVS neviena no tām pilnībā neatbilst izvirzītajiem OIĪSS kritērijiem. Vistuvāk no MDVS ir *objektu orientētais modelis* un *aktīvās datubāzes*. No DDVS tās ir *daudzaģentu sistēmas*.
- kritērijs (*i*) *skaitļošanas resursu pārvaldības sadale un decentralizācija* ir pretrunā ar MDVS uzbūves principiem. DVS, kura atbilst šim kritērijam, jābūt būvētai pēc decentralizētas DVS principiem.
- Apskatītās DVS atbilst izvirzītajiem OIĪSS kritērijiem tikai daļēji. Radās nepieciešamība pēc objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcijas izstrādes, kas atbilstu kritērijiem, kas ir izvirzīti objektu individuālajās īpašībās sakņotām decentralizētām datu vadības sistēmām.

Šīs nodaļas galvenais teorētiskais rezultāts ir izvirzītie OIĪSS kritēriji. Uz šo kritēriju pamata ir plānots veikt objektu individuālajās īpašībās sakņotas sistēmas decentralizētas datu vadības sistēmas izstrādi. Pēc koncepcijas izstrādes pārbaudīt atbilstību OIĪSS izvirzītajiem kritērijiem, lai pārliecinātos, ka promocijas darba mērķis ir sasniegts.

2. OIĪSS izmantoto tehnoloģiju izpēte

Jebkurš IS risinājums balstās uz konkrētām tehnoloģijām. Šīs nodaļas mērķis ir identificēt tehnoloģijas, kuras plānots izmantot objektu individuālajās īpašībās sakņotām sistēmām. Pamatojoties uz jaunās koncepcijas definētajiem kritērijiem, ir izvirzīti tehnoloģiju salīdzināšanas parametri, kuri ir nepieciešami, lai nodrošinātu izvirzītos OIĪSS kritērijus. Izvirzītie parametri ar apkopotajām vērtībām tiks izmantoti gala risinājuma piemērotāko tehnoloģiju izvēlei.

1.1.3. apakšnodaļā bija definēts uzdevumu klāsts, kas tiek apvienots ar objektu individuālajās īpašībās sakņotu sistēmu izmantošanu. 2.1. apakšnodaļā ir apskatīti piemēri no 1.1.3. apakšnodaļas, lai definētu tipiskas izmantotās tehnoloģijas. Pēc tehnoloģiju izmantošanas nolūkiem ir klasificētas tehnoloģiju grupas. 2.2. apakšnodaļā katrai no grupām ir definētas un apskatītas tehnoloģijas. Tehnoloģijas savā darbā izmanto datu pārraides standartus, un 2.3. nodaļā ir apskatīti tipiskie datu apmaiņas formāti. Nodaļas beigās ir dots kopsavilkums un secinājumi.

2.1. Piemēru apskats tehnoloģiju identificēšanai

Tiek ņemti piemēri no objektu individuālajās īpašībās sakņotajām sistēmām, kuri ir bija apskatīti 1.1.3. nodaļā. Uz šo piemēru pamata ir identificētas tipiskas tehnoloģijas, kas ir izmantotas risinājumu iedarbināšanai.

Autonomie kooperatīvie loģistikas procesi

Jau vairākus gadus pamattehnoloģija loģistikā ir svītrkodu tehnoloģija un tai līdzīgas tehnoloģijas (QR kodi utt.). Tā ir lēta tehnoloģija, bet svītrkodu nolasīšanai nepieciešama tieša redzamība un ir pārāk lēns pats nolasīšanas process. Tāpēc alternatīvs risinājums ir *radio frekvenču identifikācijas* (ang. *Radio Frequency Identification, RFID*) tehnoloģija. RFID tehnoloģija ir dārgāka, bet ar tās palīdzību ir iespējams nolasīt daudzus objektus vienlaicīgi no attāluma, bez tiešas redzamības un ātri (līdz 200 objektiem sekundē). Daudzos informācijas avotos [Lah2005, Hei2005, Bau2005, Jun2004, Vaj2003, Hed2007] detalizēti apskati svītrkodu un RFID tehnoloģiju lietderīgas izmantošanas gadījumi.

RFID tehnoloģiju bieži izmanto kopā ar pozicionēšanas tehnoloģijām, tādām kā *globāla vietnoteices sistēma* (ang. *Global Positioning System, GPS*). Informācija par atrašanās vietu vai nu tiek glabāta pašā *RFID* ierīcē, vai tiek sūtīta uz citu sistēmu. Savienošanai ar citām sistēmām parasti tiek izmantotas bezvadu savienošanas tehnoloģijas: *Wi-Fi* (bezvadu tīklu savienošana uz IEEE 802.11 standarta bāzes), *globālo mobilo sakaru sistēma* (ang. *Global System for Mobile Communications, GSM*), *vispārējais pakešu radio pakalpojums*

(ang. *General Packet Radio Service, GPRS*), *infrasarkanais savienojums* (ang. *Infrared Data Association, IrDA*) un *satelītu komunikācijas*.

Tīršanas, remonta un apkopes servisi

Tīršanas, remonta un apkopes servisiem ir svarīgi identificēt objektu. Automatizācijas risinājumi parasti identificē objektu ar svītrkodu tehnoloģiju. Pēc svītrkoda iespējams identificēt precīzi un atrast datubāzē nepieciešamo apstrādes algoritmu. Vēl uz RFID tehnoloģijas pamata ir risinājumi [Fer2010], kad uz laiku tiek ievietotas RFID ierīces, kurās iekšā ir ierakstīts identifikators. Pēc identifikatora sistēma pati nosaka, kā apstrādāt precīzi. Tā kā pašas preces var tikt apstrādātas specifiskos apstākļos (piemēram, paaugstināta temperatūra), tiek izmantotas RFID ierīces, kas pasargā tās no šiem apstākļiem. RFID tehnoloģijai tiek izmantotas RFID skenēšanas ierīces, kas ir savienotas ar citām sistēmām, izmantojot *Wi-Fi* vai *lokālo tīklu* (ang. *Local Area Network, LAN*).

Viedās ēkas

Viedās ēkas risinājumi (piemēram, skat. [Cha2008]) izmanto dažādu tipu *sensorus* (piemēram, audio, video, temperatūras, mitruma utt.). Viedās mājas darbības objektu savienojums tiek nodrošināts ar *bezvadu tehnoloģijām* (*Wi-Fi, GPRS, Bluetooth* utt.) vai *datortīklu*. Kad risinājumam nepieciešams nodrošināt objektu identificēšanu, to nodrošina ar sensoriem vai RFID tehnoloģiju.

Monitoringa, informēšanas un regulēšanas sistēmas

Monitoringa, informēšanas un regulēšanas sistēmu viena no pamatfunkcijām ir objektu identificēšana. Tāpēc tiek izmantotas identificēšanas tehnoloģijas: dažāda veida sensori (pirkstu un acu skanētāji, video, audio utt.). Gadījumā, kad nepieciešams identificēt vairākus objektus vienlaicīgi un bez tiešās redzamības nodrošināšanas, tiek izmantota RFID tehnoloģija [Cha2008]. Mūsdienās jau plašāk izmanto *tuvas darbības lauka komunikācijas* (ang. *Near Field Communication, NFS*) tehnoloģiju, kas ietver RFID tehnoloģijas priekšrocības, un ar katru gadu arvien biežāk šo tehnoloģiju iebūvē dažādās mobilajās ierīcēs (mobilie telefoni, piezīmjdatori utt.). Darbības objektu savienojums tiek nodrošināts ar *bezvadu tehnoloģijām* (*Wi-Fi, GPRS* utt.) vai *LAN*.

Kopsavilkums

Tika apskatīti četri piemēri, lai identificētu tipiskas tehnoloģijas, kas ir izmantotas objektu individuālajās īpašībās sakņotām sistēmām. Šīs tehnoloģijas tiks apskatītas detalizētāk, lai pēc tam identificētu parametrus, kas ir svarīgi jaunajai koncepcijai. Ir identificētas sekojošas tehnoloģijas (kas dotas ar izmantošanas aprakstiem):

- *Svītrkods* – tā ir tehnoloģija, kas tiek izmantota fizisko objektu identificēšanai, ir gadījumi kad birkā kopā ar identifikatoru tiek glabāta papildu informācija par objektu (piemēram, partijas numurs, ražotāja identifikators utt.).
- *RFID* – tā ir tehnoloģija, kas izmantota tādiem pašiem nolūkiem kā svītrkodu tehnoloģija. Parasti RFID sauc par „svītrkoda nākamās paaudzes tehnoloģiju”.
- *NFC* – tā ir RFID tehnoloģijas modifikācija, kas bieži ir iebūvēta mobilajās ierīcēs, viens no tās tipiskās izmantošanas piemēriem ir ātri maksājumi un lietotāju identificēšana.
- *Lokālais tīkls* – tā ir tehnoloģija, kas tiek izmantota datu pārraidei, tīkla nodrošinājumam nepieciešams uzbūvēt struktūru. Ar Internet izplatīšanos lokālais tīkls kļūst par populāru un pieejamu tehnoloģiju.
- *Bluetooth, IrDA* – datu pārraides tehnoloģijas, kuras parasti tiek izmantotas, ja nepieciešami savienojumi ar mobilajām ierīcēm.
- *GSM, GPRS* – mobilo sakaru standarts un mobilais datu pārraides serviss, kurš tiek izmantots datu pārraidei, tagad tas ir bieži pieejams un tiek lietots tur, kur nav pieejami uz vietas citi datu pārraides standarti.
- *Wi-Fi, WiMAX* – bezvadu datu pārraides tehnoloģijas, kuras mūsdienās ir izplatītas un to infrastruktūra tiek plaši izmantota.
- *Satelīti* – viens no populārākajiem šīs tehnoloģijas piemēriem ir GPS, kas ir piemērota pozicionēšanai. Līdzīgas tehnoloģijas, kas strādā pēc tāda paša principa ir Galileo, KOMPASS, GLONASS un citas [Bød2006].

2.2. Tehnoloģijas

Objektu individuālajās īpašībās sakņota decentralizētas datu vadības sistēmas koncepcijai jāatbilst 1.2. tabulā definētajiem kritērijiem. Tehnoloģijas apskats tiek veikts pamatojoties uz tehnoloģiju salīdzināšanas parametriem, kas izvirzīti šo kritēriju pamatā. Pamatojoties uz šiem parametriem, ir veikts tehnoloģiju apskats.

2.2.1. Tehnoloģiju salīdzināšanas parametri

Pamatojoties uz tādiem kritērijiem kā (i) *skaitļošanas resursu pārvaldības sadale un decentralizācija*, (ii) *standartu izmantošana*, (iii) *fizisko objektu daudzveidības atbalstīšana*

un (v) *autonoma darbība*, tiek piedāvāts sadalīt visas tehnoloģijas grupās pēc tehnoloģiju izmantošanas mērķiem. Tehnoloģiju salīdzināšanas parametra (i) *tehnoloģiju grupa* vērtības ir dotas 2.1. tabulā.

2.1. tabula

Tehnoloģiju grupas pēc izmantošanas mērķiem

Nr.	Nosaukums	Apraksts
1.	<i>Objektu identificēšanas tehnoloģijas</i>	Šīs grupas tehnoloģijas pamatmērķis ir identificēt objektu. Parasti tas ir fizisks objekts. Ietilpstošās definētās tehnoloģijas ir: <i>svītrkods, RFID, NFC</i> utt.
2.	<i>Datu pārraides tehnoloģijas</i>	Tās ir tehnoloģijas, kuru pamatmērķis ir pārvietot datus no vienas vietas uz citu. Šeit ietilpstošās definētās tehnoloģijas ir: <i>Wi-Fi, WiMAX, Bluetooth, GPRS, IrDA, satelīti</i> utt.
3.	<i>Uz sensoriem bāzētas tehnoloģijas</i>	Uz šo grupu attiecas tehnoloģijas, kas parasti tiek izmantotas kopā ar 1. un 2. grupas tehnoloģijām un ir vajadzīgas, lai paplašinātu standartu tehnoloģijas funkcionalitāti. Parasti tie ir dažādu veidu sensori: <i>globālas pozicionēšanas sistēma</i> (ang. <i>Global Positioning System, GPS</i>), termometrs utt.

Viena tehnoloģija var atbilst vienai grupai vai divām grupām. Piemēram, Bluetooth tehnoloģiju galvenokārt izmanto datu pārraides mērķiem un šī tehnoloģija pieder 2. grupai. Ir gadījumi (piemēram, no [Che2006, Hal2003]), kad uz šīs tehnoloģijas pamata ir izveidotas objektu identifikācijas birkas, tad šī tehnoloģija pieder 1. grupai.

(iv) kritērijam par *lokālas datu piekļuves nodrošināšanu* ir jādefinē tehnoloģiju salīdzināšanas parametri, kas attiecas uz apraksta tehnoloģiju darba parametriem: (ii) *darba rādiusu*, (iii) *datu pārraides ātrumu*, (iv) *darba frekvenci* un (v) *tiešās redzamības nepieciešamības*.

Divi no kritērijiem (vi) *nelielu risinājumu ieviešanas ekonomiskais izdevīgums* un (vii) *pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale* ietekmē izmaksas aprēķinu. Šīs nepieciešamības dēļ ir ieviesti nākamie tehnoloģiju salīdzināšanas parametri (vi) *ieviešanas izmaksas* un (vii) *tehnoloģiju iedarbināšanas mehānismu sarežģītība*. Īpaša uzmanība ir pievērsta (viii) *datu konfidencialitātes, integritātes un pieejamības nodrošināšanas jautājumam*. Šis kritērijs ir atspoguļots (viii) *datu aizsardzības* tehnoloģiju salīdzināšanas parametrā.

Izvirzīto tehnoloģiju salīdzināšanas parametri ir aprakstīti 2.2. tabulā.

Tehnoloģiju salīdzināšanas parametri

Nr.	Nosaukums	Apraksts
I.	<i>Tehnoloģiju grupa</i>	Definētas trīs tehnoloģiju izmantošanas grupas, kas attiecas uz izmantošanas mērķi: <i>objektu identificēšana</i> (1. grupa), <i>datu pārraidei</i> (2. grupa) un <i>uz sensoriem bāzētas tehnoloģijas</i> (3. grupa). Viena tehnoloģija var vienlaicīgi piederēt dažādām grupām.
II.	<i>Darba rādiuss</i>	Tehnoloģijas maksimālais darba rādiuss starp divām ierīcēm.
III.	<i>Datu pārraides ātrums</i>	Cik ātri iespējams pārsūtīt datus no vienas ierīces uz otru. Tehnoloģijām, kas attiecas uz objektu identificēšanu, ir norādīts, cik daudz laika ir vajadzīgs viena objekta identificēšanai.
IV.	<i>Darba frekvence</i>	Lielākā daļa no tehnoloģijām savā darbā izmanto konkrētu frekvenci. Svarīgi tās zināt, jo katrā valstī ir savi ierobežojumi, un ir tehnoloģijas, kuras aizliegts izmantot specifisko frekvenču izmantošanas dēļ.
V.	<i>Tiešās redzamības nepieciešamība</i>	Viens no būtiskākajiem ierobežojumiem, kas attiecas uz tehnoloģijām.
VI.	<i>Ieviešanas izmaksas</i>	Tiek sniegtas relatīvās izmaksas relatīvās vienībās, kur pirmā vienība nozīmē infrastruktūras izmaksas, otrā vienība – termināla daļas izmaksas un trešā – birku izmaksas (attiecība uz 1. grupu). Vienības izmaksas ir robežās no 0 līdz 9 (jo augstāk, jo dārgāk).
VII.	<i>Tehnoloģiju iedarbināšanas mehānismu sarežģītība</i>	Tā ir uz konkrētas tehnoloģijas balstītu ierīču darbības principu nodrošināšanas sarežģītība. Piemēram, definēts, ka viena no sarežģītākajām tehnoloģijām ir satelītu tehnoloģija, tās iedarbināšanai nepieciešams nodrošināt satelītus. Pašas vērtības ir definētas no tehnoloģijas apraksta un ekspertu viedokļiem šajā jomā.
VIII.	<i>Datu aizsardzība</i>	Tiek definēts, kāda līmeņa datu drošības un konfidencialitātes atbalsts ir nodrošināts. Tiek definētas divas pazīmes: <i>programmatūras līmenis</i> (ang. <i>Software, SW</i>) un <i>aparātūras līmenī</i> (ang. <i>Hardware, HW</i>).

2.2.2. Tehnoloģiju apskats

Svītrkods

Svītrkods [Lah2005] (ang. *Barcode*) tā ir shēma, uz kuras ir drukāti simboli, kas prezentē informāciju teksta veidā. Drukātie simboli pamatā sastāv no vertikālām svītrām, atstarpēm, kvadrātiem un punktiem.

Metodi, kas šifrē burtu un ciparu pazīmes, izmantojot šos elementu simbolus apdrukās, sauc par *simbologiju*. Divās simbologijās var lietot vienādus vai atšķirīgus simbolu elementus, šifrējot tādus pašus raksturīgus nosacījumus.

Parasti, svītrkods prezentē informāciju teksta veidā, tā pat lielākā daļa no simbologijas tiek balstīta tieši uz teksta informācijas glabāšanu. Pēc nepieciešamības šīs simbologijas ir iespējams pārvietot uz baitu informācijas glabātuvī.

Daži simbologijas raksturojumi ir sekojoši [Lah2005]:

- *Kodējuma tehnika* – simbologija ar labāku kodējuma tehniku nodrošina efektīvu simbologijas atpazīšanas iespēju gadījumā, kad ir sliktas kvalitātes svītrkods.
- *Simbolu blīvums* – simbologija, kas piedāvā labāku simbolu blīvumu, tādejādi spējot attēlot maksimāli vairāk teksta informācijas uz viena fiziskā laukuma;
- *Kļūdu pārbaudes tehnikas* – simbologija, kas piedāvā labākas kļūdu pārbaudes tehnikas, nodrošina datu korektu lasīšanu arī tādā gadījumā, kad daži no simbolu komponentiem ir bojāti vai pazaudēti.

Statistika ir apkopota, izmantojot vairākus informācijas avotus, kuros norādīts, (piemēram, [Bro2007, Lah2005]), ka eksistē apmēram 270 simbologijas, un 50 no tām ir plaši izmantotas mūsdienās. Simbologijas iespējams sadalīt trīs kategorijās:

1. *Lineārā simbologija* – sastāv no vertikālām līnijām ar dažādu platumu, kas ir sadalītas ar kontrasta līnijām. Ar šo simbologiju parasti tiek kodēti cipari. Maksimālais pieļaujamais simbolu skaits, ko ir iespējams iekodēt, ir 50 simboli.
2. *Divu dimensiju simbologija* – tai ir lielāka datu glabāšanas iespēja nekā lineārajai simbologijai. Parasti divu dimensiju svītrkodu var definēt kā vairāku lineāru simbologiju apvienošana. Lineārais svītrkods tiek rakstīts viens zem otra.
3. *Iegravētā simbologija* (ang. *bumpy bar code*) – tas ir lineārais vai divu dimensiju svītrkods, kas ir iespiests uz virsmas. Šim svītrkodu skenerim nav svarīgs krāsu kontrasts starp svītrkodu līnijām, jo tas analizē iespaidumu. Parasti tāda veida svītrkodus izmanto gadījumos, kad citas kategorijas svītrkodi ir viegli sabojājami.

2.1. attēlā ir atspoguļotas trīs kategoriju svītrkodu piemēri.



2.1. attēls. Trīs kategoriju svītrkodu piemēri

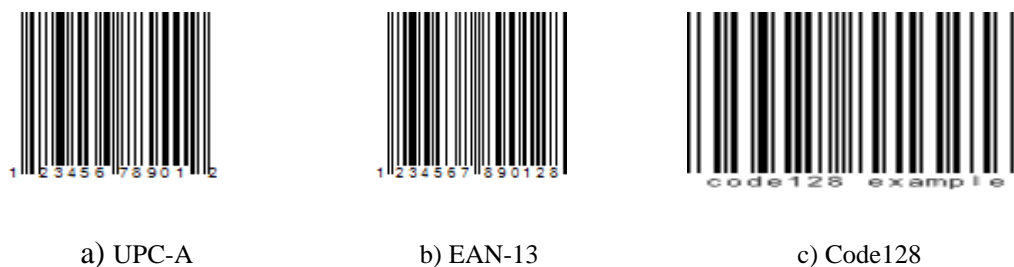
Nākamajās apakšnodaļās ir aprakstītas lineārās un divu dimensiju simbologijas. Tā kā iegravētās simbologijas kategorijai tiek izmantotas iepriekšējo divu kategoriju simbologijas, tad tai netiek piemērots atsevišķs apraksts.

Lineārie

Lineāro svītrkodu vēsture sākās 1948. gadā Amerikā. Svītrkoda principu izgudrotājs misters *Vudlands* (ang. *Norman Joseph Woodland*) ir teicis, ka doma radās no *morse koda* idejas. Pirmais patents tika reģistrēts jau 1949. gadā.

Svītrkodu radīšanas iemesls izskaidrojams ar to, ka bija nepieciešams izveidot iespēju automātiski pārbaudīt preces atrašanās vietu. Pati tehnoloģija tiek izmantota dažādu citu uzdevumu risināšanai, bet komerciāli, vispirms, sāka izplatīties tieši preču automātiskai atpazīšanai tirdzniecībā un loģistikā.

2.2. attēlā ir attēloti tipiskie lineārie svītrkodi.



2.2. attēls. Lineāro svītrkodu piemēri

Elektroniskais produkta kods

Īsta svītrkodu tehnoloģijas popularitāte sākās pēc 1973. gada, kad sākās *produkta elektroniskā koda* (ang. *Electronic Product Code, EPK*) izplatīšanās Ziemeļamerikā. Svītrkodu sāka izmantot loģistikas operācijās. Ar EPK kodu standartizēšanu nodarbojas *Elektronisko kodu padome* (ang. *Electronic Code Council*).

UPC-A (skat. 2.2.a. attēla) ir UPC modifikācija, kas nodrošina 12 ciparu glabāšanu: 1 cipars nosaka sistēmas numuru, 5 cipari nodrošina ražotāja kodu, vēl 5 cipari produkta kodu un viens ir pārbaudes cipars.

Eiropas Preču Numerācija

Eiropas preču numerācija (ang. *European Article Numbering, EAN*) tā ir eiropiešu UPC-A simbologijas ekvivalents. EAN-13 simbologijai, ja salīdzina to ar UPC-A, tika pievienots vēl viens cipars, kurš kopā ar divpadsmit cipariem nodrošina valsts kodu definēšanu.

EAN-13 izmantoto izdevējdarbības nozarē un to nodrošina *grāmatu starptautiskā standartnumerācija* (ang. *International Standard Book Number, ISBN*).

Vēl eksistēja citas līdzīgas simbologijas, tādas kā: *starptautiska preču numerācija* (ang. *International Article Numbering, IAN*), *japāņu preču numerācija* (ang. *Japan Article Numbering, JAN*) utt. 1977. gadā tika nodibināta GS1 Systems (skat. [GS12011]), kas ir bezpeļņas organizācija un apvieno vairākus standartu turētājus (t.s. ECC). Līdz ar to standartus sāka unificēt starptautiskā mērogā.

Code128

Tā ir augsta blīvuma svītrkoda simbologija. Piemērs dots 2.2.c attēlā. Ar šo simbologiju iespējams iekodēt jebkuru no 128 simboliem ANSCII kodējumā. Vispopulārākās ir trīs simbologijas modifikācijas: 128A (ANSCII simboli no 00 līdz 95: cipari, burti augstā reģistrā, vadības simboli, funkciju kodi), 128B (ANSCII simboli no 32 līdz 127: cipari, burti, speciālie simboli, funkciju kodi) un 128C (ANSCII simboli no 00 līdz 99: iekodē jebkurus divus ciparus vienā kodā un funkciju kods).

Code128 simbologijai ir definētas sešas sekcijas: *tukšā zona* (ang. *quiet zone*), *palaišanas simbols* (ang. *start character*), *iekodētie dati* (ang. *encoded data*), *pārbaudes simbols* (ang. *check character*), *apturēšanas simbols* (ang. *stop character*) un *tukšā zona* (ang. *quiet zone*). Izmantojot pārbaudes simbolu, tiek nodrošināts datu precizitāte.

Divu dimensiju

Lineārajiem svītrkodiem ir būtiski ierobežojumi no datu glabāšanas apjoma, datu ticamības, drošības un konfidencialitātes viedokļa. Nākamā svītrkodu pakāpes simbologijas tehnoloģija ir divu dimensiju svītrkodi. Klasiskie lineārie svītrkodi tiek iekodēti horizontāli, divu dimensiju svītrkodi tiek iekodēti vienlaicīgi horizontāli un vertikāli.

Viena no pirmajām divu dimensiju simbologijām bija *PDF417*. Tas ir vairāku lineāru svītrkodu apvienojums. Viens lineārais svītrkods atrodas zem otra. Nākamajā pakāpē ir mainījies svītrkodu nolasīšanas princips. Tie ir lasāmi ar video kamerām nevis ar lāzeru skeneriem, kuri lasa parastus svītrkodus, kas nozīmē, ka tos var nolasīt jebkurā orientācijā. Tad parādījās cita veida divu dimensiju svītrkodi, kas kodē informāciju šaha galda veidā ir/nav šūna. Mūsdienās visizplatītākās divu dimensiju simbologijas izmanto šo principu, tas ir *Data Matrix* un *QR Code*.

Data Matrix

Data Matrix [Ste2005] ir atbalstīts ar ISO standartu un *Elektronisko industriju asociācija* (ang. *Electronic Industries Association*) rekomendē to, lai marķētu mazus elektroniskus komponentus. ECC 200 Data Matrix kodi var glabāt vairāk nekā 3,116 ciparu, 2,335 burtu simbolus vai 1,555 baitu bināras informācijas, kuri glabāti masīvā ar 144 rindām un 144 kolonnām. Informācija ir kodēta digitāli. Te var izmantot kontrastu drukāšanu tieši uz objektiem bez nepieciešamības pēc speciālām etiķetēm. Tas piedāvā augstu informācijas blīvumu – augstāko no citiem divu dimensiju kodiem, kas nozīmē, ka ir iespējams novietot lielu informācijas apjomu nelielā platībā. Eksistē mērogojumi, kas nozīmē, ka ir iespējams drukāt un lasīt tos dažādos izmēros, kuri ir ierobežoti tikai ar izšķirtspēju, kura ir pieejama drukāšanas tehnikai un attēla apstrādes paņēmieniem. Augstā blīvuma dēļ, kurš raksturīgs Data Matrix kodiem, tie arī piedāvā iebūvēto kļūdu korekcijas paņēmieni, kas atļauj pilnīgi atjaunot ziņojumu, kurš ir iekodēts Data Matrix kodā, pat tad, ja ir sabojāta vai pazudusi 20% no simbola.

ECC 200 Data Matrix plaši lieto automobiļu, kosmosa, elektronikas, pusvadītāju, medicīnas ierīcēm un citās ražotāju ierīcēs. Data Matrix kodi parasti neaizstāj parastus svītrkodus, bet tiek lietoti tur, kur tradicionālie svītrkodi ir pārāk lieli, nenodrošina nepieciešamo datu blīvumu vai nav lasāmi.

QR Code

1994. gadā japāņu kompānija „*Danso-Wave*” [Den2010] izveidoja QR Code divu dimensijas simbolģiju. Simbolģija ātri kļuva populāra, pamatoties uz vieglu realizāciju un lielu datu glabāšanas apjomu.

QR Code datu ietilpība ir 7089 cipari, 4296 cipari ar burtiem, 2953 bināru dati. Ir iebūvēti kļūdu korekcijas mehānismi. Dati var tikt atjaunoti pat tad, ja simbols ir daļēji netīrs vai sabojāts. Maksimums 30% no koda var tikt atjaunoti. Nodrošināta viegla datu nolasīšanas iespēja, QR kodam ir kļūdu korekcijas iespēja un ir iespēja lasīt 360 grādu leņķi, lielā ātrumā. Tas tiek realizēts, pateicoties trijām speciālām pazīmēm, kuras atrodas trijos koda stūros. Šīs pazīmes garantē stabilu ātru nolasīšanu. Pats svītrkods var būt sadalīts vairākos datu apgabalos. Informācija, kas glabājas dažādos apgabalos, var tikt atjaunota kā viena vesela koda dati. Vienā QR Code apgabalā var būt līdz pat 16 iekļauto apgabalu.

Radio frekvences identifikācija

Radio frekvences identifikācija (ang. *Radio Frequency Identification, RFID*) – tas ir automātiska identifikācijas metode, kas balstās uz datu glabāšanu un datu attālinātu saņemšanu, izmantojot ierīces ar nosaukumu RFID birkas. RFID birku iespējams novietot uz

fiziska objekta, iebūvēt fiziskā objektā vai izveidot mijiedarbību ar citiem fiziskiem objektiem (piemēram, sensors).

Viens no pirmajiem darbiem par RFID bija raksts [Sto1948], kuru uzrakstīja *Harijs Stokmans* (ang. *Harry Stockman*) un kurš tika publicēts 1948. gadā. 50. gados tika veikta papildus RFID izpēte pēc veiksmīgas šīs tehnikas attīstības 30. un 40. gados. Tika izpētītas dažādas tehnoloģijas līdzīgas RFID, piemēram, tālu distanču birku sistēmas "identifikācija, draugs vai ienaids" aviācijas spēkiem. *D.B. Harijs* izveidoja [Har1960] patentu un mainīja tehnoloģijas statusu, un RFID kļuva par realitāti. 70. gados izstrādātāji, izgudrotāji, kompānijas, akadēmiskās iestādes un valdību laboratorijas aktīvi strādāja pie RFID tehnoloģijām, bet ievērojami panākumi tika sasniegti pētniecības laboratorijās un akadēmiskajās iestādēs, piemēram, *Los Alamos Scientific Laboratory*, *Northwestern University* un *Microwave Institute Foundation* Zviedrijā. Pirmā nozīmīgā izstrāde bija [Koe1975] pētījums.

Pirmais komerciālais RFID pielietojums bija XX gadsimta 60. gados, kad ar RFID moduļu starpniecību tika atzīmētas mantas pašapkalpošanās lielveikalos. Šis tehnoloģijas pielietojums ir ļoti izplatīts arī šodien un sastopams praktiski jebkurā pašapkalpošanās veikalā. XX gadsimta 90. gados RFID atrada ļoti plašu pielietojumu piekļuves kontroles sistēmās un apmaksas sistēmās.

RFID tehnoloģijas izmantošanas iespējas, priekšrocības un ierobežojumus definē frekvence. Mūsdienas izmanto četras pamata RFID tehnoloģijas frekvenču klases [Hei2005, Lah2005, Hun2007]:

- *Zemas frekvences* (ang. *Low Frequency, LF*) – tās ir frekvences no 125 KHz līdz 134 KHz. Klasi apraksta trīs standarti: ISO 11784 un 11785, ISO/IEC 18000-2, ISO 14223-1. Labi piemērota gadījumam, kad jāizmanto kopā ar fiziskiem objektiem, kuru sastāvā ir metāls, ūdens, dažādu veidu netīrumi, sniegs utt. Parasti informāciju no birkām var nolasīt attālumā < 0.5 m. Parasti lieto dzīvnieku identifikācijai, piekļuves kontrolei utt.
- *Augsta frekvence* (ang. *High Frequency, HF*) – tā ir frekvence 13.56 MHz. Klasi apraksta četri standarti: ISO/IEC 18000-3, EPC class I, ISO 15693 un ISO 14443 (A/B). Parasti informāciju no zīmēm var nolasīt apmēram 1 m attālumā. Parasti lieto piekļuves kontrolei, maksājuma identifikācijai, preces līmeņa iezīmēšanai, bagāžas kontrolei, biometrijā, bibliotēkās, transportā, tērpos utt. Vēl eksistē apakšklase *Ļoti augstas frekvences* (ang. *Very High Frequency, VHF*), kura mūsdienās netiek komerciāli izmantota.

- *Ultra augstas frekvences* (ang. *Ultra High Frequency, UHF*) – tas ir frekvences diapazons no 300 MHz līdz 1 GHz. Frekvences izmantošana ir atkarīga no valsts (piemēram, Amerikā 915 MHz, Eiropas valstīs 868 MHz) un no birkas tipa (piemēram, aktīvās birkas 375 MHz, pasīvās 868 MHz). Šīs frekvences apraksta divi standarti: ISO/IEC 18000-6, EPC class 0/class I. Parasti informāciju var nolasīt apmēram 4–5 m attālumā. Tipiski lieto iegādes ķēdes konteineru un kastu iezīmēšanai, bagāžas apstrādei, elektronisko nodevu iekasēšanai.
- *Mikroviļņu frekvence* (ang. *Microwave Frequency, MF*) – tā ir frekvence virs 1 GHz, un parasti tās ir diapazons ir no 2.45 GHz līdz 5.8 GHz. Šīs frekvences apraksta standarts ISO/IEC 18000-4. Parasti informāciju var nolasīt apmēram 1 m attālumā. Lieto elektronisko nodevu iekasēšanai, augstuma ķēžu pārvaldībai, vides uzraudzībai utt.

RFID birkas dalās sešās klasēs atkarībā no funkcionālajām iespējām [Bau2005]:

- *0. klase* – tikai lasīšanai, dati tiek ierakstīti rūpnīcā, kas ražo birkas;
- *1. klase* – ierakstīt var tikai vienu reizi, lasīt var vairakkārt, *WORM* (ang. *Write Once, Read Many*) metode – tā ir metode, kad ieraksts tiek veikts vienu reizi rūpnīcā vai to veic gala lietotājs;
- *2. klase* – lasīt un rakstīt, iespējams programmēt daudzas reizes;
- *3. klase* – lasīt un rakstīt, ar iebūvētiem sensoriem (piemēram, termometrs);
- *4. klase* – lasīt un rakstīt, ar integrētu rādītāju, komunicē neatkarīgi no RFID lasītājiem;
- *5. klase* – lasīt un rakstīt, ar integrētu rādītāju, 4. klases priekšrocības ar iespēju komunicēt ar pasīvām ierīcēm.

Eksistē vēl viens RFID tehnoloģijas klasifikācijas veids pēc birku uzbūves principa. Šis klasifikācijas veids apvieno dažādas frekvences birkas ar dažādām birkas klasēm. Eksistē trīs pamatbirku veidi:

1. *Pasīvā birka* – parasti tās ir 0., 1. un 2. klases birkas, kam nav iebūvēts enerģijas avots (akumulators). Enerģija, lai nolasītu datus no birkas, tiek saņemta no RFID skenera. Šīs birkas tiek raksturotas ar zemām izmaksām, tās ir izturīgas pret netipiskiem ārējās vides apstākļiem (piemēram, augstas temperatūras, ūdens izturība). Tipiski tiek

izmantotas LF un HF tipa frekvences. Mūsdienās tas ir viens no populārākajiem birku veidiem.

2. *Daļēji aktīvā birka* (cits nosaukums *daļēji pasīvā birka*) – tās ir 1. līdz 3. klases birkas, kam ir iebūvētais enerģijas avots (akumulators). Birkai papildus enerģija ir izmantota lielāku datu apjomu glabāšanai un darba rādiusu palielināšanai. Parasti tiek izmantotas UHF un MF frekvences. Ir būtisks ierobežojums, kas atteicas uz birkas dzīves ciklu. Parasti tās dzīves cikls ir 2-3 gadi.
3. *Aktīvā birka* – tai atbilst no 2. līdz 5. klases birkai, kurā ir iebūvēts enerģijas avots (akumulators) un kurai ir vairāk funkcionālu aparatūras un programmatūras iespēju. Būtiski ir palielinājusies birkas cena. Funkcionalitātes esamība rada iespēju komunicēties ar citām birkām (5. klases birkas), palielinās darbības diapazons (no simts metriem līdz dažiem kilometriem). Lielāka funkcionalitāte prasa vairāk enerģijas, būtiski samazinās birkas dzīves cikls.

Viena no aprakstītajām galvenajām priekšrocībām [Rob2006] ir tāda, ka RFID sistēmas ir bezkontakta un strādā arī tad, ja objekts nav redzes līnijā. Birkas var nolasīt arī dažādos vides un vizuāli problemātiskos apstākļos, tādus kā sniegs, ledus, migla, krusa, netīrumi, konteineros un transportā. Tā kā atbildes laiks RFID lasītajam ir mazāk par 100 ms, tad tas var nolasīt vienlaikus līdz pat simts birkām. Viens no lielākajiem trūkumiem [Lee2005] RFID sistēmās ir lēna birku identifikācija kolīziju dēļ, tas ir tad, kad dažas birkas vienlaicīgi strādā vienā radio frekvencē.

RFID tehnoloģija jau tagad ir izplatīta un s izplatīsies vēl vairāk nākotnē. Ar katru gadu šīs tehnoloģijas izmantošanas izmaksas pazeminās, bet tehnoloģiskās spējas palielinās un uzlabojas. No 2004. gada Nokia kompānija izstrādāja pirmo GSM telefonu ar integrēto RFID birku nolasīšanas iespēju [Nok2004]. Tagad citas lielās kompānijas – tādas kā Samsung [Sam2010], Apple piedāvā vai plāno piedāvāt RFID lasīšanas iespēju GSM telefonos. Mūsdienās RFID tehnoloģija ir aktuāls pētīšanas objekts. Bet gandrīz visi pētījumi tiek veikti ar centralizētas informācijas glabāšanās principa izmantošanu [Lah2005, Hei2005, Bau2005, Sar2010]. Darbā [Cho2007] tiek veikts pētījums par MAS veidošanu uz RFID tehnoloģijas bāzes, lai attāli glabātu datus un izgūtu tos caur bezvadu tīklu. Tika definēts, kā RFID birkas jau tiek izmantotas MAS šajā darbā [Zip2001], bet tā piedāvātā pieeja ir vienkāršota un RFID drošība netika apspriesta.

Tuva darbības lauka komunikācija

Tuva darbības lauka komunikācija (ang. *Near Field Communication, NFC*) tā ir radio frekvenču tehnoloģija, kura ir domāta komunicēšanai īsās distancēs, pārsvarā tas ir loģisks RFID paplašinājums. NFC kā tādu izstrādāja Sony un NXP Semiconductors 2002. gadā.

NFC ir savietojams ar citām RFID sistēmām, bet tam ir atšķirīgs darba princips. Galvenā atšķirība slēpjas tur, ka NFC ierīce var būt gan raidītājs, gan saņēmējs, kad tradicionālā RFID pieeja satur “vecāko”, jeb raidītāju un “jaunāko” jeb uztvērējierīci. RFID ierīces ir izstrādātas ar domu, ka tā ir vienīgā aktīvā ierīce darbības laukā, tai pašā laikā NFC ierīces izmanto, lai izvairītos no sadursmēm tīklā, kurā var atrasties citas NFC ierīces. NFC ir induktīvā pāra tehnoloģija, ar 13.56 MHz radio lauka frekvenci. Datu pārraides ātrumi ir 106 kBit/s, 202 kBit/s un 404kBit/s.

NFC šobrīd tiek ļoti plaši lietots Japānā FeliCa tehnoloģijas veidā, kuru arī izstrādāja Sony. FeliCa tika izgudrots kā RFID karšu standarts, bet sākot ar šo gadu Sony nodarbojas ar NFC un FeliCa savietojamību, tika reformēta FeliCa infrastruktūra. FeliCa ir de-facto IC karšu standarts Japānā. Edy viens no IC karšu standartiem, ir visizplatītākā elektroniskā naudas karte Japānā. Transportā lietojamo karšu vidū par izplatību spriest ir grūtāk, jo tas ir atkarīgs no reģiona, tāpēc katrā novadā ir savs karšu paveids, kurš tiek bāzēts FeliCa tehnoloģijā. Vislielākais lietotāju skaits ir Suica, PASMO, ICOCA, Kitaca un TOICA kartēm. Plaši atzītu starptautiskā NFC pielietojošā sistēma šobrīd ir grūti atrodama, bet PayPass varētu būt viens no pirmajiem starp tiem. [Ker2011]

Lokālais tīkls

Datortīkls tas ir datoru ierīces savstarpējais savienojums ar sakaru kanālu starpniecību. Datortīkli var savienot dažādas ierīces, kas nodrošina datortīkla piesaistīšanu, spēj komunicēt un interpretēt informāciju no tīkla, citiem vārdiem sakot, nodrošina tīklu protokolu atbalstu. Viens no tipiskiem datortīkliem ir *lokālais tīkls* (ang. *Local Area Network, LAN*). LAN tas ir tīkls, kurš darbojas relatīvi nelielā teritorijā un tiek izmantots, lai sniegtu savienojamību aplikācijām organizācijām vai citiem lietotājiem.

Datortīklu straujas attīstības sākumu var saskatīt XX gadsimta. 60. gados. Neskatoties uz to, ka mūsdienu datortīkli atgriežas pie centralizētās skaitļošanas principiem, tā laika tīkli attīstījās pretējā virzienā skaitļošanas iekārtu tīkliem, bez centralizēta lieldatora, kad tīkli tiek izmantoti tikai informācijas apmaiņai. Datortīklu mūsdienu veids parādījās kopā ar jēdzienu Ethernet, kurš tika izgudrots 1975. gadā Xerox korporācijā, un IEEE nosaka Ethernet starptautisko standartu, izveidojot “802” komiteju. IEEE nosaka 802.3 standartu kā LAN Ethernet. 802.2 jeb *LLC* (ang. *Logical Link Control*) tika izveidots, lai noteiktu dažādu paveidu LAN komunikācijas modeli.

Datortīklam tiek izmantotas dažādu veidu ierīces, kas ir savienotas savā starpā tīklā, tās sauc par *tīkla topoloģijām*. Daudzas priekšrocības un ierobežojumi izriet no izvēlētās tīkla topoloģijas. Izplatītākie tīklu topoloģijas veidi ir [Her2001]:

- *Kopnes topoloģija* tas ir tīkls, kas sastāv no vada, ar kuru tiek tieši savienoti tīkla mezgli. Katrs mezgls uzklausa tīkla datu plūsmas, bet adrešu filtrs ļauj katram mezglam pārtvert tikai to informāciju, kura ir nosūtīta noteiktam mezglam. No priekšrocībām ir minamas – vienkāršība, zemas izmaksas, uzticamība, pieejamība, vienkārša paplašināmība un konfigurēšana. Kā trūkumu var minēt to, kad kopne, kuras funkcijas izpilda tīkla vads, tiek bojāta, tīkls vairs nav lietojams. Tas ir galvenais iemesls, kāpēc kopnes topoloģija nav izplatīta mūsdienās.
- *Zvaigžņveida topoloģija* ir visizplatītākais tīklu topoloģijas veids mūsdienās. Tas izmanto 10BaseT, 100BaseT un 1000BaseT tehnoloģijas. Šī topoloģija tiek veidota no mezgla, pie kura tiek pievienoti visi attālie mezgli. Priekšrocības ir pārvaldīšanas un novērošanas vienkāršība un savienojamība starp noteiktiem punktiem. Ja viens kabelis, kurš savieno apakšējo mezglu ar centrālo mezglu, ir sabojāts, tas neietekmē tīkla darba spēju. Starp trūkumiem ir jāatzīmē atkarība no centrālā mezgla ātrdarbības un uzticamības. Ja centrālais mezgls iet bojā, viss tīkls kļūst nelietojams. Relatīvi augstās tīkla izveidošanas izmaksas ir vēl viens trūkums, jo katram mezglam ir nepieciešams interfeiss jeb vads, lai varētu savienoties ar centrālo mezglu, tādējādi tiek izmantots liels vadu daudzums. Turklāt tīkla paplašināmības iespējas ir atkarīgas no centrālā mezgla fiziskas iespējas apkalpot limitētu apakšējo mezglu daudzumu, vienkārši izsakoties – centrālais mezgls nevar savienot vairāk datoru nekā tam ir tīkla vadu savienotāju.
- *Gredzenveida topoloģija* ir relatīvi vienkārša, jo tā ietver savienojumu starp mezgliem ar punkta-punkta saikni, tādējādi izveidojot apli. Katrs mezgls saņemot informāciju, kura tam ir nosūtīta, saņem datus no tīkla, bet informāciju, kura nav adresēta šim noteiktajam mezglam, nosūta tālāk pa apli. Priekšrocība ir iespēja lietot optisko šķiedru kabeļus tā vienkāršajās saitēs starp mezgliem, kas nodrošina augstu datu plūsmas iespēju. Trūkums ir augstas izmaksas dubultotu detaļu dēļ, neuzticamība, jo viens mezgls, kurš gājis bojā, apstādina visu tīklu, kā arī elastīguma trūkums.
- *Kokveida topoloģija* ir sava veida kopnes topoloģijas uzlabotais veids. Tīkla vads darbojas kā kopne, kurai ir atzari. Tāpat kā kopnes topoloģijā, kad notiek pārraide, dati tiek raidīti uzreiz visiem mezgliem tīklā. Priekšrocības un trūkumi ir praktiski tie paši,

kuri ir kopnes topoloģijai, ieskaitot to priekšrocību, ka kokveida topoloģija ir elastīgāka par primitīvo kopnes topoloģiju.

Galvenie faktori, kuri ietekmē LAN drošību, ir operētājsistēma un drošības programmnodrošinājums uz katra mezgla, vai LAN ir pievienots Internetam, vai tiek organizēta tīkla segmentu loģiskā un fiziskā norobežošana drošības nolūkos. Kopumā drošība ir neautorizētas datu piekļuves un neautorizētu darbību ar datiem novēršana.

Bluetooth

Bluetooth ir bezvadu personālo tīklu standarts. Šī tehnoloģija ļauj veikt datu apmaiņu starp dažādām ierīcēm ar radiosignālu palīdzību. 1998. gadā tika nodibināta izstrādātāju grupa *Bluetooth SIG* (ang. *Bluetooth Special Interest Group*). Šajā grupā piedalījās izstrādātāji no piecām kompānijām: Ericsson, IBM, Intel, Toshiba un Nokia. 2002. gadā tika veikta IEEE sertificēšana un rezultātā tika izveidots IEEE 802.15.1 standarts.

Bluetooth tehnoloģija savā darbā izmanto frekvences diapazonu no 2.4 līdz 2.48 GHz. Tehnoloģija ir vienkārša un nav dārga realizācijas ziņā. Par pamatu tiek izmantota *FHSS* metode (ang. *Frequency Hopping Spread Spectrum*), tiek nodrošināta nesēja signāla lēcienveidīgas maiņa 1600 reizes sekundē. Sakarā ar frekvences maiņu, kas ir zināma tikai raidītājam, uztvērējam ir nodrošināta iespēja pārsūtīt tīru signālu. Tas ir gadījums, kad nav nepieciešamības pēc papildu datu pārraides plūsmas kodēšanas, kas dod iespējas izmantot paaugstinātus datu pārraides ātrumus. Piemēram, Bluetooth 1.0 līdz 0.7 Mbits/sek, Bluetooth 2.0 līdz 2.1 Mbits/sek, Bluetooth 3.0 līdz 24 Mbit/s un jaunākajā versijā Bluetooth 4.0 plānoti vēl augstāki datu pārraides ātrumi. Mūsdienās viens no populārākiem standartiem ir Bluetooth 2.0. Darbības diapazons atkarībā no versijas ir no 10 līdz 100 m.

Savā laikā tika plānots, ka Bluetooth tehnoloģija kļūs par galveno spēlētāju bezvadu datu komunikāciju tirgū tās vienkāršības un lielu kompāniju atbalsta dēļ. Mūsdienās Bluetooth SIG ietver vairāk nekā 15000 dalībnieku un tehnoloģijas atbalsts tiek nodrošināts lielākajai daļai no mobilajām ierīcēm.

Infrasarkanais savienojums

1993. gada 28. jūnijā *IrDA* (ang. *Infrared Data Association*) organizēja pirmo tikšanos ar mērķi izveidot lētu, visuresošu, seriālu infrasarkano sakaru standartu. Tika gaidīti tikai 50 pārstāvji no apmēram 20 ieinteresētiem uzņēmumiem, bet ieradās ap 150 cilvēkiem no vairāk nekā 50 uzņēmumiem, kas acīmredzami liecināja par to, ka industrijai šāds standarts ir nepieciešams. [Knu2001] Infrasarkanā savienojuma tehnoloģiju nosauca par IrDA.

No [Knu2001, Mil1998] pētījumiem ir apkopota informācija par tās darbības principiem. Šobrīd IrDA nosaka fizisko un protokola slāni, kuri ir nepieciešami katram divu ierīču pārim, kuri pakļaujas IrDA standartam, lai varētu savstarpēji notikt datu apmaiņa un

ierīču atrašana. Pirmā IrDA 1.0 specifikācija detalizēja seriālu, pusdupleksu, asinhronu sistēmu ar 2400 bit/s – 115200 bit/s pārraides ātrumu, ar darbības attālumu līdz vienam metram, ar redzamības leņķi starp 15 un 30 grādiem. Vēlāk IrDA fiziskā slāņa specifikācija tika paplašināta datu pārraides ātrumu nodrošināšanai, kura sasniedz 4 Mbit/s.

[Mil1998, Haa2006] pētījumos ir apskatīts IrDA datu drošības un konfidencialitātes jautājums. Tiek uzskatīts, ka acīmredzams IrDA drošības aspekts ir tā pārraides mehānismā, jo infrasarkanos viļņus nav iespējams efektīvi pārtvert nozīmīgos attālumos. Pārējie drošības aspekti ir tie paši, kuri ir citiem bezvadu tīkliem, jo IrDA darbojas pēc ļoti līdzīga principa, lai arī utilizē tikai divi loģiskie slāņi.

Svarīgākā IrDA priekšrocība ir relatīvs vienkāršums, zemas ierīču izmaksas un pieejamība, relatīva neatkarība no traucējumiem, jo IrDA darbojas infrasarkanās gaismas diapazonā. Biežāk pieminētie šīs tehnoloģijas trūkumi ir šādi - IrDA nav iespējams lietot bez tiešās redzamības abu ierīču starpā, lietošanas attālums ir ļoti zems un pastāv atkarība no leņķa.

IrDA popularitātes vilnis mūsdienās ir noplacis, bet joprojām tas tiek lietots mobilajās ierīcēs, printeros un portatīvajos datoros. Ir iespējami IrDA pielietojumi arī lokālo tīklu izveidei. IrDA plaši tiek lietots augsti tehnoloģiskajos mobilajos tālruņos Japānā un Korejā, kur laika gaitā ir izveidojusies IrDA apmaiņas kultūra, kas ir svarīgs sociāls elements.

Mobilās telekomunikāciju tehnoloģijas

XX gadsimta 80. gados mobilie sakari piedzīvoja strauju attīstību Eiropā. Katra valsts izveidoja savus sakaru standartus, kuri nebija savietojami ar sakaru sistēmām citās valstīs. Šī situācija bija neizdevīga ne tikai lietotājiem, kuru mobilie sakari bija limitēti ar noteiktu valsti, bet arī ekonomiski, jo sakaru iekārtu tirgus nespēja paplašināties vairāk par vienu valsti un tāpēc bija stipri ierobežots. 1982. gadā *CEPT* (ang. *Conference of European Posts and Telegraphs*) izveidoja izpētes grupu, kuru nosauca par *GSM* (fran. *Groupe Special Mobile*). Grupas mērķis bija visas Eiropas mobilo sakaru sistēmas izveidošana, ar sekojošiem kritērijiem: laba balsu pārraides kvalitāte, zema infrastruktūras cena, starptautiskas viesabonēšanas atbalsts, iespēja atbalstīt mobilas termināļu iekārtas, daudzu jaunu pakalpojumu atbalsts, *ISDN* (ang. *Integrated Services Digital Network*) savietojamība.

GSM ir populārākais mobilo sakaru standarts. Tas darbojas pēc bāzes staciju principa – mobilā ierīce meklē tai tuvāko bāzes staciju, lai pievienotos tīklam. Detalizētāka informācija par iekšējiem darbības mehānismiem ir pieejama [Ibr2005, Sak2005, Sco1996] pētījumos.

Galvenā GSM priekšrocība ir tās pieejamība, tehnoloģijai nav vajadzīga tieša redzamība, un tā strādā gan ārējā vidē, gan iekštelpās. Signāla stiprums var mainīties, atkarībā no signāla absorbējušiem šķēršļiem, bet pietiekošu darba staciju izplatības gadījumā šī

ierobežojuma sekas ir minimālas. Tiek ieviesti daudz līmeņu datu pārraides drošības mehānismi, detalizēta informācija pieejama daudzos pētījumos (piemēram, [Mar2005, Pen2008]).

Pirmā GSM specifikācijas fāze tika izstrādāta 1990. gadā, un komerciālie pakalpojumi sākās jau 1991. gada vidū, un kopš tā laika GSM ir piedzīvojis kolosālu izaugsmi. Neskatoties uz to, ka GSM tika izstrādāts kā Eiropas standarts, tehnoloģija ir izplatīta gan Tuvajos Austrumos, gan Āfrikā, gan Amerikā, gan Austrālijā un daudzviet Tālajos Austrumos, tādējādi veidojot visplašāk lietojamo mobilo sakaru tīklu kopumu. Šobrīd vislielākais GSM lietotāju skaits ir Ķīnā, Indijā un Eiropas Savienībā.

Vispārējais pakešu radio pakalpojums

Vispārējais pakešu radio pakalpojums (ang. *General Packet Radio Service, GPRS*) - tas ir serviss, kurš balstās uz GSM mobilajiem sakariem, kas nodrošina pakešu datu pārraidi. Šis serviss ļauj nodrošināt datu pārraidi starp dažādām GSM ierīcēm, tajā skaitā nodrošina pieeju Internet. GPRS paredz nosūtīto datu apjoma tarifikāciju, atšķirībā no GSM tarifikācijas veidā.

GPRS ir līdzīgas GSM tehnoloģijas priekšrocības, ierobežojumi un trūkumi. Galvenā atšķirība ir tā, ka GPRS ir paredzēts datu pārraidei un tāpēc ir nodrošināts paaugstināts datu pārraides ātrums. Ir arī tehniskas realizācijas atšķirības, piemēram, datu drošības mehānismu realizācijā (detalizētāk skat. [Pen2008]).

Wi-Fi

Wi-Fi (IEEE 802.11) ir bezvadu interneta standarts, kuru izstrādāja *Wi-Fi Alliance* [Lab2007]. Dažādu ražotāju produktu savietojamību nodrošina neatkarīga organizācija Wi-Fi Alliance, kas dibināta 1999. gadā ar sešiem kompānijas dibinātājiem: 3Com, Aironet, Intersil, Lucent Technologies, Nokia un Symbol Technologies. Mūsdienās Wi-Fi Alliance biedri ir vairāk kā 350 kompānijas, to skaitā tādi pazīstami ražotāji kā D-Link, Enterasys, Cisco, Lucent, 3Com, IBM, Intel, Apple, Compaq u.c. Ar detalizētāku informāciju par Wi-Fi Alliance var iepazīties tās oficiālajā Internet lapā [WiF2012].

Bezvadu internets iegūst arvien lielāku izplatību pasaulē. Šai tehnoloģijai ir dažādi standarti – IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n [Lab2007]. Wi-Fi tīkls tiek izveidots no noteikta skaita *piekļuves punktiem* (ang. *access point*), kas tiek savienoti tīklā un pievienoti globālajam tīmeklim ar parastu vadu savienojumu. Interference starp dažādām stacijām tiek minimizēta, izmantojot dažādus signāla kanālus un CSMA/CA piekļuves protokolu. Ierīces tiek pieslēgtas tīklam, izmantojot tā saucamās Wi-Fi kartes, kas parasti tiek saistītas ar to piekļuves punktu, kuram ir stiprāks signāls. Ierīcei ir iespēja pārvietoties starp dažādiem piekļuves punktiem, kas ļauj paplašināt Wi-Fi tīklus, lai veidotu tā saucamos

„pieslēguma mākoņus” (ang. *clouds of connectivity*) tādās vietās, kur ir liels pieprasījums pēc bezvadu interneta. [Bru2005]

Tipiskie piekļuves punkti strādā IEEE 802.11b un IEEE 802.11g standartu bāzē, tas nodrošina darbības diapazonu 32 metrus iekštelpās un 95 metrus ārpus telpām. Ar datu pārraides ātrumu no 1 līdz 2 Mbps. IEEE 802.11g standarta maksimālais definētais ātrums ir 54 Mbps un IEEE 802.11g līdz 600 Mbps.

Wi-Fi piekļuves punkti ir plaši izplatīti tirgū, tas ir saistīts ar vieglu un lētu infrastruktūras uzbūvi. Gandrīz jebkurai mūsdienīgai datorsistēmai ir iespējams pieslēgt Wi-Fi moduli. Wi-Fi parasti asociē ar mobilo Internetu, tiek iebūvēts tādās datorsistēmās kā portatīvie datori, plaukstdatori, viedtālruņi un citās ierīces. Stingra standartizācija nodrošina Wi-Fi iekārtu savietojamību.

Galvenais tehniskais trūkums, kas parasti ir pieminēts pētījumos [Lab2007, Bru2005], ir drošības jautājums. Brīva piekļūšana pie Wi-Fi piekļuves punktiem prasa papildus pievērst uzmanību datu pārsūtīšanas drošības nodrošināšanai. Ir tendence Wi-Fi piekļuves punktus izmantot kopā ar aizsardzības līdzekļiem (piemēram, WEP, WPA un WPA2 datu šifrēšanas standarti [WiF2012]) izmanto *virtuālu privātu tīklu* (ang. *Virtual Private Network, VPN*).

WiMAX

WiMAX tas ir IEEE802.16 standarta pamatā, bezvadu platjoslas pieejas tehnoloģija, kura nodrošina izpildījumu, kurš ir tuvs Wi-Fi tehnoloģijai.

Apvienojot informāciju no dažādiem pētījumiem [Kan2011, Gab2005, Sas2006, Jef2007] ir saņemta informācija par WiMAX izveidošanas vēsturi. XX gadsimta 90. gadu vidū telekomunikāciju organizācijas izstrādāja platjoslu bezvadu tīklu lietošanas ideju, izmantojot pēdējos industrijas sasniegumus, lai varētu sniegt zināmu alternatīvu esošajām tehnoloģijām. Pirmais rezultāts nāca gaismā 2001. gadā, tas bija sakaru standarts 10 Mhz līdz 66 Ghz frekvenču diapazonā, bet faktiski tas bija WLAN (ang. *Wireless LAN*) atvasinājums un tam piemita zināmi ierobežojumi, tāpēc ražotāji pievērsās 802.16 standartam 2 Ghz – 11 Ghz diapazonā. 2001. gadā tika izveidots WiMAX Forums, kurš nodarbojās ar WiMAX / IEEE 802.16 standarta veicināšanu tirgū. Tādējādi tie pieņēma s WiMAX vārdu un 2003. gadā tika izveidots IEEE 802.16a standarts, kur dati tika raidīti ar netieši notēmētiem radio signāliem no bezvirziena antenām. 2004. gadā tika izveidots IEEE 802.16-2004 standarts, bet 2005. gadā tika laista klajā pirmā mobilā WiMAX sistēma 802.16e. Šī tehnoloģijas versija izmantoja *SOFDMA* (ang. *Scalable Orthogonal Frequency-Division Multiple Access*) standartu, kurš tika optimizēts pret pakešu zaudējumiem un aizturēm, turklāt ar pastiprinātu tīkla drošību. Šobrīd WiMAX piedzīvo strauju attīstību, un tā mērķis ir palielināt pārraides ātrumu līdz 1Gbps.

WiMAX darbības princips ir līdzīgs Wi-Fi. Tam ir zināmas priekšrocības Wi-Fi priekšā, jo WiMAX spēj darboties līdz pat 50 kilometru attālumā no stacijām, tai pašā laikā Wi-Fi spēj nodrošināt tikai 30-100 metrus. WiMAX spēj nodrošināt tos pašus ātrumus, kurus nodrošina Wi-Fi, bet ar tādu atšķirību, ka WiMAX signāls ļoti stipri pavājinās, ja starp raidītāju un uztvērēju ir fiziska barjera. [Gab2005] pētījumā ir uzskatīts, ka galvenās 802.16 priekšrocības, salīdzinot ar citām bezvadu tīklu tehnoloģijām, ir augsta spektrālā efektivitāte (3.6 bps/Hz), kas nodrošina lielu kanālu kapacitāti un datu pārraides ātrumus (līdz 70 Mbps), spēja strādāt bez tiešās redzamības, liels darbības attālums un zemas aiztures, kas būtiskas reālā laika lietotnēm. Kopumā ir samērā zema moduļu cena, augsta mobilitāte, augsti pārraides ātrumi, liels attālums starp raidītāju un uztvērēju.

Satelītu tehnoloģija

Jau ilgu laiku pastāv satelītu tehnoloģijas, viens no tipiskiem piemēriem ir satelītu navigācija [Bsh2009]. Tā sākotnēji tika izveidota izmantošanai militāriem mērķiem. Pastāv dažādas sistēmas, kas ir bāzētas uz šo tehnoloģiju. Satelītu navigācijas tehnoloģija ļauj ar mazu elektronisku uztvērēju palīdzību noteikt to pozīciju līdz pat dažu metru precizitāti.

Vispazīstamākā sistēma, kas ir bāzēta uz satelītu navigāciju, ir GPS. XX gadsimta 60. gados vairākas ASV valsts organizācijas, ieskaitot *DOD* (ang. *Department of Defense*), *NASA* (ang. *National Aeronautics and Space Administration*) un *DOT* (ang. *Department of Transport*) bija ieinteresētas satelītu sistēmas izstrādāšanā, kura nodrošinātu objekta trīsdimensiālu pozīcijas noteikšanu. Optimāla sistēma tā ir sistēma ar sekojošām īpašībām: globāls pārklājums, nepārtraukta darbība neatkarīgi no laika apstākļiem, iespēja nodrošināt pakalpojumus dinamiskām platformām un augsta precizitāte. 1969. gadā *OSD* (ang. *Office of the Secretary Defense*) nodibināja *DNSS* (ang. *Defense Navigation Satellite System*) programmu ar neatkarīgu izpētes rezultātu konsolidēšanas mērķi. Kopš šī brīža tika izveidota *NAVSTAR GPS* koncepcija. Šī programma tika izstrādāta *GPS JPO* (ang. *Joint Program Office*) birojā *El Segundo* pilsētā Kalifornijā, un šis birojs turpināja izstrādāt un pilnveidot sistēmu, ražot jaunus satelītus, zemes virsmā lietojamas iekārtas un lielāko daļu no GPS uztvērējiem, kuri tika lietoti militārajām vajadzībām. [Kap2006]

Šobrīd GPS ir pilnīgi izstrādāta un atbilst kritērijiem, kuri tika noformulēti 1960. gadā, GPS izstrādes sākumā. Sistēma nodrošina precīzu, nepārtrauktu vispasaules trīsdimensiālu atrašanās vietas noteikšanu un izplata *Universālais koordinētais laiks* (ang. *Coordinated Universal Time, UTC*) standarta laiku.

Galvenais darbības princips ir objekta pozīcijas noteikšana, tas aprēķina attālumu no satelīta. Satelīts nosaka attālumu līdz objektam, izmērot laiku, kurš ir vajadzīgs, lai signāls no raidītāja tiktu līdz satelītam. Lai uzzinātu objekta pozīciju uz Zemes, nepieciešams noteikt tā

atrašanos *platumā* (ang. *latitude*) un *garumā* (ang. *longitude*), aprēķinot attālumu vismaz no trīs satelītiem. Ja nepieciešams uzzināt vēl *augstumu* (ang. *altitude*), tad nepieciešami četri satelīti (detalizētāk skat. [Dom1996, Kan1996, Kan2006]). Augstuma sākpunkts definēts Zemes centrā caur Ziemeļpolu.

Atkarībā no pavadoņu izvietojuma virs koordinātes apvidū GPS signāla precizitāte var svārstīties no 3.5 līdz 150 metriem. Faktori, kas samazina precizitāti: pavadoņu orbītas, pavadoņu izvietojuma ģeometrija GPS ierīces uztveršanas apvidū, traucējošo objektu klātbūtne, kas aizklāj nepieciešamos debesu apvidus, atmosfēras ietekme un radio viļņu atstarojums. Galvenais tehnoloģijas trūkums ir tas, ka zināmos apstākļos signāls var nerasniegt uztvērēju vai pienākt ar ievērojamiem sagrozījumiem jeb aizkavējumiem. Piemēram, praktiski nav iespējams precīzi noteikt savu atrašanās vietu, atrodoties dzelzsbetona ēkas dzīvoklī, pagrabā vai tunelī.

Pēdējā laikā šī sistēma ir kļuvusi populāra un GPS uztvērēji tiek iebūvēti visdažādākajās mobilajās ierīcēs. Eksistē arī citas analogiskas sistēmas, kas nodrošina līdzīgas funkcijas, bet tomēr nav tik izplatītas kā GPS: Galileo, KOMPASS, GLONASS un citas [Bød2006].

Sensori

Eksistē tendence IT risinājumos analizēt apkārtējo pasauli, izmantojot sensorus. Pēc definīcijas *sensors* [Lza2011] – tās ir ierīce, kuras uzdevums ir noteikt kāda objekta fizikālo stāvokli (piemēram, drukāšanas galviņas novietojumu) vai tā fizikālo kvalitāti (piemēram, temperatūru) un pārveidot to datora ieejas signālā.

Līdz šim brīdim no aprakstītajām tehnoloģijām tipiskais sensoru piemērs ir GPS tehnoloģija, kas izmantota objekta globālas pozīcijas noteikšanai. Vēl eksistē arī dažādi sensoru veidi, kas neasociējas ar konkrētu tehnoloģiju, bet parasti tiek pievienoti konkrētām ierīcēm. Piemēram, ir risinājumi (skat. [Lah2005]), kad apkārtējas vides sensorus iebūvē RFID birkās. Tās paplašina RFID tehnoloģiju ar papildu iespējām.

Parasti sensors ģenerē analogsignālu, kas pirms tā ievadīšanas datorā ar analogo diskrēto pārveidotāju pārvērš to par ciparsignālu. Sensorus iespējams klasificēt sekojošās kategorijās: *termiskie* (temperatūras devējs), *elektromagnētiskie* (voltmetrs), *mehāniskie* (spidometrs), *optiskie* (web kamera), *ķīmiskie* (elektrods), *akustiskie* (mikrofons) utt.

2.2.3. Tehnoloģiju parametru salīdzinājums

Tiek apskatītas tipiskas tehnoloģijas, kas ir izmantotas objektu individuālajās īpašībās saņemtās sistēmās. Pēc šo tehnoloģiju vispārīga apskata ir definēti nozīmīgākie tehnoloģiju parametri. Parametru analīzei ir izmantotas visbiežāk lietotie tehnoloģiju paveidi un versijas.

Piemēram, Bluetooth tehnoloģijas populārākā versija mūsdienās ir Bluetooth 2.0. Visa šī informācija ir apvienota tabulas veidā (skat. 2.3. tabulā).

2.3. tabula

Tehnoloģiju parametri salīdzinājums

(I – tehnoloģiju grupa; II – darba rādiuss (kilometros); III – datu pārraides ātrums (datu pārraides ātrums, vai sek/obj); IV – darba frekvence; V – tiešās redzamības nepieciešamība; VI – ieviešanas izmaksas (infrastruktūra/termināls/birka); VII – tehnoloģiju iedarbināšanas mehānismu sarežģītība; VIII – datu aizsardzība)

Parametrs Tehnoloģija	I	II	III	IV	V	VI	VII	VIII
<i>Svītrkods (Code128)</i>	1.	0.01	0.3-0.5 obj./sek.	n/a	ir	3/1/0	nē	SW
<i>Svītrkods (DataMatrix)</i>	1.	0.002	0.2-0.4 obj./sek.	n/a	ir	3/2/0	nē	SW
<i>Svītrkods (QR Code)</i>	1.	0.002	0.2-0.4 obj./sek.	n/a	ir	3/2/0	nē	SW
<i>RFID (pasīvais)</i>	1.	0.0001 - 0.009	1-200 obj./sek.	LF, HF	vēlama	4/2/1	nē	SW, HW
<i>RFID (daļēji aktīvais)</i>	1.	0.001 – 0.035	1-200 obj./sek.	UHF, MF	vēlama	4/2/3	nē	SW, HW
<i>RFID (aktīvais)</i>	1.	0.035 – 0.1	1-200 obj./sek.	UHF, MF	vēlama	4/2/5	viduvēji	SW, HW
<i>NFC</i>	1.un 2.	0.0015	424 Kb/s.	HF	vēlama	4/1/1	nē	SW, HW
<i>LAN</i>	2.	0.1	1 Gbit/s.	HF	nav	5/3/-	viduvēji	SW, HW
<i>Bluetooth</i>	1.un 2.	0.1	2 Mb/s.	HF	vēlama	2/2/4	nē	SW
<i>IrDA</i>	1.un 2.	0.01	4 Mb/s.	LF	ir	2/2/4	nē	SW
<i>GSM</i>	1.un 2.	35	43.3 Kb/s.	UHF	nav	7/3/5	jā	SW, HW
<i>GPRS</i>	2.	35	80 Kb/s.	UHF	nav	7/3/5	jā	SW, HW
<i>Wi-Fi</i>	1.un 2.	0.1	600 Mb/sek.	MF	vēlama	4/3/6	jā	SW, HW
<i>WiMAX</i>	1.un 2.	50	70 Mb/sek.	HF	vēlama	5/3/7	jā	SW, HW
<i>GPS</i>	3.	globālais	n/a	UHF	ir	9/2/2	jā	n/a
<i>Sensori</i>	3.	0.001	n/a	n/a	ir	2/2/2	nē	n/a

2.3. Datu apmaiņas formāti

Pirmais (i) tehnoloģijas parametrs, kas nosaka *tehnoloģiju grupu*, ietver tehnoloģijas, kas nodrošina datu pārraides funkciju. Jānodrošina (ii) kritērijs objektu individuālajās īpašībās sakņotām sistēmām par *standartu izmantošanu*. Šiem nolūkiem ir apskatīti tipiskie datu apmaiņas standarti, kas tiek izmantoti datu pārraidei.

Ar norobežotāju atdalītas vērtības

Ar norobežotāju atdalītas vērtības – tas ir datu apmaiņas un glabāšanas formāta veids, kas ļauj glabāt un modificēt datus cilvēkiem viegli saprotamā veidā. Viena no pirmajām datu pārvaldības sistēmām, kas balstījās uz ar norobežotāju atdalītas vērtības principiem, ir *izplātās datnes modelis* (detalizētāk skat. [Sin1995]). Šī modeļa pamatā tika izmantots princips, kad dati tika sadalīti pa datu grupām un teksta datnes viena rinda prezentē datus par vienu ierakstu. Uz to brīdi izmantoja divus pamatveidus, kuri prezentē datus ar norobežotāju atdalītām vērtībām:

1. *Dalītāju izmantošana* – tā ir teksta datne, kurai viens ieraksts ir prezentēts ar vienu rindu, kur katrs rindas lauks (ieraksta) atdalīts ar komatiem vai dažādām citām zīmēm (piemēram, punkts, tabulācija zīme utt.).
2. *Fiksētais lauku izmērs* – tā ir teksta datne, kurai viens ieraksts prezentēts ar vienu rindu, rinda ir sadalīta pēc izmēra (simbolu skaits) pa laukiem. Ja vienam laukam ir mazāk vietas nekā iedalīts rindā, tad pārējie simboli prezentēti ar atstarpes zīmi.

Ar norobežotāju atdalītās vērtības formātiem ir tādas priekšrocības kā: vienkārša realizācija, cilvēkam saprotams datu prezentēšanas veids, vienkārša datu modificēšana. Kopā ar priekšrocībām ir savi ierobežojumi un trūkumi, piemēram, nav paredzēts strukturālo datu prezentēšanai, viegli sabojāt (tāpēc ka viegli modificēt), nav iebūvēti datu drošības mehānismi utt.

Tas ir viens no pirmajiem datu apmaiņas un glabāšanas formātiem, kas mūsdienās transformējas citos jaunos teksta veida formātos, kuriem nav teksta datnes veida tipisku ierobežojumu un trūkumu, formātu piemēri: *paplašināmās iezīmēšanas valoda* (ang. *eXtensible Markup Language, XML*), *JavaScript objektu notācija* (ang. *JavaScript Object Notation, JSON*) utt.

Paplašināmās iezīmēšanas valoda

Paplašināmās iezīmēšanas valoda [Lza2011] (ang. *eXtensible Markup Language, XML*) – tas ir programmēšanas valodas SGML apakškopa, kas speciāli izstrādāta darbam ar tīmekļa dokumentiem. Valoda XML nodrošina globālā tīmekļa izstrādātājiem iespēju radīt savas personiskās birkas (kodus), lai nodrošinātu tādu funkciju izpildi, ko nevar nodrošināt ar

valodas HTML starpniecību. Atšķirībā no valodas HTML, kuras saites norāda tikai uz vienu dokumentu, valoda XML nodrošina saites, kurās ir atsauces uz vairākiem dokumentiem.

Vienā no pētījumiem [Oli2002] paskaidrots, ka XML ir lietots, lai attēlotu un datus un apmainītos ar datiem strukturētā veidā Internet tīklā. Atšķirībā no HTML tagiem, kuri ir izveidoti galvenokārt, lai attēlotu hiperteksta prezentācijas iespējas, XML tagus var definēt dokumenta autors, lai attēlotu informācijas semantiku, kas dod formālo informācijas aprakstu.

XML dokumentācijā (piemēram, [Cag2000]) ir norādīts, ka pirmo rindu XML dokumentā sauc par XML deklarāciju – tā ir rinda, kura norāda XML dokumenta versiju. Tāpat šajā rindā var būt norādīta simbolu kodēšanas tabula vai arī ārējo atkarību esamība. Piemēram, `<?xml version="1.1" encoding="UTF-8"?>` nosaka XML dokumenta versiju 1.1. un kodējumu UTF-8. Specifikācija prasa, lai XML procesori obligāti atbalstītu UNICODE kodēšanu UTF-8 un UTF-16 (UTF-32 nav obligāta). Ir pieļaujams lietot arī citas kodēšanas tabulas, kuras balstās uz standarta ISO/IEC 8859. Tagos parasti nelieto simbolus, kas ir atšķirīgi no latīņu alfabēta, tas ļauj izvairīties no kļūdām dokumenta apstrādes laikā.

XML pamatelements ir *elements*, tas ir iegultais elements, kuram var būt atribūti un saturs. Elements parasti sastāv no sākuma un beigu tagam, kuri ietver tekstu un citus elementus. Sākuma tags sastāv no elementa nosaukuma stūra iekavas „`<name>`”, bet beigu tags sastāv no tā paša nosaukuma, pirms kura ir simbols „`/`”. Elementa nosaukumi nevar saturēt atstarpes. Nosaukums var sākties ar burtu, pasvītrojumu vai kolu. Pārējie simboli var būt tie paši simboli un arī cipari, defise un punkts. XML ir reģistra jutīga valoda, tāpēc ir nepieciešams, lai tagu nosaukumi būtu uzrakstīti vienā reģistrā. Par elementa saturu sauc visu to, kas atrodas starp sākuma un beigu tagiem, ieskaitot tekstu un citus elementus.

Elementam var būt arī *atribūti* – pāri nosaukums–nozīme, kuri ir ievietoti sākuma tagā uzreiz pēc tā nosaukuma. Atribūtu vērtības vienmēr atrodas iekavās. Atribūtu nosaukumi nevar atkārtoties vienā elementā.

Viena no galvenajām sintaksiskajām prasībām ir tāda, ka dokumentā var būt tikai viens saknes elements. Tas nozīmē, ka tekstam vai citiem dokumenta datiem ir jābūt starp saknes elementa tagiem.

XML pielietojuma klāsts ir ļoti plašs, to izmanto, lai attēlotu Internet lapu satura iekšējo struktūru, vektoru grafikā [Ant2005], Microsoft Office kompleksa dokumentu saglabāšanai [Mic2007], informācijas apmaiņai starp dažādiem servisiem (piemēram, Google [Goo2012a]).

JavaScript objektu notācija

JavaScript objektu notācija (ang. *JavaScript Object Notation, JSON*) – teksta veida datu apmaiņas un glabāšanas standarts, kas ir paredzēts strukturēto datu prezentēšanai. Datu

prezentēšanas principi bija ņemti no JavaScript valodas, bet formāts uzskatīts par neatkarīgu un mūsdienās izmantots dažādām platformām un programmēšanas valodām. Vienkāršas struktūras datu prezentēšanai tas aizņem mazāk vietas salīdzinot ar XML, tāpēc tas tiek izmantots kā alternatīvs risinājums.

JSON tika izstrādāta balstoties uz JavaScript programmēšanas valodas, standarta ECMA-262 trešās redakcijas 1999. gadā. JSON formātu aprakstījis *Douglas Krokfords* (ang. *Douglas Crockford*) specifikācijā RFC 4627 (fakti no [Cro2006]). JSON pielietoja kompānija State Software sākot no 2001. gada. No 2005.gada Yahoo kompānija sāka piedāvāt savus WEB-servisus JSON formātā. 2006. gada Google kompānija sāka piedāvāt informācijas plūsmas JSON formātā [Goo2012b].

Pēc sintakses (detalizētāk skat. [Goo2012b]) JSON ir JavaScript objektu notācības sintakses apakškopa. Tas satur datus kā nosaukums/vērtības pārus. Visi dati ir atdalīti ar komatu, figūriekavas satur objektus, bet kvadrātiekavas satur masīvus. JSON datus pieraksta kā nosaukums/vērtība pāri. Nosaukums/vērtība pāris sastāv no lauka nosaukuma (dubultās pēdiņas), pēc kura seko kols, pēc kura seko elementa vērtība.

Galvenie JSON datu tipi ir: *numurs* (vesels skaitlis vai skaitlis ar peldošo komatu), *rinda* (dubultās pēdiņas), *loģiskā vērtība* (paties vai nepaties), *masīvs* (kvadrātiekavās) un *objekts* (figūriekavās, var iekļaut sevī daudzus nosaukums/vērtība pārus).

Mūsdienās eksistē daudzas JSON formāta modifikācijas, viena no populārākajām ir *BSON* (ang. *Binary JSON*), tas ir JSON formāts, kas prezentēts bināra veidā. Detalizētāk par dažādu JSON formātu modifikācijām un salīdzinājumu ar citiem formātiem ir iespējams apskatīt [Bay2010] pētījumā.

JSON formātu lieto, lai nodrošinātu informācijas pārraidi no vienas sistēmas uz citu. Viens no populārākajiem izmantošanas viediem ir Google GeoCoding sistēma [Goo2012b].

Protokolu buferis

Protokolu buferis (ang. *Protocol Buffers, PB*) (definējums ir adaptēts no [Bay2010]) ir formāts datu serializācijai. Google kompānija izstrādāja PB formātu, lai atrisinātu problēmu, kas bija saistīta ar interfeisu versijām. Tagad Google lieto PB formātu, lai glabātu un apmainītos ar datiem Google iekšējās aplikācijās, un publicēja C++, Java un Python realizācijas algoritmus PP formātam, kā atklāta pirmkoda programmatūru.

Google kompānija raksturo (skat. [Goo2012c]) savu piedāvāto formātu kā elastīgu, efektīvu, automatizētu mehānismu strukturēto datu serializācijai. PB salīdzina ar XML, bet kā mazāku, ātrāku un vienkāršāku.

PB formāts pašā sākumā tika izstrādāts, lai strādātu ar servera pieprasījuma/atbildes protokolu. Pirms šī formāta bija cits risinājums, kurš neatbalstīja interfeisa vecās versijas

atbalstu. Rezultātā bija neglīts, grūti modificējošs kods. Vecā tipa protokolus parasti sauc par *stingri formatētiem protokoliem*. Stingri formatēti protokoli bija pārāk sarežģīti, lai tos modificētu, tas deva iedvesmu jauna formāta izstrādei. Lai ieviestu jauno protokolu, bija nepieciešams no sākuma pārliecināties, ka visi serveri sapratīs jauno formātu, un tikai pēc tam var sākt tos lietot. PB formāts tika izstrādāts, lai atrisinātu tāda tipa problēmas, kas bija sākotnējam formātam. PB izveidošanas laikā bija izvirzītās divas galvenās prasības:

1. Jābūt vienkāršai iespējai ieviest jaunus laukus uz servera, lietotāji, kuriem nav nepieciešami jauni lauki, var vienkārši tos ignorēt un strādāt tikai ar daļu no laukiem.
2. Jauns formāts ir pietiekošs, un jābūt vienkārši realizētām dažādām valodām (C++, Java utt.)

Katrs PB ziņojums sastāv no pāriem *vārds/nozīme*. Pats formāta uzbūves princips ir tāds, ka katra ziņojuma tips sastāv no viena vai vairākiem unikāli numurētiem laukiem, un katrs lauks sastāv no vārda un nozīmes. Iespējamie lauku tipi ir: *skaitlis, loģiskais, rinda, baitu kopa* utt. Nākamais solis ir PB ziņojums, kas ir kompilēts uz programmēšanas valodu (piemēram, C++, Java utt.) pamata, radot datu klasi.

PB formāta galvenais izmantošanas nolūks ir serveru datu apmaiņas nodrošināšana. Lielākais ierobežojums, kas neļauj pilnvērtīgi aizvietot citus populāras datu apmaiņas formātus (piemēram, XML un JSON), tā ir nepieciešamība PB ziņojuma struktūru kompilēt atsevišķi (.proto datne). Un bez šīs nokompilētās datu struktūras PB izmantot nav iespējams.

Tomēr ir priekšrocības, kas motivē izmantot PB analogu vietā, salīdzinājumā ar XML formātu PB ir (fakti ņemti no [Goo2012c]): *vienkāršāks, trīs līdz desmit reizes mazāks, divdesmit līdz simts reizes ātrāks, ir mazāk divdomīgs un ģenerē datu klasi, kuru vienkārši izmantot programmēšanā*.

PB formātu lieto, lai nodrošinātu informācijas pārraidi no vienas sistēmas uz citu, piemēram, dažādu platformu integritātei (t.s. mobilajās platformās). Viens no populārākajiem izmantotiem pielietojuma veidiem ir Google iekšējo servisu iedarbināšana [Goo2012c].

2.4. Kopsavilkums un secinājumi

Šajā nodaļā ir identificētas tehnoloģijas, kas ir izmantotas OIĪSS gadījumā. Šis nodaļas galvenie jaunie teorētiskie rezultāti ir izvirzītie tehnoloģiju salīdzināšanas parametri, kas ietekmē koncepcijas izstrādi nākotnē. Informācija ir apkopota, veicot tehnoloģiju salīdzinājumu. Atsevišķi ir apskatīti datu apmaiņas standarti, lai identificētu tos datu apmaiņas standartus, no kuriem tiks izvēlēts datu apmaiņas standarts koncepcijas praktiskās realizācijas laikā.

Nodaļā paveiktais:

- ir identificētas tipiskas tehnoloģijas, kas ir izmantotas OIĪSS risinājumos;
- ir izvirzīti tehnoloģiju parametri, kas ir jāņem vērā koncepcijas izveidošanas laikā;
- ir apskatītas iepriekš identificētas tehnoloģijas, apkopojot konkrētus tehnoloģiju parametrus;
- ir apskatīti datu apmaiņas formāti.

Sasniegtie rezultāti ir šādi:

- ir identificētas 16 tehnoloģijas un tehnoloģiju modifikācijas (*svītrkods (Code128)*, *svītrkods (DataMatrix)*, *svītrkods (QR Code)*, *RFID (pasīvais)*, *RFID (daļēji aktīvais)*, *RFID (aktīvais)*, *NFC*, *LAN*, *Bluetooth*, *IrDA*, *GSM*, *GPRS*, *Wi-Fi*, *WiMAX*, *GPS* un *sensori*), kuras var izmantot kopā ar objektu individuālajās īpašībās sakņotām sistēmām;
- ir ieviests tehnoloģiju grupas parametrs, kas klasificē tehnoloģiju pēc izmantošanas nolūkiem. Tiek iedalītas trīs grupas: *objektu identificēšanas tehnoloģijas*, *datu pārraides tehnoloģijas* un *uz sensoru bāzētas tehnoloģijas*.
- ir izvirzīti 8 tehnoloģiju parametri, kas definēto OIĪSS kritērijus pamatā, kurus ir nepieciešams ņemt vērā tehnoloģijas izvēlei OIĪSS risinājuma izstrādes laikā: (i) *tehnoloģiju grupa*, (ii) *darba rādiuss*, (iii) *datu pārraides ātrums*, (iv) *darba frekvence*, (v) *tiešās redzamības nepieciešamība*, (vi) *ieviešanas izmaksas*, (vii) *tehnoloģiju iedarbināšanas mehānismu sarežģītība* un (viii) *datu aizsardzība*;
- ir apkopota informācija tabulas veidā par 16 tehnoloģijām un tehnoloģiju modifikācijām 8 tehnoloģiju parametru griezumā;
- ir identificēti 4 galvenie datu apmaiņas formāti, kurus plānots izmantot DVS koncepcijas praktiskās realizācijas laikā formāta izvēlei: *ar norobežotāju atdalītas vērtības*, *paplašināmās iezīmēšanas valoda (XML)* un *JavaScript objektu notācija (JSON)*, *protokolu buferis (PB)*.

Galvenie secinājumi ir šādi:

- Izpildot četru objektu individuālo īpašību sakņoto sistēmu analīzi (*autonomie kooperatīvie loģistikas procesi; tīrīšanas, remonta un apkopes servisi, viedās ēkas, monitoringa, informēšanas un regulēšanas sistēmas*) ir identificētas 16 tehnoloģijas un tehnoloģiju modifikācijas, ko var izmantot objektu individuālajās īpašībās sakņotās sistēmās.

- Detalizēti apskatīts (*i*) tehnoloģiju parametrs par *tehnoloģiju grupu*. Viena tehnoloģija var atbilst vienai grupai atsevišķi vai atbilst 1. un 2. grupai vienlaicīgi, un katrai tehnoloģijai ir iespējamas dažādas tehnoloģiju parametru vērtības.
- Neviena no apskatītajām tehnoloģijām neatrisina (*viii*) kritēriju par *datu konfidencialitātes, integritātes un pieejamības nodrošināšana*. Šo kritēriju nepieciešams atbalstīt programmatūras līmenī.
- Tikai 1. un 2. grupas tehnoloģijas ir iespējams izmantot kā pamata tehnoloģijas koncepcijas praktiskās realizācijas laikā. 3. grupas tehnoloģijas ir izmantotas kā papildu informācijas avoti.

Šīs nodaļas galvenie jaunie teorētiskie rezultāti ir izvirzītie tehnoloģiju parametri, kas ietekmē koncepcijas izstrādi nākotnē. Uz šo tehnoloģiju parametru bāzes ir apkopota informācija par tipiskām objektu individuālajās īpašībās sakņotās sistēmās izmantotajām tehnoloģijām.

3. Konceptijas apraksts

Ņemot vērā definētos OIĪSS kritērijus, šajā nodaļā ir piedāvāta objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcija.

3.1. apakšnodaļā ir dots koncepcijas vispārīgs apraksts. DARSIR koncepcijas pamatelements ir resurss, 3.2. apakšnodaļā tiek dota resursu iekšējā struktūra un resursu klasifikācija. 3.3. nodaļā tika piedāvātas resursu iespējamie mijiedarbības scenāriji. Objektu individuālās īpašības ir definētas ar atribūtu un noteikumu palīdzību. Tā kā jebkurai īpašībai iespējamas dažādas aprakstu shēmas, 3.4. apakšnodaļā ir klasificētas īpašības un definētas atribūtu un noteikumu grupas. 3.5. apakšnodaļā ir dota koncepcijas pārbaude izvirzīto kritēriju atbilstībai. Nodaļas beigās ir dots kopsavilkums un secinājumi.

3.1. Konceptijas vispārīgais apraksts

Objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcijas pamatprincipi pirmoreiz tika publicēti [Zur2007] publikācijā. Koncepcija ir nosaukta par *dati un noteikumi glabāti resursā* (ang. *Data And Rules Saved In Resource, DARSIR*). Tā kā DARSIR saīsinājums visos pētījumos ir dots angļu valodā (pilnais saraksts ar autora publikācijām ir pieejams Ievadā), tad turpmāk šeit arī tiks izmantots šis saīsinājums.

IS, kas izstrādāta pēc DARSIR koncepcijas principiem, dati netiek glabāti centralizēti, bet sadalīti starp fiziskiem objektiem (IS dalībniekiem). Tas nodrošina atteikšanos no klasiskajai datubāzei nepieciešamā tehniskā nodrošinājuma (piemēram, serveris, tīkls utt.). Šī koncepcija piemērota gadījumos, kad IS iespējams realizēt, izmantojot decentralizētas datu glabāšanas un pārvaldības principus.

DARSIR koncepcija bāzējas uz objektorientētas pieejas, saskaņā ar kuru jāizskata dati, kuri jāizmanto, lai sistēma pildītu savu uzdevumu. DARSIR koncepcija par tādiem datiem uzskata objektus. DARSIR koncepcijas objekts ir nosaukts par resursu. Resurss ir jebkurš dzīvās vai nedzīvās dabas objekts, kas ir iesaistīts informācijas sistēmas darbības procesā. Resursa dati tiek glabāti birkā. Birka ir koncepcijas izmantotās tehnoloģijas sastāvdaļa. Birka resursam ir piesaistīta vai iebūvēta tajā. Piemēram, RFID tehnoloģijas birka ir *RFID birka* (ang. *RFID tag*). Svītrkodu tehnoloģijā tā ir *divu dimensiju svītrkods*.

Resursa piedāvātā struktūra ir aprakstīta, lietojot kopu teorijas pierakstu. Ja B ir birka:

$$B = \{S, D\},$$

kur S ir sensoru kopa, kas satur sevī no 1 līdz i sensorus:

$$S = \{S_1, S_2, \dots, S_i\}$$

un D ir dokuments, tas ir datu glabāšanas formāts, kas nodrošina objektu īpašību (atribūti) un metodes (noteikumi) glabāšanu strukturētā veidā.

Savukārt, dokuments D :

$$D = \{A, N\},$$

kur A ir resursa atribūtu kopa, kas satur atribūtus no 1 līdz i :

$$A = \{A_1, A_2, \dots, A_i\}$$

un N ir resursa noteikumu kopa, kas satur noteikumus no 1 līdz i :

$$N = \{N_1, N_2, \dots, N_i\}.$$

Pašā birkā ir realizēta saite starp sensoriem un dokumentu. Ar sensoru saņemtajām vērtībām (piemēram, temperatūra pēc Celsija) iespējams operēt (piemēram, saņemt, apstrādāt, nodot utt.), izmantojot noteikumus, kas definēti dokumentā.

3.2. Dokumenta iekšējā struktūra

Dokuments tas ir resursa datu aprakstu veids, kas apraksta resursa atribūtus un noteikumus. Skaidrojumiem par dokumenta iekšējo struktūru par pamatu ir ņemts piemērs no loģistikas.

Eksistē konteineris, kurš tiek izmantots preču pārvietošanai. Konteinerim ir DARSIR dokuments, kurā tiek glabāti konteineru atribūti un noteikumi.

Atribūti

Objekta īpašības, dati un atribūti ir sinonīmi DARSIR koncepcijas kontekstā. Pēc formālās definīcijas [Lza2011] *dati* ir formalizētā veidā attēlota tekstuāla, skaitliska, grafiska, video vai audio informācija, kuru lietotājs vai datu apstrādes ierīces var interpretēt, apstrādāt vai pārsūtīt. Resursam ir fizisku objektu īpašības. Piemēram, ir iespējams definēt datus, kas attiecas uz konteineri – tam var būt šādi fiziski parametri: augstums, platums, ietilpība utt. Konteineros var ievietot citus objektus, piemēram, preces.

Atribūtu sastāvdaļas un izmantošanu var aprakstīt, lietojot kopu teorijas pierakstu. *Atribūts* ir objekta īpašību prezentācija dokumentā. Atribūtiem eksistē daudzi paveidi. Lai vienkāršotu standartizācijas procedūru, atribūtus var klasificēt – sadalīt *atribūtu grupas*. Ja A ir atribūtu grupa, kas s ietver sevī viena tipa atribūtus no 1 līdz i :

$$A = \{A_1, A_2, \dots, A_i\}$$

savukārt, A_i ir atribūts:

$$A = \{NS, V, P\},$$

kur NS ir atribūta nosaukums,

V ir atribūta vērtība

un P ir atribūta pazīmes kopa, kas satur sevī pazīmes no 1 līdz i :

$$P = \{P_1, P_2, \dots, P_i\}.$$

Piemēram, konteineram ir fiziskie parametri. Fizisko parametru atribūtu grupa ietver sevī atribūtus, kas apraksta konteīnera augstumu, platumu, ietilpību utt. Atribūtam, kas apraksta konteīnera augstumu ir nosaukums *con_height* (NS), vērtība 2.4 (V) un pazīme *m* (P). Dotajā gadījumā pazīme norāda, ka augstuma vērtība ir norādīta metros.

Noteikumi

Resursa *noteikums*, objektorientētās pieejas kontekstā, ir *metode*. Objektorientētajā programmēšanā tiek definētas divas lielas sastāvdaļas, kuras darbina metodes: tās ir mainīgie un metodes. DARSIR koncepcijā piedāvāto noteikumu struktūru var aprakstīt, lietojot kopu teorijas pierakstu. Ja *N* ir noteikums:

$$N = \{M, PR, FN, T\},$$

kur *M* ir mainīgie,

PR ir procedūras,

FN ir funkcijas

un *T* ir *trigeris* (ang. *trigger*) – ir operāciju kopa, kas tiek glabāta resursā un tiek palaista automātiski, kad iepriekš ir definēts notikums.

Atsevišķs paskaidrojums ir nepieciešams *trigerim*. Šis termins tiek lietots aktīvo datubāzu pieejai (piemēram, [Dat2004]). DARSIR trigerim iespējams izmantot līdzīgus aktīvas datubāzes trigeru darbības principus. Trigera struktūru var aprakstīt, lietojot kopu teorijas pierakstu. Lai *T* ir trigeris:

$$T = \{NT, NS, D\},$$

kur *NT* ir *notikums* – operācija, kas notiek ar atribūtiem (atribūtu vērtības izmaiņa, atribūtu vērtības nozīmes saņemšana, atribūtu pievienošana, atribūtu dzēšana utt.) vai notiek ar procedūrām (procedūru palaišana, procedūru pievienošana, procedūru izmaiņa utt.),

NS ir *nosacījums* – loģiska izteiksme, kura var pieņemt Bulu nozīmi (paties vai nepaties). Un ja Bulu vērtība ir patiesa, tad darbība tiek palaista. Ja netiek definēts nosacījums (nosacījums nav norādīts), tad tiek uzskatīts, ka Bulu nozīme ir patiesa un darbība jāpalaiž

un *D* ir *darbība* – operatoru kopa, kas tiek automātiski izpildīta, ja nosacījums ir paties.

Turpinot apskatīt augstāk minēto piemēru ar konteīneri no loģistikas sfēras, varam apskatīt tipiska trigeru izmantošanas piemēru. Ir definēts, ka ir jāizveido trigeri, kas piefiksē konteīnera pārvietošanas vēsturi. Pieņemsim, ka konteīnera dokumentā ir atribūts, kas apraksta vietu, kur konteīneris šobrīd atrodas. Tādā gadījumā iespējamais *notikums* (NT) ir jaunas vērtības piešķiršana šim atribūtam, bet *nosacījums* (NS) ir fakts, ka jaunā vērtība un

vecā vērtība nav vienādas. *Darbība* (D) ir fiksēt pašreizējo datumu un laiku kopā ar pašreizējo konteinera atrašanās vietu.

Resursu grupas

Nodrošināt katram resursam aparatūras iespēju fiziski mijiedarbojoties ar citiem resursiem ir ekonomiski neizdevīgi. IS eksistē aktīvie un pasīvie darbības elementi. Ņemot vērā šo principu, var aprakstīt resursu grupu klasifikāciju, lietojot kopu teorijas pierakstu. Ja R ir resurss:

$$R = \{PR, AR\},$$

kur PR ir *pasīvie resursi* (ang. *Passive Resource, PR*) – resursi, kuros tiek glabāti dati un piesaistīti sensori (tāda veida resurss tikai glabā iekšējos datus),

un AR ir *aktīvie resursi* (ang. *Active Resource, AR*) – resursi, kuros tiek glabāti dati un kuriem papildus piemīt arī funkcionalitāte. Aktīvus resursus var sadalīt:

AR ir:

$$AR = \{ASR, ADR\},$$

kur ASR ir *aktīvie statiskie resursi* (ang. *Active Static Resource, ASR*) – informācija tiek glabāta statiskā veidā (statiskas instrukcijas) un to var mainīt tikai citi resursi vai ierīces.

un ADR ir *aktīvie dinamiskie resursi* (ang. *Active Dynamic Resource, ADR*), kuri ne tikai glabā informāciju, bet arī ir spējīgi tiešā veidā ietekmēt citus resursus (piemēram, saņemt informāciju, palaist resursus utt.). Šeit tipiskākais piemērs ir RFID skeneris. Ja pieņem, ka resurss ir prezentēts kā RFID tags, tad RFID skeneris var lasīt un modificēt datus, kas atrodas resursos (RFID tagos).

DARSIR risinājumu iedarbināšana tiek nodrošināta ar ADR elementiem. ADR analogu ir iespējams atrast MAS teorijā (skat. [Woo2001, Jen1998]), kur to sauc par *aģentu*. Ņemot vērā 2. nodaļā apskatītās tehnoloģijas, ADR tas ir skeneris, kurš DARSIR koncepcijas gadījumā ir aktīvais elements, kas nodrošina resursu iekšējo funkcionalitāti gadījumā, ja tie atrodas tā uztveršanas zonā.

3.3. Resursu mijiedarbības scenāriji

DARSIR koncepcijas pamatelements ir resurss. Resursu mijiedarbības ir nodrošinātas caur aktīvajiem resursiem. ASR var glabāt sevī noteikumus, bet tikai ADR var palaist noteikumus. ADR var nosaukt par skeneri. Skeneris var mijiedarboties ar citiem resursiem, ja šie resursi atrodas tā uztveršanas zonā. Šīs apakšnodaļas mērķis ir parādīt trīs resursu mijiedarbības scenārijus.

Detalizētai resursu mijiedarbības scenāriju izmantošanas izskaidrošanai izmantosim piemēru no loģistikas, kas ir aprakstīts, lietojot kopu teorijas pierakstu. Ja N ir noliktava:

$$NL = \{DR, K, KST\},$$

kur DR ir durvis, ADR,

K ir konteineris, ADR,

un KS ir kastes kopa, kas satur sevī kastes no 1 līdz i:

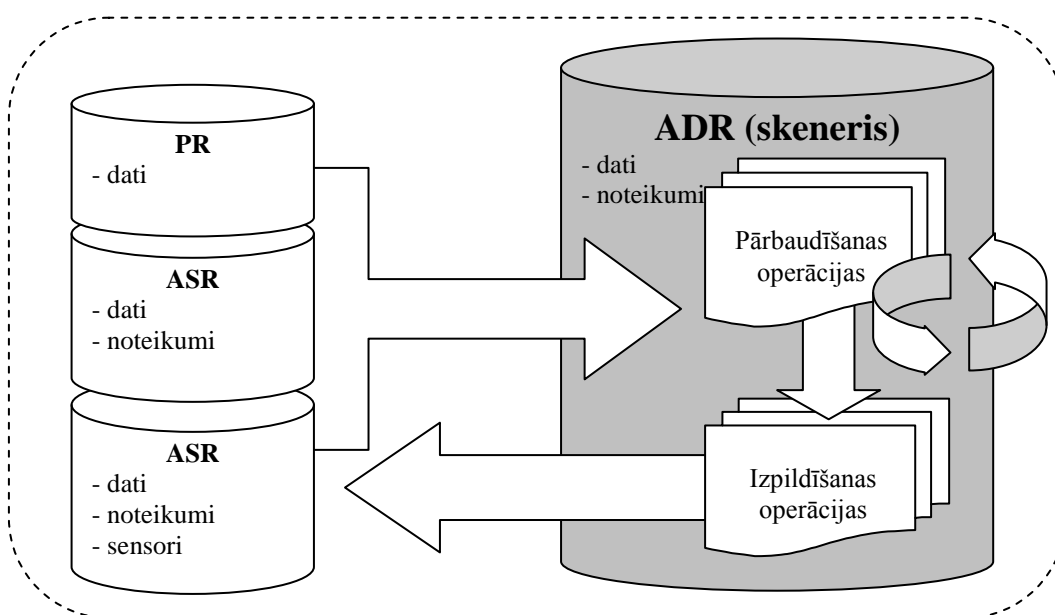
$$P = \{KS_1, KS_2, \dots, KS_i\},$$

kur KS_i ir kaste, ASR kas satur sevī preces (pasīvie resursi) no 1 līdz j:

$$P_i = \{PR_1, PR_2, \dots, PR_j\}.$$

Viena ADR mijiedarbības scenārijs

ADR pamata raksturojums ir autonoma darbība ar iespēju pašam pieņemt lēmumus uz vietas, pamatojoties uz pieejamajiem datiem. Pēc raksturojuma un funkcijas ADR ir līdzīgs aģentam no MAS. ADR darbības principu definēšanai un praktiskai realizācijai tiek izmantota aģentu izmantošanas pieredze no MAS (skat. [Woo2001, Jen1998]). Visvienkāršākais darbības scenārijs, kad ir viens ADR, ir parādīts 3.1. attēlā. ADR nolasa visus datus un noteikumus savā atmiņā no PR un ASR, kas atrodas tā nolaišanas diapazonā. No PR saņemti atribūti. No ASR iespējams saņemt arī noteikumu. Ja ASR noteikumi nosaka, ka jāizpilda iepriekš definētas darbības, tad ADR izpilda tas. Ja ADR iekšā ir ierakstīti noteikumi, tie būs arī izpildīti.



3.1. attēls. Viena ADR mijiedarbības scenārijs

Viena no praktiskajām realizācijām ir svītrkodu tehnoloģijas iebūvētie risinājumi. Eksistē datorsistēma ar pieslēgto svītrkodu skeneri. DARSIR koncepcijas terminos tas ir viens ADR. Pēc katra PR vai ASR nolasišanas atmiņā ADR izpilda noteikumus un pieņem lēmumus, kā rīkoties ar katru no resursiem.

Pamatoties uz iepriekš doto piemēru no loģistikas sfēras, konteineri ir domāti, lai pārvietotu kastes (ASR) ar precēm (PR). Kad kastes ar precēm ir ievietotas konteinerā, tiek palaisti noteikumi, kuri ir ierakstīti uz kastes un preces. Ja nav pretrunas starp konteineru uzdevumu un kastes iekšējiem noteikumiem, notiek transportēšana.

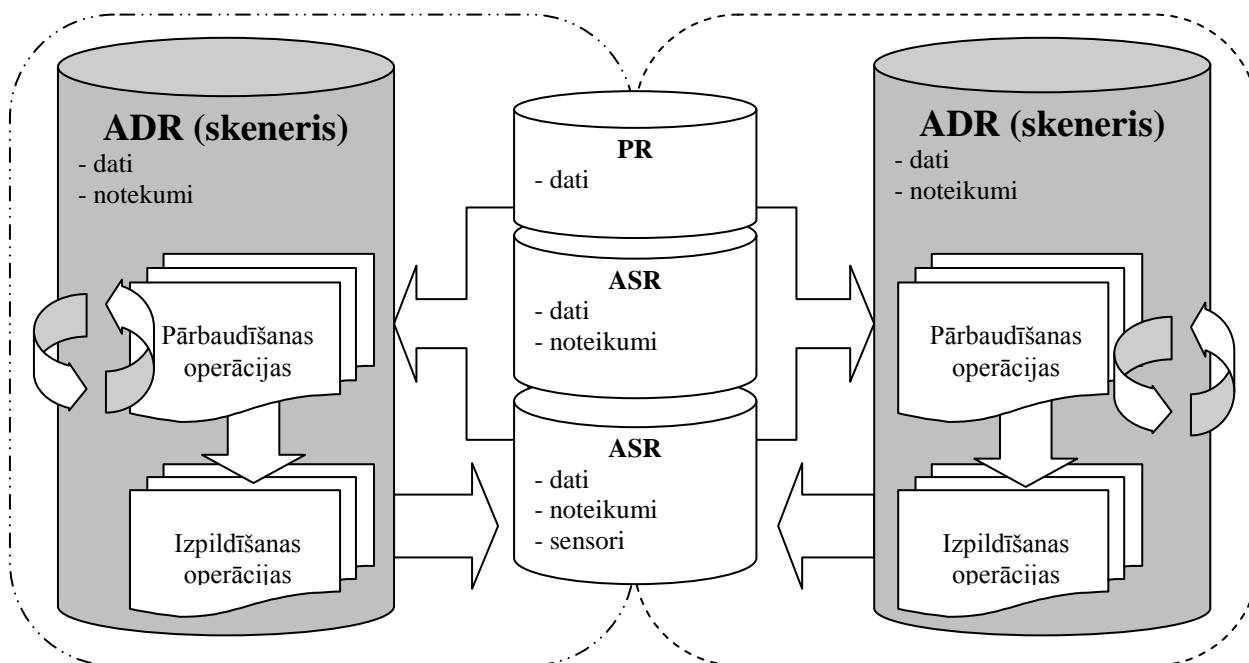
Konkurentu ADR mijiedarbības scenārijs

Viena ADR mijiedarbības scenārijs ir pamata risinājums, bet eksistē gadījumi, kad divi vai vairāki ADR atrodas kopējā darbības zonā. Tādā gadījumā vairāki ADR var vienlaicīgi ietekmēt ne tikai PR un ASR, bet arī citu ADR. Viens ADR var ietekmēt citu ADR, ja tam pietiek piekļuves tiesības.

ADR nolasa PR un ASR noteikumus, apvieno ar saviem noteikumiem un, palaižot noteikumus, realizē operācijas ar citiem resursiem asinhronā režīmā. PR un ASR noteikumus nav iespējams palaist uzreiz no pašiem resursiem. Ja nav iespējams palaist noteikumu, piemēram, notiek darbs ar šo resursu no cita ADR, tad noteikums gaida, kamēr nebūs iespējams to izpildīt. Noteikumu tiek apturēts pašā ADR, kurš vēlas tos aizpildīt.

Katrs ADR pilda savu unikālu uzdevumu. Arī gadījumā, kad eksistē divi ADR ar vienādiem vai līdzīgiem uzdevumiem, tiek paredzēts, ka tie paši strādā paralēli un mijiedarbība starp tiem nav paredzēta.

Divu ADR vienlaicīgas mijiedarbības scenārijs ir attēlots 3.2. attēlā.



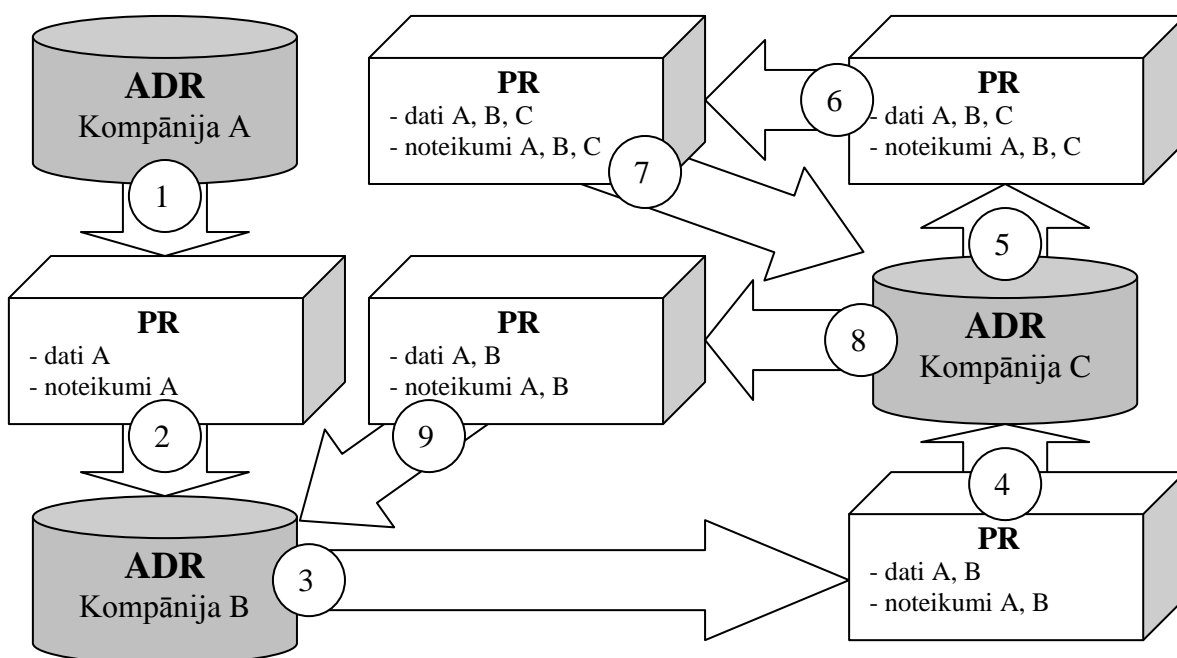
3.2. attēls. Divu ADR vienlaicīgas mijiedarbības scenārijs

ADR strādā nepārtraukti un pats var atkārtoti piemērot resursiem savus noteikumus. Noteikumi projektēti tādā veidā, lai būtu iespējams tos palaist atkārtoti. Iespējami gadījumi, kad resurss atrodas divu ADR uztveršanas zonā, bet paši ADR neredz viens otru. Vairāku ADR tuva mijiedarbība ar vienādiem uzdevumiem iespējama, bet nav vēlama, t. k. palielina sistēmas noslodzi (vairākas reizes vienlaicīgi jāpalaiž tādi paši noteikumi).

Apskatīsim doto scenāriju ar piemēru no loģistikas sfēras. Eksistē divi ADR, tas ir konteineris un durvis. Katram ir savs unikālais uzdevums. Konteinerim ir uzdevums pārvietot kastes ar precēm. Durvīm – nodrošināt durvju atvēršanu un aizvēršanu. Iespējams gadījums, kad paletē (ASR) ir noteikums, kas prasa papildu prasības transportēšanai, kurus nenodrošina konteineris. Uz šo noteikumu var reaģēt durvis (ADR): aizvērt durvis un neļaut konteinerim pārvietot šo kasti ārpus noliktavas.

Dažādu ADR secīgas mijiedarbības scenārijs

Tieša komunikācija starp dažiem ADR nav obligāta. Visa komunikācija vai cita veida mijiedarbība notiek caur resursiem (parasti PR un ASR). ADR var papildināt resursu ar datiem un noteikumiem. Līdzīgs mehānisms eksistē uz objektiem orientētā pieejā. Iespējams veidot sarežģītus objektus, izmantojot saliktu objektu konstruktorus. Pēc tam, kad šie dati nav vajadzīgi, tos iespējams dzēst no resursa. Piemēram, ir konteineris, kas paredzēts loģistikas procesa kastes pārvietošanai (skat. 3.3. attēlā).



3.3. attēls. Dažādu ADR secīgas mijiedarbības scenārijs

3.3. attēlā katrs elements ir viens solis. Ar bultiņām ir norādīta pārvietošana starp soļiem, ar numuru ir atzīmēts soļu kārtas numurs. Konteineram ir savi dati (atribūti un

noteikumi), kuriem sākuma vērtības ir definētas no ražotāja A. Kad konteineru sāk izmantot loģistikas kompānija B, tā pievieno savu daļu no datiem (jaunus atribūtus un noteikumus). Vienu no pakārtotajiem loģistikas procesiem var realizēt cita loģistikas kompānija C, kas pievieno savus datus (atribūtus un noteikumus), kuri ir vajadzīgi, lai apmainītos ar informāciju starp kompānijas C dalībniekiem. Kad kompānija C pabeidz savu daļu no darba un atgriež konteineru kompānijai B, kompānijas C dati (atribūti un noteikumi) kļūst nevajadzīgi un tiek dzēsti no konteinerā.

3.4. Atribūtu un noteikumu grupas

Resurss ir aprakstīts DARSIR risinājumā ar atribūtiem un noteikumiem. Vienam atribūtam vai noteikumam ir iespējamas dažādas apraksta shēmas. Viens no veidiem, kā atvieglot darbu ar resursiem, ir to klasificēšana un standartizēšana. Šīs apakšnodaļas ietvaros ir veikta atribūtu un noteikumu klasificēšana, sadalot tos grupās.

Dažādos informācijas avotos [Lah2005, Hei2005, Sar2010, Bro2001b, Bro2001d, Mil2008, Hun2007] ir norādīti fizisko objektu aprakstu principi; pamatā tie ir saistīts ar loģistiku, izmantojot RFID un svītrkodu tehnoloģijas. Ir norādīti dažādi principi fizisko objektu klasificēšanai, visvairāk atsaucoties uz *PML*.

Fizikālā iezīmēšanas valoda (ang. *Physical Markup Language, PML*) – XML modifikācijas formāts, kas nodrošina saglabāšanas un apmaiņas formātu par fiziskajiem objektiem, un tā tiek aprakstīta [Aut2002] kā vienkārša, vispārīga valoda fizisku objektu aprakstīšanai, uzraudzības un kontroles lietošanai fiziskajā vidē – sevišķi internetā. Tiek uzskatīts, ka PML ir iespējams izmantot industriālos, komerciālos un klientu lietojumos. Galvenokārt to izmanto loģistikas uzdevumiem.

Atribūtu un noteikumu grupēšana ir organizēta pēc izmantošanas sfērām. Atribūtiem un noteikumiem ir vienādas grupas un tie pieder vienādām grupām. Pie konkrētas atribūtu grupas ir pievienoti noteikumi, kas nepieciešami šīs grupas atribūtu iedarbināšanai. Par grupēšanas pamatu tika ņemti RFID un svītrkodu bāzes standarti un darbības principi, kā arī pētījumi par PML valodu no [Sar2000, Bro2001a, Bro2001b, Bro2001c un Bro2001d], adaptējot tos objektu individuālajās īpašībās sakņotām sistēmām.

Piedāvātais grupu sadalījums ir dots 3.1. tabulā.

3.1. tabula

DARSIR koncepcijas atribūtu un noteikumu grupas

Nosaukums	Pamatojums
<i>Resursa identifikators</i>	Resursam ir unikālais identifikators, pēc kura iespējams viennozīmīgi atšķirt vienu resursu no resursu kopas. Eksistē dažādas metodes fizisko objektu identificēšanai. Dažas no šīm metodēm ir piemērotas resursu identificēšanai.
<i>Individuālās īpašības</i>	Resursam eksistē individuālas īpašības. Tipiskākās individuālās resursa īpašības ir fiziskie parametri (garums, augstums, svars utt.), materiāla īpašības (plastmasa, metāls utt.), ķīmiskais sastāvs (kalcijs, nātrijs utt.) utt. Definēts kā strādāt ar šīm īpašībām, un kā iespējams piemērot fundamentālās bāzes mērvienības, tādas kā garums, masa, temperatūra utt.
<i>Saistības</i>	Resursam ir papildu parametri, kas resursu saista ar citiem objektiem vai vietām. Parasti tie ir resursa īpašnieks (piemēram, fiziska persona, organizācija, preču zīme utt.) vai resursa atrašanās vieta (piemēram, pilsēta, rajons, pasta indekss, precīza adrese, telpu izvietojums ēkā utt.).
<i>Laiks</i>	Tas ir mērīšanas sistēmas daļa, kuru izmanto, lai definētu notikumu secību, lai salīdzinātu notikumu ilgumu un intervālu starp notikumiem. Cilvēkam vislabāk piemēroti tādi termini kā sekunde, minūte, stunda, diena, nedēļa, mēnesis, gads utt.
<i>Vēsture</i>	Fiziskā objekta dzīves cikla gaitā notiek izmaiņas. Nepieciešams mehānisms, kas nodrošina vēstures saglabāšanu. Šī grupa cieši saistīta ar laika grupu. Vēstures atribūtā ir fiksēts laiks un fiziskā objekta stāvoklis, kas bija fiksēts norādītajā laikā.
<i>Pozicionēšana</i>	Resursam iespējams definēt atrašanās vietu. Tādās nozarēs kā loģistika, šis ir viens no būtiskajiem atribūtiem. Pozicionēšanu daļa divās lielās grupās: ārējā vide un iekštelpa.
<i>Piekļuves tiesības</i>	Datu konfidencialitātei un drošības dēļ, izmantojot DARSIR koncepciju, jāregulē piekļuves tiesības resursiem. Objektu orientētai pieejai tās tiek nodrošinātas ar iekapsulēšanas principu atbalstu. Resursu iespējams ietekmēt ar atvērtiem noteikumiem (metodēm) un

	atribūtiem. Ar šo grupu ir nodrošināta piekļuves tiesību regulēšana.
<i>Klasifikācija</i>	Resurss ir fizisks objekts ar savām unikālajām individuālajām īpašībām. Kad resursu ir daudz, tiem grūti piemērot kopējas darbības. Viens no variantiem, kā atvieglot darbu ar resursiem, ir resursu klasificēšana un kategorizēšana. Jāparedz iespēja veidot sarežģītus resursus, izmantojot saliktu resursu konstruktorus.
<i>Konfigurācija</i>	Resurss var saturēt dažāda veida citus objektus. Tipisks piemērs ir konteineris, kurā ir kastes ar precēm. Ja IS nav paredzēts visus resursus nodrošināt ar birkām, tad iespējams tās informāciju ierakstīt pamata resursā. Tad informāciju ir iespējams izmantot IS, bet ar šo grupu palīdzību.
<i>Prezentācija</i>	Iepriekš aprakstītās resursu grupas, bija nepieciešamas, lai automatizētu darbības, kas saistītas ar resursu darbu. Lietotājam arī jānodrošina informācija par resursiem, kas nav saistīta ar fiziskajām resursa īpašībām. Tie ir dažādu veidu dati (teksta, video, audio utt.), kas raksturo resursu un palīdz informēt resursa patērētāju par resursa izmantošanas īpašībām. Tipiskie piemēri, kas atbilst resursu prezentēšanai: lietotāju apmācība, garantija, reklāma (teksts, attēli, audio, video utt.) utt.

3.5. Konceptijas pārbaude definēto kritēriju atbilstībai

Šajā nodalā ir aprakstīta DARSIR koncepcija vispārīgā līmenī. Pēc DARSIR koncepcijas vispārīga apraksta to ir jāpārbauda uz atbilstību izvirzītajiem astoņiem OIĪSS kritērijiem.

Pirmais (i) kritērijs par *skaitļošanas resursu pārvaldības sadali un decentralizāciju* ir nodrošināts, jo DARSIR koncepcija ir izstrādāta uz DDVS darbības principu bāzes. Visi dati attiecas uz fiziskiem objektiem, un katrā fiziskā objektā tiek glabāti dati par pašu objektu. Šis fakts arī nodrošina (iv) kritērija par *lokālas datu piekļuves nodrošināšanu* atbalstu. Nav nepieciešamas citas vietas informācijas glabāšanai.

Viens no pamata kritērijiem, kas balstās uz mūsdienīgiem risinājumiem, ir (ii) *standartu izplatīšanās*. Eksistējošās tehnoloģijas un mūsdienīgi standarti (piemēram, skat. 2.3. apakšnodaļā apskatītos datu apmaiņas formātus), tiek apskatīti 2. nodaļā. Apskatītie standarti ir izmantoti DARSIR koncepcijas izstrādē. (iii) *fizisko objektu nevienādīgumu atbalstīšana* ir nodrošināta ar datu glabāšanu mūsdienīgās datu apmaiņas formātu (piemēram, XML, JSON u.c.) modifikācijās.

DARSIR koncepcijai ir definēti *aktīvie resursi (AR)*, ar kuru palīdzību IT sistēmu ir iespējams sadalīt neatkarīgos autonomos risinājumos. Tas atbilst (v) kritērijam par *autonoma darbību*. Lielu sistēmu ir iespējams sadalīt atsevišķās daļās un ieviest IT sistēmu pakāpeniski. Tas dod tādas priekšrocības kā ieviešanas izmaksu sadale, ekonomiskā efekta iegūšana īsā laika periodā utt. Tas nodrošina (vii) kritēriju par *pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale*. Dažādu tehnoloģiju pielietošana DARSIR koncepcijai var dot priekšrocību izmantot eksistējošas tehnoloģijas. Tas kopā ar šajā rindkopā pieminētajiem faktiem nodrošina kritēriju (vi) par *nelielu risinājumu ieviešanas ekonomiskais izdevīgums*.

Pēdējam kritērijam (viii) par *datu konfidencialitātes, integritātes un pieejamības nodrošināšana* ir nepieciešami papildu pētījumi. Šo kritēriju daļēji iespējams nodrošināt ar konkrētas tehnoloģijas īpašībām (piemēram, skat. 2.3. tabulā 8. punktā). Kritēriju atbilstību iespējams nodrošināt ar programmatūras starpniecību, šis process detalizēti apskatīts nākamajās nodaļās.

3.6. Kopsavilkums un secinājumi

Šajā nodaļā vispārīgā līmenī ir definēti DARSIR koncepcijas pamatelementi, kas nepieciešami tās izveidošanai. Objektu individuālās īpašības ir iespējams saskaņot ar sistēmas izstrādi, tādējādi atrisinot jautājumu par īpašību glabāšanu pašā objektā. Tas savukārt sniedz iespēju lēmumu pieņemšanas brīdī, kas ietekmē pašu objektu, nodrošināt visu informāciju no paša objekta. Nodaļas mērķis ir izveidot DARSIR koncepciju un pārbaudīt koncepcijas atbilstību OIĪSS kritērijiem.

Nodaļā paveiktais:

- ir izstrādāta DARSIR koncepcijas resursa iekšējā struktūra un pamatelementi;
- ir klasificētas mijiedarbības shēmas, kā arī resursu, atribūtu un noteikumu grupas;
- ir pārbaudīta DARSIR koncepcijas atbilstība izvirzītajiem OIĪSS kritērijiem.

Sasniegtie rezultāti ir šādi:

- ir dotas definīcijas DARSIR koncepcijas pamatelementiem: *resurss, atribūts, noteikums, dokuments* utt.
- ir klasificēti un aprakstīti trīs resursu tipi: *pasīvais resurss, aktīvais statistiskais resurss* un *aktīvais dinamiskais resurss*.
- ir izveidoti trīs resursu mijiedarbības scenāriji: *vienas ADR (skeneris) mijiedarbības, konkurentu ADR mijiedarbības* un *secīgas ADR mijiedarbības*;
- ir izdalītas 10 atribūtu un noteikumu grupas: *resursa identifikators, individuālās īpašības, saistības, laiks, vēsture, pozicionēšana, piekļuves tiesības, klasifikācija,*

konfigurācija un prezentācija. Atribūtu un noteikumu grupēšana ir organizēta pēc izmantošanas sfērām.

Galvenie secinājumi ir šādi:

- Izstrādāta DARSIR koncepcija, kas bāzējas uz objektu orientētu pieeju un par objektu uzskata resursu, dati par kuru tiek glabāti birkā. Birka ir piesaistīta resursam vai iebūvēta tajā un ietver sensoru kopu un dokumentu, kas pārstāv datu glabāšanas formātu, t.i. nodrošina objektu īpašību (*atribūti*) un metožu (*noteikumi*) glabāšanu strukturētā veidā.
- *Atribūts* ir objekta īpašību prezentācija dokumentā. Atribūtiem eksistē daudzi paveidi, lai vienkāršotu standartizācijas procedūru, atribūtus vajag klasificēt – sadalīt tos *atribūtu grupās*. Ir izdalītas 10 atribūtu un noteikumu grupas, kas ir organizētas pēc izmantošanas sfērām.
- Resursa *noteikums* objektorientētās pieejas kontekstā ir metode, un DARSIR koncepcijā noteikums ir pārstāvēts ar *mainīgiem, funkcijām, procedūrām un trigeriem*. *Trigeris* – ir operāciju kopa, kas tiek glabāta resursā un tiek palaista automātiski, kad iepriekš ir definēts noteikums. Tālākajos pētījumos jāizskata iespējas un jāizstrādā mehānismi noteikumu realizēšanai.
- DARSIR koncepcijā visus resursus var iedalīt *pasīvajos resursos*, kuros tiek glabāti dati un kuriem ir piesaistīti sensori, un *aktīvajos resursos*, kuros glabājas ne tikai dati, bet kuriem piemīt arī resursa funkcionalitāte, atbilstoši kurai aktīvie resursi vēl iedalās *aktīvajos statiskajos resursos* un *aktīvajos dinamiskajos resursos*.
- Resursu mijiedarbība ir nodrošināta caur *aktīvajiem resursiem*. Tikai *aktīvais dinamiskais resurss* var palaist noteikumus un to var uzskatīt par skeneri. Skeneris var mijiedarboties ar citiem resursiem, ja šie resursi atrodas tā uztveršanas zonā.
- DARSIR koncepcijas galvenās priekšrocības: *lēmumu pieņemšanu var realizēt uz vietas; atbalstīta fizisko objektu daudzveidība un unikalitāte; paaugstinās datu privātuma līmenis; iespējamība atteikties no centralizētas datu glabāšanas un pārvaldības infrastruktūras nodrošinājuma.*
- Piedāvātā DARSIR koncepcija teorētiski pilnvērtīgi atbilst objektu individuālajās īpašībās izvirzītajiem kritērijiem. Ar nosacījumu, ka (*viii*) kritērijs par *datu konfidencialitātes, integritātes un pieejamības nodrošināšanu* jāapskata detalizētāk nākamajās nodaļās.

Šis nodaļas galvenais jaunais teorētiskais rezultāts ir izstrādātā DARSIR koncepcija, kas pārbaudīta, vai tā atbilst OIĪSS kritērijiem. Tā ietver sevī koncepcijas galvenās darba elementu definīcijas, resursa iekšējās struktūras definējumu, resursu mijiedarbības shēmas, kā arī atribūtu un noteikumu klasifikāciju.

4. DARSIR koncepcijas ieviešana

3. nodaļā ir definēta koncepcijas izveidošanas pamata informācija. Tas ir izdarīts vispārīgā līmenī, un tagad nepieciešams nonākt līdz koncepcijas realizēšanai. Pirms realizācijas ir nepieciešams savākt papildu informāciju, kas var ietekmēt realizāciju. Jādefinē risinājumam nepieciešamo tehnoloģiju izvēles metode. Balstoties uz šiem kritērijiem, jādefinē vispiemērotākā tehnoloģija un jāapskata tās tipiska realizācija. Īpaša uzmanība jāpievērš drošības un konfidencialitātes jautājumu risināšanai.

Vispirms 4.1. apakšnodaļā ir izvēlēti datu apmaiņas un glabāšanas formāti. Uz tādu formātu pamata tiek izveidota DARSIR koncepcijas programmēšanas valoda. 4.2. apakšnodaļā ir aprakstīta atribūtu un noteikumu grupas ieviešana. 4.3. apakšnodaļā ir definēti koncepcijas tehnoloģiju izvēles būtiski kritēriji. Pamatoties uz tehnoloģijas izvēles kritērijiem, ir piedāvāta un pamatota svaru koeficientu metodes pielietošana, ar kuras palīdzību izvēlētas piemērotākās DARSIR koncepcijas tehnoloģijas atkarībā no IS būtiskiem darba apstākļiem. 4.4. apakšnodaļā viena no tehnoloģijām ir apskatīta detalizēti, īpaša uzmanība vērsta uz implementāciju. 4.5. apakšnodaļā ir dots datu drošības jautājumu apskats, kam pamatā ir viena no vispiemērotākajām tehnoloģijām. 4.6. apakšnodaļā ir aprakstīti promocijas darbā veiktie pētījumi, kas attiecas uz praktisku DARSIR koncepcijas pielietošanu. Nodaļas beigās ir dots kopsavilkums un secinājumi.

4.1. Datu apmaiņas un glabāšanas formātu izvēle

Tiek apskatīti datu apmaiņas formāti, lai tos izmantotu kopā ar DARSIR koncepciju. Tika pieņemts, ka datubāzi izmantot nav efektīvi, jo mērķis ir izstrādāt decentralizētu DVS. DARSIR koncepcijai ir vajadzīgs saglabāt ne tikai datus, bet arī funkcionalitāti. Ir apskatīti šādi datu apmaiņas formāti:

- *Ar norobežotāju atdalītas vērtības* – tā ir teksta datne, kas domāta tabulāro datu (skaitļi un teksts) glabāšanai. Parasti izmantota CSV (ang. *Comma-Separated Values*) metode, ar komatiem atdalītas vērtības. Nav paredzēta funkcionalitātes glabāšanai;
- *Paplašināmās iezīmēšanas valoda* (ang. *eXtensible Markup Language, XML*) – viens no populārākajiem datu apmaiņas formātiem. Iespējams izmantot;
- *JavaScript objektu notācija* (ang. *JavaScript Object Notation, JSON*) – līdzīga XML valoda, iespējams paaugstināt ātrdarbību, ja salīdzina ar XML. Iespējams izmantot;

- *Protokolu buferis* (ang. *Protocol Buffers, PB*) – labs, universāls datu apmaiņas formāts, bet datu apmaiņas struktūrai jābūt iepriekš definētai. Nav elastīguma, grūti izmantot.

Koncepcijas realizēšanai ir piedāvāti divi datu apmaiņas un glabāšanas formāti:

1. *Resursu fizikāla iezīmēšanas valoda* (ang. *Resource Physical Markup Language, RPML*) – tā ir XML tipa valoda, kas izveidota speciāli DARSIR koncepcijai. Pirmoreiz šīs valodas definējums un apraksts publicēts [Zur2007].
2. *JavaScript objektu notācija resursam* (ang. *JavaScript Object Notation for Resource, JSONR*) – tā ir JSON tipa valoda, kas izveidota speciāli DARSIR koncepcijai.

RPML un JSONR datu apmaiņas formātus iespējams nosaukt par DARSIR datu apmaiņas formātiem. Darbā aprakstā, ja tiek izmantots termins DARSIR dokuments, tiek pieņemts, ka tas ir RPML vai JSONR datu apmaiņas formāta fails.

4.2. Atribūtu un noteikumu grupas ieviešana

3.1. tabulā ir definētas desmit atribūtu un noteikumu grupas. Katrai no šīm grupām ir definēts ieviešanas apraksts. Par pamatu tiek ņemti principi, kas definēti publikācijās [Sar2000, Bro2001a, Bro2001b, Bro2001c un Bro2001d].

Resursa identifikators

Resursam ir vajadzīgs savs unikālais *resursa identifikators* (ang. *Resource Identification, RI*). Resurss ir fizisks objekts, kuram iespējams piemērot tādas pašas identifikācijas metodes kā parastiem fiziskiem objektiem. Svītrkodu tehnoloģijai objektu identificēšanai parasti tiek izmantota parastā UPC. UPC-A standarta struktūra, tā ir parādīta 4.1 attēlā.



4.1. attēls. UPC-A standarta struktūra

UPC-A standarta shēmai ir definētas četras sadaļas: numurēšanas sistēmas (dots viens cipars, iespējams definēt 10 numurēšanas sistēmas); ražotāja identifikācijas numurs (doti pieci cipari, iespējams definēt 100,000 unikālus ražotājus); produkta tips (pieci cipari, iespējams definēt 100,000 unikālus produkta tipus) un pārbaudes simbols.

UPC-A tiek izmantots Amerikā, bet ražotāju jau ir vairāk par 100,000, tāpēc šo kodu mēģina adaptēt jauniem apstākļiem. Tiek definētas divas shēmas (detalizētāk skat. [GS12011]): 9-ciparu kompānijas numurs un 6-ciparu kompānijas numurs (skat. 4.2. attēlā).



4.2. attēls. GSI US: UPC standarta shēmas

RFID tehnoloģijai bija izveidots savs formāts unikālam identifikatoram. Tas ir 96 bitu identifikators. Viena no pirmajām šī identifikatora shēmām (skat. 4.3. attēlā) ir *elektroniskā produkta koda 1. tips* (ang. *Electronic Product Code Type I, EPC Type I*). EPC Type I formāta izstrādes pamatojums ir dots [Bro2001a] pētījumā.



4.3. attēls. EPC Type I struktūra (adaptēts no [Bro2001a])

EPC Type I sastāv no: sistēmas numura (8 biti, 256 iespējamās nozīmes), ražotāja numura (28 biti, nodrošina apmēram 268 miljonus ražotāju), objekta klases numura (24 biti, iespējams definēt 16 miljonus produkta tipus vienam ražotājam) un seriālā numura (36 biti, vienam produkta tipam iespējams definēt 68 miljardus objektu).

No [Far2011] definētajiem faktiem secināts, ka EPC Type I ir pietiekams mūsdienīgai situācijai, t.i. DARSIR koncepcijai. Bet tomēr ir vēl iespējams veidot EPK jaunas shēmas un paaugstināt iespējamo objektu skaitu jebkurai no EPK sastāvdaļām. Līdzīgi kā tas tika izdarīts GS1 US: UPC standarta shēmām.

Pašu RI nepieciešams iebūvēt pašā resursā. Nepieciešama universāla funkcija, kas atgriež konkrēta resursa RI. Tā funkcija ir neatkarīga no RI glabāšanas veida un ir universāla jebkurā glabāšanās veidā.

RI iekšā (piemēram, EPC Type I) ir iekodēti dažādu veidu informācijas bloki: sistēmas numurs, ražotāja numurs, objekta klase un sērijas numurs. Ir nepieciešamas iekšējās funkcijas, kas nodrošina atsevišķu informācijas bloku saņemšanu funkcijas veidā. Piemēram, tiek palaista funkcijas GetResourceNumber, kas pēc izpildīšanas atgriež resursa ražotāja numuru.

Individuālās īpašības

Resurss ir fizisks objekts, kas atšķiras no cita resursa ar savām individuālām īpašībām. Tipiskākās individuālās resursa īpašības ir fiziskie parametri (garums, augstums, svars utt.), materiāla īpašības (plastmasa, metāls utt.), ķīmiskais sastāvs (kalcijs, nātrijs utt.) utt.

Pirms aprakstīt fiziskās īpašības, ir jādefinē mērvienības, kas tiek izmantotas, lai viennozīmīgi detektētu izmantoto vērtību. Viena standarta ieviešana nodrošina paļāvību, novērš neskaidrības, vienkāršo datu komunikācijas un iekārtas, vieno noformējuma ziņā atšķirīgas sistēmas. Piemēram, eksistē atribūts, kas apraksta fiziska objekta svaru. Tiek ierakstīta vērtība 1. Bez papildu paskaidrojuma nav iespējams saprast fizisko objekta svaru, vai tas ir gramos, kilogramos vai tonnās. Ja tiek definēta mērvienība kilograms, to iespējams viegli pārveidot nepieciešamajās vienībās: 1000 gramos vai 0.001 tonnās.

Tiek veikts apskats [MsC1999,NIS1997,Tay2008], kādā veidā tiek atrisināta šī problēma. *Starptautiskajā svaru un mēru birojā* (fran. *Le Système International d'unités*) [Tay2008] kā arī citur, piemēram, *Standartu un Tehnoloģiju Nacionālajā institūtā* (ang. *National Institute of Standards and Technology, NIST*), eksistē viedoklis, ka ar septiņām fundamentālajām mērvienībām (skat. 4.1. tabulu) ir iespējams definēt visas pārējās mērvienības.

4.1. tabula

Septiņas fundamentālās bāzes mērvienības (adaptēts no [Tay2008])

Mērvienība	Nosaukums	Simbols
Garums (ang. <i>length</i>)	Metrs	m
Masa (ang. <i>mass</i>)	kilograms	kg
Laiks (ang. <i>time</i>)	sekunde	s
Elektriska strāva (ang. <i>electric current</i>)	ampērs	A
Temperatūra (ang. <i>temperature</i>)	kelvins	K
Vielas daudzums (ang. <i>amount of substance</i>)	mols	mol
Gaismas intensitāte (ang. <i>luminous intensity</i>)	candela	cd

Katrai fundamentālajai bāzes mērvienībai ir definēts daudzums 1. Katrai no septiņām fundamentālās bāzes mērvienībām ir definēts savs simbols, kurš apzīmē kādai mērvienībai atbilst vērtība. Piemēram, ir definēts atribūts, kuram vērtība ir 0.36 un simbols kg. Simbols kg nozīmē, ka esošajā atribūtā tiek glabāta masas nozīme un izmantotās bāzes vienības nozīme ir viens kilograms. Tad iespējams definēt, ka resursa svars ir 0.36 kilogrami vai 360 grami.

Līdzīgs mērvienības izmantošanas princips realizēts PML valodā (piemēram, tas atspoguļots [Bro2001b] pētījumā). PML valoda pielāgota industrijas, biznesa un patērētāja vajadzībām, tika uzskatīts, ka nepieciešams papildināt septiņas fundamentālās mērvienības ar četrām kopējām bezdimensiju mērvienībām: *daudzums* (vienībās), *leņķis* (radiānos), *proporcijas* (procentos) un *loģika* (patiess vai nepatiess). Papildu četras mērvienības ir atspoguļotas 4.2. tabulā.

4.2. tabula

Papildu četras bāzes bez dimensiju mērvienības (adaptēts no [Bro2001b])

Mērvienība	Nosaukums	Simbols
Daudzums (ang. <i>quantity</i>)	vienības	q
Leņķis (ang. <i>angle</i>)	radiāns	rad
Proporcija (ang. <i>proportion</i>)	procents	p
Loģika (ang. <i>logic</i>)	bula	b

DARSIR koncepcijai ir izmantotas visas vienpadsmit bāzes mērvienības. Viens no piemēriem, kur ir izmantotas trīs no četrām bāzes bezdimensiju mērvienībām, ir pārtikas produktu apraksts. Vairākās pasaules valstīs (lielākajā daļā no Eiropas valstīm, Amerikā utt.) pārtikas produktu ražotāji uz iepakojuma publicē informāciju par vitamīniem, kaloriju daudzumu un to atbilstību attiecībā pret cilvēkam ieteicamo dienas devu (parasti 100 gramu masā). Pieņemsim, ka pārtikas produkta svars ir 100 grami, kas satur 250 kalorijas, kas savukārt ir 10% no vidējās cilvēka dienas devas, un dienas deva netiek nodrošināta. Kalorijas atribūtam ir vērtība 250 (jo tik daudz kaloriju ir 100 gramu masā), *masas* mērvienības vērtība ir 0.1 (kaloriju skaits 100 gramu masā), *daudzuma* mērvienības vērtība ir 1 (viena vienība satur 1 kaloriju), *proporcijas* mērvienības vērtība ir 10 (10% no vidējās dienas devas) un *loģikas* mērvienības vērtība ir nepatiess (vidējās ieteicamā cilvēka dienas deva netiek sasniegta).

Tāda tipa atribūtam ir vajadzīgas funkcijas, kas var atgriezt izmantotās mērvienības un mērvienības kombinācijas. Piemēram, ir pārtikas produkts, kura sastāvā ir minerāli un vitamīni. Par katru minerālu un vitamīnu ir norādīts: nosaukums, masa un proporcija (no vidējās rekomendētās dienas devas). Ar funkciju *GetRIIMass* (atgriež resursa individuālo

īpašību „masa”) tiks atgriezta šī pārtikas produkta masa. Ar funkciju *GetRIIName* (atgriež resursa individuālo īpašību „nosaukums”), kura dod nosaukumu konkrētai īpašībai (piemēram, *Vitamīns C*). Ar funkciju *GetRIIProportion* (atgriež resursa individuālo īpašību „proporcija”) tiks atgriezta šīs īpašības proporcija pārtikas produkta vienā svara kilogramā.

Saistības

Parasti fiziskam objektam ir savs īpašnieks. Šis īpašnieks ir fiziska vai juridiska persona, organizācija, valsts utt. Īpašniekam ir tiesības veikt darbības ar šo objektu. *Saistības* definējums atbilstoši loģistikas nozarei definē [Bro2001d], ka tā ir *persona, kompānija, organizācija vai grupa, kurai pieder, vai kas pārvalda vai pārtrauga fiziskus objektus*.

Dzīves gaitā fiziskam objektam mainās saistības. Ir iespējams definēt saistību pamata veidus: ražotājs, transportētājs, pārdevējs un patērētājs. Piemēram, ja fiziskais objekts ir prece, tad apskatīsim tipiskas preces dzīves ciklu. Pirmā saistība ir *ražotājs*, kas izveido preci. *Transportēšanas kompānija* nodrošina preces pārvietošanu līdz pārdevējam. *Pārdevējs* pārdod preci *patērētājam*.

Parasti vienam fiziskam objektam ir reģistrētas vairākas vienlaicīgas *saistības*. Piemēram, ja prece atrodas pie patērētāja un iekšā ir ierakstītas citas saistības: ražotājs, pārdevējs utt. Vai gadījumā, ja fiziskais objekts pieder vairākiem īpašniekiem vienlaicīgi.

Atkarībā no īpašnieka tipa tiek definēti atribūti. Ja tā ir fiziska persona, tas ir vārds, uzvārds, personas kods, telefons, mājas adrese utt. Ja tā ir juridiska persona, tas ir nosaukums, reģistrācijas numurs, telefons, juridiskā adrese, fiziskā adrese utt. Īpašnieka atribūtus iespējams klasificēt pēc pamata datiem (piemēram, vārds, nosaukums utt.) un kontakta informācijas (piemēram, telefons, adrese utt.).

Saistību atribūtu pārvaldīšana notiek caur darbībām. Katrai darbībai jāzina, uz kuru īpašnieka veidu ir veikts pieprasījums un kas ir pats darbības veids. Piemēram, ja nepieciešams uzzināt preces īpašnieka telefonu, tad funkcijas nosaukums ir *GetEntityOwnerPhone*. Gadījumā, kad nepieciešams mainīt īpašnieka vārdu, tad funkcijas nosaukums ir *SetEntityOwnerName*.

Laiks

Laiks ir mērīšanas sistēmas daļa, kuru izmanto, lai definētu notikumu secību, lai salīdzinātu notikumu ilgumu un intervālu starp notikumiem. Cilvēkam vislabāk piemēroti tādi termini kā sekunde, minūte, stunda, diena, nedēļa, mēnesis, gads utt. Parasti tiek definēti formāti, kā pasniegt šo informāciju. Piemēram, 2012. gada 1. janvāris 10 stundas un 10 minūtes pēc *ddmmyyyy hh:mm* formāta, tas ir *01012012 10:10*.

Datorsistēmas (operētājsistēmās [Tra2000, Keg1984], programmēšanas valodās [Ora2011, Orl2004] utt.) ir pieņemts cits formāts iekšējai laika glabāšanai. Par pamata vienību ir pieņemta sekunde. Datumi un laiks ir prezentēti ar sekunžu skaitu.

Datumu reprezentācijai ir definēts 32-bitu vai 64-bitu skaits, kur tiek glabāts sekunžu skaits no 1970. gada 1. janvāra pusnakts pēc UTC standarta. Piemēram, 2012. gada 1. janvārim vērtība ir 1325436230. Laiks pirms datuma ir ar negatīvu vērtību, aiz datuma ar pozitīvu vērtību. Par pamatu tiek pieņemts, ka vienā diennaktī ir 86400 sekundes (60 sekundes 60 minūtēs un 24 stundās). UTC katru gadu ir papildus viena vai divas sekundes, kas tiek sauktas par „*otro lēcienu*” (ang. *leap second*). Papildus sekunde piesaistās pēdējās sekundes konkrētai minūtei. Un šī minūte parasti ir 31. decembrī un 30. jūnijā. Datorsistēmās papildus sekundes ne visi datori var atpazīt. Tādēļ, definēsim, ka laika atribūtam ir 86400 sekundes diennaktī.

Jāizvēlas datu glabāšanas tips starp 32-bitiem un 64-bitiem. 32-bitu vērtībā iespējams glabāt datumu līdz 2038. gada 19. janvārim. Pēc šī datumam vērtība kļūst negatīva un sāk uztvert mazāk par 1970. gada 1. janvāri. 64-bitu vērtībā iespējams glabāt datumu līdz 292277026596 gadam. Tiek pieļauts, ka tas ir pietiekami.

Tiek pieņemts, ka atribūtam „laiks” nozīme tiek glabāta kā absolūta vērtība nevis formula (aprēķinu metode). Šis ierobežojums ieviests, lai nodrošinātu ātrāku datu apstrādi. Piemēram, ja produktam ir dots derīguma termiņš, tad tas ir prezentēts kā gala datums, līdz kuram produktu ir iespējams realizēt, nevis ražošanas datums un laiks, cik ilgi vēl produktu iespējams izmantot.

Laika atribūta īstenošanai jādefinē iekšējā funkcionalitāte, kas nodrošina datuma prezentēšanu noteiktā formātā. Ir dažādu veidu funkcijas, kurās jāizmanto laika atribūts: pašreizējā laika definēšana, operācijas ar datumiem (maksimālā datuma atrašana, diennaktis/mēnešus/gadu skaits starp diviem datumiem, utt.), operācijas ar laiku (maksimālā laika atrašana, sekundes/minūtes/stundas skaits starp diviem laikiem, utt.) utt.

Vēsture

Fiziskā objekta dzīves cikla gaitā notiek izmaiņas. Mehānisms, kas nodrošina vēstures saglabāšanu, tā ir *vēstures* atribūta grupa. Šī atribūta grupa saistīta ar *laika* atribūtu grupu. *Vēstures* atribūtā ir fiksēts laiks un fiziskā objekta stāvoklis, kas bija fiksēts norādītājā laikā.

Iepriekš jau definēts, ka DARSIR koncepcijai datu glabāšanas apjoms ir ierobežots. Un tas ir viens no atribūtiem, kas būtiski var ietekmēt DARSIR dokumenta apjomu. Tāpēc šiem parametriem ir ieviests obligātais ierobežojums par ierakstu skaitu.

Vēstures atribūtu īstenošanai vajag definēt atribūtu kopu, kurai jāseko līdzi un jāglabā vēsture. Jebkuru definēto parametru (vai uzreiz dažādu) maiņa tiek fiksēta vēsturē, bet tikai

par tiem parametriem, kas ir mainījuši savu stāvokli. Pirms vēstures saglabāšanas ir saglabāts pirmais ieraksts, kur ir fiksēts visu iepriekš definēto parametru stāvoklis uz vēstures glabāšanas procesa sākumu. Šim atribūtam ir iespējamie sekojoši režīmi:

1. *Nepārtrauktais* – ir nodrošināts režīms, kad nepārtraukti tiek saņemta informācija par fizisko objektu.
2. *Uz notikumu aktivizētais* – ir definēti noteikumi, kad nepieciešams saglabāt vēsturi. Piemēram, tas ir sensoru darbības rezultāts, kas reaģē uz objekta pārvietošanu (piemēram, uz noliktavu, uz mašīnu utt.).
3. *Periodiskais* – ir definēts periods (piemēram, katru stundu, katru dienu utt.) kad notiek vēstures saglabāšana.

Funkcionalitātei jānodrošina funkcijas katram no trijiem vēstures glabāšanas režīmiem (nepārtrauktais, uz notikumu aktivizētais un periodiskais). Birkas datu glabāšanas apjoms ir ierobežots, tiek pieņemts, ka ar vēstures glabāšanas funkcijas palaišanu jānodod skaitlis, cik lielu vēsturi ierakstos ir nepieciešams glabāt. Ir gadījumi, kad nav iespējams saglabāt vēsturi, parastais iemesls ir, ka nav pietiekamas brīvas vietas birkā. Šajā gadījumā eksistē trīs darba režīmi, kas attiecas uz tālākajām iespējamajām darbībām:

1. *Ar veco vēsturi* – ir gadījumi, kad ir svarīgi glabāt visu veco vēsturi, ja sistēma konstantē, ka nevar pievienot vēl vienu vēstures ierakstu, funkcija pārtrauc savu darbu.
2. *Ar jaunāko vēsturi* – ja ierakstu vēstures glabāšanas gadījums tiek detektēts, ka nav iespējams saglabāt vēsturi, tad tiks dzēsts vecākais ieraksts un pievienots jaunais.
3. *Ar prioritātes vēsturi* – parastā tendence attiecībā uz vēstures saglabāšanu ir - jo vairāk, jo labāk. Birkas iespējamo datu glabāšanas apjoma dēļ tas ir grūti realizējams. Iespējams klasificēt vēstures saglabāšanas gadījumus un ieviest svarīguma prioritātes. Par to atbild atsevišķs funkcijas parametrs, kas katram konkrētam gadījumam definē prioritātes no 0 līdz 9 (deviņi ir visaugstākā prioritāte). Ja nepietiek brīvas vietas, vispirms tiek dzēsti ieraksti ar zemāko prioritāti, ja paliek tikai ar vienu prioritāti, tad strādā identiski 2. režīmam (ar jaunāko vēsturi).

Pozicionēšana

Vispazīstamākā tehnoloģija, kas mūsdienās tiek izmantota pozicionēšanai, ir *globālā vietnoteices sistēma* (ang. *Global Positioning System, GPS*). GPS balstās uz satelītu tehnoloģiju, kura nodrošina satelītu navigāciju (piemēram, skat. [Bsh2009]). Galvenais tās darbības princips ir objekta pozīcijas noteikšana, aprēķinot attālumu no satelīta. Attālums tiek

aprēķināts no brīža, cik ilgi signāls iet no satelīta līdz objektam. Lai uzzinātu objekta pozīciju uz Zemes, nepieciešams definēt tā atrašanās vietu *platumā* (ang. *latitude*) un *garumā* (ang. *longitude*), aprēķinot attālumu vismaz no trim satelītiem. Ja nepieciešams uzzināt vēl arī *augstumu* (ang. *altitude*), tad nepieciešami četri satelīti (detalizētāk skat. [Dom1996]). Augstuma sākumpunkts definēts Zemes centrā caur Ziemeļpolu. Tā kā parasti tiek izmantots platums un garums, sauksim tās par *koordinātēm*. Ja koordinātes jāpapildina ar augstumu, tad nosauksim tās par *3D koordinātēm*. Augstums bieži tiek ignorēts, jo parasti tiek izmantots augstums nevis no Zemes centra, bet no jūras līmeņa, tādiem mērķiem izmanto altimetru.

GPS (un citu veidu satelītu navigācija) nodrošina pozicionēšanu no 10 metriem līdz 10 cm. Tehnoloģija parastam lietotājam ir lēta, GPS nemaksā dārgi, bieži tā jau ir iebūvēta mobilajās ierīcēs (mobilie telefoni, piezīmjdatori utt.), uzturēšanas maksa netiek paredzēta.

Satelītu navigācijai arī ir ierobežojums - nav iespējams definēt pozīciju iekštelpās (detalizētāk skat. [Fen2002]). Iekštelpu pozicionēšanai ir iespējams izmantot daudzas tehnoloģijas: *Wi-Fi* [Jek2010, Jek2011a, Jek2011b, Bød2006], *RFID* [Chi2009, Jah2009, Man2010], *Bluetooth* [Che2006, Hal2003], *GSM* [Dul2009], *ultraskaņu* [Hex2010] utt. Iespējams pieminēt divas pamatpozicionēšanas aprēķinu matemātiskās metodes: pēc signāla stipruma (piemēram, [Jek2010, Jek2011a, Jek2011b, Suz2008, Bad2007, Bru2005, Tho2005]) un izmantojot trilaterāciju (tā ir GPS izmantotā metode, detalizētāk skat. [Bah2000, Ser2004]).

Iekštelpu pozicionēšanai nav stingras piesaistes GPS koordinātēm, sauksim tās par iekštelpas koordinātēm. Iekštelpu koordinātēm vajadzīgs sākumpunkts. Pieņemts, ka koordinātes sākumpunkts tiek definēts atkarībā no nepieciešamības. Piemēram, tiek ņemts konteineris, kura iekšienē ir kastes. Katrai kastei iekšienē ir ierakstītas koordinātes, atrašanās vieta konteinerā. Savukārt, kastē ir preces, un katrai precei ir ierakstītas koordinātes, atrašanās vieta kastē.

Attālums ir norādīts metros, vienība pēc noklusējuma ir viens metrs. Ir ierakstītas trīs koordinātes, kas atbilst X, Y un Z asīm. Koordinātēm ir iespējamas pozitīvas un negatīvas vērtības atkarībā no tā, kādā virzienā jāpārvieto koordinātes.

Tiek definēta divu veidu pozicionēšanas atribūtu izmantošana, ko iespējams izmantot noteikumu īstenošanai, tās ir GPS koordinātes (vai GPS 3D koordinātes) un iekštelpu koordinātes (kad ierakstīta pārvietošanās pa X, Y un Z asīm metros).

GPS koordinātu prezentēšanai ir definēti trīs pamata formāti (detalizētāk skat. [Kap1996, Kap2006]: *DDD° MM' SS.S"* (ang. *Degrees, Minutes and Seconds*), *DDD° MM.MMM'* (ang. *Degrees and Decimal Minutes*) un *DDD.DDDDD* (ang. *Decimal Degrees*). Pamatojoties uz mazāk izmantoto simbolu skaitu formātā ir izvēlēts *DDD.DDDDD* (ang. *Decimal Degrees*) formāts. Nepieciešamas transformācijas funkcijām no esošā formāta uz

$DDD^{\circ} MM' SS.S''$ (ang. *Degrees, Minutes and Seconds*) vai $DDD^{\circ} MM.MMM'$ (ang. *Degrees and Decimal Minutes*), kā arī pretēja virziena transformācijas funkcijas.

GPS koordinātes iespējams saņemt no GPS moduļa, jābūt funkcijai, kas pēc pieprasījuma var saņemt GPS koordinātes un ierakstīt dokumentā. Piemēram, iespējams definēt funkciju GPS koordinātes detektēšanai *GetGPScoordinates*, kas atgriež esošās GPS koordinātes vai kļūdas paziņojumu.

Iekštelpu pozicionēšanai tiek izmantotas iekštelpu koordinātes. Pieņemts, ka ir speciāli moduļi, no kuriem iespējams pieprasīt un saņemt kāda objekta iekštelpu koordinātes atbilstoši sākumpunktam. Sākuma punktu iespējams definēt kā moduļa atrašanās vietu vai zināmu attālumu (pēc X, Y un Z asīm) no definētā sākuma punkta. Piemēram, konteinerā iespējams pieņemt konteineru zemāko stūri par sākumpunktu. Ja nepieciešams norādīt konteinerā ievietotai kastei konkrētas koordinātes, tas būs attālums metros pa X, Y un Z asīm no šī sākumpunkta (konteineru zemākais stūris).

Piekļuves tiesības

Viens no svarīgiem koncepcijas kritērijiem ir (*viii*) *datu konfidencialitātes, integritātes un pieejamības nodrošināšana*. Piekļuves pieeju nodrošināšanai (piemēram, skat. [Keg1984]) ir paredzētas lietotāju grupas definēšana un lietotāju pievienošana lietotāju grupām. DARSIR koncepcijai nav centralizētas pārvaldības, un šiem noteikumiem iespējams piemērot resursa unikālo identifikatoru.

Katram resursam ir unikāls identifikators, kas sastāv no: sistēmas numura, ražotāja numura, objekta klases numura un sērijas numura. Pamatojoties uz šiem datiem ir iespējams definēt piekļuves tiesības četros līmeņos:

1. *Sistēmas* – tiek iekodēts sistēmas numurs 8 bitos, līdz 256 sistēmām.
2. *Ražotāja* – definēts ražotāja numurs atsevišķā sistēmā 28 bitos, līdz 268 miljoniem ražotāju.
3. *Produkta* – atsevišķa ražotāja produkts, glabājas 24 bitos, līdz 16 miljoniem produktu.
4. *Elementa* – atsevišķa produkta izveidotais resurss, glabājas 36 bitos, līdz 68 miljardiem elementu.

Resursa identifikatora glabāšanai tiek izmantots heksadecimālais kods, jo tas aizņem mazāk vietas. Viena heksadecimāla skaitļa prezentēšanai ir vajadzīgi 4 biti. Visi iepriekš definētie līmeņi bez atlikuma dalās ar četri. Izmantojot regulāras izteiksmes [Keg1984, Lov2005, Rob2005b], ir iespējams iekodēt resursa identifikatoru diapazonu. Piemēram, nepieciešams iekodēt visus A sistēmas, FF ražotāja, visus produkta tipus no AF0 līdz AFF. Tas būs $A.FF.AF?.*$, kur „?” ir viens cipars un „*” visi cipari.

Viens no tipiskiem datu glabāšanas veidiem ir glabāšana datnē. No *Unix* operētājsistēmas vides [Rob2005a, Keg1984] ir definētas trīs pamata operācijas ar datiem (datnēm): *lasīt* (ang. *read*), *rakstīt* (ang. *write*) un *palaist* (ang. *execute*). Palaišanas operācija ir paredzēta noteikumu realizēšanai. Kopā ar trijām pamata operācijām ir nepieciešama piekļuves tiesību mainīšanas operācija. Visas šīs operācijas ir iespējams atļaut vienlaicīgi, tāpēc tiek uzskatīts, ka katrai operācijai iespējamās divas nozīmes (0 – aizliegts un 1 – atļauts). Katrai operācijai ir dots bits un bināru veidā iespējams iekodēt kombinācijas no *0000* (viss aizliegts) līdz *1111* (viss atļauts). Katram bitam ir sava operācija: *1-bits* – lasīt, *2-bits* rakstīt, *3-bits* mainīt piekļuves tiesības un *4-bits* palaist.

Heksadecimālā veidā šos 3 bitus iespējams iekodēt ar cipariem no 0 līdz F. Tā kā ir doti četri piekļuves tiesību līmeņi, tad piekļuves tiesību operāciju iekodēšanai nepieciešami četri cipari.

Pamatojoties uz resursa dokumentu piekļuves tiesībām ir iespējams definēt sekojošas sastāvdaļas:

1. *Atribūtu* – tā ir vismazākā dokumenta sastāvdaļa. Tikai vienam atribūtam nav paredzētas izmaiņas - resursu identifikatoram.
2. *Bloks* – iespējams uzstādīt piekļuvi blokam, un bloka piekļuves tiesības automātiski manto visi bloka apakšbloki un atribūti.
3. *Dokuments* – iespējams vienlaicīgi regulēt piekļuves tiesības visam dokumentam.

Piekļuves tiesības tiek mantotas no visaugstākā līmeņa uz viszemāko līmeni. Piemēram, ja ir pieņemts, ka dokumentam ir augstākais līmenis un bloks ir zemākais, tad piekļuves tiesības uz dokumentu attiecas arī uz dokumenta blokiem. Tas nozīmē, ka, ja tieši tiek uzstādītas piekļuves tiesības konkrētam elementam, tad visi iekļautie elementi manto šīs piekļuves tiesības.

Definēsim, ka ir funkcija *SetAccessRights*, ar kuras palīdzību iespējams definēt piekļuves tiesības atribūtam, blokam vai uzreiz visam dokumentam. Ja uzstāda piekļuves tiesības visam dokumentam, tad automātiski šīs piekļuves tiesības tiek piemērotas visām dokumenta sastāvdaļām (blokiem un atribūtiem). Mainīt piekļuves tiesības iespējams no resursa, kam ir atļauts veikt tāda tipa operāciju. Ja jau ir uzstādītas piekļuves tiesības atsevišķam dokumenta blokam, tad funkcijai *SetAccessRights* jāzina, kā jārikojas šajā situācijā. To iespējams realizēt ar atsevišķu atribūtu *Change*. Ja atribūts ir *patiens*, tad tas maina arī citas uzstādītās piekļuves tiesības (ja vien pašai funkcijai ir pietiekamas piekļuves tiesības).

Piemēram, eksistē kontainers, kuram definēta piekļuves tiesību vērtība *F.AOF.AFE01.86578A* ar *0008* (8 tas ir *1000*; tikai lasīt), un tās ir kastē iekšā, tiek definētas piekļuves tiesības *F.AOF.*.** ar *ODDD* (*D* tas ir *1111*; var lasīt, rakstīt, palaist un mainīt piekļuves tiesības). Konteineru datus iespējams nolasīt tikai resursam ar resursa identifikatoru *F.AOF.AFE01.86578A*. Visiem resursiem, kam ir definēta sistēma *F* un ražotājs *AOF* ar paletēm, kas atrodas konteinerā, ir atļautas visas operācijas. Bet, ja resurss *F.AOF.A1.B2* piemēro funkciju *SetAccessRights* ar atribūta *change* vērtību *nepatiess*, tad nekas netiks mainīts, tāpēc ka piekļuves tiesības jau ir uzstādītas. Ja *change* atribūtam būs vērtība *patiess*, tad piekļuves tiesības tiks izmainītas tikai kastēm, jo šim resursam ir pietiekamas piekļuves tiesības tikai uz kastēm, lai veiktu šīs izmaiņas.

Klasifikācija

Resurss ir fizisks objekts ar savām individuālajām īpašībām. Kad resursu ir daudz, tiem grūti piemērot kopējas darbības. Viens no variantiem, kā atvieglot darbu ar resursiem, ir resursu klasificēšana un kategorizēšana.

Pats resursa identifikators jau ir klasificēts, atsevišķi definēta sistēma, tālāk ir sadalīšana starp ražotājiem, katram ražotājam ir objekta klases, un katrai objekta klasei ir elementi. Tas ir viens no veidiem, kā iespējams klasificēt resursu. Pilnīgi balstīties uz šo klasificēšanas veidu nav iespējams. Bieži pēc ražotāja objekta klases nav iespējams klasificēt objektu (piemēram, plastmasas maisīnam iespējama dažādu veidu klasificēšana atkarībā no izmantošanas nolūkiem: preces pakošana, ražošana utt.). Iespējams ieviest atribūtu, kas atbild par resursa klasi. Klases variāciju ir daudz, tāpēc tiek ieviests *klases identifikators*, kas līdzīgs RI (nav tikai elementa numura), sastāv no: sistēmas numura, ražotāja numura un klases.

Izņemot resursa klašu identificēšanu (pēc resursa identifikatora vai klašu identifikatora), vēl eksistē gadījumi, kad papildus jāpievieno dati un noteikumi, kas attiecas uz konkrētu klasi. Tā kā visiem nepieciešamajiem atribūtiem un noteikumiem jābūt pieejamiem uz vietas, nepieciešams mehānisms, kas nodrošina datu un noteikumu pievienošanu un dzēšanu no resursa. Viena no metodēm, kādā veidā to realizēt, ir uz *objektu orientētu* principu izmantošana.

Gadījumos, kad resursa identifikatora sastāvdaļas nav pietiekamas resursu klasificēšanai (piemēram, tāpēc ka resursa izmantošanas nolūki atšķiras no ražotāja definētajiem), ir ieviests atsevišķs atribūts, kas atbild par resursa *klases identificēšanu*. Nepieciešama funkcija, kas ieraksta šo klašu numuru (*SetClassNumber*), dzēš (*DeleteClassNumber*) un nolasa to (*GetClassNumber*). Kad izsaukta funkcija par klašu piesaisti (*SetClassNumber*), tiek pārbaudīts, vai resursam eksistē klašu numurs. Ja nav piešķirts klašu numurs (eksistē tikai resursa numurs), tad tiek pievienots jauns atribūts, kuram

piešķirta klašu numura nozīme. Ja klašu numura atribūts eksistē, tad klašu numura atribūtam ir pievienota papildu vērtība (jauna klase). Ja nepieciešams nolasīt resursa klašu numuru, tad, ja definēts klašu numurs, no turienes tiek ņemta vērtība, pretējā gadījumā tā tiek ņemta no resursa identifikatora.

Resursa klasificēšanas brīdī iespējams arī pievienot atsevišķus atribūtus un noteikumus, kas atbilst šai klasei. Standarta mehānisms klasificēšanai ir svarīgs, lai nodrošinātu iespēju ne tikai piešķirt resursam klasi, bet pēc nepieciešamības arī noņemt to nost. Atribūti un noteikumi klasei ir saglabāti atsevišķā blokā. Tiek paredzētas divas funkcijas: klases pievienošana (*AddClass*) un klases noņemšana (*DeleteClass*).

Konfigurācija

Fiziskais objekts var ietvert sevī dažāda veida citus fiziskus objektus vai sastāvēt no dažiem fiziskiem objektiem. [Bro2001d] pētījumā ir piedāvāts aprakstīt konfigurācijas atribūta veidu ar hierarhisku struktūru. Hierarhiskas struktūras aprakstam parasti ir piedāvāta analogija ar koku. Kokam ir zari, zari var saturēt citus zarus, zaros ir lapas. Šī analogija ir veiksmīgi piemērojama datorzinātnē, piemēram, failu sistēmas realizācijā. Pēc šīs analogijas mape ir zars un datne ir lapa.

Viens no tipiskiem konfigurācijas atribūta piemēriem ir uzdevums no loģistikas. Tādos pētījumos kā [Bro2001a, Bro200b] ir dots piemērs ar konteineru. Konteinerā iekšā ir paletes, uz paletēm atrodas kastes un kastēs ir preces. Uz šo piemēru pamata var parādīt vietas, kur glabāt konfigurācijas atribūtu. Viens no variantiem ir glabāt visu aprakstu konteinerā. Alternatīvais variants ir glabāt konteinerā informāciju par paletēm, paletēs par kastēm un kastēs par precēm. Ja apskata tipiskas IS realizācijas (piemēram, [Lah2005]) bieži tas ir atkarīgs no tehnoloģiskām iespējām. Piemēram, ja birka ir tikai konteinerā, tad tā ir vienīgā vieta, kur glabāt visus datus par fiziskiem objektiem.

Ir gadījumi, kad konfigurācijas atribūts ir saistīts ar fiziska objekta datu glabāšanu. Ja eksistē fiziskais objekts bez birkas, bet pats objekts ir sastāvdaļa no cita objekta. Šajā gadījumā iespējams glabāt ne tikai konfigurācijas aprakstu, bet arī paša resursa aprakstu konfigurācijas atribūtā.

Konfigurācijas atribūta iedarbināšanai ir nepieciešams ieviest divus terminus: vecāks un bērns. Vecāks tas ir pamatelements, kas sastāv no bērniem – apakšelementiem. Katram fiziskam objektam ir resursa identifikatori. Pamatelementā ir veikts konfigurācijas apraksts, un saite uz apakšelementiem ir nodrošināta ar fizisko objektu resursa identifikatora palīdzību. Ja iespējams norādīt, ka viens fiziskais objekts satur citu objektu, tad iespējams apvienot objektus ar *CreateNode* funkciju. Ja objekti sadalās neatkarīgos objektos, tad iespējams sadalīt

objektus ar *DeleteNode* funkciju. Katrai funkcijai ir divi ieejas parametri: *vecāks* (galvenais elements) un *bērns* (apakšelements).

Prezentācija

Iepriekš aprakstītās resursa atribūtu grupas bija nepieciešamas ar resursa saistīto darbu automatizāciju. Lietotājam vajadzīga informācija par resursiem, kas nav saistīti ar resursa fiziskajām īpašībām, tā ir *resursa prezentācija*. *Resursa prezentēšana* tie ir dažādu veidu dati (teksta, video, audio utt.), kas raksturo resursu un palīdz painformēt resursa patērētāju par resursa izmantošanas īpašībām. Tipiskie piemēri, kas atbilst resursu prezentēšanai: pamācība lietotājam, garantija, reklāma (teksta, attēli, audio, video utt.) utt.

Resursa prezentācijai ir trīs sastāvdaļas:

1. *Nosaukums* – šeit ir definēts kādiem nolūkiem ir vajadzīga šo datu prezentēšana. Ir nepieciešams definēt standarta gadījumus, tad būs iespējams definēt standarta noteikumus, kas nodrošina darbu ar šiem atribūtiem. Tipiskie gadījumi ir resursu: instrukcija, garantija, logo, reklāma utt.
2. *Avots* – jādefinē kāda tipa informācija tiek prezentēta. Iespējamie varianti: teksts, attēls, audio, video utt. Katram tipam atsevišķi jādefinē, kāds formāts tiek izmantots: teksta kodējums, attēla formāts utt.
3. *Vērtība* – šeit tiek saglabāta vērtība, kas atbilst *avota* definētajam formātam.

DARSIR koncepcijai ir svarīgs datu izmantošanas apjoms. Tiek uzskatīts, ka, ja nepieciešams glabāt grafiskos, audio un video datus, tad tas paaugstina prasības pret tehnoloģijas izmantošanu.

Resursu prezentācijas noteikumi ir nepieciešami patērētāju nodrošināšanai ar nepieciešamo informāciju ar vizuālās vai audio informācijas prezentēšanas palīdzību (piemēram, grafiskais attēls, audio reklāma, video klips utt.). Tipisks piemērs, kad veikalā ir speciāls birkas skenētājs, kad pēc preces noskenēšanas lietotājs var ieraudzīt skenera cenu skenētāja ekrānā. Preču birkā iespējams ierakstīt tādu informāciju kā: reklāmas klips, cena, instrukcija, garantija, logo utt. Kad pircējs izvēlas preci, par kuru vēlas uzzināt vairāk, viņš pārvieto uz birkas skenētāja aktīvo zonu. Birku skenētājs saņem prezentēšanas atribūtus no birkas, un tas var atspoguļot šos atribūtus pircējam (piemēram, parādīt preču reklāmas klipus).

Ja pats resurss ir nodrošināts ar funkciju atspoguļot prezentācijas atribūtu lietotājam, tad attiecīgajiem noteikumiem ir jābūt ierakstītiem pašā resursā. Jābūt atribūta atspoguļošanas palaišanas funkcijai (*StartPresentationAttr*) un pārtraukšanas funkcijai (*StopPresentationAttr*). Vēl nepieciešamas darbības ar prezentācijas atribūtu: pievienot (*AddPresentationAttr*) un dzēst (*DeletePresentationAttr*).

4.3. Tehnoloģijas izvēle

DARSIR koncepciju ir iespējams izmantot ar dažādām tehnoloģijām. Ir nepieciešama vienkārša un ticama metode, ar kuru iespējams ātri izvēlēties optimālo tehnoloģiju, pamatojoties uz risinājuma vajadzībām. Risinājuma analīzes un izstrādes laikā var rasties jaunas prasības, taču ar šo metodi būs iespējams ātri novērtēt, kā tās ietekmē tehnoloģijas izvēli.

Ir apskatīti galvenie kritēriji, kas var ietekmēt šo izvēli. Pamatojoties uz šiem kritērijiem ir definēta tehnoloģijas izvēles aprēķinu metode. Tiek definētas visiem kritērijiem vienādas ietekmes nozīmes un tiek izvēlēta DARSIR koncepcijai pēc noklusējuma vispiemērotākā tehnoloģija.

4.3.1. Ietekmes kritēriji

Nepieciešams klasificēt *ietekmes kritērijus* un definēt kritēriju grupas:

1. *Fiziskie parametri un ārējā vide* – nepieciešams definēt parametrus, kas ietekmē risinājuma obligāto ieviešanu, tas ir: birkas parametri (izmērs, datu glabāšanas apjoms, pārrakstīšanas iespēja utt.), skeneru parametri (skenēšanas ātrums, attālums līdz objektam), vides apstākļi (temperatūra, mitrums utt.) utt.
2. *Izmaksas* – tās ir risinājumu kopējās palaišanas un uzturēšanas izmaksas.
3. *Drošība un konfidencialitāte* – tās ir dažādu veidu datu aizsardzības metodes vai metožu kombinācijas ieviešana.

Viens no pamata parametriem, kāda veidā ir iespējams aprēķināt izvēlētajā tehnoloģijas efektivitāti, ir kopējās risinājuma izmaksas. Un gandrīz jebkura parametra izvēli, pirmkārt, ietekmē risinājuma izmaksas. 1. un 3. punkta kritēriju grupas definē pamata funkcionalitāti, kura obligāti jānodrošina, pretējā gadījumā risinājumu nebūs iespējams ieviest vispār.

4.3.1.1. Fiziskie parametri un ārējā vide

Birkas datu apjoms

Birka ir noteicošais elements DARSIR koncepcijā, tajā glabājas resursa atribūti un noteikumi. Par pamata vienību tiek uzskatīts teksta simbols, jo datu glabāšanas formāti tiek prezentēti teksta vidē. Kritēriju vērtība vieninieks - tā ir iespēja nodrošināt neierobežotu datu apjomu vai tuvu tam. Pareiza kritēriju izvēle var samazināt risinājuma ieviešanas un uzturēšanas izmaksas. Piemēram, mūsdienās loģistikas procesos parasti tiek izmantota svītrkodu tehnoloģija. Ja konkrētam risinājumam ir pietiekams svītrkoda datu apjoms, tas samazina risinājuma ieviešanas izmaksas, jo tiks piemērota eksistējoša tehnoloģija nevis ievadīta jauna tehnoloģija.

Viens no būtiskiem kritērijiem, ja nebūs pietiekami vietas datu glabāšanai, tad risinājumam būs būtiski ierobežojumi, kas paaugstinās blakus izmaksas (piemēram, vajag resursam pievienot divas birkas nevis vienu), vai arī nebūs iespējams piemērot tehnoloģiju šim risinājumam.

Pārrakstīšanas iespēja

Katrai tehnoloģijai ir savas ierobežojumi, un viens no būtiskākajiem ierobežojumiem ir datu pārrakstīšanas iespēja. Iepriekš apskatītajās svītrkodu un RFID tehnoloģijās šī kritērija atbalsts var būtiski paaugstināt prasības risinājumam (datu glabāšanas apjomu, procesora jaudu utt.) un rezultātā kopējās izmaksas (ieviešanas, infrastruktūras būvēšanas, uzturēšanas utt.).

Eksistē tehnoloģijas, kas nenodrošina datu pārrakstīšanu. Tomēr eksistē alternatīvi risinājumi, tādi kā jaunas birkas izveidošana, datu pārrakstīšanas gadījumā. Piemēram, svītrkodu tehnoloģijai nav tādas iespējas, bet, ja pārrakstīšanas process notiek reti, iespējams paredzēt jaunas birkas izveidošanu, kurā tiek ierakstīta jauna informācija. Tādas realizācijas ir pamatotas, kad ir liels birku skaits un birkas datu pārrakstīšana jāveic reti.

Birkas izmērs

Tipiska tendence, jo vairāk funkciju jānodrošina bircai (datu glabāšana, iebūvētie sensori utt.), jo bircai ir lielāks izmērs. Jo lielāks ir birkas izmērs, jo vairāk enerģijas ir vajadzīgs, lai nodrošinātu birkas funkcionēšanu (datu nolasīšana, mainīšana utt.). Viena no pareizajām pieejām ir izmantot tik daudz funkcijas, cik ir vajadzīgs konkrētam risinājumam. Piemēram, svītrkodu tehnoloģijā, jo lielāks birkas izmērs, jo vairāk datu iespējams glabāt vienā birkā. Ja ir pietiekams lineārais svītrkods, nav nepieciešamības ieviest 2D svītrkodu.

Ja resurss, kam pieder birka, ir liela izmēra, var ievietot arī lielu birku. Ja gadījumā bircai jābūt mazai (piemēram, resurss ir drēbes), tad iespējamo tehnoloģiju skaits samazinās, un paaugstinās kopējās risinājuma izmaksas.

Birkas svars

Ar šo kritēriju ir līdzīga situācija kā birkas izmēra kritērijam, tikai tas attiecas uz svaru nevis uz izmēru. Ir gadījumi, kad birkas svars ir kritiskais kritērijs, un pretēji, kad tam nav lielas ietekmes uz risinājumu. Piemēram, ja tiek ņemts konteiners ar kastēm, kur kastes ir atsevišķas preces. Konteinerā atļauts ievietot birku, kuras svars kilograms. Tomēr kastei birkas svars nevar būt lielāks par simts gramiem un precei ne vairāk par 10 gramiem.

Parasti pašai tehnoloģijai ir definēts tipisks birkas izmērs, bet ir gadījumi, kad atkarībā no nepieciešamajām funkcijām, arī vienas tehnoloģijas ietvaros birkas svars var būtiski atšķirties (piemēram, RFID tehnoloģijas pasīvais tags sver dažus gramus, bet aktīvā birka var svērt dažus kilogramus). Ir tehnoloģijas, kurās birkas svars tik mazs, ka šo kritēriju var

ignorēt, piemēram, svītrkodu tehnoloģija. Pati birka ir attēls, ir gadījumi, kad birku uzreiz attēlo uz resursa iepakojuma. Ja svītrkods ir papīra uzlīme, kas uzlīmēta precei, tad svītrkoda svars ir tik mazs, ka to iespējams ignorēt.

Netipiskie vides apstākļi

Eksistē dažāda veida vides apstākļi: temperatūra, mitrums, spiediens utt. Vides apstākļiem ir definēti savi tipiskie parametru diapazoni, bet ir gadījumi, kad nepieciešams nodrošināt specifisku vides apstākļu atbalstu. Piemēram, pieņemsim, ka ir resurss kreklis, kuram ir pievienota birka. Kreklu ir plānots mazgāt ūdenī ar temperatūras režīmu +60C ar veļas pulveri un veļas mīkstinātāju. Tas nozīmē, ka birkai jāatbilst šiem vides apstākļiem, birkai jābūt izturīgai pret: ūdeni, augstu temperatūru (+60C) un ķīmiskiem līdzekļiem (veļas pulveris un veļas mīkstinātājs).

Risinājuma darbības procesā, kurš izveidots DARSIR koncepcijas pamatā, ir divi galvenie darbības elementi - birka un skeneris. Skenera diapazonam nav obligāti visu laiku jābūt pieejamai birkai, tāpēc parasti skenerus mazāk ietekmē vides apstākļi. Pati birka visu laiku ir pievienota resursam, tai jābūt izturīgai pret tādiem pašiem vides apstākļiem, kuriem piemērots pats resurss. Birku ražotāji jau tagad ražo birkas, kas atbilst netipiskiem vides apstākļiem. Piemēram, RFID tehnoloģijai ir pasīvie tagi (tās ir birkas), kurus iespējams mazgāt veļas mašīnā [Lah2005], tie izturīgi pret ūdeni, augstu temperatūru un veļas mazgāšanā izmantoto ķīmiskiem līdzekļiem.

Skenera nolasīšanas ātrums

Tipiskais DARSIR koncepcijas risinājums paredz vienas vietas vienu vienlaicīgu skeneru darbību. Atkarībā no datu glabāšanas apjoma birkā un kopējā birku skaita, kas pieejams pie viena skenera vienlaicīgi, ir atkarīgs birkas nolasīšanas ātrums. Skeneri parasti neatbalsta paralēlu birku nolasīšanu, nolasīšanas process notiek secīgi no vienas birkas uz otru. Šis kritērijs ietekmē tehnoloģijas izvēli, kad ir nepārtrauktais darbības process un nepietiekošs skenera nolasīšanas ātrums var apturēt risinājuma darbības procesu. Piemēram, loģistikas procesos bieži izmanto RFID tehnoloģiju. Skeneris (RFID skeneris) iebūvēts visos vārtos. Kad automašīna izbrauc vai iebrauc caur vārtiem, skeneris nolasa datus no birkas (RFID tags). Ja pieņem, ka skenera ātrums nav pietiekošs, lai nolasītu datus no birkas, tas nozīmē, ka automašīnai nepieciešams apturēt savu kustību pie vārtiem, kamēr skeneris nolasa datu no taga pilnībā.

Skenēšanas attālums

Attālumu no birkas līdz skenerim sauc par *skenēšanas attālumu*. Šis kritērijs ietekmē skenera nolasīšanas ātrumu, jo tuvāk birka skenerim, jo lielāks ātrums. Skenēšanas attālumu ietekmē ārējās vides apstākļi – temperatūra, mitrums, gaisa piesārņojums utt.

Skenēšanas režīma tieša redzamība

Eksistē tehnoloģijas, kurām ir nepieciešama tieša redzamība no skenera līdz birkai (piemēram, svītrkods). Izpildot šo kritēriju, ir definēts, vai ir iespējams izmantot tehnoloģijas ar skenēšanas režīma ierobežojumu (obligāta tieša redzamība). Eksistē arī ārējās vides apstākļi, kas paredz obligātu izpildi šim kritērijam. Piemēram, ja darbības procesā tiek izmantota ēka ar signālu absorbējošām sienām, tad šis kritērijs obligāts izpildei.

Skenera daudzkārtēja birku lasīšana

DARSIR koncepcijas risinājumam tipiskais scenārijs ir viens skeneris vienā vietā un viena vai vairākas birkas. Ja skeneris var nolasīt tikai vienu birku savā uztveršanas zonā, tad risinājumam jāparedz šis ierobežojums. Tādas tehnoloģijas kā svītrkodu parasti neparedz vienlaicīgi lasīt daudzas birkas, un pats nolasīšanas ātrums skeneriem ir neliels. RFID tehnoloģijai katrai no ierīcēm ir savi parametri šim kritērijam, viens skeneris (RFID skeneris) var nolasīt tikai vienu birku (RFID tagu) vienlaicīgi savā uztveršanas zonā, citam birku nolasīšanas ātrums ir 200 birku/sekundē (skat. [Lah2005, Hei2005, Bau2005]).

Tehnoloģiju atpazīšana, izplatīšana un pieejamība

Viena no pamatīpašībām, kas tiek plānota, sasniedzot ar DARSIR koncepcijas izmantošanu, ir pašreizējā risinājuma tehnoloģijas izmantošana un evolucionāra pieeja tālākai IT risinājumu uzlabošanai. Tiek uzskatīts, ka ekonomiski izdevīgāk izmantot tehnoloģijas, kas jau pieejamas uz vietas, kur plānots ieviest risinājumu. Piemēram, ir noliktava, kur nepieciešams ieviest jaunu risinājumu. Pēc analīzes ir definēts, ka ir pieejamas Wi-Fi un svītrkodu tehnoloģijas. Tas nozīmē, ka šim konkrētam gadījumam šīs tehnoloģijas jau ir atzītas, izplatītas un pieejamas. Citā gadījumā, kad risinājumu nepieciešams ieviest vietā, kur tehnoloģijas infrastruktūras nav (piemēram, avāriju loģistika [She2007]), šis kritērijs neietekmē tehnoloģijas izvēli.

4.3.1.2. Izmaksas

Aprēķinot izmaksu ietekmes kritērijus, ir ņemts vērā fakts, ka, jo lielāka izmaksu summa kritērijam, jo vairāk ietekmes kritēriju vērtība tiecas uz nulli.

Tehnoloģiju ieviešanas izmaksas

Pirms tehnoloģijas ieviešanas nepieciešams uzbūvēt tehnoloģijas infrastruktūru. Galvenās izmaksas veido skenera un pietiekoša birku skaita piegāde, elektrības tīkla nodrošināšana. Infrastruktūras nodrošināšana ir viens no būtiskiem izmaksu veidiem jaunu risinājumu ieviešanas gadījumā. DARSIR koncepcija var izmantot dažādas tehnoloģijas un tiek uzskatīts, ka jau eksistējošu infrastruktūru izmantošana var būtiski samazināt šīs izmaksas. Tomēr eksistē gadījumi, kad tomēr jāuzbūvē infrastruktūra no nulles.

Tehnoloģiju uzturēšanas izmaksas

Nav pietekami ieviest tehnoloģiju risinājumam, nepieciešams arī aprēķināt kopējās izmaksas uzturēšanai. Uzturēšanas izmaksas sastāv no: diagnostikas (regulāras pārbaudes), remonta, ierīču maiņas (jaunu ierīču uzstādīšana sabojāto ierīču vietā), elektroenerģijas patēriņa, licences gada maksas utt.

Tehnoloģijas uzturēšanas izmaksas parasti tiek aprēķinātas, kā summa gadā no ierīces. DARSIR koncepcijas risinājumam galvenās darbības ierīces ir skeneri un birkas.

Skeneru izmaksas

Parasti pamatojoties uz skeneru fiziskiem parametriem ir iespējams definēt skeneru skaitu, kas nepieciešams risinājumam. Parasti viena tehnoloģija arī nodrošina daudzas ierīces ar dažādiem darbības parametriem un jāatrod balanss starp funkcionalitāti un skeneru skaitu. Piemēram, definēsim, ka nepieciešams nodrošināt skenēšanas funkcionalitāti telpā. Iespējams nopirkt dažus lētākus skenerus pie katrām telpas durvīm vai vienu dārgāku, kurai aktīvai darbības zonai ir pieejama visa telpa.

Katram skenerim ir sava cena un blakus izmaksas (remonts, instalēšana, uzturēšana utt.). Ir gadījumi, kad būtisks ir skeneru skaits, un skenera modeļa izvēlei jāpamatojas ne tikai uz darbības funkcionalitāti, bet arī uz skenera kopējām izmaksām.

Birkas izmaksas

Birkas cena var būtiski ietekmēt kopējās risinājuma izmaksas. Parasti birku skaits ir liels. Kā gadījumā ar skeneriem, tas arī šajā gadījumā ir kompromiss starp funkcionalitāti un birkas cenu. Viens no būtiskiem birku parametriem ir datu glabāšanas apjoms. Ir gadījumi, kad divu (vai vairāku) birku kopēja izmantošana var samazināt šī faktora ietekmi. Piemēram, loģistikā bieži tiek izmantota pieeja, kad svītrkodu izmantošanas gadījumā ir daudz svītrkodu, un katrs svītrkoda veids glabā konkrēta veida informāciju: produkta unikālais kods, partija, adrese utt.

Birkas dzīves cikls

Parasti birkai savai funkcionēšanai ir vajadzīga enerģija. Tipisks enerģijas avots ir akumulators. Birkām, kurām ir uzstādīts akumulators, tiek definēts scenārijs, kā jāīrkojas, ja birka vairs nefunkcionē. Iespējamie scenāriji nosaka birkas dzīves ciklu (piemēram, ekspluatācijas periods 2 gadi), kamēr akumulatora enerģija nebeidzas; vai izmantot birku tādā veidā, kad birkas akumulatora izlādēšanās gadījumā, tas neietekmē risinājuma darbības procesu (piemēram, birku dublē cita birka ar analogisku funkcionalitāti).

Eksistē arī tehnoloģijas, kuras neizmanto savā darbā akumulatora enerģiju, bet tās ietekmē citu faktori. Pamatoties uz šiem faktoriem ir aprēķināts birkas dzīves cikls. Piemēram, svītrkodu var ietekmēt krāsas sastāvs, no kura izveidots attēls (tieša saule var samazināt birkas

dzīves ciklu); RFID pasīvais tags var sabojāties fiziski, vai arī, ja aktīvajam tagam ir vajadzīgs akumulators, tad tas var izlādēties ar laiku.

4.3.1.3. Datu drošība un konfidencialitāte

Aprēķinot konkrētus datu drošības un konfidencialitātes ietekmes kritērijus, tiek definēts, jo zemāks datu aizsardzības līmenis tiek nodrošināts, jo vairāk ietekmes kritēriju vērtība tiecas uz nulli.

Iebūvētie kļūdu detektēšanas līdzekļi

Katrai tehnoloģijai ir realizēti savi datu drošības līdzekļi. Ja dati tiek sabojāti, nepieciešamas metodes, kuras nodrošina viennozīmīgu datu pārbaudi, lai detektētu šo faktu. Viens no parastajiem datu kļūdu detektēšanas līdzekļiem ir *cikliska atlaišanu pārbaude* (ang. *Cyclic Redundancy Check, CRC*) koda izmantošana. Par CRC skatīties detalizētāk [Sti2006]. CRC – tas ir no 8 līdz 32 bitu skaits, kurš aprēķināts pēc matemātiska algoritma no datu bloka un pievienots šim datu blokam. Gadījumā, kad tiek sabojāti dati (vai pats CRC kods) tos var detektēt. Ja salīdzina pašreizējo CRC skaitli ar aprēķināto vērtību no pašreizējā datu bloka, ja vērtības atšķiras, tas nozīmē, ka informācija ir sabojāta un dati jāatjauno.

Šis kritērijs parāda, cik droši kļūdu detektēšanas līdzekļi ir izmantoti tehnoloģijai. Ir gadījumi, kad vienas tehnoloģijas robežās dažādas realizācijas nodrošina dažāda līmeņa iebūvētus līdzekļus. Pamatojoties uz kritērija svarīgumu konkrētai koncepcijai, ir gadījumi, kad nepieciešams veikt papildu darbības datu drošības nodrošināšanai. Piemēram, ja šim kritērijam ir nepietekoši iebūvēti kļūdu detektēšanas līdzekļi, nepieciešams piesaistīt papildu līdzekļus.

Iebūvētie datu atjaunošanas līdzekļi

Tiek paredzēts, ka DARSIR koncepcijai dati tiek glabāti lokāli pašā resursā un netiek organizēta centralizēta datu glabāšana un atjaunošana. Šajā gadījumā datu atjaunošanas līdzekļiem ir pievērsta papildu uzmanība. Vienu no datu atjaunošanas līdzekļiem ir iespējams nodrošināt failu sistēmas vidē, piemēram, *NTFS* (ang. *New Technology File System*) ir pat atjaunota faila sistēma (detalizētāk skat. [Rus2005, Mic2003]). Galvenā šīs failu sistēmas priekšrocība ir darbības ātrums, kurš arī dod iespēju atjaunot datus, ja bija kādi trūkumi darbā. Failu sistēmas atjaunošanas mehānisms strādā, ja ir sistēmas fatālas problēmas. Tad fails veic trīs operācijas: analizē, atkārtoti un atjauno. Analīzes procesā tas pamatojas uz reģistrācijas failu, uzzina visas kļūdas un problēmas, sistēma uzzina kādus klasterus vajag atjaunot. Otrās operācijas laikā tiek veiktas visas transakcijas iterācijas no kontroles punkta. Tiek veikta nepabeigtu transakciju atjaunošana. NTFS sistēmā ir svarīgs datu atjaunošanas mehānisms, ko sauc par *atlikto pārraidi* (ang. *lazy commit*). Eksistē reģistrācijas fails, kurā ir ierakstītas nepieciešamās darbības, ko sauc par atliktiem ierakstiem. Tas ļauj samazināt apstrādes laiku,

visi darbietilpīgie darbi tiek veikti vēlāk. Ja notiek darba pārtraukšana NTFS sistēmas darba laikā, iespējams atjaunot transakcijas, pamatoties uz informāciju no reģistrācijas faila. Ja ir bijusi fatāla kļūda pirms transakcijas pabeigšanas, tad NTFS pārbauda, vai bija/ nebija pilnīgi pabeigta dotā transakcija, ja analīze dod atbildi, ka ir problēmas, tad sistēma automātiski atjauno transakciju. Ir aizliegts atstāt nepabeigtas transakcijas. Ja eksistē nepabeigtas transakcijas, NTFS sistēma cenšas tās pabeigt vispirms.

Vēl viens princips datu atjaunošanai ir iebūvētie papildu dati, kas dublē daļu no datiem, kas savukārt nodrošina ātru datu atjaunošanu. Tāds princips ir svītrkodu tehnoloģijai, piemēram DataMatrix simboloģijai ir iespējams nolasīt pareizi datus no svītrkoda, ja ir sabojāti līdz 20-40% no svītrkoda (procents atkarīgs no svītrkoda izmēra). Pats princips ir, ka svītrkods tiek papildināts ar papildu datiem, ar kuru palīdzību iespējams atjaunot pamata datus.

Šis kritērijs definē, kura no metodēm ir pieejama konkrētai tehnoloģijai, un tā tiek nodrošināts kopējais aizsardzības līmenis. Būtiski ir, kad datu atjaunošanas līdzekļi ir iebūvēti tehnoloģijā, jo vairākām tehnoloģijām ir ierobežojumi papildu līdzekļu nodrošināšanai.

Iebūvētie nesankcionētas datu izmantošanas līdzekļi

Nesankcionētu datu izmantošanu iespējams sadalīt divās grupās. Pirmā, kad datus nepieciešams aizsargāt no datu nolasīšanas, un otra, kad nepieciešams pārlicināties, ka dati ir uzticami un tos saglabā uzticamā avotā. Ja nepieciešams aizsargāt datus no nolasīšanas, tiek izmantotas šifrēšanas metodes – tas ir datu apstrādes process, kad datus pēc definēta algoritma samaina, izmantojot elektronisko atslēgu. Un iespējams ar tādu pašu atslēgu datus atjaunot pirmsšifrēšanas stāvoklī. Otrā gadījumā, kad nepieciešams pārlicināties, ka datus sagatavoja ticams avots, tiek izmantota *publiskās atslēgas* (ang. *public key*) metode. Tā ir metode, kur ir atslēga [Lza2011], ko publiskās atslēgšifrēšanas procesā lietotājs izplata saviem potenciālajiem korespondentiem un ko šie korespondenti izmanto, lai šifrētu lietotājam adresētos ziņojumus un atšifrētu lietotāja signatūru, kas šifrēta ar lietotāja privātu atslēgu.

Kritērijam ir definētas iebūvētas aizsardzības metodes (algoritmi) ar kurām ir nodrošināta datu aizsardzība no nesankcionētas datu piekļuves. No algoritmu kopas ir aprēķināts konkrētās tehnoloģijas aizsardzības līmenis.

Identificēšanas metodes

DARSIR koncepcijā ir paskaidrots, ka resursa identifikators ir EPC Type I, šis identifikators ir paredzēts svītrkodu un RFID tehnoloģijām. Tādas tehnoloģijas kā Bluetooth, LAN, Wi-Fi izmanto citu identifikatoru – *mediju piekļuves kods* (ang. *Media Access Code, MAC*). Tas ir 48-bitu kods, kas iepriekš izmantots tīkla ierīcēs, bet tagad daudzu datorsistēmu ierīcēs ir iebūvēts šis identifikators. Eksistē šis identifikatora paplašinājumi, kas aizņem 64-

bitus. MAC ir piesaistīts ierīcei ražošanas brīdī, tas ir unikāls, un parasti to nav iespējams izmainīt.

Šis kritērijs definē, vai ir iespējams tehnoloģijai izmantot EPC Type I identifikatoru, vai eksistē cits identifikators, kurš obligāti jāizmanto resursa identificēšanai. Jebkurai tehnoloģijai ir birkas datu apjoms, kurā papildus iespējams pievienot EPC Type I identifikatoru, cita identifikatora (piemēram, MAC) papildus izmantošana var nodrošināt drošu resursa līdzekļu identificēšanai.

Vislabākais ir gadījums, kad tiek nodrošināts resursa identifikators EPC Type I. Viduvējais variants ir, kad izmantoti cita veida unikālie identifikatori (piemēram, MAC). Visliktākais variants, kad ir nepieciešams atsevišķi nodrošināt resursa identifikatora ģenerēšanu un glabāšanu birkas pamata datos.

Skenēšanas precizitāte

Aizsardzības mehānisms, kurš var atpazīt kļūdas gadījumu birkas nolasīšanā no skenera, ir nosaukts par skenēšanas precizitāti. Šim kritērijam iespējams definēt procenta daļu no simts procentiem, kas parasti nozīmē varbūtību procentos uz simts gadījumiem. Piemēram, ja ir tiek analizēts svītrkodu skeneris, kurš no 100 gadījumiem nolasa 5 reizes nepareizi, tad tiek uzskatīts, ka svītrkodu skenerim skenēšanas precizitāte ir 95%.

Eksistē netipiski vides apstākļi, kas var papildus ietekmēt šo kritēriju, piemēram, tādi kā: augsta temperatūra, paaugstināts mitrums, gaisa piesārņojums utt. Parasti šis kritērijs ir aprēķināts no vidējiem vides apstākļiem, ja risinājuma darbība jānodrošina netipiskos vides apstākļos, tad nepieciešams nodrošināt papildu eksperimentu šo kritēriju noskaidrošanai.

4.3.2. Tehnoloģiju izvēles aprēķinu metode

DARSIR koncepciju iespējams iedarbināt ar dažādu veidu tehnoloģijām (skat. detalizētāk 2. nodaļā). Konkrētu tehnoloģiju izvēle notiek, izvērtējot IT risinājuma nepieciešamību. Parasti tiek atrasts kompromiss starp iespējām un izmaksām. Tiek definēti tehnoloģiju kritēriji, un katrai tehnoloģijai ir savs tehnoloģijas tipa parametrs. Katram tehnoloģijas tipa parametram tiek definēta vērtība diapazonā no 0 līdz 1, jo lielākā parametru vērtība, jo labāk parametrs piemērots DARSIR koncepcijas risinājumam. Piemēram, birkas datu glabāšanas apjoms, *lineāra svītrkodu tehnoloģija*. Tās apjoms ir 10 baitu un vērtība ir 0.1. *Divu dimensiju svītrkodiem* 100 baitu un vērtība ir 0.6. *RFID daļēji aktīvām birkām* apjoms ir 1000000 baitu un vērtība ir 0.8.

Vēl tiek definēts tehnoloģijas parametru ietekmēšanas koeficients, kuram ir pieļaujama vērtība diapazonā no 0 līdz 1. Piemēram, ja konkrētam risinājumam jau izmantota *svītrkodu tehnoloģija*, tad tehnoloģijas tipa ietekmēšanas koeficients ir 1. Gadījumos, kad būs

nepieciešamība aprēķināt divu dimensiju svītrkodus, tad ietekmēšanas koeficients ir 0.5, bet *RFID* tehnoloģijai vajadzīgas jaunas izmaksas, tāpēc vērtība ir 0.1.

Katrai tehnoloģijai ir definēti raksturojumi un ierobežojumi. Ir gadījumi, kad būtiski tehnoloģijas ierobežojumi neļauj izmantot konkrētu tehnoloģiju. Ja nozīmīgam tehnoloģijas tipa parametram uzstāda mazāko iespējamo vērtību, tad šie ierobežojumi tiks aprēķināti tehnoloģijas izvēles laikā. Piemēram, iepriekšējā gadījumā ar birkas datu glabāšanas apjomu. Pieņemsim, ka IT risinājumam ir nepieciešams datu apjoms 90 baitu, kas atbilst vērtībai 0.5. *Lineāriem svītrkodiem* šī vērtība ir 0.1, *divu dimensiju svītrkodiem* ir 0.6, un *RFID daļēji aktīvajiem birkām* ir 0.8. Tas nozīmē, ka visas tehnoloģijas atbilst uzdevumam, izņemot *lineāros svītrkodus*.

IKKATN (saīsinājums no *Ietekmes Kritēriju Kombinācijas Aprēķins Tehnoloģiju Novērtējumam*) tiek aprēķināts pēc formulas:

$$IKKATN = \sum_{i=1}^n a_i h(c_i, t_i), \quad (4.1a)$$

kur n ir tehnoloģiju skaits, i ir tehnoloģijas tipa numurs, a_i ir i -tās tehnoloģijas tipa ietekmēšanas koeficients, c_i ir i -tās tehnoloģijas tipa parametrs, t_i ir i -tās tehnoloģijas tipa parametra pieļaujamo vērtību sliekšnis un h ir sliekšņa funkcija, kas saņem vērtības atbilstoši formulai:

$$h(c, t) = \begin{cases} c, & \text{ja } c \geq t \\ +\infty, & \text{ja } c < t. \end{cases} \quad (4.1b)$$

Svarīgu parametru koeficienti ir nosaukti par *ietekmes koeficientiem* (ang. *Impact Factor, IK*). DARSIR koncepcija pamatā ietekmē objektu identificēšanas tehnoloģijas (1. grupa, skat. 3. nodaļā). IK uz objektu identificēšanas tehnoloģijām ir apvienoti 4.3. – 4.4. tabulās. Tabulās ir norādīta vispārīgā informācija, kas nozīmē ka, katrai tehnoloģijai ir dažādas modifikācijas un katrai modifikācijai ir iespējams definēt savus IK.

4.3. tabula

Fizisko parametru un ārējās vides tehnoloģiju IK

(1 – birkas datu apjoms; 2 – pārrakstīšanas iespēja; 3 – birkas izmērs; 4 – birkas svars; 5 – netipiskie vides apstākļi; 6 – skenera nolastīšanas ātrums; 7 – skenēšanas attālums; 8 – skenēšanas režīma tiešā redzamība; 9 – skenera daudzkārtēja birku lasīšana; 10 – tehnoloģiju atpazīšana, izplatīšana un pieejamība)

IK	1	2	3	4	5	6	7	8	9	10
Tehnoloģija										
<i>Svītrkods (Code128)</i>	0.2	0.1	0.9	0.9	0.9	0.4	0.3	0.1	0.1	0.8
<i>Svītrkods (DataMatrix)</i>	0.6	0.2	0.9	0.9	0.9	0.5	0.2	0.1	0.4	0.8
<i>Svītrkods (QR Code)</i>	0.6	0.2	0.9	0.9	0.9	0.5	0.2	0.1	0.4	0.8
<i>RFID (pasīvais)</i>	0.6	0.1	0.9	0.9	0.9	0.7	0.6	0.7	0.9	0.8
<i>RFID (daļēji aktīvais)</i>	0.8	0.7	0.8	0.8	0.2	0.7	0.8	0.7	0.9	0.6
<i>RFID (aktīvais)</i>	0.9	0.9	0.2	0.5	0.1	0.5	0.9	0.8	0.7	0.4
<i>NFC</i>	0.6	0.5	0.9	0.9	0.7	0.7	0.6	0.7	0.4	0.6
<i>Bluetooth</i>	0.9	0.9	0.3	0.3	0.1	0.5	0.6	0.7	0.2	0.6
<i>IrDA</i>	0.9	0.9	0.3	0.3	0.1	0.5	0.6	0.3	0.2	0.5
<i>GSM</i>	0.8	0.6	0.3	0.3	0.1	0.3	0.8	0.7	0.7	0.8
<i>Wi-Fi</i>	0.9	0.7	0.2	0.2	0.1	0.3	0.7	0.7	0.6	0.9
<i>WiMAX</i>	0.9	0.9	0.2	0.2	0.1	0.2	0.9	0.7	0.6	0.5

4.4. tabula

Tehnoloģiju izmaksas IK

(1 – tehnoloģiju ieviešanas izmaksas; 2 – tehnoloģiju uzturēšanas izmaksas; 3 – skeneru izmaksas; 4 – birkas izmaksas; 5 – birkas dzīves cikls)

IK	1	2	3	4	5
Tehnoloģija					
<i>Svītrkods (Code128)</i>	0.7	0.8	0.6	0.9	0.9
<i>Svītrkods (DataMatrix)</i>	0.7	0.8	0.7	0.9	0.9
<i>Svītrkods (QR Code)</i>	0.7	0.8	0.7	0.9	0.9
<i>RFID (pasīvais)</i>	0.6	0.8	0.6	0.8	0.9
<i>RFID (daļēji aktīvais)</i>	0.5	0.7	0.6	0.7	0.7
<i>RFID (aktīvais)</i>	0.4	0.5	0.5	0.6	0.6
<i>NFC</i>	0.7	0.8	0.6	0.8	0.9
<i>Bluetooth</i>	0.5	0.5	0.6	0.6	0.4
<i>IrDA</i>	0.4	0.5	0.6	0.6	0.4
<i>GSM</i>	0.3	0.4	0.4	0.6	0.3
<i>Wi-Fi</i>	0.5	0.5	0.4	0.5	0.3
<i>WiMAX</i>	0.4	0.4	0.3	0.4	0.2

Datu drošības un konfidencialitātes IK

(1 – iebūvētie kļūdu detektēšanas līdzekļi; 2 – iebūvētie datu atjaunošanas līdzekļi; 3 – iebūvētie nesankcionētas datu izmantošanas līdzekļi; 4 – identificēšanas metodes; 5 – skenēšanas precizitāte)

IK Tehnoloģija	1	2	3	4	5
<i>Svītrkods (Code128)</i>	0.4	0.1	0.1	0.9	0.2
<i>Svītrkods (DataMatrix)</i>	0.7	0.7	0.7	0.4	0.6
<i>Svītrkods (QR Code)</i>	0.7	0.7	0.7	0.4	0.6
<i>RFID (pasīvais)</i>	0.4	0.1	0.5	0.9	0.7
<i>RFID (daļēji aktīvais)</i>	0.5	0.4	0.6	0.9	0.8
<i>RFID (aktīvais)</i>	0.6	0.6	0.7	0.9	0.8
<i>NFC</i>	0.4	0.1	0.5	0.9	0.7
<i>Bluetooth</i>	0.5	0.4	0.8	0.7	0.7
<i>IrDA</i>	0.5	0.4	0.8	0.7	0.7
<i>GSM</i>	0.5	0.3	0.7	0.6	0.8
<i>Wi-Fi</i>	0.5	0.2	0.8	0.7	0.7
<i>WiMAX</i>	0.5	0.2	0.8	0.7	0.7

Vienas tehnoloģijas robežās iespējams izmantot dažādu ražotāju ierīces, kurām savukārt iespējami savā starpā dažādi parametri (tehnoloģijas vērtības). Vienas ierīces nomaiņa ar citu arī ietekmē IKKATN. Piemēram, mums ir viens kritērijs - svītrkodu birku izmērs, otrs ir svītrkodu birku datu apjoms. Ja paaugstina prasības datu apjomam, parasti tas ietekmē svītrkodu birkas izmēru.

4.3.3. Ietekmes koeficientu vienvērtīgās nozīmes aprēķins

Tiek definētas tehnoloģiju IK, kas ietekmē objektu individuālajās īpašībās sakņotas sistēmas ar objekta identificēšanas funkcionalitāti. Tālākajos pētījumos ir jāaprēķina, kuras no piedāvātajām tehnoloģijām ir vispiemērotākās DARSIR koncepcijai.

Pieņemsim, ka šim uzdevumam ir abstrakts risinājums, kuram nav būtisku ierobežojumu, un visiem IK vienvērtīgas nozīmes. Aprēķināsim IKKATN (pēc 4.1. formulas) visus IK, kas tiek definēti 12 pamata objektu identificēšanas tehnoloģijām. IK vērtības ir ņemtas no 4.3. – 4.5. tabulām.

Pirmajā IK grupā ir apvienota informācija par fiziskajiem parametriem un ārējo vidi (skat. 4.3. tabulā). Vislielākais IKKATN ar vērtību 7.1 ir pie RFID tehnoloģijas ar pasīvām birkām. Vislielākās priekšrocības ir birkas fiziskajiem parametriem (svars un izmērs). Turklāt

ir iespēja strādāt ar netipiskiem vides apstākļiem, ko pilnā apmērā nevar nodrošināt neviena cita tehnoloģija.

4.4. tabulā ir apkopoti dati par tehnoloģiju izmaksu IK grupām. Pēc IKKATN aprēķina vislielākā nozīme ir 2D svītrkodiem (vērtība 4) un NFC (vērtība 3.7) tehnoloģijām. Šis rezultāts ir saņemts pamatoties uz šīs tehnoloģijas ilgu dzīves ciklu un nelielām birkas izmaksām.

Trešajā IK grupā ir apvienota informācija par datu drošības un konfidencialitātes nodrošināšanu (skat. 4.5. tabulā). Vislielākais IKKATN ar vērtību ir 3.6 pie RFID tehnoloģijas ar aktīvām birkām. Tas rezultāts pamatojas uz to, ka RFID tehnoloģija ir vairāk piemērota objektu identificēšanai nekā citas.

Pēc IKKATN aprēķiniem ar vienvērtīgām IK nozīmēm ir secināts, ka DARSIR koncepcijai vispiemērotākās ir 2D svītrkodu un RFID tehnoloģijas. Tomēr RFID tehnoloģijai ir iespējama lielāka funkcionalitāte, salīdzinot to ar 2D svītrkodu tehnoloģiju. Piemēram, svītrkodiem grūti realizēt datu pārrakstīšanas iespējas, jādrukā jauna birka virs vecās.

4.4. Radio Frekvences Identifikācijas tehnoloģijas apskats

Iepriekšējā apakšnodaļā tika definēti DARSIR koncepcijas IK tehnoloģijas izvēlei un tika secināts, ka, ja ir vienādi ietekmes kritēriji, tiek rekomendēts izmantot RFID tehnoloģiju. Šis nodaļas ietvaros jāapskata populārākie ceļi RFID tehnoloģijas ieviešanai.

4.4.1. RFID tehnoloģijas standarti

RFID industrijā de-facto standarts ir EPCglobal standarts, tas ir konsorcijs, kas sastāv no tādiem industrijas līderiem kā Wal-Mart, Cisco, Hewlett-Packard un citi [EPC2010a].

Eksistē daudz uzņēmumu RFID izmantošanas stratēģiju, katram lielam uzņēmumam parasti ir sava stratēģija (skat. [Bau2005, Flo2005, Sap2009, EPC2010a]). Par pamatu bieži izmanto EPCglobal standartu, bet eksistē arī tendence, kad lielas programmatūru ražošanas kompānijas izstrādā pašas savu programmatūru, terminoloģiju un datu apmaiņas standartus. Piemēram, IBM izmanto kopā ar XML arī savu standartu MQ [Jun2004]. SAP kompānija izmanto savu izstrādātu arhitektūru, kura tiek iebūvēta SAP aplikācijā [Sap2009]. Ir pētījumi, kuros izmanto vienkāršotu RFID starpprogrammatūras pieeju (piemēram, SmartRF [Gha2008], IBM Lightweight RFID bus framework [Jun2004]), kuru raksturo ar vienkāršu arhitektūru un izmantošanas iespējām, kas piemērotāka maza biznesa vajadzībām.

RFID tehnoloģija ir interesanta priekš analīzes, kas attiecas uz konkrēto problēmu risināšanu; ir iespēja samainīt ne tikai programmatūru, bet arī tehnisko nodrošinājumu.

Tipiskie tehnisko nodrošinājumu risinājumi, kuri ir apraksti literatūrā [Bau2005, Fer2010, hei2005], ietver sevī viedo paliktņi, nolasītāju, RFID printeri, drošības karti, viedo ieroci utt.

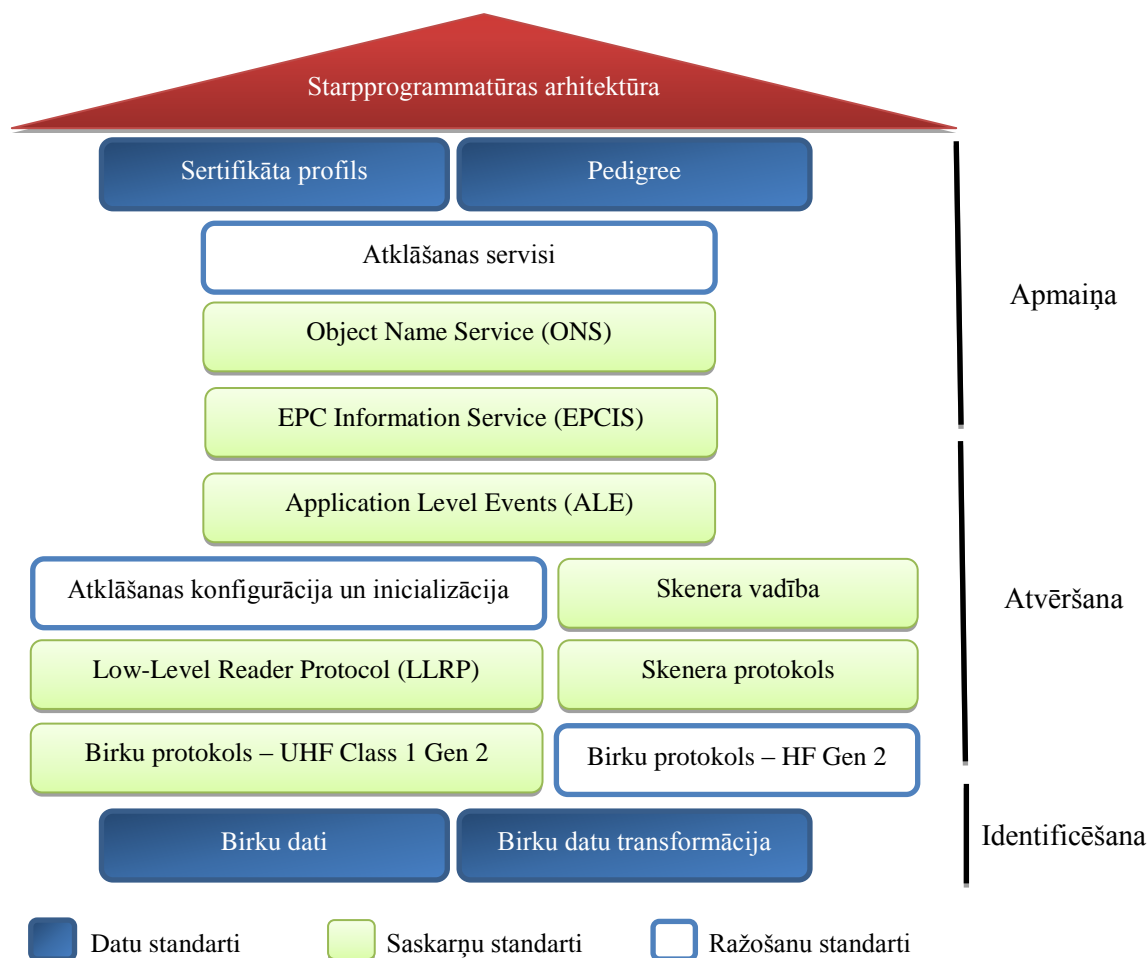
RFID tehnoloģijā fiziska objekta identificēšanai izmanto *elektroniskais produkta kods* (ang. *Electronic Product Code, EPK*), parasti tiek izmantots *EPC Type I* (detalizētāk skat. [Bro2001a, Bro2001c]) formāts. Šis identifikators ir pilnīgi identisks DARSIR koncepcijas izmantotajam. Datu apmaiņas mehānisms tiek nodrošināts ar *elektronisku produkta koda starpprogrammatūru* (ang. *Electronic Product Code Middleware, EPC Middleware*) – tie ir pakalpojumi, kas realizē datu apmaiņu starp EPK lasītāju (vai tīklu lasītājiem) un lietojumprogrammatūru, kura balstās uz RFID tehnoloģiju. Starpprogrammatūras komponenti nodrošina darbojošos notikumu filtrēšanu par EPK fiziskajiem objektiem, un biznesa lietojumprogramma tikai saglabā ienākošo un pamesto lasītāju rindu.

Praktiskā RFID tehnoloģijas lietojumprogramma ietekmē starpprogrammatūras arhitektūru, tādēļ tālāk šī tēma tiks detalizēti izskatīta. Lai tālāk aprakstītu RFID tehnoloģijas aspektus, nepieciešams definēt dažus terminus:

- *Arhitektūras struktūra* – ir definēta standarta atbalsta struktūra, uz kuras bāzes daži projekti varētu tikt organizēti un izstrādāti.
- *Starpprogrammatūras arhitektūra* – komunikācijas slānis, kurš ļauj lietojumprogrammām savstarpēji iedarboties starp aparatūras un tīkla vidi. Starpprogrammatūra nodrošina komunikāciju starp atšķirīgām sistēmām.

4.4.2. EPCglobal rekomendētais risinājums

RFID industrijā izmanto EPCglobal standartu kā faktisko standartu. EPCglobal tas ir industrijas konsorcijs, tādi līderi kā Wall-Mart, Cisco, Hewlett-Packard un citas. Arhitektūras struktūra parādīta 4.4. attēlā [EPC2010a].

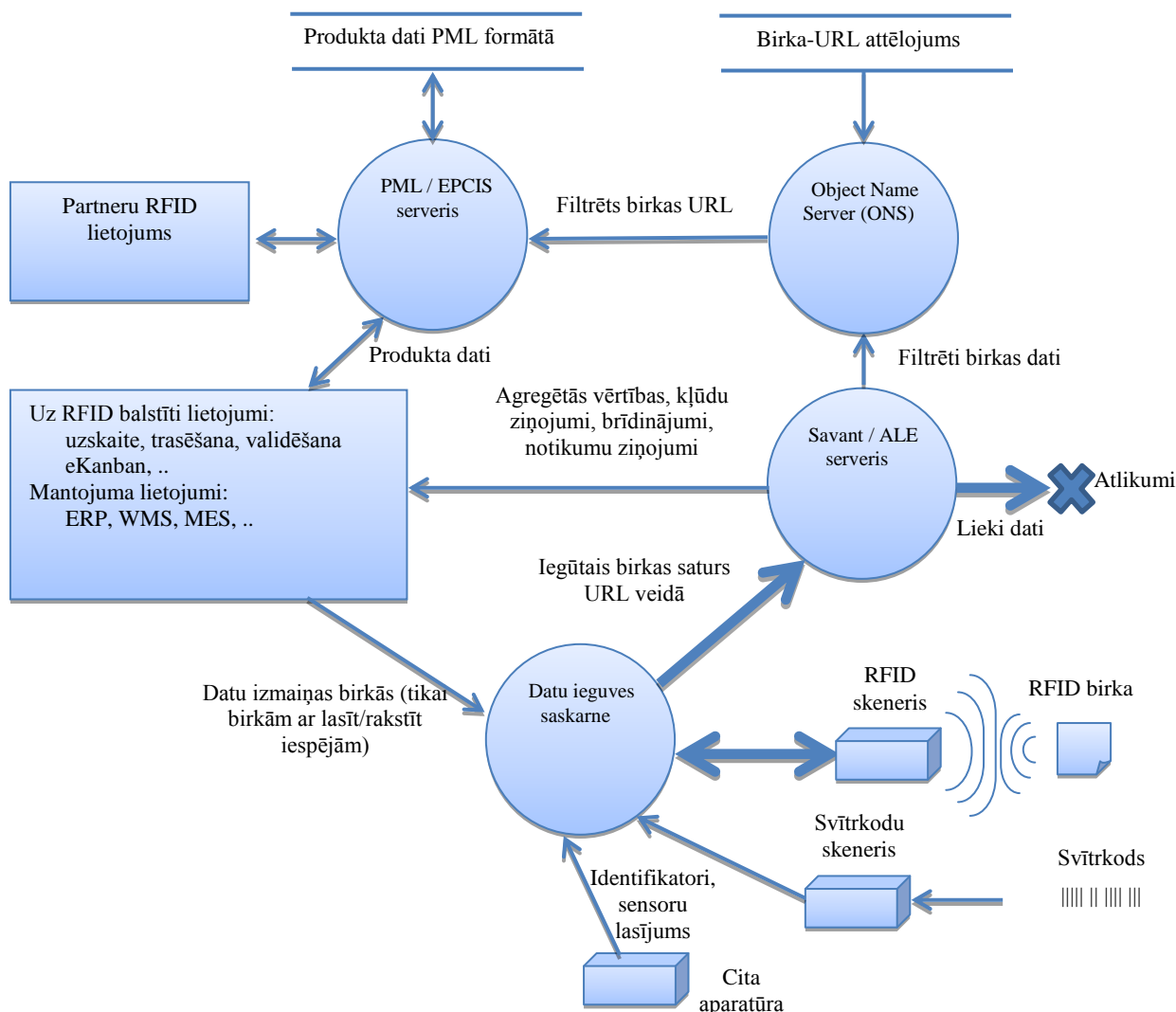


4.4. attēls. EPCglobal standarts (adaptēts no [EPC2010a])

Tālākam RFID tehnoloģijas arhitektūras apskatam interesē nākamās starpprogrammatūras arhitektūras elementi. Pamatelementi, kas izmantoti starpprogrammatūras arhitektūrā:

- *EPCIS* (ang. *EPC Information Services*) [EPC2010b] – datu apmaiņas standarts, kas definē apmaiņas mehānismu, kuru varētu izmantot EPK bāzētām lietojumprogrammām, lai apmainītu EPK datus un notikuma informāciju.
- *ALE* (ang. *Application Level Events*) [EPC2010c] – interfeisa definīcijas standarts, kurš definē, ka iespējams filtrēt un apvienot EPK datus no dažādiem avotiem. Interfeisa definīcija, kas definē, ka uz EPK balstīta lietojumprogramma savstarpēji iedarbojas ar RFID tīkla lasītājiem.
- *ONS* (ang. *Object Name Service*) [EPC2010d] – tas ir servisa standarts, kurš ir nepieciešams, lai pārveidotu EPK kodu par Internet adresi (URL'u), kurā glabājas objektu dati.

EPCglobal rekomendētā starpprogrammatūras arhitektūra tika parādīta 4.5. attēlā.



4.5. attēls. EPCglobal starpprogrammatūras arhitektūra (adaptēts no [Bau2005])

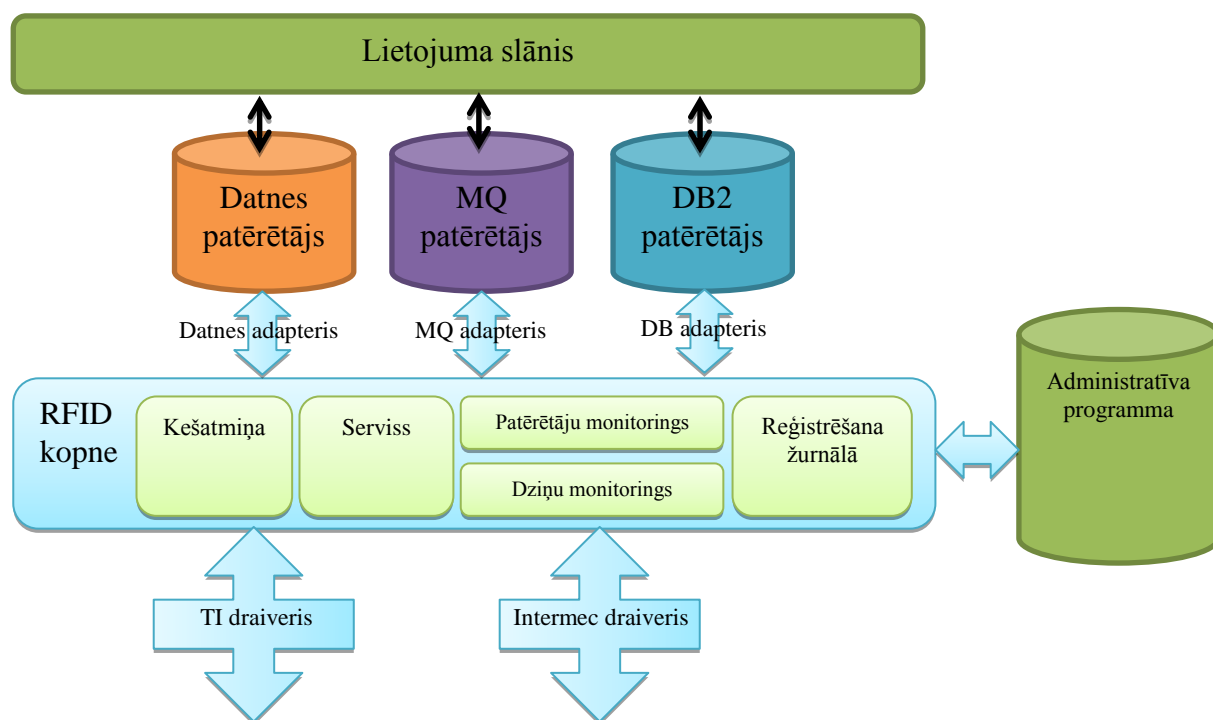
Šajā starpprogrammatūras arhitektūrā iespējams saņemt informāciju no dažādiem avotiem (tādiem kā RFID tagi, svītrkodi, nolasītāji utt.). Datus, ja ir nepieciešams, varētu kopēt PML'ā komunikācijām ar citām lietojumprogrammām. Nav tik svarīgi, lai izmantotu priekš biznesa lietojumprogrammas, bet pamatā ir vietas bez datu iegūšanas interfeisa, ALE serveris, ONS, EPCIS serveris un uz RFID balstītas lietojumprogrammas.

Šo starpprogrammatūras arhitektūru lieto lielajā biznesā, jo pastāv nepieciešamība strādāt ar lielāku datu apjomu un dažādiem partneriem uz RFID balstītām lietojumprogrammām.

4.4.3. Risinājums ražošanai

Cik daudz pastāv kompāniju, kuras realizē RFID risinājumus, tik daudz arī pastāv iespējamo atšķirīgo stratēģiju RFID lietošanai. Vairākums no kompānijām lieto EPCglobal standartus, bet dažreiz lieto atšķirīgu terminoloģiju un datu apmaiņas standartus (piemēram, IBM lieto ne tikai XML, bet arī savu datu apmaiņas standartu MQ; SAP kompānijā ir sava arhitektūra, kas ir iebūvējama SAP lietojumprogrammā).

Bet detalizētai starpprogrammatūras arhitektūru izstrādāšanas iepazīšanai ir izvēlēts IBM kompānijas “Lightweight RFID bus framework” produkts mazajam biznesam. Izvēle balstās uz mērķauditoriju, kurai tiek izstrādāts risinājums, tas ir paredzēts maziem biznesa risinājumiem vai risinājumiem ar atvieglotu programmatūras arhitektūru. IBM RFID sistēmas starpprogrammatūras arhitektūra parādīta 4.6. attēlā.



4.6. attēls. “Lightweight RFID bus framework” starpprogrammatūras arhitektūra (adaptēts no [Jun2004])

Attēlā parādīta RFID kopne, kas ietver visus komponentus, kuri ir nepieciešami, lai vieglāk būtu strādāt ar RFID lasītājiem (tādiem kā TI, Intermec, utt.). RFID lasītāji lasa taga informāciju, pēc tam apkopo to un sūta uz RFID kopni. Diskdziņa monitors ir iebūvēts RFID kopnes serdē, saņem taga informāciju no RFID lasītājiem. Patērētāja monitors saņem taga informāciju un sūta to uz klienta definēto adapteri (tādu kā Files Adapter, MQ Adapter, DB2 Adapter utt.) vai pārvieto caur RFID kopni.

Klientam RFID kopne ir kā melnā kaste, kas dod informāciju par birkām. Lietojumprogrammas klientam ļoti svarīgi definēt datu glabātavu (klienta definēto adapteri) un datu formātu taga informācijas apmaiņai.

4.5. Drošības jautājumu apskats

DARSIR koncepcijas jānodrošina ar (*viii*) kritēriju par *konfidencialitāti, integritāti un pieejamību* (skat. 1.2. tabulā). Vispirms ir jāapskata tehnoloģiju drošības riski. Tāpat kā iepriekšējās nodaļās par pamatu tiek ņemta RFID tehnoloģija un apskatīta RFID tehnoloģijas specifiska uzbrukumu taksonomija. Tiek veikta analīze, cik aktuālas DARSIR koncepcijai ir RFID tehnoloģijas informācijas drošības problēmas. Rezultātā tiek saņemti aktuālie DARSIR koncepcijas drošības riski un tiek definēti jaunie aktuālie riski, kas ir specifiski DARSIR koncepcijai. Pēdējais solis ir definēt DARSIR koncepcijas RFID tehnoloģijas pamatrisku mazināšanas pasākumus.

4.5.1. RFID tehnoloģijai specifiskā uzbrukumu taksonomija

Informācijas drošības sistēmas funkcionēšanas process ir iteratīvs (ciklisks) un realizē risku pārvaldības metodes, kas ietver sevī riska identifikāciju un prioritātes definēšanu, aizsardzības metožu izstrādi, implementāciju un atlikušo risku novērtēšanas pasākumus [Itg2006]. RFID sistēmām, kā arī visiem citiem tehnoloģiskiem risinājumiem ir daži specifiski informācijas drošības riski šo risinājumu arhitektūras un izmantoto tehnoloģiju dēļ. Vispārējie RFID tehnoloģijas specifiskie riski, saistītie uzbrukumi un riska samazināšanas mehānismi (pretpasākumi) tika izstudēti un aprakstīti dažādās publikācijās [Lah2005, Hei2005, Bau2005, un Tho2006]. Tajā pašā laikā drošības risku stāvoklis mainās, ja tiek izmantota jauna arhitektūra vai tehnoloģiskais risinājums (detalizētāk skat. [Kry2008]). Daži riski paliek aktuāli, jo risku radīšanas notikums atkarīgs no tehnoloģijas izmantošanas nepieciešamības. Daži riski kļūst nederīgi. Tajā pašā laikā jaunā tehnoloģija un jaunā arhitektūra ienes jaunus informācijas drošības riskus, kas var ilglaicīgas ignorēšanas gadījumā būtiski ietekmēt jauno ideju un koncepciju ieviešanu un tālāku izmantošanu.

Tālāk ir apskatīti galvenie uzbrukumu tipi, kas tiek veikti RFID sistēmās. Par pamatu tika ņemta informācija no pētījuma [Tho2006]. Ērtību dēļ aprakstam par piemēru tika ņemtas tirdzniecības sistēmas. Dažreiz kāds, kas uzbrūk RFID sistēmai, var izdarīt tā, lai nozagtu vienu objektu, kamēr otrs var uzbrukt, lai uzreiz paralizētu veikala vai pat veikalu tīkla darbu. Uzbrucējs var mēģināt ievietot dezinformāciju konkurenta aizmugursistēmas datubāzē tā, lai šī datubāze kļūtu nederīga. Urķi var mēģināt krāpt fiziskās piekļuves kontroli bez jebkādas

intereses par datiem. Tāpēc ikvienam ir nepieciešams zināt, kā viņu darbs/sistēma ir aizsargāta, no kurienes uz tiem var uzbrukt.

Pastāv daudz veidu, kā var uzbrukt RFID sistēmai. Šeit ir aprakstīti daži no tiem:

- *Izlikšanās uzbrukumi* (ang. *spoofing attack*) piegādā nepareizo informāciju, kas izskatās pareiza un kuru sistēma pieņem. Izlikšanās uzbrukumiem ir raksturīgi izmantot viltotu domēna vārdu, *interneta protokola* (ang. *Internet Protocol, IP*) adresi vai *mediju piekļuves koda* (ang. *Media Access Code, MAC*) adresi. Izlikšanās uzbrukuma piemērs RFID sistēmā ir nepareiza EPK numura apraide, kad derīgs numurs tiek gaidīts [Tho2006].
- *Ievietošanas uzbrukumi* (ang. *insert attack*) ievieto sistēmas komandas tur, kur parasti tiek gaidīti dati. Šie uzbrukumi darbojas, jo ir pieņemts, ka noteiktajās vietās vienmēr tiek ievadīti dati un netiek veikta validācija šiem ievadītajiem datiem. Ievietošanas uzbrukumi ir raksturīgi Internet lapām, kur ļaunprātīgo kodu injicē Interneta lietojumprogrammatūrā. Tipiskais šī tipa uzbrukums ir injicēt SQL komandu pieprasījumā datubāzei. Tas pats princips var tikt izmantots RFID tehnoloģijas gadījumā, ierakstot tagā sistēmas komandu nevis derīgos datus (piemēram, EPK numura vietā) [Tho2006].
- *Atkārtotās uzbrukuma* (ang. *replay attack*), derīgs RFID signāls tiek notverts un tā dati tiek ierakstīti; šie dati pēc tam tiek sūtīti RFID lasītājam, kur tiek “atskaņoti”. Tāpēc ka dati izskatās derīgi, sistēma akceptē tos [Tho2006].
- *Servisa atteikšanas uzbrukumi* (ang. *Denial-Of-Serviss Attack, DOS*), arī zināmi kā *surogātpasta* (ang. *flood*) uzbrukumi. Tie notiek, kad signāls ir pārpildīts ar daudz lielāku datu apjomu nekā tos var apstrādāt. Šī uzbrukuma variācija ir *radiofrekvences traucēšana* (ang. *jamming*), kas ir ļoti labi zināma radio pasaulē. Tā notiek tad, kad radiofrekvence tiek pārpildīta ar trokšņainu signālu. Jebkurā gadījumā rezultāts ir tāds pats: sistēma nespēj pareizi apstrādāt ienākošos datus. Jebkura variācija var tikt izmantota, lai uzbruktu RFID sistēmai [Tho2006].
- *Tagu datu manipulācijas uzbrukumi*. Atkarībā no taga izcelsmes, tā cena, tirgu numurs un jebkuri citi dati var tikt mainīti. Kad objekts ar modificētiem birkām tiek nopirkts, izmantojot pašpalpošanās kases, neviens nevar noteikt izmaiņas. Tikai fiziskā inventarizācijas atklās, ka pārdošanas žurnālā dotais objekts tika pārdots par citu cenu. RF Dump programma ir uzrakstīta Java valodā. Programma skenē RFID tagus ar lasītāju, kas ir pievienots datoram caur seriālo portu. Kad lasītājs noteic tagu, programma prezentē taga datus tabulas veidā uz ekrāna. Lietotājs var šos datus ievadīt

vai mainīt un atspoguļot šīs izmaiņas tagā. Šī programmatūra var arī tikt izmantota mobilajās ierīcēs [Tho2006].

- *Starpprogrammatūras uzbrukumi* var notikt jebkurā posmā starp lasītāju un aizmugursistēmu. Tomēr iespējams, ka vājākais punkts ir LAN. Šī ierīce var zagt (ang. sniff) datus no LAN, lai pēc tam izmantotu tos atkārtotā uzbrukumā, vai tā arī var injicēt datus LAN, izraisot DOS uzbrukumu pret maksāšanas sistēmu. Šī ierīce var arī veikt neautorizēto pārraidi. Cita iespēja var būt, ka tehniski izglītota persona, kas izpilda darbu organizācijā, var darīt to tikai tāpēc, lai iegūtu piekļuvi starpprogrammatūrai. Daži tādi uzbrukumi notiek, kad kāds iegūst zemu apmaksātu darbu, kas dod viņam piekļuvi mērķa sistēmai [Tho2006].
- *Aizmugursistēmas uzbrukumi* (ang. *back-end attack*). Tā kā aizmugursistēmas datubāze bieži ir attālinātākais no RFID taga punkts sistēmā, datu jomā un fiziskās distances jomā, tas var izskatīties par nepiemērotu mērķi uzbrukumiem, bet tomēr tas ir un būs uzbrukumu mērķis. Datubāzēs var būt vērtīga informācija, ja, piemēram, tajā ir tāda informācija kā klientu kredītkaršu numuri. Datubāzē var arī būt tāda vērtīga informācija kā pārdošanās atskaites vai uzņēmējdarbības noslēpumi, kas ir ļoti pievilcīga konkurentiem. Manipulēšana ar datubāzi var arī nodarīt reāli daudz postošākās sekas nekā klientu zaudēšana [Tho2006].
- *Jauktie uzbrukumi*. Uzbrukumi var tikt izmantoti arī kombinācijās. Daudzu veidu uzbrukumi, kas tika mērķēti uz RFID sistēmām, tika veikti arī pret atsevišķām apakšsistēmām. Tomēr uzbrucēju attapība un gudrība iespējams novedīs pie jauktiem uzbrukumiem. Uzbrucējs var uzbrukt pārdēvēja radio frekvenču interfeisam ar savu vīrusa tagu, kas var pēc tam izmantojot tuneli caur starpprogrammatūru un pavēlēt aizmugursistēmas datubāzei veikt datu saglabāšanu kādā nezināmā Internet lapā caur anonīmu serveri [Tho2006].
- *Cilvēks, kurš veido uzbrukumu no sistēmas iekšienes* (ang. *Man in the Middle Attack, MITM*) ir tāds uzbrukums, kas izmanto savstarpējās uzticības priekšrocību. MITM uzbrukumus veic nezināma puse komunikācijā, kas nodod informāciju turp un atpakaļ, radot visiem pārējiem priekšstatu, ka tā ir cita puse. RFID ir īpaši jūtīga pret MITM uzbrukumiem tās mazo izmēru un zemās cenas dēļ. Vairums RFID tehnoloģiju sazinās ar jebkuru lasītāju, kas atrodas pietiekami tuvu, lai uztvert signālu. Taga nolasīšanā nav lietotāja mijiedarbības un nenotiek nekādas lasītāja autentifikācijas. Tāpēc ir iespējams pietuvoties kādam RFID tagam ar lasītāju, kas ir noregulēts uz īpašu frekvenci, un nolasīt vai mijiedarboties ar tagu, un taga saimnieks par to pat nezina, kamēr viņa tags tiek emulēts vai ar to veic atkārtošānu [Tho2006].

- *Desinhronizācijas uzbrukuma* mērķis ir sagraut galveno modernizāciju, atstājot tagu un lasītāju desinhronizētā stāvoklī un padarot autentifikāciju neiespējamu [Van2009].

4.5.2. Informācijas drošības problēmu piemērotība DARSIR koncepcijai

IS, kas ir bāzētas uz RFID tehnoloģiju, parasti izmanto centralizētas datu vadības principus (detalizētāk skat. [Lah2005, Hei2005, Bau2005]). Tā ir galvenā DARSIR koncepcijas risku atšķirība no riskiem, kas piemīt vienkārši RFID tehnoloģijas realizācijai. Riski, kas attiecas ārējiem datu avotiem nav aktuāli DARSIR koncepcijai. Tomēr riski, kuru mērķis ir RFID tagi (resursi) kļūst par ļoti bīstamiem.

Parastā DARSIR koncepcijas realizācija ir slēgtā sistēma, tāpēc uzbrukuma risks no Interneta vai LAN ir zemāks. Iepriekš tika izstudēta RFID tehnoloģijai specifisko uzbrukumu taksonomija. Nepieciešams izpētīt identificētos uzbrukumus RFID tehnoloģijai un tās piemērotību DARSIR koncepcijai:

- *Izlikšanās uzbrukumi* var tikt mērķēti uz DARSIR resursu. DARSIR resurss var tikt aizstāts ar krāpnieku resursu, kas rada pārkāpumu sistēmā un veic kaitīgas operācijas. Krāpnieks, izmantojot šādu uzbrukumu, var mainīt informāciju DARSIR bāzētā informācijas sistēmā un veikt kaut kāda veida manipulācijas ar to.
- *Ievietošanas uzbrukumi* var būt īstenoti DARSIR sistēmā, tomēr šo uzbrukumu nav iespējams veikt ar skenera palīdzību, jo skeneris parasti nesatur izpildīšanas kodu. Iespējams realizēt uzbrukumu resursam (RFID tags), un ievietot bīstamo kodu tur, bet, ja sistēma tiek labi administrēta no resursu piekļuves tiesību viedokļa, šis risks ir minimāls. Tas nav tipisks ievietošanas uzbrukuma veids, tādus uzbrukumus sauc par tagu datu manipulācijas uzbrukumiem.
- *Atkārtēšanas uzbrukums* ir iespējams DARSIR sistēmā, un visiem ir jāizpilda visi pretpasākumi, lai novērstu sistēmas bojājumus.
- *DOS uzbrukumi* var tikt izmantoti DARSIR sistēmā. Daži DARSIR resursi var palikt nerasāmi un nepieejami uz kādu laiku, padarot sistēmu nelietojamu uz šo laiku, kad tiek veikti šāda tipa uzbrukumi.
- *Tagu datu manipulācijas uzbrukumi* arī var mainīt informāciju tagā, veicot nelielas izmaiņas DARSIR sistēmā, bet šo izmaiņu bīstamību nedrīkst nepietiekami novērtēt, šo izmaiņu apjoma dēļ tas var lielā mērā ietekmēt DARSIR sistēmu, ja cilvēks, kas veic šo uzbrukumu, ir pietiekami kompetents. Tāpēc, kad tiek izvēlētas drošības metodes, ir jāņem vērā šī tipa uzbrukumi.
- *Starpprogrammatūras uzbrukumi* var notikt DARSIR sistēmā, un šo uzbrukumu risks var tikt samazināts, ja piekļuves tiesības ir pareizi sadalītas starp visiem lietotājiem.

- *Aizmugursistēmas uzbrukumi* nav iespējami DARSIR sistēmā, jo DARSIR koncepcija neparedz centralizētu datubāzi, visi dati tiek glabāti resursos (RFID tagos).
- *Jauktie uzbrukumi* ir iespējami DARSIR koncepcijā. Uzbrucējam ir nepieciešams spēt sūtīt informāciju. Tātad viņam būs, piemēram, jāizveido savienojums ar Internetu caur Wi-Fi, kas palielina uzbrucēja identificēšanas varbūtību.
- *MITM uzbrukumi*. Ja objekti nav pietiekami aizsargāti, DARSIR sistēma var kļūt viegli ievainojama, jo, ja krāpniekam ir pilna pieeja sistēmai, viņš var darīt gandrīz jebko, ko viņš var iedomāties. Lai novērstu to DARSIR sistēmā, ir jāizstrādā paroļu sistēma.

Pastāv arī specifiski uzbrukuma veidi, kas ir raksturīgi DARSIR koncepcijai:

- *Uzbrucējtags*. Visa informācija (atribūti un noteikumi) tiek glabāta lokāli resursā. Ja resurss nav pietiekami aizsargāts (piemēram, informācija nav šifrētā), tad pastāv iespēja pievienot uzbrucējtagu, kas ievieto uzbrukuma noteikumus informācijas sistēmā. Kad uzbrukuma noteikumi tiek izpildīti, ir iespējams paņemt dažu funkciju kontroli informācijas sistēma vai saņemt konfidenciālu informāciju.
- *Falsificēts tags*. Resursā (RFID tagā) var tikt glabāti atribūti, kas apraksta neeksistējošas resursa īpašības vai bojātas resursa īpašības. Piemēram, paliktņis, kuram ir atribūti, kas apraksta paliktņa izmērus un svaru. Ja šī informācija tiek bojāta, tas var novest pie tehniskām problēmām, kad šis paliktņis tiek transportēts.

4.5.3. RFID kontrole informācijas drošības risku samazināšanai

Dati, kas atrodas RFID tagos un sistēmā nav pasargāti no daudzu veidu uzbrukumiem, tāpēc ir jāveic pareizi drošības pasākumi, lai izvairītos no pārkāpumiem, kas izraisa sistēmas nestabilitāti vai iznīcināšanu. Ir jāatceras, ka neviens drošības pasākums nevar atvairīt visus iespējamus uzbrukumus – tā mērķis ir minimizēt draudu iespējamību vai minimizēt drauda iespējamus zaudējumus (ietekmi) gadījumā, ja tas notiks.

Daži no drošības pasākumiem, kas paredz sistēmas aizsardzību ir aprakstīti tālāk:

1. Lai novērstu visparastākos uzbrukumus DARSIR sistēmai, var tikt izmantota *publiskās atslēgas infrastruktūra* (ang. *Public Key Infrastructure, PKI*), tā, lai dati, kas atrodas tagā nebūtu atvērti un pieejami krāpniekiem. PKI tiek izmantota, lai autentificētu „rokaspiedienu” starp RFID tagu un lasītāju, un RFID starpprogrammatūra tiek izmantota, lai autentificētu „rokaspiedienu” starp lasītāju un tīklu. PKI ir komponentes, kas tiek izmantotas, lai izplatītu un pārvaldītu šifrēšanu un digitālā paraksta atslēgas caur centralizēto servisu.

2. Katram tagam ir jābūt savam unikālajam numuram (resursa identifikators, EPC Type I), lai tas varētu tikt inicializēts un lai DARSIR sistēma varētu pārbaudīt, ka tas ir derīgs tags (unikālais numurs tiek glabāts lietotāja atmiņā). EPK ir šifrēts lētā RFID tagā. EPK tīkls arī uztver un padara pieejamo (caur Internetu un autorizētajiem pieprasījumiem) citu informāciju, kas attiecas uz doto objektu. EPK informācijas serviss padara ar EPK tīklu saistītos datus pieejamus jebkuram servisam, kad to pieprasa RPML vai JSONR formātā.
3. Dati, kuri ir pieejami caur EPK informācijas servisu, iekļauj sevī taga nolasītos datus, kas tika saņemti no RFID starpprogrammatūras (piemēram, objektu monitoringa nodrošināšanas gadījumā); instances līmeņa dati, piemēram, izgatavošanas datums, derīguma datums utt.; objekta klases līmeņa dati, piemēram, produkta kataloga informācija. Atbildot uz pieprasījumiem, EPK informācijas serviss balstās uz dažādiem datu avotiem, kas eksistē uzņēmumā, pārveidojot šos datus RPML formātā. Kad EPK informācijas servisa datu avoti ir izplatīti visā piegādes ķēdē, jebkura nozare var veidot EPK pieejas reģistru, lai darbotos kā glabātava EPK informācijas servisa interfeisa aprakstiem. [Aut2002] definē protokolu piekļuvei EPK informācijas servisam.
4. RFID starpprogrammatūra ir programmatūra, kas tika izstrādāta, lai apstrādātu tagu vai sensoru datu plūsmas (notikumu datus), kas nāk no vienas vai vairākām ierīcēm. Tas veic filtrēšanu, agregāciju un tagu datu skaitīšanu, samazinot datu apjomu pirms to sūtīšanas uz *uzņēmumu programmatūru* (ang. *enterprise applications*).
5. Lasīšanas un rakstīšanas kontrole var būt īstenota tā, lai tikai autorizētie lietotāji ar paroli varētu ierakstīt informāciju sistēmā, bet visi citi lietotāji varētu tikai nolasīt šo informāciju. Visi lietotāji arī varētu būt sadalīti vairākās grupās, un visām grupām ir jābūt dažādām tiesībām. Katrai grupai ir jābūt tai pieejamo funkciju sarakstam. Tas palīdzētu šādas sistēmas pārvaldei.
6. Parolei un visai informācijai ir jābūt šifrētai tagos, lai tikai pareizie lasītāji/rakstītāji varētu atšifrēt un izgūt šo informāciju no birkām. Paroles pārbaude tiks veikta katram objektam (tags/lasītājs), kas plāno mainīt kaut ko DARSIR sistēmā.
7. Piekļuves kontroles mehānismi. Daži tagi īsteno piekļuves kontroles mehānismu savai nolasīšanas/rakstīšanās atmiņai. Pašreizējie RFID tagi neaizsargā unikālo identifikatoru. Daži tagi uzspiež autentifikācijas mehānismus pirms piešķirt lasīšanas vai rakstīšanas piekļuvi specifiskajiem atmiņas blokiem. Šeit praktiski tiek realizēta vai nu parasta paroles autentifikācija vai nu vienpusēja vai divpusēja uzdevumu

reagēšanas autentifikācija (piemēram, ISO 9798-2) ar simetriskajām atslēgām [Kno2004].

8. [Ish2003] publikācijā tiek piedāvāts tags, kas emitē tikai „Anonymous EPC”. Pēc tam aizmugursistēmas drošības centrs sniedz skaidrā tekstā EPK caur drošo kanālu, lai autorizētu entītijas. Paplašinātā versijā lasītāji var atkārtoti sūtīt anonīmo pieprasījumu drošības centram, kas ģenerē jauno ‘Anonymous EPC’. Pēc tam tags tiek papildināts ar to identifikatoru.
9. Tagu autentifikācija. Pastāv priekšlikumi protokoliem, kas autentificē tagu lasītājā un novērš tagu falsifikāciju. Darbā [Vaj2003] tika piedāvāti un analizēti daži parastie tagu autentifikācijas protokoli, piemēram, no darba [Fel2004] ņemts *SASL* (ang. *Simple Authentication and Security Layer*) protokols.
10. [Ayo2006] darbā tika piedāvāta ideja, kas tiek saukta par *APF* (ang. *Authentication Processing Framework*), kā viena no metodēm aizsardzībai pret neautorizētiem lasītājiem. Tas ir ļoti svarīgi aizsargāt sistēmu pret neautorizēto piekļuvi birkām, lai novērstu pārkāpumus pret informācijas konfidencialitāti, kas tiek glabāta šajos tagos. *APF* ir savstarpējā autentifikācija, kas veido sistēmu, kura spēs aizsargāties pret neautorizēto piekļuvi.

Drošības resursi, kuri tika minēti iepriekš, nav pietiekami, lai nodrošinātu drošību visiem iepriekš minētajiem riskiem. Pastāv divi riski, kas ir specifiski DARSIR koncepcijai: uzbrucējtags un falsificēts tags. Iespējamie virzieni drošības pasākumiem, lai samazinātu šos riskus ir:

- *Uzbrucējtags*. DARSIR koncepcijā resurss (RFID tags) glabā sevī noteikumus, kas var mainīt informācijas sistēmas funkcionalitāti. Jāizstrādā laba sistēma sazināšanai starp resursiem, kur tiek glabāta informācija par to, kuras funkcijas ar kuru resursu (RFID tagu) var tikt izpildītas un kuras nevar. Visiem mēģinājumiem izpildīt nelegālas funkcijas ir jābūt identificētiem un novērstiem un jāziņo par to drošības darbiniekam. Resursiem arī ir nepieciešams piedāvāt autentifikācijas iespēju informācijas sistēmā. Tas var tikt izdarīts, izmantojot iepriekš minētās metodes (piemēram, PKI).
- *Falsificētais tags*. Šī problēma var tikt atrisināta izmantojot vismaz divas metodes. Pirmā metode ir datu šifrēšana, izmantojot standarta metodes. Otrā, piedāvātās informācijas pārbaude.

4.6. DARSIR koncepcijas praktiskie pētījumi

Promocijas darba ietvaros ir aprakstīti teorētiskie pētījumi, kuru rezultāts ir DARSIR koncepcijas izstrāde. Papildu bija veikti praktiskie DARSIR koncepcijas pētījumi (pirmais bija [Zur2007]). Pamatā ir pielietotas JSONR un RPML valodas realizācijas. Pirmkārt tika definēti iespējamie datu tipi un to prezentācijas formāti. Rezultāts ir aprakstīts 1. pielikumā. Otrkārt ir parādīta atribūtu grupas un noteikumu grupas realizācija. Procesa apraksts ir dots 2. un 3. pielikumos. Pamatoties uz šo praktisko DARSIR koncepcijas realizācijas piedāvājumu tika izveidota JSONR un RPML programmēšanas valodas realizācija. Īpaša uzmanība tika pievērsta datu aizsardzības jautājumam, šo pētījuma rezultāts ir publicēts [Zur2011b] ziņojumā.

Nākamais solis bija programmatūras aprobācija reālos piemēros. Tika pētīts veids, kā izveidot palīgprogrammatūru DARSIR risinājuma izveidošanai. 4. pielikumā ir piedāvāts izmantot *vizuālo interaktīvo simulāciju*. Kā galvenais praktiskais rezultāts, pamatoties uz šīs nodaļas teorētisko pamatojumu, ir izveidots „*DARSIR simulation tool*” simulācijas rīks. Apraksts tika veikts ar komentāriem, kas attiecas uz DARSIR koncepcijas izmantošanu uz RFID tehnoloģijas bāzes. Piedāvātās pieejas rezultāti ir atspoguļoti [Zur2011c, Zur2008c] ziņojumos.

Koncepcijai ir piedāvāts viens tipisks risinājums "*Viedā veļas mazgātava*" (skat. 5. pielikumā) un [Zur2008b] ziņojumā. Risinājums, kas pilnībā atbilst DARSIR koncepcijas izveidošanas prasībām. Ir aprakstīts DARSIR risinājuma izveidošanas process. No risinājuma prasības sastādīšanas līdz risinājumu izveidošanai mobilajā ierīcē. Papildus ir ņemts netipisks uzdevums "*Ārštata pakalpojumu pārvaldības sistēmas uzbūve*", kas aprakstīts 6. pielikumā. Risinājumā ir veidota tikai programmatūra, neizmantojot fiziskus objektus. Ir veidots sfēras apskats un pētījumu virzieni. Tiek pielietots *daudzaģentu sistēmas* darbības princips koncepcijas pamatā. Realizēta simulācija ar NetLogo [Wil1999] programmatūru. Tiek sasniegti divi galvenie rezultāti. Pirmais, ka DARSIR koncepciju iespējams piemērot arī risinājumos, kuros nav izmantoti fiziskie objekti. Otrais, ka jaunu risinājumu izveidošanas pētījumi var prasīt papildus DARSIR koncepcijas atribūtu un noteikumu izveidošanu. Šo pētījumu rezultāti ir prezentēti [Zur2011a, Zur2012a, Zur2012b] ziņojumos. Papildus šim pētījumiem tika veikta DARSIR koncepcijas praktiskā pārbaude ar citiem uzdevumiem. Piemēram, [Zur2008a] pētījumā ir publicēti rezultāti no "*Ceļu satiksmes monitorings un vadīšana*" uzdevuma.

7. pielikumā ir parādīts piemērs jaunas funkcijas ieviešanai. Ir funkcijas, kas nepieciešamas DARSIR koncepcijai, bet pēc noklusējuma nav realizētas konkrētai tehnoloģijai. Ir parādīts, ka pielietojot universālu metodi, var realizēt pozicionēšanas funkciju

RFID tehnoloģijai. Daudzas šīs funkcijas īpašības ir atspoguļotas [Jek2010, Jek2011a, Jek2011b] ziņojumos. Šo metodi iespējams izmantot arī citās tehnoloģijās.

4.7. Kopsavilkums un secinājumi

Šajā nodaļā ir apskatīti aktuālie jautājumi DARSIR koncepcijas ieviešanai. Īpaša uzmanība ir pievērsta (*viii*) kritēriju nodrošināšanai (skat. 1.2. tabulā), kas paredzēti *datu konfidencialitātes, integritātes un pieejamības jautājumu nodrošināšanai*. Ir aprakstīti DARSIR koncepcijas praktiskie pētījumi ar atsaucēm uz pielikumiem un starptautiskiem zinātniskiem ziņojumiem.

Nodaļā paveiktais:

- ir veikta datu apmaiņas un glabāšanas formātu izvēle;
- ir aprakstīta atribūtu un noteikumu grupas ieviešana;
- ir pielietota metode tehnoloģiju izvēlei;
- ir apskatīti RFID tehnoloģijas tipiskie risinājumi;
- ir veikts drošības jautājumu apskats uz vienas tehnoloģijas pamata;
- ir parādīti paveiktie DARSIR koncepcijas praktiskie pētījumi.

Sasniegtie teorētiskie rezultāti ir šādi:

- ir piedāvātas trīs galvenās ietekmes kritēriju grupas: *fiziskie parametri un ārējā vide; izmaksas; drošība un konfidencialitāte*. Klasificēti 20 tehnoloģijas ietekmējošie kritēriji;
- ir pielietota svaru koeficientu metode, lai palīdzētu izvēlēties konkrētam risinājumam piemērotāko tehnoloģiju, izmantojot tehnoloģiju ietekmējošos kritērijus;
- ir identificēti 8 uzbrukuma veidi, kas ir aktuāli DARSIR koncepcijas pielietošanai uz RFID tehnoloģijas bāzes;
- ir detektēti divi papildu uzbrukuma veidi (*uzbrucējtags* un *falsificēts tags*), kuri ir aktuāli DARSIR koncepcijai, bet RFID tehnoloģijas tipisks izmantošanas scenārijs būtiski neietekmē risinājumu;
- ir dotas rekomendācijas aizsardzības mehānismu ieviešanai diviem DARSIR koncepcijas specifiskiem uzbrukuma veidiem;
- ir sniegtas desmit rekomendācijas DARSIR koncepcijas RFID tehnoloģiju informācijas drošības risku samazināšanai.

Sasniegtie praktiskie rezultāti ir šādi:

- ir paveikts pētījums, uz kura pamata praktiski ir izveidoti divi jauni formāti: *resursu fizikālā iezīmēšanas valoda* (ang. *Resource Physical Markup Language, RPML*) un *JavaScript objektu notācija resursam* (ang. *JavaScript Object Notation for Resource, JSONR*);
- ir nodrošināta astoņu galveno atribūtu grupas realizācija;
- ir izstrādāts rīks ar nosaukumu „*DARSIR simulation tool*”, kas ir vizuālās interaktīvās simulācijas pamatā;
- ir aprakstīti divi iespējamie DARSIR koncepcijas pielietojumi ar izveidotu programmatūru. Pirmajā gadījumā ir realizēts koncepcijas tipiskais uzdevums „*Viedā veļas mazgātava*”. Otrā gadījumā, lai pārbaudītu netipiska uzdevumu realizāciju, ir realizēts „*Uz daudzāģentu sistēmas balstītas ārstata pakalpojumu pārvaldības sistēmas*” uzdevums.
- ir parādīts, ka, izmantojot universālas metodes, iespējams izveidot funkcijas konkrētām tehnoloģijām, kas iepriekš nebija realizētas, bet vajadzīgas DARSIR koncepcijas iedarbināšanai.

Galvenie secinājumi ir šādi:

- *IKKATN* (saīsinājums no *Ietekmes Kritēriju Kombinācijas Aprēķins Tehnoloģiju Novērtējumam*) metodes iedarbināšanai ir aizpildītas divdesmit ietekmes kritēriju nozīmes. *IKKATN* pielietota ar vienvērtīgiem ietekmes kritērijiem, un secināts, ka vispiemērotākās ir 2D svītrkodu un RFID tehnoloģijas.
- RFID tehnoloģija ir piemērotākā no apskatītajām tehnoloģijām. Tas secināts pēc *IKKATN* metodes pielietošanas ar nosacījumu, ka ir svarīgi nodrošināt funkciju par resursu atribūtu un noteikumu izmaiņām risinājuma darba procesā.
- Viena no iespējām, lai nodrošinātu lietotājam intuitīvi un viegli veidot IT risinājumu, ir simulācijas rīku izmantošana. Piedāvāta pieeja IT risinājuma izveidošanai tieši no simulācijas (piemēram, izmantojot *vizuālu interaktīvu simulāciju*), kas dod iespēju atvieglot procesu IT risinājuma izveidošanai.
- Pēc netipiska uzdevuma "*Ārstata pakalpojumu pārvaldības sistēmas uzbūve*" risinājuma ir saņemti sekojoši rezultāti:
 - DARSIR koncepciju iespējams piemērot arī risinājumos, kuros nav izmantoti fiziskie objekti.
 - Izmantojot universālas metodes, iespējams veidot papildu funkcijas tehnoloģijām, kuras izmanto DARSIR koncepcijā.

- Jaunu risinājumu izveidošanas pētījumi var prasīt papildu DARSIR koncepcijas atribūtu un noteikumu izveidošanu.

Šīs nodaļas galvenie jaunie teorētiskie rezultāti ir pētījums par DARSIR koncepcijas drošības jautājumu nodrošināšanu un sagatavotais teorētiskais materiāls DARSIR koncepcijas praktiskai izveidošanai. Tas ietver sevī pamata atribūtu un noteikumu grupas, IKKATN metode utt.

Galvenie rezultāti un secinājumi

Promocijas darbā tika izvirzīts mērķis, balstoties uz vadības sistēmas analīzi un ierobežojumu un trūkumu identificēšanu tajās, izstrādāt objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepciju un pārbaudīt tās praktiskās realizācijas iespējas.

Izvirzītā mērķa sasniegšanai tika izpildīti šādi uzdevumi:

- Identificēti kritēriji *objektu individuālajās īpašībās sakņotām sistēmām (OIĪSS)*. Izpētītas vispazīstamākās datu vadības sistēmas koncepcijas un novērtēta to atbilstība izvirzītajiem OIĪSS kritērijiem. No apskatītajām datu vadības sistēmu koncepcijām ir apkopoti labās prakses principi un metodes, kas ir izturējušas laika pārbaudi. Apkopotie rezultāti ir izmantoti OIĪSS koncepcijas izstrādei.
- Identificētas tehnoloģijas, kuras ir iespējams izmantot OIĪSS. Pamatojoties uz OIĪSS definētajiem kritērijiem, ir izvirzīti tehnoloģiju salīdzināšanas parametri, kuri tika pielietoti piemērotākās tehnoloģijas izvēlei IS izveidošanai, balstoties uz izstrādāto koncepciju.
- Ņemot vērā izvirzītos OIĪSS kritērijus, tiek piedāvāta objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcija. Aprakstīti koncepcijas galvenie elementi un elementu mijiedarbības scenāriji un klasificētas objektu īpašības, definētas atribūtu un noteikumu grupas. Tika veikta koncepcijas pārbaude iepriekš izvirzītām OIĪSS kritērijiem.
- Veikts detalizēts atribūtu un noteikumu grupas pētījums, rezultāts tiek izmantots koncepcijas praktiskās realizācijas laikā. Ir izvirzīti būtiskākie tehnoloģijas izvēles kritēriji, kurus nepieciešams ņemt vērā IS risinājuma tehnoloģijas izvēles gadījumā. Ņemot vērā kritērijus, ir piedāvāta tehnoloģiju izvēles metode, kas ļauj izvēlēties koncepcijai piemērotākās tehnoloģijas atkarībā no IS būtiskiem darba apstākļiem.
- Veikts izstrādātās koncepcijas datu privātuma un drošības jautājumu pētījums. Identificēti tipiskie uzbrukuma veidi, detektēti koncepcijas specifiskie uzbrukuma veidi. Pamatoties uz pētījuma rezultātiem dotas rekomendācijas izstrādās koncepcijas informācijas drošības risku samazināšanai.
- Veikti izstrādātās koncepcijas praktiskie pētījumi ar nolūku realizēt piedāvātas koncepcijas programmatūras elementu praktiskas realizācijas iespējas pārbaudi.

Pētījumu rezultātā tika identificēti šādi kritēriji objektu individuālajās īpašībās sakņotām decentralizētām datu vadības sistēmām:

- Skaitļošanas resursu pārvaldības sadale un decentralizācija;
- Standartu izmantošana;
- Fizisko objektu daudzveidības atbalstīšana;
- Lokālas datu piekļuves nodrošināšana;
- Autonoma darbība;
- Mazu risinājumu ieviešanas ekonomiskais izdevīgums;
- Pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale;
- Datu konfidencialitātes, integritātes un pieejamības nodrošināšana.

Izstrādājot datu vadības koncepciju, kas apmierina minētos kritērijus, ir sasniegti šādi **jauni teorētiskie rezultāti**:

- Ir identificēti 8 kritēriji, kuriem jāatbilst *objektu individuālajās īpašībās sakņotai sistēmai (OIĪSS)*;
- Ir pētītas 9 monolītas un 5 decentralizētas datu vadības sistēmas koncepcijas un novērtēta to atbilstība OIĪSS izvirzītajiem kritērijiem. Ir noskaidrots, ka neviena no apskatītajām datu vadības sistēmām neatbilst visiem izvirzītajiem OIĪSS kritērijiem, radot nepieciešamību pēc objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcijas izstrādes.
- Ir identificēti datu vadības sistēmu trūkumi, ierobežojumi un priekšrocības no datu glabāšanas un pārvaldības viedokļa, kas tiek ņemts vērā jaunās koncepcijas uzbūvē.
- Ir identificētas 16 tehnoloģijas un tehnoloģiju modifikācijas, kas parasti tiek izmantotas OIĪSS risinājumos, un piedāvātas trīs tehnoloģiju grupas, kas klasificē tehnoloģijas pēc OIĪSS izmantošanas nolūkiem. Ir izvirzīti 8 tehnoloģiju salīdzinājuma parametri, kuri ietekmē OIĪSS risinājuma tehnoloģijas izvēli;
- Ir izstrādāta objektu individuālajās īpašībās sakņotas decentralizētas datu vadības sistēmas koncepcija DARSIR, kas atbilst izvirzītajiem OIĪSS kritērijiem. Ir klasificēti piedāvātās koncepcijas 3 objektu pamattipi: *pasīvie resursi*, *aktīvie statistiskie resursi* un *aktīvie dinamiskie resursi*. Ir piedāvāti trīs DARSIR koncepcijas resursu mijiedarbības scenāriji un aprakstīti *vienas ADR (skeneris) mijiedarbības*, *paralēlas ADR mijiedarbības* un *secīgas ADR mijiedarbības scenāriji*. Ir izdalītas 10 atribūtu un noteikumu grupas, grupēšana ir organizēta pēc izmantošanas sfērām. Rezultāts ir izmantots DARSIR koncepcijas praktisko realizāciju īstenošanai. Piedāvātajai koncepcijai ir šādas priekšrocības: lēmumu pieņemšana ir realizēta uz vietas fiziskajā objektā; tiek atbalstīta fizisko objektu daudzveidība un unikalitāte; ir paaugstināts datu

privātuma līmenis; ir izdarīta atteikšanās no centralizētas datu glabāšanas un pārvaldības infrastruktūras nodrošinājuma.

- Ir piedāvātas trīs galvenās ietekmes kritēriju grupas un klasificēti 20 tehnoloģiju ietekmējošie kritēriji, kuru pamatā ir izveidota *IKKATN* (saīsinājums no *Ietekmes Kritēriju Kombinācijas Aprēķins Tehnoloģiju Novērtējumam*) metode. *IKKATN* metode tika izmantota, lai izvēlētos konkrētam risinājumam piemērotāko tehnoloģiju. Ir veikta *IKKATN* metodes pielietošana 12 tehnoloģijām no objektu identificēšanas tehnoloģiju grupas. Ar ietekmes koeficientu vienvērtīgas nozīmes aprēķinu ir identificēts, ka *DARSIR* koncepcijai piemērotas ir *RFID* un *svītrkodu* tehnoloģijas.
- Ir konstatēts, ka neviena no apskatītajām tehnoloģijām neizpilda kritēriju par *datu konfidencialitātes, integritātes un pieejamības nodrošināšanu* – šo kritēriju ir nepieciešams izpildīt programmatūras līmenī. Ir veikti papildu pētījumi, kuru rezultāts ir:
 - Noteikti 8 uzbrukuma veidi, kas ir aktuāli *DARSIR* koncepcijas pielietošanai uz *RFID* tehnoloģijas bāzes, un divi papildu uzbrukuma veidi (*uzbrucējtags* un *falsificēts tags*), kuri ir aktuāli *DARSIR* koncepcijai, bet *RFID* tehnoloģijas tipisks izmantošanas scenārijs būtiski neietekmē risinājumu.
 - Piedāvātas 10 rekomendācijas *DARSIR* koncepcijas ar *RFID* tehnoloģiju informācijas drošības risku samazināšanai.
- Ir piedāvāti divi jauni formāti – *RPML* (balstīts uz *XML* valodu) un *JSONR* (balstīts uz *JSON* valodu), kas ir izmantoti *DARSIR* koncepcijas praktiskajā realizācijā.

Iegūtie teorētiskie rezultāti praktiski ir realizēti programmatūras kompleksā, un darbā ir aprakstīti divi eksperimenti:

- Koncepcijas tipiskais uzdevums – objektu individuālajās īpašībās sakņotais *DDVS* risinājums „*Viedā veļas mazgāšanas mašīna*”. Ir dots standarta ceļvedis jaunu risinājumu izveidošanai;
- Koncepcijas netipiskais uzdevums – tiek realizēts programmatūras risinājums „*Uz daudzāģentu sistēmu principiem balstītas ārstata pakalpojumu pārvaldības sistēmas*”. Ir izveidotas divas jaunas atribūtu grupas, kas nepieciešamas pētījumu praktiskai realizēšanai. Tika dots ceļvedis programmatūras risinājumu izveidošanas gadījumā, kuriem izstrādātā koncepcija sākotnēji nebija paredzēta, bet ir veiksmīgi pielietojama, paplašinot ar to risināmo uzdevumu klāstu.

Programmatūras kompleksa izstrāde un eksperimentālās pārbaudes **rezultāti ļauj izdarīt šādus secinājumus:**

- *Objektu individuālajās īpašības sakņotām sistēmām (OIĪSS) ir identificēti šādi kritēriji: skaitļošanas resursu pārvaldības sadale un decentralizācija; standartu izmantošana; fizisko objektu daudzveidības atbalstīšana; lokālas datu piekļuves nodrošināšana; autonoma darbība; nelielu risinājumu ieviešanas ekonomiskais izdevīgums; pakāpeniskas ieviešanas iespēja un ieviešanas izmaksu sadale; datu konfidencialitātes, integritātes un pieejamības nodrošināšana.*
- Klasiskas datu vadības sistēmas atbilst izvirzītajiem OIĪSS kritērijiem tikai daļēji. Radās nepieciešamība pēc objektu individuālajās īpašībās sakņotas decentralizētās datu vadības sistēmas koncepcijas izstrādes, kas apmierinātu izvirzītos OIĪSS kritērijus.
- Piedāvātā DARSIR koncepcija pilnvērtīgi atbilst izvirzītajiem OIĪSS kritērijiem.
- Pēc *IKKATN* (saīsinājums no *Ietekmes Kritēriju Kombinācijas Aprēķins Tehnoloģiju Novērtējumam*) metodes pielietošanas ir identificēts, ka no OIĪSS identificētām tehnoloģijām un tehnoloģiju modifikācijām, vispiemērotākās ir:
 - *RFID* tehnoloģija, ja resursa iekšējo informācijas izmaiņas jānodrošina jebkura IS laikā posmā;
 - *2D svītrkodu* tehnoloģija, ja resursā informācija ir statiska, un izmaiņas veikti reti IS dzīves ciklā.
- *DARSIR* koncepcijas pamata priekšrocības: *lēmumu pieņemšana ir realizēta uz vietas; atbalstīta fizisko objektu daudzveidība un unikalitāte; paaugstinās datu privātuma līmenis; atteikšanās no centralizētas datu glabāšanas un pārvaldības infrastruktūras nodrošinājuma.*

Turpmāko pētījumu iespējamie virzieni:

1. Dažādu veidu risinājumu izveidošana, balstoties uz promocijas darbā piedāvāto koncepciju, kā arī, balstoties uz jaunajiem risinājumiem, piedāvāt koncepcijai jaunas atribūtu un noteikumu grupas un iebūvētas funkcijas.
2. Izpētīt praktisko risinājumu izveidošanas ceļus, pamatojoties uz piedāvāto koncepciju.

Literatūra

- [Aut2002] Auto-ID Center MIT, The Physical Markup Language. Auto-ID Center MIT, July 2002 / Internets. – <http://web.mit.edu/mecheng/pml>
- [Adl2001] Adler D.W., "IBM DB2 Spatial Extender - Spatial Data Within the RDBMS," Proceedings of 27th International Conference on Very Large Data Bases, Italy, Rome, 2001.
- [Alt1980] Alter S., "Decision Support Systems: Current Practice and Continuing Challenges," Reading, Mas.: Addison-Wesley, 1980.
- [And1998] Anderson T., Breitbart Y., Korth H.F. and Wool A., "Replicaton, Consistency, and Practicality: Are These Matually Exclusive?" Proceedings of 1998 ACM SIGMOD International Conference on Management of Data, Seattle, Washington, 1998.
- [Ant2005] Antoniou B. and Tsoulos L., "The potential of XML encoding in geomatics converting raster images to XML and SVG," Cartography Laboratory, School of Rural and Surveying Engineering, National Technical University of Athens, Zografos Campus, Greece, 2005.
- [AOS2011] Autonomous Decision-Making Software, "Jack Documentation" / Internets. – <http://www.agent-software.com.au/products/jack/>
- [Atk1987] Atkinson M. and Buneman O.P., "Types and Persistence in Database Programming Languages," ACM Computer Survey, 1987.
- [Atk1990] Atkinson M., "The Object-Oriented Database System Manifesto," Proceedings of 1st International Conference on Deductive and Object-Oriented Databases," USA, New York, Elsevier Science, 1990.
- [Ayo2006] Ayoade J. Security Implications in RFID and Authentication Processing Framework. Computers & Security, 2006, p. 207 – 212.
- [Bad2007] Badawy O.M. and Hasan M. "Decision tree approach to estimate user location in WLAN based on location fingerprinting," In: Proceedings of 24th National Radio Science Conference, Ain Shams Univ., Egypt, 2007, pp. 1-10.
- [Bah2000] Bahl P. and Padmanabhan V. N., "RADAR: An In-Building RF-based User Location and Tracking System" // INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 2, 2000, p. 775-784.
- [Bar1996] Barry D.K., "The Object-Oriented Database Handbook:How to Select, Implement, And Use Object-Oriented Databases," USA, New York, John Wiley & Sons, 1996.

- [Bau2005] M. Baudin and A. Rao, "RFID applications in manufacturing," USA, Palo Alto: MMTI, 2005, pp. 1-12.
- [Bay1980] Bayer R., Heller M., and Reiser A., "Parallelism and Recovery in Database Systems," ACM TODS, 1980.
- [Bay2010] Bayer T., "Protocol Buffers, Etch, Hadoop and Thrift Comparison", predic8 GmbH, Germany, Bonn, 2010.
- [Bel1987] P. C. Bell and R.M. O'Keefe, "Visual Interactive Simulation - History, Recent Developments, and Major Issues," *Simulation*, 49(3), 1987, pp. 109-116.
- [Bel1999] P. C. Bella, C. K. Andersona, D. S. Staplesb and M. Elderc, "Decision-makers' perceptions of the value and impact of visual interactive modelling," *Journal Omega*, Volume 27, Issue 2, Apr. 1999, pp. 155-165.
- [Bem2005] Belemeit B., Lorenz M., Schumacher J., and Herzog O., "Risk Management for Transportation of Sensitive Goods," *Proceedings of the 10th International Symposium on Logistics*, England: Nottingham, 2005, p. 492-498.
- [Ben1981] Bennet J.L., "Building Decision Support Systems," Reading, Mass.: Addison-Wesley, 1981.
- [Ben2004] Beynon-Davies P. "Database Systems," 3rd Edition, Palgrave, Houndmills, Basingstoke, 2004, pp. 617.
- [Ber1980] Berenson H. and Goodman N., "Timestamp-Based Algorithms for Concurrency Control in Distributed Database Systems," *Proceedings of 6th International Conference On Very Large Data Bases*, Canada, Montreal, 1980.
- [Ber1992] Berners-Lee T., "The Hypertext Transfer Protocol," World Wide Web Consortium, 1992.
- [Ber1993] Berners-Lee T. and Connolly D., "The Hypertext Markup Language," World Wide Web Consortium, 1993.
- [Ber1995] Berenson H., "A Critique of ANSI SQL Isolation Levels," *Proceedings of 1995 ACM SIGMOD International Conference on Management of Data*, USA, California, San Jose, 1995.
- [Ber1996] Bernstein P.A., "Middleware: A Model for Distributed System Services," CACM, 1996.
- [Ber1999] Berners-Lee T. and Fishetti M., "Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor," USA, California, San Francisco, Garper San Francisco, 1999.

- [Bis1990] J. L. Bishop and O. Balci, "General Purpose Visual Simulation System: a Functional Description," Winter Simulation Conference, New Orleans, Louisiana, 1990, pp. 504-512.
- [Bjo1973] Bjork L.A., "Recovery Scenario for DB/DC System," Proceedings of ACM National Conference, USA: Atlanta, 1973.
- [Bla1979] Blasgen M.W., Gray J.N., Mitoma M. and Price T.G., "The Convoy Prehomenon," ACM Operating Systems Review, 1979.
- [Blo2002] Blott S. and Korth H.F., "An Almost-Serial Protocol for Transaction Execution in Main-Memory Database Systems," Proceedings of 28th International Conference On Very Large Data Bases, Hong Kong, 2002.
- [Bød2006] Bødker K., Jørgensen B., Pedersen E., Vøge M. - „Community based WiFi Positioning”, Department of Computer Science, Aalborg University, 2006
- [Bon1981] Bonczek R.H., Holsapple C.W. and Whinston A., "Foundations of Decision Support Systems," Orlando, Academic Press, 1981.
- [Boo1991] Booch G. "Object-Oriented Design with Applications" // The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1991.
- [Bro1989] Brogie M., "Future Intelligent Information Systems: AI and Database technologies Working Together," Readings in Artificial Intelligence and Databases, USA: San Francisco, 1989.
- [Bro2001a] Brock D. L., The Electronic Product Code (EPC) A Naming Scheme for Physical Objects, Massachusetts Institute of Technology, USA: Cambridge, 2001.
- [Bro2001b] Brock D. L., The Physical Markup Language, Massachusetts Institute of Technology, USA: Cambridge, 2001.
- [Bro2001c] Brock D. L., Integrating the Electronic Product Code (EPC) and the Global Trade Item Number (GTIN), Massachusetts Institute of Technology, USA: Cambridge, 2001.
- [Bro2001d] Brock D. L., Milne T. P., Kang Y. Y. and Lewis B., The Physical Markup Language Core Components: Time and Place, Massachusetts Institute of Technology, USA: Cambridge, 2001.
- [Bro2002] Brodie M.L., "Data Management Challenges in Very Large Enterprises," Proceedings of 28th International Conference on Very Large Data Bases, Hong Kong, 2002.

- [Bro2007] Brown E. D., "RFID implementation," McGraw-Hill,USA: New York, 2007, p. 466
- [Bru2005] Brunato M. and Battiti R. "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks and ISDN Systems*, 47(6), Elsevier, 2005, pp. 825-845.
- [Bsh2009] Bshara M., Vershraegen R., Van Biesen L. – “Cell-ID positioning in WiMAX networks Analysis of the Clearwire network in Belgium” // *ICT-MobileSummit 2009 Conference Proceedings*, 2009
- [Buf2008] Buford J.F., Yu H., Lua E.K., "P2P Networking and Applications," ISBN 30-12374-214-5, Morgan Kaufmann, 2008, p. 408.
- [Bur2005] Burnette E., "Eclipse IDE Pocket Guide ," O'Reilly Media, 2005, p. 128.
- [Cag2000] Cagle K., Gibbons D., Hunter D., Ozu N., Pinnock J. and Spencer P., "Beginning XML," WROX, 2000.
- [Car1999] Carey M.J., "O-O, What Have They Done to DB2?" *Proceedings of 25th International Conference on Very Large Data Bases*, Scotland, Edinburgh, 1999.
- [Cel1995] Celko J., "SQL for Smarties: Advanced SQL Programming," USA: San Francisco, Morgan Kaufmann, 1995.
- [Cha1974] Chamberlin D.D. and Boyce R.F., "SEQUEL: A Structured English Query Language," *Proceeding of ACM SIGMOD Workshop on Data Description, Access, and Control*, Ann Arbor, Mich, 1974.
- [Cha1977] Chamberlin D.D. et al., "SEQUEL/2: A Unified Approach to Data Definition, Manipulation, and Control," *IBM J. R&D*, 1977.
- [Cha2005] Chapman S., "Fundamentals of Production Planning and Control," Prentice-Hall, USA: Englewood Cliffs, 2005.
- [Cha2008] Marie Chan, Daniel Estève, Christophe Escriba, Eric Campo, "A review of smart homes—Present state and future challenges," *Computer Methods and Programs in Biomedicine*, Volume 91, Issue 1, 2008, p. 55-81.
- [Cha2009] Marie Chan, Eric Campo, Daniel Estève, Jean-Yves Fourniols, "Smart homes — Current features and future perspectives," *Maturitas*, Volume 64, Issue 2, 2009, p. 90-97.
- [Che2006] Cheung K. C., Intille S. S., and Larson K. – „An Inexpensive Bluetooth-Based Indoor Positioning Hack”, 2006.

- [Chi2009] Ching-Sheng W., Chun-Hong H., Yong-Si C., and Li-Jie Z. "An implementation of positioning system in indoor environment based on active RFID," *Pervasive Computing (JCPC)*, 2009 Joint Conferences on, Dec 2009, pp. 71-76.
- [Cho2007] Chow H.K.H., Choya K.L., and Lee W.B. A dynamic logistics process knowledge-based system – An RFID multi-agent approach. *Knowledge-Based Systems* 20, 2007, pp. 357–372.
- [Cod1969] Codd E.F., "Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks," IBM Research Report RJ599, 1969.
- [Cod1970] Codd E.F., "A Relational Model of Data for Large Shared Data Banks", *Communications of the ACM* 13 (6), USA, 1970, p. 377–387.
- [Cod1980] Codd E.F., "Data Models in Databases Management," *Proceedings of Workshop on Data Abstraction, Database and Conceptual Modelling*, Pingree Park, Colo, 1980.
- [Cod1990] Codd E.F., "The Relation Model for Database Management Version 2," Reading, Mass.: Addison-Wesley, 1990.
- [Con2009] Connolly T.M. and Begg C.E., "Database Systems: A Practical Approach to Design, Implementation and Management (5th Edition)", Addison Wesley, 2009, p. 1400.
- [Cro2006] Crockford D., "The application/json Media Type for JavaScript Object Notation (JSON)," *JSON.org*, 2006, p. 10.
- [Dar2010] Dargie W. and Poellabauer C., "Fundamentals of wireless sensor networks: theory and practice", John Wiley & Sons, ISBN 978-0-470-99765-9, 2010.
- [Dat1971a] Date C.J. and Hopewell P., "Storage Structure and Physical Data Independence," *Proceedings of ACM SIGFIDET Workshop on Data Definition, Access, and Control*, USA: San Diego, 1971.
- [Dat1971b] Date C.J. and Hopewell P., "File Definition and Logical Data Independence," *Proceedings of ACM SIGFIDET Workshop on Data Definition, Access, and Control*, USA: San Diego, 1971.
- [Dat1987] Date C.J., "Twelve rules for a distributed databases," *ComputerWorld*, 1987, p. 75-81.
- [Dat1990] Date C.J., "What is a Distributed Database System?", Reading, Mass.: Addison-Wesley, 1990.

- [Dat1993] Date C.J. and Colin J.W., "A Guide to DB2 (4th edition)," Reading, Mass.:Addison-Wesley,1993.
- [Dat1997] Date C.J. and Darwen H.,"A Guide to the SQL Standard (4th edition)," Reading, Mass.: Addison-Wesley, 1997.
- [Dat1998] Date C.J., "Why the Object Model' Is Not a Data Model," Reading, Mas.: Addison-Wesley, 1998.
- [Dat2000a] Date C.J., Darwen H., "Foundation for Object/Relational Databases: The Third Manifesto (2th edition)," Reading, Mass.: Addison-Wesley, 2000.
- [Dat2000b] Date C.J., "Great News, The Relational Model Is Very Much Alive!", / Internets. – <http://www.dbdebunk.com>
- [Dat2001a] Date C.J., "There's Only One Relational Model!" / Internets. – <http://www.dbdebunk.com>
- [Dat2001b] Date C.J., "The Database Relational Model: A Retrospective review and Analysis," Reading, Mass.: Addison-Wesley, 2001.
- [Dat2003] Date C. J., "An Introduction to Database Systems (8th Edition)", Addison Wesley, USA, 2003.
- [Dat2004] Date C. J., "An Introduction to Database Systems (8th Edition)", Addison Wesley, USA, 2004, p. 1327.
- [Del1988] Delvin B.A. and Murphy P.T., "An Architecture for a Business and Information System," IBM Systems, 1988.
- [Den1979] Denning D.E. and Denning P.J., "Data Security," ACM Comp. Surv., 1979.
- [Den2010] "QR code", Denso Wave Incorporated, 2010 / Internets. – <http://www.denso-wave.com/qrcode/index-e.html>
- [Dom1996] Dommety G., and Peter M., "Potential networking applications of global positioning systems (GPS)," OSU Technical report, Networking and Internet Architecture (cs.NI), 2007.
- [Dul2009] Dulya B., "GSM-Positioning, Outdoorpostioning: Technologies, Characteristics and Limitations", 2009.
- [Eck2003] Eckel B. "Thinking in Java. Third Edition." // President, MindView, Inc., U.S., 2003.
- [EPC2006] Eiropadomes prezidentūras secinājumu 30. punkts, 2006. gada 14.–15. decembris.

- [EPC2010a] EPCglobal Standards Overview. Prepared by GS1 EPCglobal: GS1 EPCglobal. / Internets. – <http://www.epcglobalinc.org/standards/>
- [EPC2010b] EPCIS Standard v. 1.0. Prepared by GS1 EPCglobal: GS1 EPCglobal. / Internets. – http://www.epcglobalinc.org/standards/epcis/epcis_1_0-standard-20070412.pdf
- [EPC2010c] ALE Standard v. 1.0. Prepared by GS1 EPCglobal: GS1 EPCglobal. / Internets. – http://www.epcglobalinc.org/standards/ale/ale_1_0-standard-20050915.pdf
- [EPC2010d] ONS Standard v. 1.0. Prepared by GS1 EPCglobal: GS1 EPCglobal. / Internets. – http://www.epcglobalinc.org/standards/ons/ons_1_0-standard-20051004.pdf
- [Era2002] Erasala N. and Yen D. C., "Bluetooth technology: a strategic analysis of its role in global 3G wireless communication era," *Computer Standards & Interfaces*, Volume 24, Issue 3, 2002, p. 193-206.
- [Far2011] Farighetti R., *The World Almanac and Book of Facts 2011*, World Almanac Books, 2011.
- [Feb1999] Ferber J., "MultiAgent System: An Introduction to Distributed Artificial Intelligence," Harlow: Addison Wesley Longman, ISBN 0-201-36048-9, 1999.
- [Fel2004] Feldhofer M. A Proposal for Authentication Protocol in a Security Layer for RFID Smart Tags. In: *The 12th IEEE Mediterranean Electrotechnical Conference (MELECON)*, Dubrovnik, 2004.
- [Fen2002] Feng S., and Law C.L., "Assisted GPS and its impact on navigation and intelligent transportation systems," *IEEE 5th International Conference on Intelligent Transportation Systems*, 2002, p. 923-931.
- [Fer2010] Geraldo Ferrer, Nicholas Dew, Uday Apte, When is RFID right for your service? Original Research Article *International Journal of Production Economics*, Volume 124, Issue 2, April 2010, Pages 414-425
- [Fis1997] Fishman N, "SQL du Jour," *DBP&D*, 1997.
- [Flo2005] C. Floerkemeier and M. Lampe, "RFID middleware design - addressing application requirements and RFID constraints," *Smart Objects & Ambient Intelligence*, France, Grenoble, Oct. 2005.
- [Fre2011a] How it Works / Internets. – <http://www.freelancer.com/info/how-it-works.php>
- [Fre2011b] Help / Internets. – <http://www.free-lance.ru/help/>
- [Fre2011c] API Overview / Internets. – http://developer.freelancer.com/API_Overview
- [Gab2005] Gabriel C., "Research Report (Part of the BluePrint Wi-Fi substruction package)," *Blueprint*, 2005, p. 33.

- [Gen2009] International Association for Textile Care Labeling. / Internet – http://www.sartex.ch/ginetex_web/index.html
- [Gha2008] A. Ghayal, "SmartRF: A Flexible and Light-weight RFID Middleware," A Thesis Report Submitted in Partial Fulfillment of the Requirement for the Degree of Master of Technology, Indian Institute of Technology, Kanpur, June 2008.
- [Gol1994] Goldring R.A., "Discussion of Relational Database Replication Technology," InfoDB, 1994.
- [Goo2011] Google, Spade2 Smart Python Agent Development Environment / Internets. – <http://code.google.com/p/spade2/>
- [Goo2012a] "The Google Geocoding API", Google, Google Maps API Web Services, 2012 / Internets. – <http://code.google.com/intl/en/apis/maps/documentation/geocoding/#KML>
- [Goo2012b] "Using JSON in the Google Data Protocol", Google, Google Data API's , 2012 / Internets. – <http://code.google.com/intl/ru/apis/gdata/docs/json.html>
- [Goo2012c] "Protocol Buffers", Google, Developer Guide, 2012 / Internets. – <http://code.google.com/intl/ru/apis/protocolbuffers/docs/overview.html>
- [Gra1993] Gray J. and Reuters A., "Transaction Processing: Concepts and Techniques," USA: San Mateo, Morgan Kaufmann, 1993.
- [Gra1996] Gray J., Helland P., O'Neil P., Shasha D., "The Dangers of Replication and a Solution," Proceedings of 1996 ACM SIGMOD International Conference on Management of Data, Canada, Montreal, 1996.
- [Gre1996] Greenfield L., "Don't let the Data Warehousing Gotchas Getcha," Datamation, 1996.
- [Gre2007] Greenwald R., Stackowiak R. and Stern J., " Oracle Essentials: Oracle Database 11g . Fourth Edition edition ," O'Reilly Media, 2007, p. 416.
- [GS12011] Uniform Code Council (UCC), Uniform Product Code (U.P.C.) <http://www.gs1us.org>
- [Gur2011] Employer FAQs / Internets. – http://www.guru.com/help/employer/emphelp_Left.htm
- [Haa1989] Haas L.M., Freytag J.C., Lohman G.M. and Pirahesh H., "Extensible Query Processing in Starburst," Proceedings ACM SIGMOD International Conference on Management of Data, USA, Portland, 1989.

- [Haa2006] Haataja K.M.J., "Security in Bluetooth, WLAN and IrDA: a comparison," Kuopio Universit ate, Somija, 2006, p. 14.
- [Hal2003] Hallberg J., Nilsson M., Synnes K. – "Positioning with Bluetooth", 2003.
- [Har1960] Harris D. B., "Radio transmission systems with modulatable passive responder," U.S. Patent 2 927 321, 1960.
- [Har1983] Harder T. and Reuters A., "Principles of Transaction-Oriented Database Recovery," ACM Comp. Surv., 1983.
- [Hed2007] Hedgepeth W. O., "RFID METRICS Decision Making Tools for Today's Supply Chains," USA: CRC Press, ISBN: 978-0-8493-7979-6, 2007, p. 131
- [Hei2005] Heinrich C. RFID and Beyond: Growing Your Business Through Real World Awareness – Chichester, England: Wiley, 2005.
- [Her2001] Hernandez L., "Back to basics: LAN technologies," Agilent Technologies, ASV, 2001.
- [Hes2010] Hesmondhalgh D. and Baker S., "'A very complicated version of freedom': Conditions and experiences of creative labour in three cultural industries," Poetics 38, Australia: Queensland, 2010, pp. 4-20.
- [Hex2010] Hexamite, HX5 Ultrasonic Positioning System / Internets: <http://www.hexamite.com/hx5.htm>, 2010
- [Hig2009] Highsmith J., Agile Project Management: Creating Innovative Products, 2nd Edition, Addison-Wesley Professional, ISBN-13: 978-0321658395, 2009, p. 432.
- [Hof2002] Hoffmann E. , Walwe U., "Wandel der Erwerbsformen: Was steckt hinter den Veranderungen?", IABKompodium Arbeitsmarkt - und Berufsforschung, Germany: Nurnberg, 2002, pp. 135-144.
- [Hot2008] Hotek M., "Microsoft® SQL Server® 2008 Step by Step (Step by Step (Microsoft)) ," Microsoft Press; Pap/Cdr edition, 2008, p. 544.
- [Hun2007] Hunt V. D., A. Puglia, M. Puglia, "A guide to radio frequency identification", ISBN: 978-0-470-10764-5, USA, NJ: Wiley-Interscience, 2007, p. 214.
- [Ibr2005] Ibrahim I. K. and Taniar D., "Mobile Multimedia for Business Data Communication," Monash University, Australia, 2005.
- [Inm1988] Inmon W.H., "Data Architecture: The Information Paradigm," Wellesley, Mas.: QED Information Sciences, 1988
- [Inm1992] Inmon W.H., "Building The Data Warehouse," New York, John Wiley & Sons, 1992.

- [Inm1993] Inmon W.H., "Building the Data Warehouse," New York, John Wiley & Sons, 1993.
- [Inm1994] Inmon W.H. and Hackathorn R.D., "Using the Data Warehouse", New York, John Wiley & Sons, 1994.
- [Ish2003] Ishikawa T., Yumoto Y., Kurata M., Endo M., Kinoshita S., Hoshino F., Yagi S., Nomachi M. Applying Auto-ID to the Japanese Publication Business. 2003
- [ISO1990a] International Organization for Standardization (ISO), "Information Processing Systems, Open Systems Interconnection, Remote Data Access Part1: Generic Model, Service, and Protocol," Document ISO DIS 9579-1, 1990.
- [ISO1990b] International Organization for Standardization (ISO), "Information Processing Systems, Open Systems Interconnection, Remote Data Access Part1: SQL Specialization," Document ISO DIS 9579-2, 1990.
- [ISO1999] "International Organization for Standardization (ISO): Information technology - Database Languages - SQL, Document ISO/IEC 9075:1999.
- [Itg2006] Information Technology Governance Institute. Information Security Governance: Guidance for Boards of Directors and Executive Management. – 2006
- [Jah2009] Jahng H. P., Young-Kwan J. "Location estimation of mobile systems using passive RFID tags in a indoor environment , "International Journal of Modern Physics C, Vol. 20, No. 4, 2009, pp. 619–632.
- [Jef2007] Jeffrey G. A., Arunabha G. and Rias M., "Fundamentals of WiMAX Understanding Broadband Wireless Networking," Prentice Hall, ASV, 2007.
- [Jek2010] Jekabsons G., Zuravlyov V. Refining Wi-Fi based indoor positioning // Proceedings of the international scientific conference Applied Information and Communication Technologies, Latvia, Jelgava, 22.-23. April, 2010. - pp 87-95.
- [Jek2011a] Jekabsons G., Kairish V., Zuravlyov V. Analysys of Wi-Fi based indoor positioning accuracy // Proceedings of the 6th internal conference on electrical and control technologies, ISSN 1822-5934, Lithuania, Kaunas, 5.-6. May, 2011. – pp. 45-50.
- [Jek2011b] Jekabsons G., Karish V., Zuravlyov V. An analysis of WI-FI Based Indoor Positioning Accuracy // Proceedings of the international conference on Applied Computer Systems, ISSN 1407-7493, Latvia, Riga, 2011., pp. 138-145.
- [Jen1998] Jennings N., Sycara K. and Wooldridge M., "A Roadmap of Agent Research and Development, Autonomous Agents and Multi-Agent Systems", 1998, p. 275–306.

- [Jun2004] C. Junwei, "Lightweight RFID framework," IBM, 2004.
- [Kan2011] Kang J. S., Lee H. Y. and Tsai J., "An analysis of interdependencies in mobile communications technology: The case of WiMAX and the development of a market assessment model," *Technology in Society*, Volume 33, Issues 3-4, Taiwan, 2011, p. 284-293.
- [Kap1996] Kaplan E. D., "Understanding GPS: Principles and Applications," Artech House, Boston, 1996.
- [Kap2006] Kaplan E. D. and Hegarty C. J., "Understanding GPS Principles and Applications," Artech House, ASV, 2006, p. 703.
- [Kee1978] Keen P.G.W. and Morton M.S.S., "Decision Support systems: An Organizational Perspective," Reading, Mas.:Addison-Wesley, 1978.
- [Keg1984] Kernighan B. W., Pike B., "The Unix Programming Environment," Prentice Hall, Inc., 1984, p. 357.
- [Ken2000] Beck K. and Fowler M., "Planning Extreme Programming," Addison-Wesley Professional, ISBN: 978-0201710915, 2000, p. 160.
- [Ken2010] Beck K., "Manifesto for Agile Software Development," Agile Alliance, 2010.
- [Ker2011] Kerschberger M., "Near Field Communication, A survey of safety and security measures," Austrija, 2011.
- [Kle2011] Klein L., "WordPress Power Guide - Using WordPress to Blog Your Way to Success - Blogging Guide," Kindle, 2011, p.62.
- [Kno2004] Knospe H., Pohl H. RFID Security, Information Security Technical Report, Volume 9, Issue 4, December 2004, pp. 39-50.
- [Knu2001] Knutson C. D. and Diviney G., "Infrared Data Communications with IrDA," Counterpoint Systems Foundry, Corvallis, ASV, 2001.
- [Koe1975] Koelle A.R., Depp, S.W. and Freyman, R.W., "Short-range radio-telemetry for electronic identification, using modulated RF backscatter," *Proceedings of the IEEE*, Vol. 63 Issue 8, 1975, pp. 1260-1261.
- [KOM2004a] "ESAO Informācijas tehnoloģijas perspektīva, 2004. gads".
- [KOM2004b] "Komisijas paziņojums padomei, Eiropas raplamentam, Eiropas ekonimikas un sociālo lietu komitejai un reģionu komitejai. Izaicinājumi Eiropas informācijas sabirdrībai pēc 2005. gada," KOM(2004) 757, Brisele.

- [KOM2005] Komisija iesniegusi Padomei priekšlikumu attiecībā uz Padomes Pamatlēmumu par personas datu, ko apstrādā, sadarbojoties policijas un tiesu iestādēm krimināllietās, aizsardzību (COM (2005) 0475, galīgā redakcija).
- [KOM2007] "Komisijas paziņojums padomei, Eiropas raplamentam, Eiropas ekonomikas un sociālo lietu komitejai un reģionu komitejai. Radiofrekvenču identifikācija (RFID) Eiropā:ceļā uz politikas īstenošanas pasākumiem," SEC(2007) 312, Brisele, 2007.
- [Kor1990] Korth H.F., Levy E. and Silberschatz A., "A Formal Approach to Recovery by Compensating Transactions," Proceedings of 16th International Conference on Very Large Data Bases, Australia, Brisbane, 1990.
- [Kry2008] Kryukov D. Concept of information security system evaluation model. Scientific proceedings of Riga Technical University. – Riga: RTU, 2008.
- [Kun1981] Kung H.T. and Robinson J.T., "On Optimistic Methods for Concurrency Control," ACM TODS, 1981.
- [Lab2007] Labiod H., Afifi H., De Santis C. – "Wi-FiTM, BluetoothTM, ZigBeeTM and WiMaxTM" // Springer, 2007. – 316. lpp.
- [Lah2005] Lahiri S. RFID sourcebook. – London, England: Upper Saddle River, 2005.
- [Lai2011] Joseph H.K. Lai, "Comparative evaluation of facility management services for housing estates", Habitat International, Volume 35, Issue 2, 2011, p. 391-397.
- [Lar1999] Larsen E., Morecroft J., and Thomsen J., "Complex Behaviour in a Production-Distribution Model," European Journal of Operations Research, 1999, p. 61-74.
- [Lee2005] Lee S. R., Joo S. D. and Lee C. W., "An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification," Dept. of Electr. & Comput. Eng., Ajou Univiversity, Suwon, South Korea, 2005, pp. 166-172.
- [Lin1987] Lindsay B., McPherson J. and Pirahesh H., "A Data Management Extension Architecture," Proceedings of ACM SIGMOD International Conference on Management of Data, USA, San Francisco, 1987.
- [Lom1995] Lomet T. and Tuttle M.R., "Redo Recovery after System Crashes," Proceedings of 21th International Conference On Very Large Data Bases, Switzerland, Zurich, 1995.
- [Lor1977] Lorie R.A., "Physical Integrity in a Large Segmented Databases," ACM TODS, 1997.
- [Lov2005] Love P., Merlino J., Zimmerman C. and Reed J.C., "Beginning Unix (Programmer to Programmer) ," Wrox, 2005, p. 480.

- [Lua2005] Lua E.K., Crowcroft J. Pias M., Sharma R. and Lim S., "A survey and comparison of peer-to-peer overlay network schemes", IEEE Communications Surveys and Tutorials - COMSUR , vol. 7, no. 1-4, 2005, pp. 72-93.
- [Lza2011] LZA TK ITTEA terminu datubāze / Internets. – <http://termini.lza.lv>
- [Mai1983] Maier D, "The Theory of Relation Databases," Rockville, Md.: Compiter Science Press, 1983.
- [Man2010] Manzoor F., Huang Y., Menzel K. "Passive RFID-Based Indoor Positioning System, An Algorithmic Approach," RFID-Technology and Applications (RFID-TA), 2010 IEEE International Conference on, 2010, pp. 112-117.
- [Mar1990] R. Marshall, J.Kempf, S. Dyer, and C.-C. Yen, "Visualization methods and simulation steering for a 3D turbulence model of Lake Erie," in SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics. New York, NY, USA: ACM Press, 1990, pp. 89–97.
- [Mar2005] Margrave D., "GSM Security and Encryption," George Mason University, 2005, p. 6.
- [Mey1998] Meyer B., "The Future of Object Technology," IEEE Computer, 1998.
- [Mic2003] "How NTFS Works", Microsoft, TechNet, 2003 / Internets. – [http://technet.microsoft.com/en-us/library/cc781134\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc781134(WS.10).aspx)
- [Mic2007] "Introducing the Office (2007) Open XML File Formats", Microsoft, MSDN Library, 2007 / Internets. – <http://msdn.microsoft.com/en-us/library/aa338205%28v=office.12%29.aspx>
- [Mil1998] Millar I., Beale M., Donoghue B. J., Lindstrom K. W. and Williams S., "The IrDA Standards for High-Speed Infrared Communications", Hewlet-Packard company, ASV, 1998.
- [Mil2008] Miles S. B., Sarma S. E. and Williams J. R., "RFID Technology and Applications," Cambridge University Press, USA: New York, 2008, p. 218.
- [Moh1992] Mohan C., Haderle D., Lindsay B., Pirahesh H. and Schwartz P, "ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging," ACM TODS, 1992.
- [Moh1995] Mohan C., "Repeating History Beyond ARIES," Proceeding of 25th International Conference on Very Large Data Bases, England, Scotland, Edinburgh, 1999.

- [Mon2007] Simon Moncrieff, Svetha Venkatesh, Geoff West, Stewart Greenhill, "Multi-modal emotive computing in a smart house environment," *Pervasive and Mobile Computing*, Volume 3, Issue 2, 2007, p. 74-94.
- [Mor1971] Morton M.S.S., "Management Decision Systems: Computer-Based Support for Decision Making," Harvard UNiversity, Division of Research, Graduate School of Business Administration, 1971.
- [MsC1999] McCracken J.B. and Lobrano B.D., "U.S. Manufacturers With Products Conforming To Metric Standards: An Analysis," National Institute of Standards and Rechnology, NIST GCR 99-783, 1999 / Internets. – <http://www.nist.gov/pml/wmd/metric/upload/thomas2.pdf>
- [Nel1965] Nelson T.H., "A File Structure og the Complex, the Changing, and the Indeterminate," *Proceedings of National ACM COnference*, USA, Ohio, Cleveland, 1965.
- [NIS1997] National Institute of Standards and Rechnology, "The United States And The Metric System: A Capture History", NIST LC 1136, 1997 / Internets. – <http://www.nist.gov/pml/wmd/metric/upload/1136a.pdf>
- [Nok2004] Task and Work Flow Management with One Touch - Nokia Mobile RFID Kit. Prepared by Nokia / Internets. – <http://press.nokia.com/2004/03/17/task-and-work-flow-management-with-one-touch-nokia-mobile-rfid-kit/>
- [Oli2002] Oliboni B. and Tanca L., "A visual language should be easy to use: a step forward for XML-GL," Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo de Vinci 32, Milan, Italy, 2002.
- [Ora2011] Java Platform Standard Edition 7, Oracle / Internets. –, <http://docs.oracle.com/javase/7/docs/api/>
- [Orl2004] Технологии разработки программного обеспечения: Учебник для вузов. 3-е изд./ С.А.Орлов. – СПб.: Питер, 2004. – 527 с.: ил.
- [Pen2008] Peng C., "GSM and GPRS security," Helsinki technology university, Finland, 2008.
- [Pic2010] Pichler R., *Agile Product Management with Scrum: Creating Products that Customers Love*, Addison-Wesley, ISBN: 978-0321605788, 2010, p. 160.
- [Poo1990] J. P. Poorte and D.A. Davis, "Computer Animation with CINEMA," *Winter Simulation Conference*, New Orleans, Louisiana, 1990, pp. 123-127.
- [Ren1997] Rennhackkamp M., "Mobile Database Replication," *DBMS*, 1997.

- [Rob2005a] Robbins A., "Unix in a Nutshell. Fourth Edition," O'Reilly Media, 2005, p. 902.
- [Rob2005b] Robbins A. and Beebe N.H.F., "Classic Shell Scripting ," O'Reilly Media, 2005, p. 560.
- [Rob2006] Roberts C.M., "Radio frequency identification (RFID)," Computer & Security 25, Department of Information Sciences, Otago University, New Zealand, 2006, pp. 18-26.
- [Roo1993] M. Rooks, "A User-Centered Paradigm for Interactive Simulation," Simulation, 1993, pp. 168-177.
- [Ros1994] Ross R.G., "The Business Rule Book: Classifying, Defining, and Modeling Rules (Version 3.0)," Boston, Mass.: Database REsearch Group, 1994.
- [Ros1998] Ross R.G., "Business Rule Concepts," USA, Houston, Tex.: Business Rule Solutions Inc., 1998.
- [Rot1977] Rothnie J.B. and Goodman N., "A Survey of Research and Development in Distributed Database Management," Proceedings of 3th International Conference on Very Large Data Bases, Japan, Tokyo, 1977.
- [Rot1980] Rothnie J.B., "Introduction to a System for Distributed Databases (SDD-1)," ACM TODS, 1980.
- [Row1987] Rowe L.A. and Stonebraker M.R., "The Postgres Data Model," Proceedings of 13th International Conference on Very Large Data Bases, UK, Brighton, 1987.
- [Rus2005] Russon R. and Fledel Y., "NTFS Documentation," 2005, p. 138.
- [Sak2005] Sakaguchi T., Koyama D., Nakamichi R. and Imamura S., "FOMA Core Network Circuit/Packet Switching Separation Technology," NTT DoCoMo Technical Journal Vol. 6. No.2, Japan, 2005.
- [Sam2010] SAMSUNG's New Near Field Communication Chip Offers Increased Wireless Connectivity for Mobile Handsets. Prepared by Samsung / Internets. – http://www.samsung.com/global/business/semiconductor/newsView.do?news_id=1211
- [Sap2009] "Enable real-world awareness with SAP solutions for Auto-ID and Item Serialization", SAP AG, 2009 / Internets. – SAP AG, www.sap.com/solutions/auto-id/index.epx
- [Sar2000] Sarma S., Brock D. L. and Ashton K., The Networked Physical World Proposals for Engineering the Next Generation of Computing, Commerce & Automatic-

Identification, Massachusetts Institute of Technology, USA: Cambridge, 2001
<http://www.autoidlabs.org/uploads/media/MIT-AUTOID-WH-001.pdf>

- [Sar2010] Sarac A., Absi N. A literature review on the impact of RFID technologies on supply chain management. *Integrating the Global Supply Chain*, vol. 128, issue 1, 2010, pp. 77-95.
- [Sas2006] Sassan A.. "Introduction to Mobile WiMAX Radio Technology: RHY and MAC architecture, *Wireless Standards and Technology*," Intel Corporation, ASV, 2006.
- [Saw2008] J. Saw and M. Butler, "Exploring graphical user interfaces and interaction strategies in simulations," Melbourne, 2008.
- [Sch2002] Schollmeier R., "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications," *Proceedings of the First International Conference on Peer-to-Peer Computing*, 2002.
- [Sch2005] Schoder D., Fischbach K. and Schmitt C., "Core Concepts in Peer-to-Peer (P2P) Networking," *P2P Computing: The Evolution of a Disruptive Technology*, Idea Group Inc, Hershey, 2005.
- [Sco1996] Scourias J. , "Overview of GSM, The Global System for Mobile Communications," Waterloo University, 1996.
- [Ser2004] Serrano O., María Cañas J., Matellán V, and Rodero L. , "Robot localization using WiFi signal without intensity map" // *Proceedings of the 5th Workshop on Physical Agents (WAF-2004)*, 2004.
- [She2007] Jih-Biing Sheu, An emergency logistics distribution approach for quick response to urgent relief demand in disasters Original Research Article *Transportation Research Part E: Logistics and Transportation Review*, Volume 43, Issue 6, November 2007, p. 687-709.
- [Sim1995] Simon A.R., "Strategic Database Technology: Management for The Year 2000," Morgan Kaufmann Publishers Inc., USA, San Francisco, 1995, p. 479.
- [Soh2007] Sohraby K., Minoli D. and Znati T., "Wireless sensor networks: technology, protocols, and applications," John Wiley & Sons, ISBN 978-0-471-74300-2, 2007.
- [Sor2003] Sorkin A.R., "Software Bullet Is Sought to Kill Musical Piracy," *New York Times*, 2003.
- [Sta2003] Statistisches Bundesam, "Bevölkerung und Erwerbstätigkeit 2002," *Beruf, Ausbildung und Arbeitsbedingungen der Erwerbstätigen (Ergebnisse des Mikrozensus)*, Germany: Wiesbaden, 2003, pp. 72–73.

- [Ste2005] Stevenson R., "Laser Marking. Matrix Codes on PCBS," *Circuit design & Manufacture*, Cmslaser, 2005, pp. 32-36.
- [Sti2006] Stigge M., Plötz H., Müller W., Redlich J. P., "Reversing CRC – Theory and Practice," HU Berlin Public Report, SAR-PR-2006-05, 2006, p. 24.
- [Sto1986] Stonebraker M. and Rowe L.A., "The Design of Postgres," *Proceedings of ACM SIGMOD International Conference on Management of Data*, Washington, 1986.
- [Sto1987] Stonebraker M., "The Design of the POSTGRES Storage System," *Proceedings of 13th International Conference on Very Large Data Bases*, UK, Brighton, 1987.
- [Sto1991] Stonebraker M. and Kemnitz G., "The Postgres Next Generation Database Management System," *CACM*, 1991.
- [Sto2005] Storey J., Salaman G. and Platman K., "Living with enterprise in an enterprise economy: Freelance and contract workers in the media," *Human Relations* 58, Germany: Hagen, 2005, pp.1033–1054.
- [Suh2005] Suh N, "Complexity in Engineering," *Annals of the CIRP*, 2005, p. 581-598.
- [Suß2010] Suß S. and Kleiner M., "Commitment and work-related expectations in flexible employment forms: An empirical study of German IT freelancers," *European Management Journal* 28, 2010, pp. 40–54.
- [Suz2008] Suzuki R., Furusawa T., Sezaki K. and Konomi S., "An RFID-Based Positioning System for Ad-Hoc Networks", 2008
- [Tan2012] Kenichi Tanaka, Akihiro Yoza, Kazuki Ogimi, Atsushi Yona, Tomonobu Senjyu, Toshihisa Funabashi, Chul-Hwan Kim, "Optimal operation of DC smart house system by controllable loads based on smart grid topology", *Renewable Energy*, Volume 39, Issue 1, 2012, p. 132-139.
- [Tay2008] Taylor B.N. and Thompson A., "U.S. Manufactures with Product Conforming to Metric Standards: An Analysis", United States version of the English text of the eighth edition (2006) of the International Bureau of Weights and Measures publication *Le Système International d' Unités (SI)*, National Institute of Standards and Rechnology, 2008.
- [Tel2011] Telecom Italia SpA, Java Agent DEvelopment Framework / Internets. – <http://jade.tilab.com/>
- [Ten2000] Tennenhouse, D. Proactive computing. *Communications of the ACM (CACM)*, Vol. 43, (no. 5), Japan, 2000, pp. 43–50 / Internets. – http://portal.acm.org/supp_gateway.cfm?id=332837&type=pdf&path=%2F340000

%2F332837%2Fsupp%2Fp43%2Dtenhouse%2Ejp%2Epdf&coll=GUIDE&dl=GUIDE&CFID=99527770&CFTOKEN=89181945

- [Tho2005] Thomas F. and Ros L. "Revisiting trilateration for robot localization," IEEE Transactions on Robotics, 21(1), 2005, pp. 93-101.
- [Tho2006] Thornton F., Haines B. RFID Security. – USA: Syngress, 2006
- [Tra2000] Tansley D., "Linux and Unix Shell Programming," Addison-Wesley, 2000, p. 528.
- [Tsa2012] Ming-Chih Tsai, Kee-hung Lai, Alison E. Lloyd, Hung-Ju Lin, The dark side of logistics outsourcing – Unraveling the potential risks leading to failed relationships
Original Research Article Transportation Research Part E: Logistics and Transportation Review, Volume 48, Issue 1, January 2012, p. 178-189.
- [UNI2012] The Unicode Consortium, The Unicode® Standard: A Technical Introduction, 2012 / Internets. – <http://www.unicode.org/standard/principles.html>
- [Vaj2003] Vajda I., Buttyan L. Lightweight Authentication Protocols for Low-Cost RFID Tags. Second Workshop on Security in Ubiquitous Computing (UbiComp), Seattle, 2003
- [Van2009] Van Deursen T., Radomirovic S. Security of RFID Protocols – A Case Study. Electronic Notes in Theoretical Computer Science 244, 2009, pp. 41–52.
- [Vas2005] Vasilescu I., Kotay K., Rus D., Dunbabin M., and Corke P., "Data collection, storage, and retrieval with an underwater sensor network," In Proceedings of the 3rd international conference on Embedded networked sensor systems, SenSys'05, ACM, USA, New York, 2005, p. 154-165.
- [Vid2010] Vidal J., "Fundamentals of Multiagent Systems," 2010 / Internets. – <http://www.damas.ift.ulaval.ca/~coursMAS/ComplementsH10/mas-Vidal.pdf>
- [Wag1996] P. R. Wagner, C.M.D.S Freitas and F.R. Wagner, "A New Paradigm for Visual Interactive Modeling and Simulation," European Simulation Symposium, 1996.
- [Wid1996] Widom J., and Ceri S., "Active Database Systems: Triggers and Rules for Advanced Database Processing," USA, San Francisco, Morgan Kaufmann, 1996.
- [Wid2007] Widyawan, Klepal M. and Pesch D. "Influence of Predicted and Measured Fingerprint on the Accuracy of RSSI-based Indoor Location Systems," In: Proceedings of 4th Workshop on Positioning, Navigation, and Communication 2007 (WPNC'07), 2007, pp.145-151.
- [WiF2012] Wi-Fi Alliance / Internets. – <http://www.wi-fi.org>

- [Wil1999] Wilensky U., "NetLogo", CLL, 1999 / Internets. – <http://ccl.northwestern.edu/netlogo/>
- [Woo2001] Wooldridge M., "An introduction to MultiAgent Systems," John Wiley & Sons Ltd., ISBN 0-471-49691-X.,2001.
- [Woo2002] Wooldridge M., "An Introduction to Multiagent Systems," John Wiley & Sons, England, Chichester. 2002, p. 348.
- [Yim2010] Yim J., Jeong S., Gwon K. and Joo J. "Improvement of Kalman filters for WLAN based indoor tracking," Expert Systems with Applications, 37(1), 2010, pp. 426-433.
- [Zip2001] Ziparo V. A., Kleiner A., Nebel B., Nardi D. RFID Based Exploration for Large Robot Teams, Robotics and Automation, 2007 IEEE International Conference, 2007, pp. 4606-4613.
- [Zur2007] Zuravlyov V. Main Principles of a New Concept of Designing Data Management Systems // Scientific Journal of RTU. 5. series., Datorzinātne. - 30. vol. (2007), pp 38-46.
- [Zur2008a] Zuravlyov V., Latisheva E. Various Aspects of RFID Technology and Their Use in Monitoring and Management of Traffic // Scientific Journal of RTU. 5. series., Datorzinātne. – 34. vol., Latvia, Riga, 2008, pp. 238–245.
- [Zur2008b] Zuravlyov V. Various Aspects of 'Smart Laundry' Task Implementation, Using 'RFID' Technology // Proceedings of papers 'Informatics in the scientific knowledge 2008', Varna, Bulgaria, 26–28 June, 2008, pp. 150–156.
- [Zur2008c] Latisheva E., Zuravlyov V. Various Aspects of Simulation's Usage for Creation of Real Solutions with „DARSIR” Conception as Example // Proceedings of Conference 'Modeling of business, industrial and transport systems 2008; MBITS 2008', ISBN 978-9984-818-04-7, Latvia, RIGA, 7.-10. May, 2008. - pp 132-135.
- [Zur2011a] Zuravlyov V., Matrosov A. Multi-agent system built using RFID technology // Proceedings of the 6th international conference on electrical and control technologies, ISSN 1822-5934, Lithuania, Kaunas, 5–6 May, 2011. – pp. 15–20.
- [Zur2011b] Zuravlyov V., Kryukov D., Kairish V. Security problems related to RFID-based concept usage and methods to counter them // Proceedings of the 6th internal conference on electrical and control technologies, ISSN 1822-5934, Lithuania, Kaunas, 5.-6. May, 2011. – pp. 76-81.

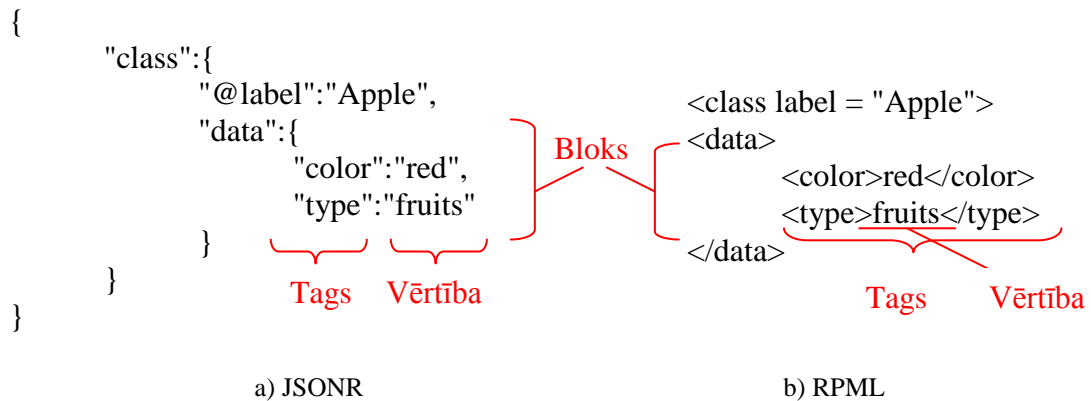
- [Zur2011c] Zuravlyov V., Latisheva E., Karish V. The Theoretical Basis and Practical Usage of the Simulation for Creating IT solutions for Data Management Systems // Proceedings of the international conference on Applied Computer Systems, ISSN 1407-7493, Latvia, Riga, 2011., pp. 138-144.
- [Zur2012a] Zuravlyov V., Matrosov A., Rutko D., Behavior Pattern Simulation of Freelance Marketplace, Workshop on Agents and Multi-agent systems for Enterprise Integration, ZOCO 2012, Salamanca, Espana, March, 2012.
- [Zur2012b] Zuravlyov V., Matrosov A., Rutko D., Freelance resource management system optimization, The 2nd International Conference on Computer and Management, CAMAN 2012, Wuhan, China, March, 2012.
- [Zur2012c] Zuravlyov V., Latisheva E., Lavendels J. Concept of Data Management Systems Based on Individual Object Properties // Proceedings of papers 'Informatics in the scientific knowledge 2012', Bulgaria, Varna, 27–29 June, 2012, pp. 269–279.
- [Zur2012d] Zuravlyov V., Latisheva E. Research of Data Storage and Management Technology Applied to DARSIR Concept // Proceedings of papers 'Informatics in the scientific knowledge 2012', Bulgaria, Varna, 27–29 June, 2012, pp. 280–294.

Pielikumi

DARSIR koncepcijas datu apmaiņas tehniskā realizācija

Loģisku bloku struktūras un sintakse

Tags ir pamatelements DARSIR dokumentā. Tags sastāv no divām daļām: taga nosaukums un taga vērtība. Viena tipa informācija ir apvienota *blokos*. Bloku un tagu sintakse *RPML* un *JSONR* formātiem ir shematiski parādīta P1.1. attēlā.



P1.1. attēls. „Apple” resursa piemērs

Atribūti tiek definēti „*data*” bloka iekšienē.

Datu tipi un to prezentācijas formāti

DARSIR koncepcijas realizācijai nepieciešams definēt iespējamo datu tipus. Noteikumam darba gaitā ir nepieciešams *loģiskais* tips, kas nosaka, vai jādara kaut kādas darbības (pārbaude, cikls utt.) vai nē. Vispopulārākais datu tips ir *teksts*. Visi dati tiek prezentēti šajā tipā. Datus iespējams konvertēt no teksta tipa uz citiem (loģiskais, cipars utt.), bet tādā gadījumā teksta tipam jābūt attiecīgajā datu tipa formātā.

Tiek definēti galvenie datu tipi un to prezentācijas formāti:

- *Loģiskais (bināru)* – tas ir datu tips, kas var pieņemt divas iespējamās nozīmes: *patiess* (ang. *true*) vai *nepatiess* (ang. *false*). Šis datu tips ir svarīgs nosacījumiem. Piemēram, iespējams definēt triggeri ar nosacījumu. Kad tiek palaists triggeris, tiek pārbaudīts nosacījums, un, ja nosacījums ir *patiess*, tad triggeris palaiž trigeru darbību. Pretējā gadījumā trigeru darbība netiek palaista. Nosaukums datu tipam ir *bool*.
- *Teksta* – tas ir datu tips pēc noklusējuma. Formāts ir simbolu rinda, katrs atsevišķais simbols ir prezentēts Unicode [UNI2012] veidā. Iespējams izmantot dažādu valodu burtus. Nosaukums datu tipam ir *string*.
- *Skaitļu* – tā ir ciparu prezentācija teksta veidā. Teksta rindā iespējamie simboli: cipars no 0 līdz 9, punkts (dalītājs), komats (dalītājs) un mīnus (negatīvais ciparu gadījumā).

Ciparu prezentāciju teksta veidā iespējams transformēt (konvertēt) uz skaitli. Ja transformācijas laikā tekstā eksistē neatļauti simboli, tad skaitļa vērtība būs *nulle* (ang. *null*). Ja teksta rinda, kura jātransformē uz skaitli, sākas ar dalītāju (punkts vai komats), tad tiek uzskatīts, ka pirms dalītāja ir nulle (piemēram, 0.123, tas pats, kas .123). Datu tipu nosaukumi un vērtības diapazoni tiek ņemti par pamatu no [Eck2003]: *integer, real, float* utt.

- *Baitu* – ir gadījumi, kad nav pietiekams teksta datu tips, nepieciešams prezentēt baitu. Formāts ir divi simboli, kas prezentē heksadecimāla veidā bitu (decimālā veidā tas ir cipars no 0 līdz 255), tas ir diapazons no 00 līdz FF. Piemēram, tas ir vajadzīgs, ja nepieciešams glabāt attēlu. Attēls - tā ir baitu kopa, kas *dokumentā* prezentēta baitu veidā. Nosaukums datu tipam ir *byte*.
- *Mērvienības* – ja ir nepieciešams definēt nozīmi mēru sistēmā, tad tiek definēts atsevišķs formāts datu prezentācijai. Tiek definētas vienpadsmit pamata vienības, ar kurām iespējams definēt citu veidu mērvienības. Specifikācija ir aprakstīta 4.2. apakšnodaļā 4.1. un 4.2. tabulās.
- *Pašu definētais* – iespējams definēt savu formātu, bet ar ierobežojumu, ka šo formātu iespējams definēt tekstā veidā.

2. pielikums

Resursu atribūtu grupas realizācija

DARSIR koncepcijai visvairāk piemērotas ir RFID un svītrkodu tehnoloģijas. Atribūtu klasificēšanai tika ņemti RFID un svītrkodu bāzes standarti un darbības principi, arī pētījumi [Sar2000, Bro2001a, Bro2001b, Bro2001c un Bro2001d], kuru rezultāts ir *fizikāla iezīmēšanas valoda* (ang. *Physical Markup Language, PML*) standarts. Vairākos informācijas avotos [Lah2005, Hei2005, Sar2010, Bro2001b, Bro2001d, Mil2008, Hun2007] norādīti dažādi fizisko objektu klasificēšanas principi. Lielākā daļa no norādītajiem avotiem apraksta PML izmantošanu.

Pamatojoties uz augsta līmeņa atribūtu grupu definējumu (skat. 4.2.1. apakšnodaļā), šīs apakšnodaļas ietvaros ir aprakstīta viena no iespējamajām atribūtu grupas realizācijām.

Iepriekš definējām, ka DARSIR dokumentam ir izveidoti divi datu apmaiņas formāti RPML un JSONR. Katram formātam ir sava specifikācija, kas balstās uz bāzes valodu (XML vai JSON). Katram atribūtu grupas paskaidrojumam ir doti piemēri RMPL un JSONR formātos.

Resursa identifikators

Tika definēts, ka *resursa identifikators* (ang. *Resource ID, RI*) ir EPC Type I shēmas tipa identifikators (skat. 4.3. attēlu). Tas ir 96 bitu identifikators, kas sastāv no četrām daļām: sistēmas numurs (8 biti), ražotāja numurs (28 biti), objekta klases numurs (24 biti) un seriālais numurs (36 biti).

JSONR un RMPL datu apmaiņas formāti reprezentēti teksta vidē. RI daļās dalītājs ir punkts. Daļu sadalīšana nepieciešama lietotājam, lai pēc RI vizuālā izskata būtu iespējams identificēt: sistēmas, ražotāja, objekta klases un seriālo numuru.

RI nav nepieciešams definēt atsevišķu bloku, pietiek ar vienu atribūtu ar nosaukumu *epc*. Katra daļa prezentēta heksadecimālā veidā, ja cipars daļā satur nulles priekšā, nulles tiek ignorētas. Piemēram, no [Bro2001a] ņemts RI paraugs *01.0000A89.00016F.000169DC0*, tas tiks atspoguļots *1.A89.00016F.000169DC0* veidā (skat. P2.1. attēlā).

"epc": "1.A89.16F.169DC0"

<epc>1.A89.16F.169DC0</epc>

a) JSONR

b) RPML

P2.1. attēls. RI prezentācijas piemērs

EPC Type I shēmas izmantošana nodrošina tādu informācijas glabāšanu kā sistēmas numurs, ražotāja numurs, objekta klase un seriālais numurs. Ar šīs informācijas apvienošanu tiek nodrošināts unikālais identifikators, turklāt iespējams šo informāciju glabāt ar nelielu datu apjomu.

Individuālās īpašības

Katram resursam ir savas *resursa individuālās īpašības* (ang. *Resource Individual Properties, RII*). Tiek uzskatīts, ka visas nepieciešamās RII iespējams prezentēt izmantojot bāzes mērvienības: *garums* (m), *masa* (kg), *laiks* (s), *elektriskā strāva* (A), *temperatūra* (K), *vielas daudzums* (mol), *gaismas intensitāte* (cd), *daudzums* (q), *leņķis* (rad), *proporcijas* (p) un *loģiska* (b). Pieņemsim, ka RII saglabāšanai definēsim atsevišķu dokumenta bloku *rii*.

Tika definēts, ka šīs mērvienības, kas ir ārpus vienpadsmit definētajām bāzes mērvienībām, jādefinē, kā attiecības starp dažādām bāzes mērvienībām. Piemēram, definēsim, ka eksistē automašīna, kurai ir vajadzīga norādīt RII ātrumu 300 km/h, kas savukārt ir 83.3 m/s. *Ātrumu* nepieciešams definēt, izmantojot bāzes mērvienības *garums* un *laiks* (skat. piemēru P2.2. attēlā).

```
"epc":"09.34F.16F.1384D32",           <epc>09.34F.16F.1384D32</epc>
"rii":{"speed":{"m":"1","s":"-1","q":"83.3"}
      }
}
```

a) JSONR

b) RPML

P2.2. attēls. Resursa individuālo īpašību prezentācijas piemērs

Eksistē īpašības, kuras iespējams raksturot ar vienu tagu, ir tagi, kuriem ir nepieciešama tagu kopa. Piemēram, ņemsim vitamīnu kompleksu „*Vitabiotics Feroglobin B12*”. Tas ir uztura bagātinātājs, kas sastāv no 4 minerāliem un 2 vitamīniem, detalizēta informācija ir atrodama marķējumā, kas atrodas uz iepakojuma. (skat. P2.3. attēlā).

Uztura informācija	Vid. vienā kapsulā	% RDA
Iron (as Fumarate)	17 mg	121
Zinc	12 mg	120
Copper	2000 µg	200
Folacin (as Folic Acid)	400 µg	200
Vitamin B12	10 µg	400
Vitamin B6	5 mg	357

*RDA – Rekomendēta dienas deva
µg – mikrogramms mg – miligramms

P2.3. attēls. Vitamīnu kompleksa „*Vitabiotics Feroglobin B12*” marķējums, kur atspoguļoti dati par uzturvērtībām

Katrs no elementiem marķējumā sastāv no trijām sastāvdaļām: minerālvielas/vitamīnu nosaukums, deva un procents no *rekomendētās dienas devas* (ang. *Recommended Daily Allowance, RDA*). Kopumā ir sešas RII, kuras tiek iedalītas atsevišķos rii blokos. Katrā blokā ir definēti trīs tagi: *nosaukums* (*name*), *svars* (*mass*) un *RDA* (*rda*). Svaram definēta *m* (*kilograms*) bāzes mērvienība ar vienas vienības nozīmi 0.00001 kg. RDA ir norādīts, izmantojot *p* (*procents*) bāzes mērvienības ar vienības nozīmi 1%. Rezultāts ir prezentēts P2.4. attēlā.

<pre>"epc": "09.34F.16F.1384D32", "rii": [{ "name": "Iron (As Fumarate)", "mass": { "m": "0.00001", "q": "17" }, "rda": { "p": "1", "q": "121" }, }, { "name": "Zinc", "mass": { "m": "0.00001", "q": "12" }, "rda": { "p": "1", "q": "120" }, }, { "name": "Cooper", "mass": { "m": "0.00001", "q": "0.02" }, "rda": { "p": "1", "q": "200" }, }, { "name": "Folacin (as Folic Acid)", "mass": { "m": "0.00001", "q": "0.004" }, "rda": { "p": "1", "q": "200" }, }, { "name": "Vitamin B12", "mass": { "m": "0.00001", "q": "0.0001" }, "rda": { "p": "1", "q": "400" }, }, { "name": "Vitamin B6", "mass": { "m": "0.00001", "q": "5" }, "rda": { "p": "1", "q": "357" } }]</pre>	<pre><epc>09.34F.16F.1384D32</epc> <rii> <name>Iron (As Fumarate)</name> <mass m=0.00001>17</mass> <rda p=1>121<rda> </rii> <name>Zinc</name> <mass m=0.00001>12</mass> <rda p=1>120<rda> </rii> <name>Cooper</name> <mass m=0.00001>0.02</mass> <rda p=1>200<rda> </rii> <name>Folacin (as Folic Acid)</name> <mass m=0.00001>0.004</mass> <rda p=1>200<rda> </rii> <name>Vitamin B12</name> <mass m=0.00001>0.0001</mass> <rda p=1>400<rda> </rii> <name>Vitamin B6</name> <mass m=0.00001>5</mass> <rda p=1>357<rda> </rii></pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

a) JSONR

b) RPML

P2.4. attēls. Vitamīnu kompleksa „*Vitabiotics Feroglobin B12*” marķējuma prezentācijas piemērs

Saistības

Definējam, ka vienam fiziskam objektam iespējamas vairākas saistības. Visas saistības ir apvienotas atsevišķā dokumentu blokā *ent* (ang. *entity*). Katram saistību veidam ir savu atribūtu kopa, lai atdalītu vienu saistību veidu no cita, ir ieviests *tp* (ang. *type*) tags. Tipiskie saistību veidi ir ražotājs (ang. *manufacturer*), pārdevējs (ang. *seller*), īpašnieks (ang. *owner*) utt.

Piemēram, eksistē prece vitamīnu komplekss „*Vitabiotics Feroglobin B12*”, kuru izveidoja *Vitabiotics Ltd.* kompānija, kuras adrese ir *1 Apsley Way, London NW2 7HF, United Kingdom*. Ar šīs kompānijas klientu servisu var sazināties pa telefonu *+44 (0)20 89552662*

vai elektronisko pastu info@vitabiotics.com. Šīs preces īpašnieks ir *Vadims Žuravļovs*, personas kods 999999-99999, viņa telefons +371 28346517. Aprakstītais piemērs ir prezentēts P2.5. attēlā.

<pre>"ent":{ "manufacturer":{ "product":"vitaminu komplekss „Vitabiotics Feroglobin B12””, "name":"Vitabiotics Ltd.", "address":"1 Apsley Way, London NW2 7HF, United Kingdom", "phone":"+44 (0)20 89552662", "email":"info@vitabiotics.com" }, "owner":{ "name":"Vadims Zuravlovs", "id":"999999-99999", "phone":"+371 28346517" } }</pre>	<pre><ent> <manufacturer> <product>vitaminu komplekss „Vitabiotics Feroglobin B12”</product> <name>Vitabiotics Ltd.</name> <address>1 Apsley Way, London NW2 7HF, United Kingdom</address> <phone>+44 (0)20 89552662</phone> <email>info@vitabiotics.com</email> </manufacturer> <owner> <name>Vadims Zuravlovs</name> <id>999999-99999</id> <phone>+371 28346517</phone> </owner> </ent></pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

a) JSONR

b) RPML

P2.5. attēls. Saistību atribūta piemērs

Laiks

Tika apskatītas metodes, kā saglabāt laiku atribūtos un pieņemt lēmumu, izmantojot 64 bitu skaitli sekundes glabāšanai, kas tiek prezentēts decimālā veidā. Šajā 64 bitu skaitlī tiek saglabāts skaits no 1970. gada 1. janvāra pusnakts pēc UTC. Piemēram, *2012. gada 1. janvāra* ir vērtība *1325436230*. Laiks pirms šī datuma ir ar negatīvu vērtību, pēc šī datumam - ar pozitīvu vērtību. Tiek glabātas absolūtās vērtības nevis aprēķinu metodes, lai samazinātu datu apstrādes laiku.

Pieņemsim, ka laika saglabāšanai definēsim atsevišķu dokumenta bloku *tm* (ang. *time*). Piemēram, eksistē darbinieks, kuram jāstrādā visas dienas no 8:00 līdz 17:00, viņam ir pusdienas pārtraukums no 12:00 līdz 12:30. Šim gadījumam ir ieviesti četri atribūti: *wfrom* (darba laiks no), *tfrom* (darba laiks līdz), *dfrom* (pusdienas laiks no) un *dto* (pusdienas laiks līdz). Tiek izmantota bāzes mērvienība laiks *t*, kurai definētā vienība ir vienāda ar 3600 sekundēm, kas vienāda ar vienu stundu sekundēs. Ja nepieciešams definēt 12:30, tad jānorāda vērtība 12.5 (12.5 stundas * 3600 sekundes= 45000 sekundes).

Rezultāts ir prezentēts P2.6. attēlā.

<pre>"epc":"09.34F.16F.1384D32", "tm":{ "t":"3600", "wfrom":"8", "wto":"17", "dfrom":"12", "dto":"12.5"}</pre>	<pre><epc>09.34F.16F.1384D32</epc> <tm> <wfrom t=3600>8</wfrom> <wto t=3600>17</wto> <dfrom t=3600>12</wfrom> <dto t=3600>12.5</wto> </tm></pre>
--------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

a) JSONR

b) RPML

P2.6. attēls. Laika atribūta piemērs

Iespējams nedaudz modificēt piemēru un noteikt, ka darbiniekam jāstrādā tikai darba dienās. Tādam gadījumam iespējams ieviest divus papildu atribūtus: *dayfrom* (sākuma diena, piemēram, pirmdiena vienāda ar 0) un *dayto* (beidzamā diena, piemēram, piektdiena vienāda ar 432000).

Vēsture

Šim atribūtam ir divi režīmi ierakstīšanai. Ja ir tikai viens atribūts, kuram jāpiefiksē vēsture, tad tas tiek definēts ar papildu atribūta parametru *hs* (saīsinājums no angļu vārda *history*) ieviešanu. Ja ir atribūtu kopa, kurai jāpiefiksē vēsture, tad atribūta parametru "hs" jādefinē blokam.

Šim atribūtam ir iespējami sekojoši režīmi:

1. *Nepārtrauktais* – ir nodrošināts režīms, kad nepārtraukti tiek saņemta informācija par fizisko objektu.
2. *Uz notikumu* – ir definēti notikumi, kad nepieciešams saglabāt vēsturi. Piemēram, tas ir sensoru darbības rezultāts, objekta pārvietošana (piemēram, uz noliktavu, uz mašīnu utt.).
3. *Periodiskais* – ir definēts periods (piemēram, katru stundu, katru dienu utt.), kad notiek vēstures glabāšana.

Piemēram, apskatīsim 2. režīma piemēru. Mums ir ābols ar *EPK* numuru *01.A89.16F.169DC0*, kuram jāpiefiksē vēsture par ābola krāsu un derīguma termiņu. Un ir sensoru sistēma, kas seko līdz ābola stāvoklim. Uzskatām, ka 2012. gada 1. janvārī ābols ir zaļš un tiek definēts ābola derīguma termiņš 3 mēneši. Ābolam nav nodrošināts saglabāšanas režīms. 2012. gada 1. februārī sensori noskaidroja, ka ābols kļūst dzeltens. Un 2012. gada 15. februārī sensori noskaidroja, ka beidzas derīguma termiņš, tāpēc ābols kļūst brūns un nav ēdams. Pieņemsim, ka vēstures saglabāšanai definēsim atsevišķu dokumenta bloku *hs*.

Šī piemēra paraugs ir attēlots P2.7. attēlā.

"epc":"09.34F.16F.1384D32",	<epc>09.34F.16F.1384D32</epc>
"hs":{"s":"1325436230","color":"green","to":"1333298630"},	<hs s=1325436230>
"hs":{"s":"1328114630","color":"yellow"},	<color>green</color>
"hs":{"s":"1329324230","color":"brown","to":"0"}	<to>1333298630</to>
	</hs>
	<hs s=1328114630>
	<color>yellow</color>
	</hs>
	<hs s=1329324230>
	<color>brown</color>
	<to>0</to>
	</hs>

a) JSONR

b) RPML

P2.7. attēls. Vēstures atribūta piemērs

Pozicionēšana

Definējam, ka pozicionēšanas atribūtam iespējams izmantot GPS koordinātes. Parasti tiek izmantotas divas koordinātu asis *platums* (ang. *latitude*) un *garums* (ang. *longitude*). Vēl iespējams izmantots *augstumu* (ang. *altitude*), kuru parasti neizmanto, bet tā vietā izmanto atsevišķo altimetra moduli.

Pieņemsim, ka pozicionēšanas saglabāšanai nav nepieciešams atsevišķi norādīt bloku. Iespējams izmantot atsevišķo atribūtu, nosauksim to par *gps*. Tiek definēts platums un pēc komata garums. Vēl iespējams norādīt pēc komata augstumu, ja tas nav norādīts, tad šis parametrs tiek ignorēts. Piemēram, Rīgas pilsētas *Brīvības pieminekļa* koordinātes prezentētas P2.8. attēlā.

```
"epc":"09.34F.16F.1384D32",           <epc>09.34F.16F.1384D32</epc>  
"gps":"56.9515504,24.1133876"        <gps>56.9515504,24.1133876</gps>
```

a) JSONR

b) RPML

P2.8. attēls. Pozicionēšanas *GPS* koordinātu piemērs

Iekštelpu pozicionēšanai ir izmantots cits darbības princips, tiek definēts attālums metros pa X, Y un Z asīm no sākuma punkta. Norādīt sākuma punktu nav obligāti, tas tiek darīts pēc vajadzības. Piemēram, ir noliktava, kurā par sākuma punktu tiek definēta vieta, kur norādīts iekštelpu pozicionēšanas modulis. No moduļa viena metra pa X asi, trīs metrus pa Y asi un 4.5 metru augstumā (piemēram, cita konteinera augšā) stāv konteiners.

Tāda piemēra prezentēšana ir attēlota P2.9. attēlā.

```
"epc":"09.34F.16F.1384D32",           <epc>09.34F.16F.1384D32</epc>  
"crd":"1,3,4.5"                       <crd>1,3,4.5</crd>
```

a) JSONR

b) RPML

P2.9. attēls. Pozicionēšanas iekštelpu koordinātu piemērs

Piekļuves tiesības

Tiek uzskatīts, ka piekļuves tiesības tiek regulētas, pamatojoties uz resursa identifikatoru. Pēc resursa identifikatora iespējams definēt četras lietotāju grupas: sistēmu, ražotāju, objekta klasi un pašu elementu (seriālais numurs). Ar regulāru izteiksmju (skat. [Keg1984, Lov2005, Rob2005b]) palīdzību ir definēti konkrēti resursi. Tipiskie izmantotie simboli regulārās izteiksmēs: „*” – visi elementi (piemēram, *A.B.C.** - visi seriālie numuri, kas attiecas uz A sistēmas B ražotāju, C objekta klasi), „?” - jebkurš viens numurs (piemēram, *A.B.C.?* – no 0 līdz F seriālie numuri, kas attiecas uz A sistēmas B ražotāju, C objekta klasi) utt.

Katrai no četrām lietotāju grupām ir definētas savas iespējamās operāciju kombinācijas. Ir četras pamatoperācijas: lasīt, rakstīt, palaist un mainīt piekļuves tiesības. Katra no operācijām ir iekodēta ar vienu bitu: 1 – izmantota un 0 – neizmantota. Ar četriem bitiem (vienu heksadecimālu ciparu) ir iespējams iekodēt visas četras operācijas: *1-bits* – lasīt, *2-bits* rakstīt, *3-bits* mainīt piekļuves tiesības un *4-bits* palaist. Katrai grupai ir definēts savs heksadecimālais cipars, kopumā četri heksadecimālie cipari, pa vienai katrai lietotāju grupai. Ja visām lietotāju grupām ir vienādas tiesības, tad tās iespējams ierakstīt ar vienu ciparu.

Definēsim, ka piekļuves tiesības ir definētas ar pazīmi *acs* (ang. *access*). Šīm pazīmēm vērtība ir resursa numurs (iespējams izmantot regulāras izteiksmes), komats un četri heksadecimālie cipari (kas definē piekļuves tiesības). Pazīmi iespējams ievietot dažādos līmeņos: dokumenta, bloka, vai atribūta. Piemēram, ir konteiners. Konteineram ir piešķirts unikālais identifikators *09.34F.16F.1384D32*, kuru aizliegts mainīt. Definēts fiziskais stāvoklis *rii* blokā, kur definēts, ka A sistēmas B ražotājs var darīt visu, izņemot piekļuves tiesību izmaiņu, bet tādā pašā ražotājam no *CO* līdz *CF* objektu klasei atļautas visas operācijas. Vēstures glabāšanas modulī *hs* jebkurš resurss var saglabāt vēsturi, bet nav iespējams lasīt, palaist vai mainīt piekļuves tiesības šim blokam.

Šī piemēra iespējamā realizācija ir dota P2.10. attēlā.

<pre>"epc":"09.34F.16F.1384D32","acs":"*0", "rii":{ "acs":"A.B.C?.DD1.0DFF","color":"green"}, { "acs":"*4", "hs":[{"s":"1325436230","color":"yellow"}, {"s":"1328114630","color":"green"}] }</pre>	<pre><epc acs="*0">09.34F.16F.1384D32</epc> <rii acs="A.B.C?.DD1.0DFF"> <color>green</color> </rii> <hs acs="*4"> <color s=1325436230>yellow</color> <color s=1328114630>green</color> </hs></pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

a) JSONR

b) RPML

P2.10. attēls. Piekļuves tiesību izmantošanas piemērs

Klasifikācija

Viens no klasificēšanas veidiem ir resursa identifikatora sastāvdaļu izmantošana. Šāda veida klasificēšana nodrošina klasificēšanu pēc sistēmas, ražotāja, objekta klases un elementa. Ja tas nav pietiekami, tiek saglabāts klašu identifikators, kuram ieviest jauns atribūts *cls* (ang. *class*). Klašu identifikators tiek izmantots, ja resursa klasi nav iespējams definēt pēc resursa identifikatora. Piemēram, eksistē resurss, kuru izveidoja A sistēmas B ražotājs, kuram piešķirta C objekta klase un *AABBCC* unikālais numurs. Nepieciešams klasificēt resursu pēc D sistēmas D ražotāja, kurš definējis A1 objekta klasi tāda tipa resursiem.

Piemēra rezultātu skatīties P2.11. attēlā.

```
"epc":"A.B.C.AABBCC",  
"cls":"D.D.A1"
```

```
<epc>A.B.C.AABBCC</epc>  
<cls>D.D.A1</cls>
```

a) JSONR

b) RPML

P2.11. attēls. Klašu identifikatora izmantošanas piemērs

Otrais gadījums, kad nepieciešams kopā ar resursa klasificēšanu pievienot papildus atribūtus un noteikumus, kas atbilst jaunai klasei. Piemēram, ir kontainers, kurš nodots loģistikas kompānijai pārvietošanai no punkta A uz punktu B. Šim nolūkam ir nepieciešama papildus atribūtu *adrese* (ang. *address*), kurā saglabāta piegādes adrese. Pieņemsim, ka ir resurss, kuru izveidoja A sistēmas B ražotājs, kuram piešķirta C objekta klase un *AABBCC* unikālais numurs. Klasificējam resursu ar klasi *D.D.A1*, kurš papildina resursu ar jaunu *address* atribūtu (skat. P2.12. attēlu).

```
"epc":"A.B.C.AABBCC",  
"cls":{"address":"56.9515504,24.1133876"}
```

```
<epc>A.B.C.AABBCC</epc>  
<cls>  
  <address>56.9515504,24.1133876</address>  
</cls>
```

a) JSONR

b) RPML

P2.12. attēls. Piemērs resursa klasificēšanai ar klašu datu pievienošanu

Konfigurācija

Viens no konfigurēšanas atribūta izmantošanas veidiem ir fizisko objektu iekļaušana citā fiziskā objektā. Realizācija tiek nodrošināta, izmantojot hierarhisku struktūru. Kā analogiju apskatīsim faila sistēmas realizēšanu. Katrs fiziskais objekts ir datne. Ja fiziskais objekts iekļauj sevī citu fizisko objektu, tad šis fiziskais objekts kļūst par mapi. Mape ir nosaukta par *mezglu* (ang. *node*).

Saite starp diviem objektiem ir nodrošināta ar resursa identifikatora izmantošanu. Resursa identifikatora norādīšanai ir izmantots *epc* tags. Pieņemsim, ka eksistē kontainers (resursa identifikators *1.A89.1F.169DC0*), kurā iekšā ir palete (resursa identifikators *1.A89.2D.1634C*), uz paletes atrodas kaste (resursa identifikators *1.A89.16F.134334*) ar precēm (resursa identifikatori *2.A8333.18C.01*, *2.A8333.18C.02* un *2.A8333.18C.03*). *Iepriekš apskatītā* piemēra rezultātu skatīties P2.13. attēlā.

<pre> "epc":"1.A89.1F.169DC0", "node":{ "epc":"1.A89.2D.1634C", "node":{ "epc":"1.A89.16F.134334", "node":{"epc":["2.A8333.18C.01", "2.A8333.18C.02", "2.A8333.18C.03"]} } } } </pre>	<pre> <epc>1.A89.1F.169DC0</epc> <node> <epc>1.A89.2D.1634C</epc> <node> <epc>1.A89.16F.134334</epc> <node> <epc>2.A8333.18C.01</epc> <epc>2.A8333.18C.02</epc> <epc>2.A8333.18C.03</epc> </node> </node> </node> </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

a) JSONR

b) RPML

P2.13. attēls. Konfigurācijas izmantošanas piemērs

Prezentācija

Resursu prezentēšanai jā saglabā dažādu veidu dati tagā. Datus parasti daļa teksta datos un speciāla formāta datos. Teksta datus vienkārši pievieno tagam, cita veida datus nepieciešams glabāt teksta vidē, viena no populārākajām metodēm ir prezentēšana heksadecimālos kodos.

Prezentācijas atribūts sastāv no trim sastāvdaļām: *nosaukums* (ang. *name*), *avots* (ang. *source*) un *vērtība* (ang. *value*). Katrai sastāvdaļai ir paredzēta sava pazīme. Tipiski nosaukumi ir jāstandartizē, tad būs iespējams viennozīmīgi izmantot prezentācijas atribūtus, kā arī tiek definēti: resursa nosaukums, pamācība lietotājam, apraksts, logo, reklāma, garantijas, brīdinājumi utt. Teksta datiem eksistē dažādi kodējumi, citiem datu veidiem ir formāti. Ja dati tiek glabāti datnēs, tipisks veids datu formāta definējumam ir datnes paplašinājums. Izmantosim analogisku pieeju. *Avota* pazīmē tiek glabāts datu paplašinājums. Ar šo pieeju iespējams definēt augsta līmeņa formātu (grafisku, teksta utt.), papildu informāciju par izmantoto datu formātu iespējams glabāt pašos datos. Tas arī ir standarta datu glabāšanas mehānisms (piemēram, par unicode formātu skat. [UNI2012]).

Tiek definēts tags *shw* (ang. *show*). Piemēram, paņemsim vitamīnu kompleksa „*Vitabiotics Feroglobin B12*” resursu. Papildu nosaukumam birkā jāglabā resursa apraksts un logo (grafiskā datne).

Rezultāts ir atspoguļots P2.14. attēlā.

```
"epc": "09.34F.16F.1384D32"
"shw": {
  "name": "title", "source": "txt", "value": "Vitaviotics Feroglobin B12"},
  {
    "name": "description", "source": "txt", "value": "The Feroglobin-B12
    range of products is specially formulated to help maintain health and
    vitality with nutrients for the formation of haemoglobin for red blood
    cells. The range includes Feroglobin-B12 slow release capsules and
    delicious Feroglobin-12 Liquids."},
  {
    "name": "logo", "source": "BMP", "value": "454D761000000000000036
    0000002800000000a0000000a0000000A001800000000004001000000
    00000000000000000000000000000000FCFCFDFFFDFBDFBFS9642
    4D7601000000000000360000000280000000A0000000A0000000010
    01800000000004001000000000000000000000000000000000000FCFCFD
    FFFFDFBDFBFE964224D76010000000000003600000028000000a
    000000a0000000010018000000000400100000000000000000000
    0000000000FCFCFDFFFDFBDFBFE964224D76" }
  }
}
```

a) JSONR

```
<epc>09.34F.16F.1384D32</epc>
<shw name="title" source="txt">Vitaviotics Feroglobin
B12<shw>
<shw name="description" source="txt">The Feroglobin-B12
range of products is specially formulated to help maintain health
and vitality with nutrients for the formation of haemoglobin for
red blood cells. The range includes Feroglobin-B12 slow release
capsules and delicious Feroglobin-12 Liquids.<shw>
<shw name="logo"
source="BMP">454D76100000000000003600000028000000
a0000000a0000000A001800000000040010000000000000
0000000000000000000000FCFCFDFFFDFBDFBFS96424D7
601000000000003600000028000000A0000000A0000000
1001800000000040010000000000000000000000000000
FCFCFDFFFDFBDFBFE964224D7601000000000036000
00028000000a000000a000000001001800000000040010000
00000000000000000000000000000000FCFCFDFFFDFBDFBFE9
64224D76</shw>
```

b) RPML

P2.14. attēls. Prezentācijas atribūtu piemērs

3. pielikums

Resursu noteikumu tehniskā realizācija

DARSIR koncepcijā funkcionalitātes nodrošināšanas pamatelements ir noteikums. DARSIR dokumentam ir piedāvāti divi datu apmaiņas formāti RPML un JSONR. Katram formātam ir sava specifikācija, un katrs apakšnodaļas piemērs ir norādīts RPML un JSONR atsevišķi.

Procedūras un funkcijas

Tā kā visi nepieciešamie dati tiek glabāti vienā vietā (pašā resursā), tas nodrošina visu nepieciešamo informāciju lēmumu pieņemšanai uz vietas. Galvenais noteikumu veids ir *procedūras*, tās ir iepriekš definētas resursam aizpildāmās operācijas. Procedūras parasti izveido lietotājs, bet ir gadījumi, kad procedūras jau ir iebūvētas resursā (piemēram, ja resurss ir lampa, tad iebūvētās procedūras ir lampas ieslēgšana un atslēgšana.). Ja procedūra atgriež vērtību pēc sava darba pabeigšanas, tad to sauc par *funkciju*. Piemēram, funkcijas atgriešanas vērtībā iespējams ierakstīt 0, ja darbība veiksmīga tika izpildīta un cita vērtība norāda, ka ir kļūda, un numurs ir kļūdas kods.

Katrs atsevišķais noteikums ir prezentēts atsevišķā blokā ar nosaukumu *tsk*. Noteikumā jāprezentē sekojoši dati:

1. *Nosaukums* – tā ir noteikuma saite, pēc kuras iespējams atšķirt vienu nosaukumu no cita un izsaukt nepieciešamo noteikumu. Ja nosaukums ir *main*, tad noteikums tiks izpildīts uzreiz, kad atradīsies skenera uztveršanas zonā;
2. *Ieejas parametri* – tie ir mainīgie, kuri tiek doti noteikumam palaišanas brīdī. Nav iespējams nosaukt mainīgo ar nosaukumu, kas jau ir izmantots citam mainīgajam vai pazīmes nosaukumā (piemēram, fiziskā stāvokļa grupas pazīmes).
3. *Darbību kopa* – tā ir procedūru kopa, kura ir iepriekš norādīta un kura ir jāpalaiž pēc noteikuma izsaukšanas.
4. *Atgrieztā vērtība* – ja noteikums ir funkcija un pēc sava darba pabeigšanas atgriež vērtību izsauktajam noteikumam.

Pieņemsim, ka mums ir noteikums *switchOn*, kurš palaiž citu nosaukumu *switchLightOn*. Noteikums *switchOn* saņem divus ieejas parametrus *variable1* ar *integer* tipa vērtību un *variable2* ar *string* tipa vērtību. Šie ieejas parametri tiek nodoti *switchLightOn* noteikumam, mainīgie *variable1* un *variable2* (caur *var* pazīmi), un tiek saņemta atbilde *value1* (caur *result* pazīmi). Noteikums ir noformēts funkcijas veidā un atgriež *boolean* tipa vērtību *value1*. Rezultāts ir attēlots P3.1 un P3.2 attēlos.

```
"task":"switchon", "params":[{"variable1":"integer"}, {"variable2":"string"}],
"cmd":
[
{"run":{"func":"switchLighton", "params":["variable1", "variable2"], "result":"value1"}},
{"return":{"value1":"bool"}}
]
```

P3.1. attēls. Noteikuma *JSONR* piemērs

```
<task name="switchon" variable1="integer" variable2="string">
  <run var="variable1,variable2" result="value1">switchLighton</run>
  <return type="bool">value1</return>
</task>
```

P3.2. attēls. Noteikuma *RMPL* piemērs

Noteikumi

Noteikumu galvenais darbības princips ir organizēts pēc iepriekš definētas reakcijas (darbību izpildīšanas), pamatojoties uz konkrētu notikumu. Notikumu darbība tiek organizēta caur *trigeriem*, kuri sastāv no trim sastāvdaļām:

1. *Notikums* – ir operācija, kas notiek ar atribūtiem (atribūtu vērtības izmaiņa, atribūtu vērtības nozīmes saņemšana, atribūtu pievienošana, atribūtu dzēšana utt.) vai darbības, kas notiek ar resursa iekšējiem noteikumiem (procedūru palaišana, procedūru pievienošana, procedūru izmaiņa utt.).
2. *Nosacījums* – ir loģiska izteiksme, kura var pieņemt Bulu nozīmi (paties vai nepaties). Un ja Bulu vērtība ir patiesa, tad darbība tiek palaista. Ja netiek definēts nosacījums (nosacījums nav norādīts), tad tiek uzskatīts, ka Bulu nozīme ir patiesa un darbība jāpalaiž.
3. *Darbība* – tas ir iepriekš definēta operatoru kopa, kas tiek palaista, ja nosacījums ir paties.

Eksistē priekš definēti notikumi, kuri pēc noklusējuma nav definēti. Un sistēmas lietotājs var definēt savu darbību konkrētam notikumam. Resursa dzīves ciklā mainās atribūtu vērtības, un tipisks gadījums notikumu izmantošanai ir atribūtu vērtības maiņa. Uzskata, ka ir četras pamatdarbības, kuras iespējams veikt ar atribūtu: *pievienot* (ang. *insert*), *modificēt* (ang. *change*), regulēt piekļuves *tiesības* (ang. *rights*) un *dzēst* (ang. *delete*). Notikuma definēšanai atkarībā no atribūtu darbībām ir sekojoša sintakse: *event* + *darbības veids* (*insert*, *change*, *rights*, *delete*) + *atribūta nosaukums*.

Piemēram, eksistē kaste, kura tiek izmantota loģistikas procesā un ir nepieciešams saglabāt datumu un laiku, kad kaste pēdēja reizi tika pārvietota. Katru reizi, kad kaste maina savu pārvietošanās vietu, tiek atjaunota atribūtu vērtība, kas glabā GPS koordinātes. Iespējams definēt notikumu, kurš tiek palaists automātiski pēc GPS koordinātes atribūtu vērtības

izmaiņām un saglabā pašreizējo datumu un laiku. Viena no iespējamajām uzdevumu realizācijām ir attēlota P3.3. attēlā.

<pre>"epc": "09.34F.16F.1384D32", "gps": "56.9515504,24.1133876", "lasttime": "13333298630", "tst": "eventChangeGps", "params": [{"time": "integer"}, {"value": "string"}], "cmd": [{"save": {"name": "lasttime"}, {"value": "time"}}]</pre>	<pre><epc>09.34F.16F.1384D32</epc> <gps>56.9515504,24.1133876</gps> <lasttime>13333298630</lasttime> <tsk name="eventChangeGps" time="integer" newvalue="string"> <save name="lasttime">time</save> </tsk></pre>
a) JSONR	b) RPML

P3.3. attēls. Noteikuma realizācijas piemērs

Piemērā ir definēti trīs atribūti: *epc* (resursa identifikators), *gps* (GPS koordinātu glabāšanai) un *lasttime* (pēdējas pārvietošanas datums un laiks). Definēts notikums *eventChangeGps*, kad tiek palaistas pēc *gps* atribūtu jaunu vērtību izmaiņas. Notikuma ieejas parametri ir *time* (pašreizējais datums un laiks) un *value* (jauna atribūta nozīme). Ar *save* tagu ir nodrošināta atribūta *lasttime* izmaiņa uz *time* definēto nozīmi.

Vadības struktūras un iebūvētās operācijas

DARSIR koncepcijai ir iekšēja programmēšanas valoda, kas nodrošina funkcionalitātes iedarbināšanu. Jebkurai programmēšanas valodai (piemēram, Java [Eck2003]) nepieciešamas iekšējās vadības struktūras, vismaz nosacījuma un ciklu struktūras. Vēl ir nepieciešamas iebūvētās operācijas, kuras nodrošina galveno funkcionalitāti, kuru atbalsta jebkurš vadības elements.

Nosacījumu struktūras

Parasti nosacījumu struktūras sauc par *IF* struktūrām, tās ir pārbaudes struktūras, kurām ir nepieciešams nosacījums un darbību kopa, kura tiks palaista nosacījumu izpildīšanas vai neizpildīšanas gadījumā. Ar tagu *if* atzīmēsim tādas struktūras. Parasti (piemēram, Java, C, C++ utt.) nosacījumiem tiek izmantotas speciālas zīmes (piemēram, =, >, < utt.), kuru izmantošana ir ierobežota datu apmaiņas formātu gadījumā (piemēram, ar < zīmi sākas tagu definējums). Tādiem gadījumiem ir izvēlētas simbolu nosacījuma zīmes alfabēta kombinācijas apzīmējumi: *EQ* (=), *GT* (>), *LT* (<), *GE* (=>), *LE* (<=), *NE* (<>). Nosacījuma sintakse ir dota P3.4. attēlā.

<pre>"if": "condition": "value", {"true": [...]}, {"false": [...]}</pre>	<pre><if condition="value"> <true>..</true> <false>..</false> </if></pre>
a) JSONR	b) RPML

P3.4. attēls. Nosacījuma realizācijas piemērs

Sintakses paskaidrojums ir nākamais, ja nosacījums ir pareizs, tad darbības bloks, kas novietots tagā „*true*”, tiks izpildīts. Pretējā gadījumā taga bloks “*false*” varētu tikt izpildīts. Ja neeksistē tags “*true*” un “*false*”, tad tiek pieņemts, ka visas iekšējās darbības jāizpilda, ja nosacījums ir patiess.

Drīkst ierakstīt nosacījumus iekšā citā nosacījumā, ar šo atļauju iespējams organizēt selektīvas struktūras, parasti tās sauc par *switch* tipa nosacījuma struktūrām (piemēram, Java [Eck2003]). Tāda tipa vadības struktūras netika piemērotas koncepcijas realizācijas atvieglošanai. Ja lietotājam *if* tipa struktūras nav pietiekošas, tad iespējams organizēt *switch* realizāciju augsta līmeņa valodā un pēc transformācijas DARSIR funkcionalitātes nodrošināšanas valodā pārvietot uz *if* nosacījumu struktūru.

Ciklu struktūras

Ar ciklu struktūrām iespējams nodrošināt ciklisku darbību kopu izpildi, kur atkārtotības reižu skaits atkarīgs no nosacījuma. Eksistē dažādas metodes ciklu realizēšanai, tiek uzskatīts par pietiekošu *if* līdzīgas vadības struktūras realizācija. Cikla definējumam ir izmantots *while* tags, kur iekšā ir definēts nosacījums, kamēr nosacījums ir patiess, darbību kopa cikliski tiek izpildīta. Viens no variantiem, kā nodrošināt nosacījumu organizēšanu ciklu gadījumā, ir ciparu mainīgo izmantošana ar vērtības palielināšanu (*inc* operācija) vai samazināšanu (*dec* operācija). Ciklu struktūras piemērs ir attēlots P3.5. attēlā.

<pre>"i": "0", "while": "condition": "i NE 10", { "inc": "i" }</pre>	<pre><i>0</i> <while condition="i NE 10"> <inc>i</inc> </while></pre>
a) JSONR	b) RPML

P3.5. attēls. Ciklu realizācijas piemērs

Tiek definēts mainīgais *i*, kuram tiek piešķirta vērtība 0. Atsevišķā *while* blokā ir definēts nosacījums, ka ciklam jāstrādā līdz brīdim, kamēr *i* mainīgajam nav vērtības 10. Tika izpildīta operācijas kopa, un katras iterācijas beigās ar *inc* tagu tiek palielināts *i* mainīgais uz 1. Kad *i* mainīgajam būs vērtība 10, iterācijas tiek pabeigtas.

Iebūvētās operācijas

DARSIR funkcionalitātes nodrošināšanas valodai ir nepieciešamas iebūvētās komandas, kas ir standartizētas un kuras atbalsta visi darbības elementi (aktīvie resursi). Tas nozīmē, ka katram vadības elementam jāprogrammē šīs operācijas. Piemēram, tiek definēts, ka nepieciešams nodrošināt funkciju nosūtīt elektronisko pastu. Funkcijai tiek nodrošināta atgrieztā ciparu vērtība, ja nosūtīt elektronisko pastu izdevās, tad atgrieztā vērtība ir 0; ja nosūtīt elektronisko pastu nav iespējams kļūdas dēļ (piemēram, nav savienojuma), tad tiek

atgriezts kļūdas numurs; ja vispār šī funkcija netiek nodrošināta (piemēram, nav šīs funkcijas atbalsta), tad vērtība ir -1.

Iebūvētās operācijas ir definētas angļu valodā, faktiski tas ir programmēšanas starptautiskais standarts. Operāciju sintakse (iekavas ir pieņemtie parametri), kas paplašina DARSIR koncepcijas funkcionalitātes nodrošināšanas valodu:

- *Inc(value1, value2)* – funkcija atgriež mainīgā *value1* nozīmi palielinātu uz mainīgā *value2*, ja mainīgais *value2* nav norādīts, tad palielina mainīgo *value1* uz 1.
- *Dec(value1, value2)* – funkcija atgriež mainīgā *value1* nozīmi samazinātu uz mainīgā *value2*, ja mainīgais *value2* nav norādīts, tad samazina mainīgo *value1* uz 1.
- *Min(value1, value2)* – atgriež patieso vērtību (*true*), ja *value1* mazāka par *value2*;
- *Max(value1, value2)* – atgriež patieso vērtību (*true*), ja *value1* lielāka par *value2*;
- Utt.

Operāciju sintakse (iekavas ir pieņemtie parametri), kas standartizē tipiskas izpildīšanas operācijas:

- *LocaleMin(objekts, attālums)* – atgriež patieso vērtību (*true*), ja *objekta* atrašanās vieta ir tuvāk par norādīto *attālumu* (vērtība norādīta metros);
- *LocaleMax(objekts, attālums)* – atgriež patieso vērtību (*true*), ja *objekta* atrašanās vieta ir tālāk par norādīto *attālumu* (vērtība norādīta metros);
- *Open(id)* – nosūtīt resursam ar identifikatoru *id* komandu „atvērt” ;
- *Close(id)* – nosūtīt resursam ar identifikatoru *id* komandu „aizvērt” ;
- *On(id)* – nosūtīt resursam ar identifikatoru *id* komandu „ieslēgt” ;
- *Off(id)* – nosūtīt resursam ar identifikatoru *id* komandu „izslēgt” ;
- *SendMail(adrese, ziņojums)* – nosūtīt tekstu *ziņojums* uz *adrese* elektronisko pastu;
- Utt.

4. pielikums

Vizuālā interaktīvā simulācija (VIS)

Viena no iespējām, kura ļauj parastam lietotājam intuitīvi un viegli izveidot jaunu IT risinājumu, ir simulācijas rīku izmantošana. Tomēr, veidojot IT risinājumus, parasti tiek izmantoti programmēšanas rīki. Tiek piedāvāts IT risinājumu izveidot no jau eksistējošas risinājuma simulācijas. Viena no metodēm, kā iespējams to nodrošināt, ir simulācijas rīku izveidošana, kuri tiek adaptēti konkrētai DVS. Pamatojoties uz programmatūras IS attīstības 4. pamatatendenci *reālās pasaules modelēšana* (skat. 1.1. apakšnodaļā, kur ir rekomendēts izmantot lietotāju grafiskās saskarnes), tika izvēlēta *vizuālā interaktīvā simulācija (VIS)*, kuras galvenie principi tika apkopoti [Bel1987] 1987. gadā. VIS tika izvēlēta kā viena no progresīvām simulācijas tendencēm. Tas ir redzams no pētījumiem, kas izmanto vizuālus interaktīvus modeļus [Saw2008, Bel1999].

VIS ir datu vizualizācijas rīku integrācija un interaktīvā resursu regulēšana. Lietotājs var apskatīt starprezultātus un modificēt modeli eksperimenta laikā. Modeļa parametri un mainīgie, kā arī ar vizuālās prezentācijas palīdzību, var tikt modificēti eksperimenta laikā ar tūlītēju ietekmi uz simulācijas procesu [Wag1996].

1990. gadā [Mar1990] tika definētas trīs vizualizācijas un mijiedarbības klases: *pēcapstrāde* (ang. *post-processing*), *kontrolē* (ang. *tracking*) un *vadība* (*steering*).

Tālāk seko katras klases definīcija:

- *Pēcapstrāde* – vides simulācija, kuras darbības laikā lietotāji nevar sekot eksperimenta gaitai un modificēt to. Eksperimenta darbības rezultāts tiek glabāts izejas datu veidā datubāzē. Pēc tam dati tiek izgūti no datubāzes, izmantojot vizuālo rīku. Galvenā izmantošanas sfēra ir pēcapstrādes analīze. Piemēram, CINEMA sistēma [Poo1990].
- *Kontrolē* – simulācijas vide, kuras darbības laikā tūlītējie rezultāti var tikt apskatīti. Lietotājs var sekot eksperimentam, izmantojot dažādus vizualizācijas instrumentus: animāciju, diagrammas, vairākus logus ar starpstāvokļiem, pārvietošanos laikā eksperimenta gaitā utt. Parasti lietotājs definē laika periodu, kurā eksperiments tiek veikts. Viņš arī definē mainīgos, kuriem vēlas sekot. Galvenais ierobežojums ir simulācijas modeļa maiņa. Piemēram, GPVSS sistēma [Bis1990].
- *Vadība* – simulācijas vide, kura ļauj lietotājam mainīt simulācijas modeli eksperimenta laikā. Šī simulācija [Roo1993] var tikt izmantota kā piemērs. Parasti vadības simulācijas vide tiek saukta par vizuālo interaktīvo simulācijas vidi.

Tiek uzskatīts, ka mūsu gadījumā piemērotākais variants ir kontroles simulācijas vide. Izmantojot kontroles simulācijas vidi, būs iespēja realizēt parastu un ātru IT risinājumu veidošanu.

IT risinājumu un simulācijas darbības principi

Pirmais solis simulācijas izmantošanai risinājuma veidošanai ir koncepcijas atslēgas elementu definēšana. DARSIR koncepcijas atslēgas elementi ir resursi. Resurss ir jebkurš dzīvs vai nedzīvs dabas objekts, kas ir iesaistīts informācijas sistēmas darbības procesā. Visi konkrētā resursa dati un to attiecības ar citiem objektiem (vai objektu tipiem) ir jāglabā resursā. 4.3.3. apakšnodaļā ir secināts, ka vispārīgos gadījumos DARSIR koncepcijai vispiemērotākā ir RFID tehnoloģija. Šajā apakšnodaļā visi piemēri doti, izmantojot RFID tehnoloģiju.

Nākamais solis procesā ir koncepcijas darbības principu definēšana un simulācijas izmantošana jauna reāla risinājuma veidošanā. DARSIR koncepcijas darbības procesā atslēgas elements ir aktīvais resurss (RFID lasītājs). Aktīvajā resursā ir iebūvēta iegultā sistēma, kas ielādē un izpilda noteikumus, kuri ir aprakstīti resursos (RFID tags). Šiem mērķiem resursiem ir jābūt aktīvajā resursā operatīvajā apgabalā. Ja aktīvie resursi ir savienoti aktīvo resursu tīklā, tad aktīvais resurss realizē resursu informācijas (dati un noteikumi) apmaiņu.

Resursi (RFID tagi) ir nozīmīgi reālajam risinājumam un simulācijai tāpēc, ka strādājoša procesa izveidošanai ir nepieciešams mainīt informāciju (atribūtus un noteikumus) tikai resursos (RFID tagos). Citiem strādājošiem procesa elementiem (tādi kā RFID lasītāji) parasti nav vajadzīgas nekādas modifikācijas. Lai izmantotu simulāciju IT risinājuma izveidošanai, nepieciešams, lai atbilstoša informācija (katra resursa dati un noteikumi) tiktu ievadīta.

Datu saglabāšana

Nepieciešams definēt veidu, kā saglabāt informāciju (dati un noteikumi) simulācijā. Iepriekšējās nodaļās tiek definēts, ka DARSIR koncepcija saglabā visu informāciju decentralizēti atbilstošos resursos (RFID tags). Un ir nepieciešams visu nepieciešamo informāciju IT risinājuma izveidei ieviest simulācijas shēmā. Nav iespējams izveidot IT risinājumu no simulācijas shēmas, ja šis nosacījums netiek izpildīts.

DARSIR koncepcijai par datu glabāšanas formātu tiek izmantoti RPML un JSONR standarti. IT risinājuma izveidošanai nepieciešams transformēt simulācijas shēmu uz atsevišķiem RPML vai JSONR dokumentiem, katram simulācijas shēmas resursam.

Piedāvātās pieejas izmantošanas lietderība

Pastāv daudzi simulācijas rīki [Bel1987, Mar1990, Poo1990, Bis1990, Roo1993, Wag1996, Bel1999, Saw2008]. Lai nodrošinātu simulācijas izmantošanu risinājumu

veidošanu, DARSIR simulācijas rīkā tika iebūvēts IT risinājumu veidošanas modulis. Šis modulis veido IT risinājumu vairāku RPML vai JSONR dokumentu veidā. Simulācijas rīks ir izveidots, izmantojot DARSIR koncepcijas uzbūves principus.

Kad jau eksistējošais rīks tiek adaptēts, nepieciešams izveidot transformācijas moduli, kurš konvertē simulāciju uz IT risinājumu. Katra simulācijas rīka un datubāzu vadības sistēmu kombinācijai ir nepieciešams papildus pētījums. Katram konkrētam pētījumam ir nepieciešams definēt datus un noteikumus. Šie dati un noteikumi pēc tam ir jāpievieno simulācijas rīkam, lai izveidotu IT risinājumu. Pieņemts, ka piedāvātās pieejas izmantošanas lietderība ir jādefinē katram konkrētam gadījumam atsevišķi.

Lietderība tiek definētā sekojošā veidā:

- Simulācijas rīkam ir jāatkārto datubāzes vadības sistēmas darbības principi;
- Simulācijas detalizācijas līmenis ir pietiekams, lai aprakstītu IT risinājumu, neizmantojot papildus;
- Simulācijas rīks ļauj pievienot papildus datu un noteikumu tipus;
- Simulācijas rīks ir izbūvēts pēc kontroles simulācijas vides principiem.

Simulācijas izveidošana no eksistējošas informācijas sistēmas

Dažreiz lietotājiem ir situācijas, kad viņiem ir gatavs reāls risinājums un nepieciešams veikt dažas izmaiņas šim risinājumam. Šai funkcionalitātei ar simulācijas izmantošanu ir iespējamas sekojošas situācijas un iespējamie risinājumi:

1. *Reālais risinājums tika izveidots DARSIR koncepcijas ietvaros* – vislabākais risinājums šajā gadījumā ir pārvietot visus atribūtus un noteikumus no resursiem uz simulācijas shēmu;
2. *Reālais risinājums visiem elementiem ar iebūvētām RFID birkām* – šajā situācijā ir iespējams ielādēt daļu no atribūtiem simulācijas shēmā kā fiziskus objektus (ja pastāv RFID identifikācijas īpašnieka tipa apraksts), sensorus (ja pastāv), entītijas (ja tā ir RFID tagā), vietu (iespējams definēt RFID lasītājā) utt.;
3. *Reālais risinājums visiem elementiem bez iebūvētām RFID birkām* – šajā situācijā pirmais solis ir iebūvētie RFID lasītāji. Ir iespējams ielādēt daļu no atribūtiem simulācijas shēmā (bet parasti mazāk nekā 2. risinājumā), kā fiziskus objektus (ja pastāv RFID identifikācijas īpašnieka tipa apraksts), sensorus (ja pastāv), vietu (iespējams definēt RFID lasītājā), konfigurāciju (iespējams definēt RFID lasītājā) utt.

Praktiskā simulācijas rīka realizācija

Balstoties uz šīs nodaļas teorētisko pamatojumu, ir izveidots „*DARSIR simulation tool*” simulācijas rīks. Detalizēta informācija par simulācijas rīku un tā praktisku pielietojumu ir atspoguļota [Zur2008c, Zur2011c] ziņojumos.

5. pielikums

Koncepcijas pārbaude ar tipisku uzdevumu

Šajā apakšnodaļā tiek realizēta koncepcijas pārbaude no uzdevuma apskata līdz simulācijas DARSIR projekta izveidošanai. 1.4.1.2. apakšnodaļā tiek apskatīti tīrīšanas, remonta un apkopes servisi, kas ir viens no objekta individuālo īpašību sistēmas uzdevumiem. Tiek izvēlēta viena no šo servisu realizācijām „*Viedā veļas mazgātava*”, uz šo uzdevumu bāzes tiek nodrošināta DARSIR koncepcijas aprobācija.

Lai parādītu, ka šī uzdevuma izpildīšanai nepieciešama DARSIR koncepcija, šajā apakšnodaļā tiek nodrošināts: uzdevuma apraksts, aprakstīti pašreizējie risinājumi un standarti šajā nozarē, tiek doti iespējamie tehnoloģiskie risinājumi un ir aprakstītas jaunas koncepcijas izmantošanas priekšrocības.

Sfēras apskats un pētījumu virzieni

“*Viedā veļas mazgātava*” – tā ir intelektuālas automatizētas veļas mazgātavas servisa prasības apgērba tīrīšanai, kas sastāv no mazgāšanas, žāvēšanas, gludināšanas un profesionālas auduma kopšanas. Galvenā šīs sistēmas prasība ir minimizēt cilvēka kļūdas, kas varētu sabojāt apgērba veļas mazgāšanas procesā. Tehniskās prasības nodrošina maksimāli atļauto auduma apstrādi un uzturēšanas norādījumiem atbilstošu kopšanu.

Veļas mazgātavas servisu darbību var sadalīt divās darbību grupās: sagatavošana un servisa pakalpojumu veidošana. Pamata darbība, kas jāautomatizē, ir sagatavošanas process, kas sastāv no:

1. *Apgērba īpašību saņemšana* – svarīgs etaps ir apgērba analīze, šāda informācija ir būtiska nākamo punktu pareizai izvēlei. Parasti to dara cilvēks.
2. *Nepieciešamās darbības izvēle* –katrai apgērba vienībai ir jāizvēlas konkrēta darbība vai darbību kopa, kas jāizpilda. Mazgāšanas procesa darbības ir mazgāšana, balināšana, žāvēšana, gludināšana un profesionāla auduma kopšana. Darbību grupas nepieciešams izvēlēties lietotājam, bet sistēmai jāpārbauda, vai ir iespējams nodrošināt darbību izpildi konkrētai apgērba vienībai un ar kādu režīmu. Piemēram, ir apgērba audumu veidi, kuriem nav paredzēta mazgāšana, bet ir iespējams nodrošināt šo darbību, izmantojot profesionālu auduma kopšanu;
3. *Ķīmisko līdzekļu izvēle* – pamatojoties uz apgērba īpašībām, ar darbības izvēli nepieciešams nodrošināt pareizo ķīmisko līdzekļu izvēli;
4. *Režīma izvēle* – pamatoties uz visiem iepriekš minētajiem punktiem, jānodrošina katrai darbībai pareiza darbības režīma izvēle.

Lai realizētu “*Viedās veļas mazgātavas*” uzdevumu, vispirms tiks apskatīti standarti un metodes, kas tiek izmantotas šajā nozarē. Vispirms ir veļas mazgāšanas standarta simboli, lai definētu tipiskus lēmumus, kuri tiek izmantoti dotajā brīdī, lai palīdzētu lietotājam izvēlēties pareizus darbības režīmus.

Tālāk tiek apskatīti mazgāšanas šķidrums standarti, lai zinātu kādas vēl sastāvdaļas tiek izmantotas veļas mazgāšanas darbā. Pēc tam tiek aprakstīti galvenie parametri, kuri jāizvēlas “*Viedās veļas mazgātavas*” vadības blokam, lai nodrošinātu pareizu mazgāšanas režīmu.

Veļas mazgātavas simbolu standarti

Veļas mazgāšanas simboli [Gen2009], vēl saucami par kopšanas simboliem, tās ir piktogrammas, kas simbolizē veļas mazgāšanu, žāvēšanu, sauso tīrīšanu un gludināšanu.

Simboli tiek rakstīti uz etiķetēm, kas tiek pievienotas apģērbiem, lai detalizēti norādītu, kā vislabāk tīrīt apģērbus. Kopšanas etiķetēm ir dažādi standarti dažādās valstīs / reģionos pasaulē. Dažādās standartu specifikācijās ir aprakstīts, ka piktogrammas tiek dublētas vai papildinātas ar rakstisku instrukciju.

Eiropā visvairāk ir noderīgs standarts, kas pārstāv *Internacionālo auduma kopšanas marķēšanas asociāciju* [Gen2009] (ang. *Groupement International d'Etiquetage pour l'Entretien des Textiles, GINETEX*), kas pārstāv auduma izgatavotājus un mazumtirdzniecības nozares un citas ieinteresētas organizācijas, kas ir nodarbinātas auduma kopšanā, dažādās valstīs izgudrota piemērota kopšanas marķēšanas sistēma audumiem, kur izmanto simbolus un piktogrammas.

Būtiski tehniski elementi to izpildīšanai ietverti tehniskajos noteikumos. Tiek iedalītas piecas pamata darbības, kurās tiek standartizētas veļas mazgāšanas darbības robežas:

1. *Mazgāšana* – simbols norāda, kāda mazgāšana ir iespējama un izmantojama mazgāšanai gan ar rokām, gan ar mašīnu.
2. *Balināšana* – simbols norāda - drīkst vai nedrīkst veļu balināt.
3. *Žāvēšana* – simbols, kas stingri norāda žāvēšanas apstrādes temperatūru, mazgāšanas un gludināšanas simboli dod rekomendācijas žāvēšanai.
4. *Gludināšana* – simbols regulē gludināšanas temperatūru.
5. *Profesionāla auduma kopšana* – piktogrammas regulē sausās tīrīšanas un mitrās tīrīšanas procesus audumam, kas attiecas uz profesionālu auduma kopšanu.

Visiem tehniskajiem noteikumiem nav tipisku simbolu, kuri ir rakstīti uz apģērba kā instrukcijas. Piemēram, ir tādas etiķetes ar nestandarta veida noteikumiem, kuras apraksta

sekojošo instrukciju: „izgriezt veļu, kamēr slapja / mazgāt kopā ar tādām pašām vai līdzīgām krāsām/ izgriezt uz āru pirms mazgāšanas un gludināšanas / sausā žāvēšana”.

Mazgāšanas līdzekļu standarti

Veļas mazgāšanas procesā tiek lietoti dažāda veida šķīdumi un veļas pulveri, balināšanas līdzekļi, mazgāšanas līdzekļi, veļas mīkstinātāji. Katrs mazgāšanas šķidrums un pulveris lietojams saskaņā ar lietošanas noteikumiem. Dažreiz tas ir stingri noteikts, bet ir gadījumi, kad netiek definēts. Kā arī bieži visi simboli un attēli dublējami angļu valodā vai / un vietējā valodā. Viens pārdevējs savam produktam var pielietot vienas iespējas, cits pārdevējs var pielietot citas iespējas.

Mazgāšanas parametri

Pirmā darbība, kura jāveic “*Viedās veļas mazgātavas*” uzdevuma robežās ir “*Mazgāšanas tips*”. Pamatojoties uz fiziskajiem apģērba parametriem, nepieciešams nodrošināt pareizu “*Mazgāšanas tipa*” izvēli. “*Mazgāšanas tips*” nodrošina sekojošu parametru izvēli:

- *Iepildsvars* – tas ir apģērba kopējais svars vienam mazgāšanas ciklam, tipiskas pazīmes: mazs, vidējs, liels, ļoti liels;
- *Mazgāšanas temperatūra* – tā ir temperatūra, kura izmantota mazgāšanas ciklam, tipiskas pazīmes: auksta, karsta, ļoti karsta;
- *Skalošanas temperatūra* – tā ir temperatūra, kura izmantota skalošanas ciklam, tipiskas pazīmes: auksta, karsta, ļoti karsta;
- *Mazgāšanas tips* – veļas mašīnas funkcija atkarībā no apģērba tipa un netīrības pakāpes, tipiskas pazīmes: smalkvilna, vilna, parasta, intensīva;
- *Mazgāšanas ilgums* – vienas mazgāšanas cikla ilgums.

Arhitektūras risinājums

DARSIR koncepcijas pamatelements ir resurss, tas ir jebkurš fizisks objekts, kurš tiek izmantots risinājumā. Definēsim risinājuma resursus:

- *Veļa* – tas ir galvenais elements, ar kuru iespējams piemērot mazgāšanas servisa pamata darbības (mazgāšana, balināšana, žāvēšana, gludināšana un profesionāla auduma kopšana).
- *Mazgāšanas līdzekļi* – tie ir dažāda veida līdzekļi, kas nepieciešami mazgāšanas servisa darbības ciklā. Šos līdzekļus var sadalīt divās grupās: šķidrums (piemēram, balinātājs) un vielas (piemēram, mazgāšanas pulveris).

- *Servisa nodrošinājuma iekārtas* – tās ir mazgāšanas servisa iekārtas, kas realizē servisa iedarbināšanu ar manipulācijām ar veļas resursu. Tipiskas iekārtas ir: veļas mazgājamā mašīna, veļas žāvēšanas mašīna utt.

Pamata resurss ir *veļa*, kurai jānodrošina *mazgāšanas serviss*. Veļā ir iebūvēta birka, kur tika glabāta informācija par auduma parametriem (piemēram, tips, svars, krāsa utt.), nepieciešamās darbības, kas jāveic ar veļu (mazgāšana, balināšana, žāvēšana, gludināšana un profesionāla auduma kopšana) un noteikumi, kas attiecas uz veļu un neietilpst standarta mazgāšanas ciklā. Visiem servisa resursiem (servisa nodrošinājuma iekārtām un mazgāšanas līdzekļiem) arī ir pievienotas birkas ar aprakstiem.

Pamata darbības resurss ir servisa nodrošinājuma iekārtas, tās tiek iedarbinātas, kad veļa tiek ievietota konkrētā iekārtā. Pieņemsim, ka ir iekārta (veļas mazgājamā mašīna) ar veļu iekšā. Iekārta saņem atribūtus un noteikumus no veļas, analizē informāciju un palaiž iekārtas noteikumus. Pēc lēmumu pieņemšanas notiek pārbaude ar veļas noteikumiem, vai nav nekādu pārkāpumu izvēlētajiem servisa parametriem. Pēc tam notiek mazgāšanas līdzekļu izvēle un tiek piemēroti noteikumi izvēlētajiem mazgāšanas līdzekļiem. Tiek izvēlēts optimālais darbības režīms un notiek veļas apstrāde. Ja ir iestrādāti ierobežojumi, ja iekārta var detektēt, kā labot situāciju un palaist darbības ciklu, tā izvēlas optimālu režīmu, pretējā gadījumā veļas īpašniekam vai mazgāšanas servisa darbiniekam tiek piedāvāti iespējamie varianti par nākamajiem soļiem atkarībā no servisa iekšējām instrukcijām.

Tipisks algoritms veļas analīzei un piemērotāko darbības režīmu izvēlei:

1. Pārbaudīt, vai visus resursus, kas atrodas veļas mašīnā, ir iespējams izmazgāt (parametrs *Washable*), ja nav, tad parādās kļūdas paziņojums ar tekstu: „*Not all items in the washing machine are washable*”.
2. Pārbaudīt, vai visu resursu, kas atrodas veļas mašīnā, svaru (parametrs *weight*) summa nav lielākā par veļas mašīnas maksimālo pieļaujamo svaru. Ja svars ir pārsniegts, tad parādās kļūdas paziņojums ar tekstu: „*Washing machine is overloaded*”.
3. Pārbaudīt, vai resursu materiālus var mazgāt kopā. Ja ne, tad parādās kļūdas paziņojums ar tekstu: „*Those materials shouldn't be washed together*”.
4. Izvēlēties mazgāšanas temperatūru.
5. Izvēlēties apgriezību intensitāti.
6. Izpildīt mazgāšanu.
7. Resursu, kas izmazgājās, vēsture parādās atskaitē par to, kas bija izmazgāts.

Nākamais solis ir izvēlēties tehnoloģiju, kuras pamatā būs “Viedās veļas mazgātavas” uzdevumu realizācija. Iepriekš definēts, ka birkas jāpievieno veļai, tas nozīmē, ka ir būtisks tehnoloģisks ierobežojums, nepieciešama birka, kas ļautu darboties netipiskos darba apstākļos (ūdenī, augstās temperatūrās, utt.). Mūsdienās pastāv divas tehnoloģijas, kas ir tuvas šai funkcionalitātei - tie ir *svītrkods* un *RFID* tehnoloģija.

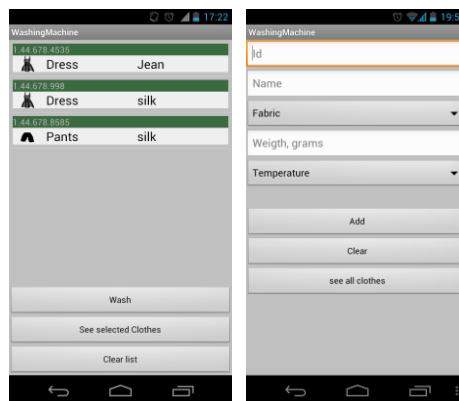
Šī uzdevuma izpildei ieteicams izvēlēties RFID tehnoloģiju, jo svītrkoda tehnoloģijai šī uzdevuma izpildei ir būtiski ierobežojumi: svītrkoda birku viegli sabojāt, nepieciešama birkas redzamība datu lasīšanas procesā, nav iespējams modificēt datus birkā utt.

DARSIR projekta praktiskās realizācijas apraksts

Tika sagatavota pamata informācija veļas mazgāšanas risinājumu praktiskai realizēšanai. Risinājumu izveidošanai eksistē „DARSIR simulation tool” simulācijas rīks (detalizētāk skat. 4. pielikumā). Viedās mazgāšanas risinājums „DARSIR simulation tool” simulācijas rīkā ir attēlots P5.1.a attēlā.



a) Simulācijas rīks



b) Mobilā lietotne

P5.1. attēls. Viedās mazgāšanas mašīnas risinājums

Pamatojoties uz šī pielikuma arhitektūras risinājuma aprakstu, ir izveidots DARSIR risinājums. „DARSIR simulation tool” simulācijas rīkam ir izveidoti trīs veidu objekti:

1. *Veļa* – tas ir pasīvais resurss, pamatelements, ar kuru iespējams piemērot mazgāšanas servisa pamata darbības. Resursam ir ierakstīti atribūti, kas apraksta veļas īpašības (piemēram, krāsa, svars, materiāls utt.).
2. *Mazgāšanas līdzekļi* – tas ir aktīvais statiskais resurss, līdzekļi, kas nepieciešami mazgāšanas servisa darbības ciklā. Tāda tipa resursam ir ierakstīti iekšēji noteikumi, kas apraksta mazgāšanas līdzekļu pieļaujamās un aizliegtās darbības. Piemēram, balinātāju iespējams izmantot tikai baltajai veļai.
3. *Veļas mazgājamā mašīna* – servisa nodrošinājuma iekārta, aktīvais dinamiskais resurss, kas ir paredzēts veļas mazgājamajai mašīnai. Šim resursam ir pievienoti

noteikumi, bāzēti uz veļas mazgājamās mašīnas darbības algoritmu. Noteikumu galvenais mērķis ir izvēlēties pareizu veļas mazgāšanas režīmu.

Viedās mazgāšanas risinājumam ir izveidoti divdesmit resursi „*DARSIR simulation tool*” simulācijas rīkā (skat. P5.1.a. attēlā). Veļas tipa resursiem ir ierakstītas veļas īpašības. Mazgāšanas līdzekļiem un veļas mazgājamajai mašīnai ir ierakstītas īpašības un noteikumi. Simulācijas rīka izveidotais risinājums ir pārnests uz mobilās iekārtas aplikāciju (skat. P5.1.b. attēlā).

Koncepcijas pārbaude ar netipisku uzdevumu

DARSIR koncepcija ir piemērota risinājumiem, kas attiecas uz fizisko objektu mijiedarbības organizēšanu, un atbilst OIĪSS kritērijiem. Atkarībā no specifisku uzdevumu risinājumam DARSIR koncepcija var paplašināties ar jaunu funkcionalitāti, piemēram, ar jaunām atribūtu un noteikumu grupām. Pieņemts, ka nepieciešams realizēt uzdevumu par ārštata pakalpojumu pārvaldības sistēmas uzbūvi.

Sfēras apskats un pētījumu virzieni

Mūsdienās eksistē tendence, ka darbinieki strādā neatkarīgi no kompānijām, viņi ir pašnodarbinātas personas. Piemēram [Hof2002] pētījumā ir norādīts, ka 40% no visiem Vācijas IT darbiniekiem nav tipisku darbinieks-darba devējs attiecību.

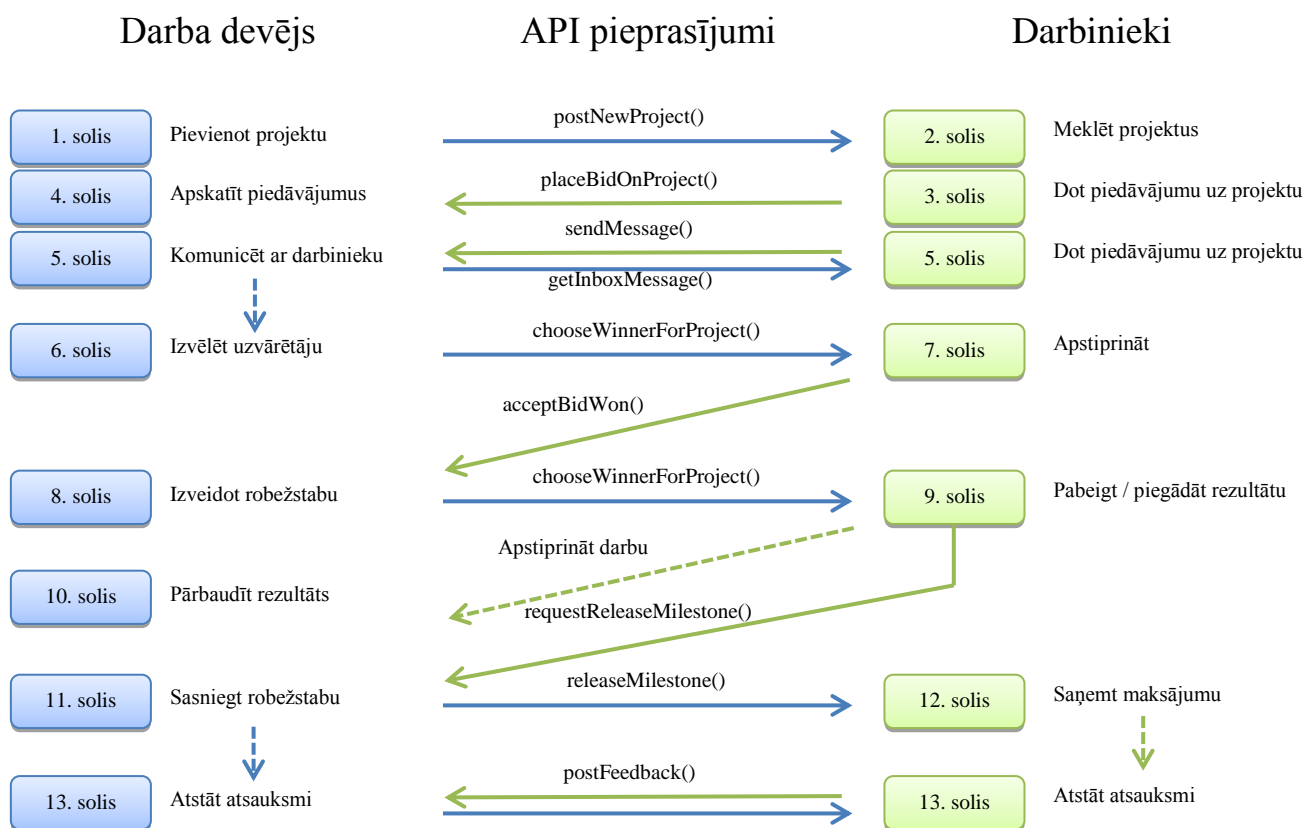
Parasti pašnodarbinātas personas sauc par *ārštata darbiniekiem* (ang. *freelancer*). Terminam „*ārštata darbinieks*” nav stingra definējuma, šis definējums atkarīgs no likumdošanas un ekonomiskajiem definējumiem, kas katrai valstij ir specifiski. Šim terminam un citiem saistītajiem terminiem nepieciešami stingri definēti formulējumi:

- *Ārštata pakalpojuma pārvaldības sistēmas (APPS)* – tās ir sistēmas, kas nodrošina mijiedarbību starp ārštata darbiniekiem un darba devējiem. Ārštata darbinieks pievieno sistēmai informāciju par savām spējām un kvalifikāciju. Darba devējs pievieno darba piedāvājumus.
- *Ārštata darbinieks (darbinieks)* – ir pašnodarbināta persona, kura izpilda jebkura tipa projektus īsā laikā.
- *Darba devējs (klients)* – tas ir APPS dalībnieks, kas reģistrē darba piedāvājumu APPS'ā.
- *Projekta pieteikums (PP)* – tas ir ieviestais APPS darba piedāvājumu apraksts, kuru izpilda darba devējs. Parasti tiek ieviests izsoles laiks (cik ilgi gaidīt darbinieku pieteikšanos, maksa par projektu (samaksa par stundu vai par darba apjomu) un detalizēts darba apraksts brīvā formā.
- *Atribūts* – tās ir īpašības, kas apraksta ārštata darbinieku, darba devēju vai projektu. Šīs īpašības ir publiski pieejamas un parasti apraksta ārštata darbinieka spējas un projekta prasības.
- *Uzvedības modelis* – tie ir uzvedības noteikumi, pēc kuriem strādā jebkurš APPS dalībnieks, šie noteikumi nav pieejami citiem APPS dalībniekiem.
- *Profils* – darbiniekiem un darba devējiem.

Detalizēti tika apskatītas dažādas APPS [Gur2011,Free2011a,Free2011b], šīs sistēmas pamatprincips ir ārštata darbinieku un darbu piedāvājumu datu centralizēta pārvaldība. APPS pamatā strādā trīs atribūtu grupas (detalizētāk skat. [Hes2010]): maksa, darba stundas un saistības (organizācijas, īpašnieks utt.). Atsevišķos pētījumos [Sta2003, Sto2005, Suß2010] ir izdalītas trīs galvenās sfēras, kur ārštata darbinieku ir visvairāk: žurnālistika (prese, radio, televīzija un Interneta vide), konsultāciju kompānijas un IT industrija.

Katram APPS dalībniekam ir jāveic daudz roku darba. Process sākas ar PP izveidošanu, ko veic darba devējs. Kad PP ir nopublicēts, uz APPS piesakās darbinieki. Laiku, kad PP ir aktīvs un darbinieki var veikt savus *cenu piedāvājumus* (ang. *bid*), nosauc par izsoli. Darba devējs kontrolē procesu, apskatot darbinieku profilus un beidzot izvēloties PP izpildītāju. Parasti darbinieks vienlaicīgi seko līdž vairākiem PP un izvēlas sev piemērotāko.

Mūsdienas APPS tiek raksturotas ar nelielu atribūtu veidu skaitu un mazām automatizācijas iespējām. Šajā situācijā APPS darbības process ir paredzēts lieliem un vidējiem PP, ar nelielu projektu skaitu datubāzē. Gadījumos, kad nepieciešams izvēlēties daudzus nelielus darbus, tas ir grūti izpildāms vienam darbiniekam. Turklāt ir aktuāls datu drošības un konfidencialitātes jautājums, centralizācijas gadījumā visi dati tiek glabāti centralizēti pie APPS servisa piegādātāja.



6.1. attēls. API darbības shēma no [Fre2011a], starp darbinieku un darba dēvēju

Tomēr jau tagad eksistē mehānismi, caur kuriem iespējams nodrošināt automatizāciju APPS procesam. Daļu no automatizācijas iespējams realizēt ar *aplikācijas programmēšanas interfeisu* (ang. *Application Programming Interface, API*), kurš jau tagad ir realizēts APPS. Piemēram, API no [Fre2011a] starp darbinieku un darba devēju ir attēloti P6.1. attēlā.

Izmantojot *Daudzaģentu sistēmas* (ang. *Multi Agent System, MAS*) uzbūves principus, ir iespējams definēt katru APPS dalībnieku kā atsevišķu aģentu. To mijiedarbības automatizāciju iespējams realizēt, izmantojot API. Daudzi neatkarīgi objekti, kuri saņem informāciju no ārējās vides (APPS), strādā autonomi un pieņem lēmumus patstāvīgi. MAS ir aprakstīts 1.3.1.2. apakšnodaļā.

Pētījumā ir izvirzīta hipotēze, ka APPS izmantotās atribūtu grupas (detalizētāk skat. [Hes2010]): maksa, darba stundas un saistības (organizācijas, īpašnieks utt.) nepieciešams papildināt, kas dos iespēju paaugstināt APPS efektivitāti. Ja APPS strādā efektīvi, tas nozīmē, ka darbinieks var labāk novērtēt projekta izmaksas un nodrošināt mazāku maksu par projektu, kas savukārt ir izdevīgi darba dēvējam.

MAS principu izmantošana APPS uzbūvei

Tiek piedāvāts piemērot MAS modeļa principus APPS uzbūvei, kad katrs APPS dalībnieks (darba devējs un darbinieks) ir atsevišķs aģents. APPS API ir izmantoti, lai nodrošinātu mijiedarbību starp APPS (projektu datubāzi), darba devējiem un darbiniekiem. Katram aģentam ir savi atribūti un uzvedības modeļi (noteikumi), kas tiek saglabāti pašā aģentā ar DARSIR koncepcijas palīdzību. Tas nozīmē, ka iekšējie dalībnieku atribūti un noteikumi nav pieejami citiem APPS dalībniekiem un tos var izmantot tikai pats dalībnieks. Atkarībā no uzvedības modeļa realizācijas iespējams realizēt pilnīgi patstāvīgu aģenta darbību, vai pusautomātisku, kad dalībnieks tiek iesaistīts procesā tikai rezultātu apstiprināšanas stadijā.

Tiek uzskatīts, ka uzvedības modeļiem jābūt dinamiskiem, kas maina savu funkcionalitāti kopā ar dalībnieka pieredzi. Izmaiņas uzvedības modeļos tiek ieviestas ar dalībnieku manuālo izmaiņu palīdzību vai automātiski, analizējot informāciju no vēstures.

Darbinieka uzdevums ir izvēlēties piemērotākos projektus no projektu piedāvājuma, bet darba devējs izvēlas un apstiprina darbinieku. Abus uzdevumus iespējams sadalīt divās fāzēs: prioritāšu fāze un lēmumu pieņemšanas fāze. Vispirms, APPS dalībnieks nosaka piedāvājumu (projektu piedāvājumi vai cenu piedāvājumi no darbiniekiem) prioritātes un pēc tam pieņem lēmumu. Tāpēc uzvedības modeļus sadala divās grupās. Pirmā ir veltīta piedāvājumu prioritāšu noteikšanai, bet otra ir vērsta uz lēmumu pieņemšanu.

Prioritāšu uzvedības modeļiem formāli ir definēti noteikumi, kas regulē piedāvājumu prioritātes noteikšanu. Katrs uzvedības modelis ir saistīts ar atribūtu vai atribūtu grupām.

Katrs APPS dalībnieks var vienlaicīgi definēt dažādus uzvedības modeļus, un katrs modelis satur savu svaru koeficientu. Piemēram, viens no prioritāšu uzvedības modeļa teksta aprakstiem ir „*darbinieks izvēlas projektu, kura budžets ir lielāks par 100 LVL un atrodas ne tālāk no 10 km no darbinieka mājām*”.

Otra uzvedības modeļu grupa ir veltīta lēmumu pieņemšanai. Šis modelis parasti ir saistīts ar darbībām, kas ietekmē APPS. Lēmumu pieņemšanas uzvedības modeli var definēt: uz cik projektiem darbinieks var vienlaicīgi pretendēt, cik projektos piedalīties (piemēram, uzreiz, kad izvēlas projektu no citiem, pēdējā brīdī utt.), saistības ar citiem dalībniekiem (piemēram, pārspēt konkurentus vai kooperēties ar partneriem) utt.

Atribūtu grupas

Piedāvātās pieejas galvenais elements ir atribūts. Šajā kontekstā tās ir īpašības, kas attiecas uz darbinieku, darba devēju un projektu. Klasificēsim atribūtus un sadalīsim tos pa atribūtu grupām. Klasiskā APPS darbība pamatojas uz trim atribūtu grupām [Hes2010]:

- *Maksa* – tā ir summa, kuru darba devējs var samaksāt darbiniekam par izpildīto darbu, vai pretēji, summa, par kuru darbinieks ir ar mieru izpildīt darbu. Ir gadījumi, kad darbinieks var izdarīt darbu bez maksas, piemēram: dodot atlaidi darba devējam, pieredzes trūkuma dēļ utt. Šim atribūtam ir definēta vienība: par projektu, par stundu, par platību utt.
- *Darba stundas* – tie ir dažāda veida atribūti, kas saistīti ar laiku:
 - *Darbs stundās* – cik daudz stundu nepieciešams darbiniekam projekta izpildīšanai vai „ideālais stundu skaists”, lai pabeigtu uzdevumu. Parasti tas ir pamata parametrs projekta izmaksu definēšanai.
 - *Projekta pabeigšanas datums* – cik ilgi tiek izpildīts projekts, to var definēt ar: beigu datums, stundu skaits, dienu skaits utt.
- *Saistības* – ar šo komplekso terminu parasti apzīmē, kādai dalībnieku grupai pieder konkrētais dalībnieks, darba pieredzi (piemēram, izpildītie projekti), rekomendācijas (darbiniekiem no darba devējiem, darba devējiem no darbiniekiem, no partneriem) utt.

Detalizēti tika apskatītas dažādas APPS [Gur2011,Free2011a,Free2011b] un tiek definēti pamata atribūti, kas tiek izmantoti projektu aprakstam (atribūti un atribūtu nozīmes ir ņemtas no [Free2011a]):

- *Budžets* (ang. *budget*) – darba devēja maksa, kurš viņš var samaksāt par projekta izpildīšanu.

- *Izveidots* (ang. *created*) – projekta piedāvājuma publikācija, ir iespējams definēt konkrētu datumu kad piedāvājums ir redzams visiem APPS dalībniekiem. Šis datums var nesakrist ar reālo projekta piedāvājuma izveidošanu.
- *Cenu noteikšanas datums* (ang. *biddings ends*) – projekta piedāvājuma beigu datums. Šis atribūts tiek definēts dienās un nevar būt ilgāks par 60 dienām.
- *Projekta izveidotājs* (ang. *project creator*) – tieša savienība ar darba devēju, kurš reģistrēja projekta piedāvājumu.
- *Darbinieka reitings* (ang. *buyer rating*) – pēc darba izpildīšanas darba devējs novērtē darbu ar vērtību no 0 līdz 5 (jo lielāks, jo labāks). Iespējams pasūtīt no APPS papildu pakalpojumu, kad iespējams saņemt papildu informāciju par darbinieku: darba devēju komentārus, izpildīto projektu kopējo skaitu, atteikto projektu skaitu utt.
- *Apraksts* (ang. *description*) – projekta piedāvājuma apraksts brīvā formā.
- *Darba veids* (ang. *job type*) – definēts kāda veida darbs jāizpilda.

Pētījuma hipotēze prasa definēt atribūtu grupas, ko ir iespējams izmantot APPS darbību nodrošināšanai. Apvienojot iepriekš definētos atribūtus, kas tiek izmantoti apraksta definējumam, darbinieka un darba devēja atribūti tiek saņemti atribūtu kopā. Tiek pētīts, kā visus šos atribūtus iespējams izmantot kopā ar DARSIR koncepcijas atribūtu grupām. Tiek uzskatīts, ka ir piemērotas sekojošas sešas DARSIR atribūtu grupas:

- *Individuālās īpašības* – tas ir APPS projekta objektu fiziskās īpašības. Iespējams definēt fiziskās īpašības pašam projektam (piemēram, zālāju pļāvējam norādīta platība, kura jāapstrādā) un objektiem, kuri jāizmanto projektā (piemēram, degviela zāles pļāvējam).
- *Saistības* – projektam ir īpašnieks (fiziska persona, organizācija, valsts utt.), kuru iespējams aprakstīt ar tādiem atribūtiem, kā: personas vārds, organizācijas nosaukums utt.
- *Laiks* – tie ir visi atribūti, kuriem definēti datumi, laiki vai stundu skaits: projekta izveidošanas datums, projekta novērtējums stundās, projekta pabeigšanas datums utt.
- *Pozicionēšana* – tā ir fiziska vieta, kur atrodas projekta dalībnieki (projekta darba vieta, darbinieka atrašanās vieta utt.). Parasti to definē ar GPS koordināti, bet iespējams norādīt atrašanās vietu arī ar citām metodēm (istabas numurs, reģions, pilsēta, pasta indekss utt.).
- *Konfigurācija* – projektu var sastādīt arī no citiem projektiem (piemēram, mājas tīrīšanu var sadalīt pa tīrīšanas veidiem vai istabām). Viens no principiem, kā lielu projektu iespējams sadalīt starp partneriem.

- *Vēsture* – tie ir dažādu veidu atribūti, kuriem ir piesaistīts laika komponents. Šis parametrs ir svarīgs, lai saglabātu vēsturi, uz kuras pamata tiek mainīti dalībnieku uzvedības modeļi.

Tika secināts, ka iepriekš definētās DARSIR koncepcijas atribūtu grupas nepietiek APPS atribūtu grupas definējumam, ir nepieciešams DARSIR koncepciju papildināt vēl ar divām atribūtu grupām:

- *Vadība* (ang. *management*) – tā ir īpašību kopa, kas tiek izmantota uzvedības modeļa iedarbināšanai. Uz šī atribūta pamata ir pieņemti dažādu veidu lēmumi, piemēram, pēc šī atribūta notiek projekta vai darbinieka novērtējums. Tipiski piemēri šai atribūtu grupai ir: maksājuma summa (piemēram, likme, kas atbilst darbinieka darba stundai, darba ilgums stundās (piemēram, definēts, ka darbs obligāti jāizpilda dienas laikā) utt.
- *Prasmes* (ang. *skills*) – tās ir darbinieka kompetences. Kompetenci var noteikt pēc atbilstības (ir vai nav), vai ir definēts zināšanu līmenis (piemēram, sertifikāts norāda valodu zināšanas līmeni skaitliskā vērtībā).

Piemēra apraksts

Pētījumā izvirzītais risinājums jāpārbauda praksē. Pieņemsim, ka ir darbinieks, kas nodarbojas ar zāles pļaušanu, nosauksim viņu par *zāles pļāvēju (ZP)*. Parasti ZP darbs aizņem no 2 līdz 10 stundām vienam projektam. ZP cenšas saņemt lielākus projektus, kas aizņem 8-10 stundas, tas viņam ir izdevīgi pēc blakus izmaksām (laiks ceļā līdz vietai, zāliena pļaušanas rīku piegāde, sagatavošanās darba procesam utt.), un tad ZP var samazināt cenu par darbu. Mūsu pētījuma mērķis ir palīdzēt ZP atrast labākos projektus no visiem piedāvājumiem. Ideāli jācenšas automātiski sastādīt ZP darba grafiku visai nedēļai.

Klasiskajā APPS'ā galvenie parametri, kurus *darba devējs* ievada, ir apmaksa un darba stundas. Un pats darbinieks katram projektam var definēt zemāku summu, par kuru viņš var izpildīt projektu. Pati samaksa ietver sevī visas darbinieka izmaksas un peļņu. Citiem vārdiem sakot, darbinieks var izvēlēties, vai ir vērts piedalīties projektā, ja tas nav izdevīgi, viņš var ignorēt piedāvājumu.

Ja ievieš detalizētāku informāciju par projektu, tad darbinieks var samazināt savas izmaksas, apvienojot projektus un definējot, ka izmaksas būs mazākas. Piemēram, darba devējs pierēģistrē projekta piedāvājumu ZP un uzreiz norāda, ka pašam darba devējam ir benzīns un zāles pļāvējs. Ja konkrētam ZP ir zināma šāda veida informācija, viņš var aprēķināt savas izmaksas, neiekļaujot sava zāles pļāvēja transportēšanu līdz darba vietai, un kopējā ZP likme samazināsies. Projekta cenu samazinājums motivē darba devēju norādīt vairāk informācijas par projektu. Vairāk informācijas par projektu nodrošina darbiniekam labāku

savu izmaksu novērtējumu, izdevīgāku cenu piedāvājumu. Rezultātā paaugstinās APPS servisa efektivitāte.

Simulācija un rezultātu analīze

Pētījuma sākumā tika definēta hipotēze, ka var paaugstināt tipisku APPS darbību, palielinot projekta atribūtu skaitu, kurus iespējams izmantot darbinieku projektu izmaksu novērtējumam. Hipotēzes pārbaudei ir izveidota APPS simulācija, kas emulē tipisku mūsdienīgu APPS darbību. DARSIR koncepcijas pamatā ir nodrošināti projekta piedāvājumā atribūtu apraksti un darbiniekiem ir nodrošināti uzvedības modeļi.

Tiek pieņemts, ka ZP gadījumā jāspecificē *apraksta* atribūts, kurā ir ierakstīts projekta piedāvājuma apraksts brīvā formā. *Apraksta* atribūtu specificējam ar diviem atribūtiem:

- *Platība* – parametrs, no kura iespējams aprēķināt ZP nepieciešamo darba izpildīšanas apjomu. Katram ZP ir norādīta likme uz vienu kvadrātmetru, un, ja kopējo platību sadala ar šo parametru, iespējams aprēķināt projekta cenu.
- *Atrašanās vieta* – pieņemts, ka ar šo parametru iespējams aprēķināt attālumu no plānotās darbinieka atrašanās vietas uz projekta sākuma datumu. Šis parametrs tiek izmantots blakus izmaksu aprēķinam: laiks, cik ilgi jābrauc līdz darba vietai, un ceļa izdevumi.

Pirms simulācijas izveidošanas ir nepieciešams definēt, pēc kura principa tiks novērtēts simulācijas efektivitātes aprēķins. Tas ļauj nodrošināt simulācijas variācijas salīdzinājumu un konkrētas simulācijas variācijas novērtējumus. Tiek definēti divi galvenie parametri simulācijas efektivitātes novērtējumam:

- *Vidējās projekta izmaksas* – tā ir samaksātā naudas summa par izpildīto projektu, šis ir parametrs svarīgs darba devējam, norādītas viņa izmaksas par izpildīto darbu.
- *Vidējā darbinieka peļņa* – tas ir parametrs, kas tiek aprēķināts pēc formulas: projekta summa mīnus darbinieka blakus izmaksas. Pēc šī parametra tiek novērtēta darbinieku projekta izvēles efektivitāte.

Šajā pētījumā ir paredzēts izmantot vienkāršu novērtēšanas metodi, kas pamatojas uz vidējām projekta izmaksām un vidējo darbinieka peļņu. Vēl iespējams piemērot novērtēšanas metodēm arī citus parametrus, piemēram: apmierinājums (darba devēja un darbinieka), darbinieka neplānotas blakus izmaksas, projekta izpildes ātrums, projekta izpildes kvalitāte, nepabeigti projekti utt.

Simulācija ir izveidota, izmantojot MAS iebūvētu ar NetLogo rīka palīdzību. NetLogo [Wil1999] ir MAS programmēšanas modelēšanas vide. Simulācijas parametri (piemēram,

vidējā projekta cena, APPS dalībnieku proporcijas, projektu daudzums utt.), tika izveidoti, pamatojoties uz mūsdienīgu [Fre2011a] APPS servisu.

Darbinieku peļņa tiek aprēķināta no projekta maksas, no kuras ir noņemtas nost visas blakus izmaksas:

- *Dienas izmaksas* – tās ir visas darbinieka izmaksas, kas ir vajadzīgas darba vides nodrošinājumam (piemēram, komunālie maksājumi, ēdiens, apģērbi utt.). Šis parametrs ir fiksēts un unikāls katram darbiniekam.
- *Projekta izmaksas* – visas izmaksas, kuras maksā darbinieks pats projekta izpildes gaitā. Šis parametrs jāaprēķina pēc stundas likmes. Tipiskās tāda veida izmaksas: darba rīku amortizācija, apdrošināšana, utt.
- *Ceļa izdevumi* – tās ir izmaksas, kas jāveic darbiniekam, lai pārvietotos līdz darba vietai. Tiek aprēķināta standarta likme par vienu kilometru.

Simulācijas tiek izveidotas pamatoties uz sekojošiem noteikumiem:

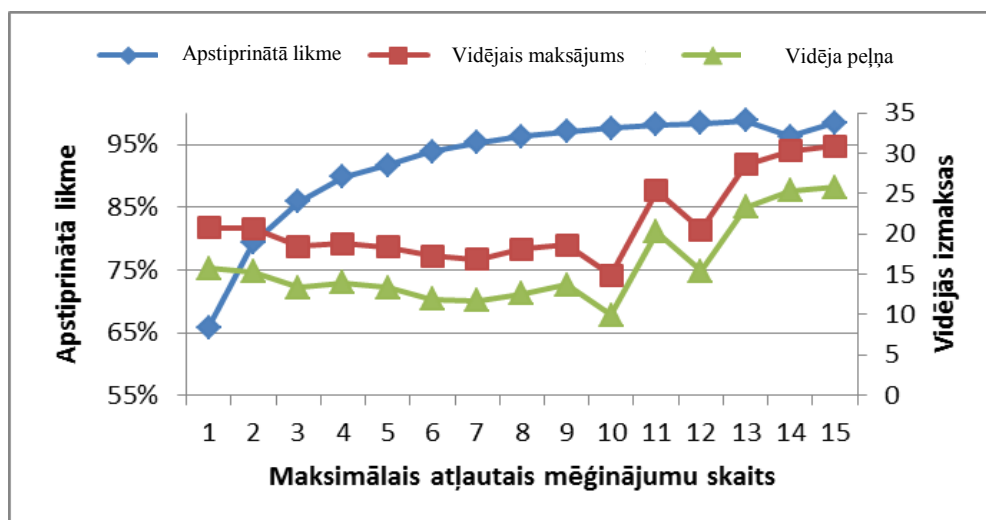
- N darbinieki, aģenti;
- M darba devēji, nodrošina projektu izveidošanu;
- $Z=M*K$ (maks.) projekta piedāvājumi;
- Katram projekta piedāvājumam ir atrašanās vieta, definēta ar X un Y koordinātēm;
- Katram darbiniekam ir sava atrašanās vieta, pēc attāluma no darbinieka atrašanās vietas līdz darba vietai ir aprēķinātas blakus izmaksas;
- Katram darbiniekam ir definēta sava likme, izmantojot kuru tiek aprēķināta cena par projekta izpildi;
- Katrs darbinieks var dot B piedāvājumus uz Z projekta piedāvājumiem;
- Darbinieka peļņa = projekta summa – blakus izmaksas (dienas izmaksas, projekta izmaksas un ceļa izdevumi).
- Katrs darbinieks var izpildīt W projektus dienā;
- Darba devējs projekta izsoles beigās apskata visus darbinieku cenu piedāvājumus un izvēlas vislētāko piedāvājumu, un atmaksā darbiniekam naudas summu atbilstoši viņa cenu piedāvājumam;
- Pēc R iterācijas ir aprēķināti divi parametri: vidējās projekta izmaksas un vidējā darbinieka peļņa;
- Tiek ieviesta darbinieka apmācības stratēģija: pamatojoties uz konkrētu vidējo darbinieka peļņu, tiek pieņemts lēmums par darbinieka likmes paaugstināšanu vai samazināšanu.

Simulācijas izstrādes laikā ir ieviests viens atribūts „atrašanās vieta” un apskatīts, cik daudz šī atribūta ieviešana ietekmē vidējās projekta izmaksas un vidējo darbinieka peļņu. Tiek izveidotas divas simulācijas:

- *Simulācija I* – atribūts „atrašanās vieta” nav zināms, un darbinieks izmanto vidējo ceļa izdevumu izmaksu aprēķinam.
- *Simulācija II* – atribūts „atrašanās vieta” ir ieviests, un darbinieks izmanto šo atribūtu izmaksu aprēķinam.

Simulācijas I rezultāts grafiskā veidā ir atspoguļots P6.2. attēlā. Tas ir gadījums, kad atribūts „atrašanās vieta” nav zināms. Grafikā ir redzams, ka apstiprinātā likme paaugstina savu vērtību atbilstoši mēģinājumu skaita palielinājumam. Jo vairāk mēģinājumu izdara aģents, jo lielāka varbūtība, ka likme tiks apstiprināta.

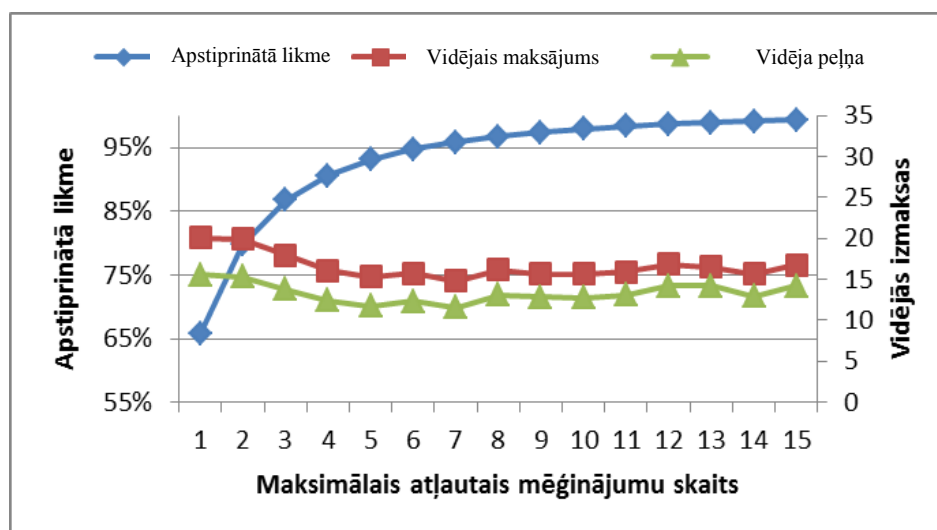
Cita sakarība ir starp vidējo maksājumu (darba devējam) un vidējo peļņu (darbiniekam). Šīs līnijas iet paralēli, jo ir noteikta atšķirība attiecībā uz vidējām ceļošanas izmaksām. Bet grafikā redzams, kad mēģinājumu skaits sasniedz noteiktu sliekšni (grafikā mēģinājumu skaits ir 10), tad darbinieki pieprasa augstāku samaksu par savu darbu un darba devējam tā jāmaksā.



P6.2. attēls. Apstiprinātā likme un vidējās izmaksas atkarībā no mēģinājumu skaita (Simulācija I)

Nākamajā grafikā, kas ir redzams P6.3 attēlā, ir atspoguļota *Simulācija II*, kurai ir tādi paši ieejas parametri kā *Simulācija I*, bet visiem dalībniekiem ir redzama ģeogrāfiska informācija par vietu, kur realizēts projekts. Tas dod iespēju darbiniekiem precīzi aprēķināt savas izmaksas.

P6.3 attēla kopējās sakarības starp apstiprināto likmi, vidējo maksājumu un vidējo peļņu ir līdzīgas P6.2. attēlam. Galvenā atšķirība ir tā, ka stabilizējas projekta vidējās izmaksas un darbinieka peļņa.



P6.3. attēls. Apstiprinātā likme un vidējās izmaksas atkarībā no mēģinājumu skaita (Simulācija II)

Tālākajos eksperimentos var ietvert papildu simulācijas noteikumus, kas var būtiski ietekmēt simulācijas rezultātus. Piemēram, ieviest sodu par darbinieka atteikšanos no uzdevuma izpildes, tas motivē darbinieku rūpīgi izvēlēties uzvedības modeļa stratēģiju. Cita tipa eksperimenti ir papildu atribūtu ieviešana, kas ļauj APPS dalībniekiem precīzāk novērtēt rezultātu pirms lēmumu pieņemšanas. Piemēram, jo vairāk darbinieks zina par projekta izmaksām, jo izdevīgāku cenu viņš var piedāvāt savam pakalpojumam. Citā gadījumā darba devējs, definējot projekta izpildītāju, var balstīties ne tikai uz darbinieka piedāvāto cenu, bet izmantot kvalitātes parametrus (piemēram, darbinieka atteikšanās no projekta piedāvājumiem, veiksmīgi pabeigti projekti, iepriekšējo darba devēju apmierinātība utt.), un tas var paaugstināt darba devēja apmierinātību par projekta rezultātiem un motivēt darbiniekus sniegt kvalitatīvus pakalpojumus.

Jaunas funkcijas ieviešana

DARSIR koncepcijai ir iespējams izmantot dažādas IT tehnoloģijas. 3.4. nodaļā tiek definētas noteikumu grupas, kas ir nepieciešamas sekmīgai DARSIR koncepcijas iedarbināšanai. Bet ir gadījumi, kad konkrētai IT tehnoloģijai pamatfunkcijas netiek nodrošinātas. Viena no iespējam nodrošināt nepieciešamās funkcijas ir izveidot tās pašam. Viena no tādām funkcijām ir pozicionēšanas iespēja. Šīs apakšnodaļas ietvaros ir parādīts gadījums, kad vienas specifiskas funkcijas dēļ tiek piedāvāta metode, kas dot iespēju ieviest šo funkciju lielākajai daļai no DARSIR koncepcijas izvēlētajām tehnoloģijām: RFID, Bluetooth, Wi-Fi utt. (pilnais tehnoloģiju saraksts ir dots 2.2. apakšnodaļā).

Pozicionēšanas funkciju iespējams nodrošināt vairākām IT tehnoloģijām, balstoties uz *signālu izplatīšanas modeļiem*. Piemērotākās RFID tehnoloģijas ir iespējams realizēt, balstoties uz aktīvām RFID birkām [Chi2009] vai pasīvām RFID birkām [Man2010, Jah2009]. Līdzīgos pētījumos bija izpētīta Bluetooth tehnoloģijas izmantošana pozicionēšanā [Hal2003, Che2006].

Detalizētai analīzei tika izvēlēta Wi-Fi tehnoloģija, kas galvenokārt balstās uz infrastruktūras pieejamību un tehnisko ierīču iepirkšanas izmaksām. Vairākums no piedāvātajām Wi-Fi pozicionēšanas sistēmām izmanto vai nu radiosignālu izplatīšanās modeļus [Tho2005, Wid2007], vai arī pozīcijas noteikšanas tehnikas, izmantojot signālu stipruma datubāzi [Bad2007, Bru2005], kas dod precīzākus rezultātus.

Pozicionēšanas funkcijas darbības principa izvēle

Wi-Fi nebija piemērota pozicionēšanai, radiosignāli var tikt izmantoti pozīcijas noteikšanai, izmantojot signālu stipruma vērtības, kas var tikt izmērītas ar jebkuru jauno mobilo ierīci, kurai ir Wi-Fi modulis – un nav nepieciešama nekāda papildus aparatūra. Precīza signāla izplatīšanās modeļa izstrāde katram Wi-Fi piekļuves punktam reālos apstākļos ir ļoti sarežģīta, un parasti rezultātā šī metode dod relatīvi zemu precizitāti [Wid2007, Yim2010].

No otras puses, signālu stiprumu datubāzes veidošanas tehnikas izmanto empīriskus datus, lai aprēķinātu pozīciju. Pirmkārt, tiek konstruēta tā saucamā radio karte, izmērot signālu stiprumus dažās zināmās pozīcijās – tie ir kalibrācijas punkti. Pēc tiem tiek aprēķināta lietotāja pozīcija, salīdzinot iegūtās signālu stiprumu vērtības ar tām, kuras ir radio kartē (datu krājumā). Tas nodrošina precīzu pozicionēšanu pat sarežģītā vidē. Cita šīs metodes priekšrocība ir tā, ka tā neprasa zināt precīzas piekļuves punktu pozīcijas.

Galvenā priekšrocība, kas ietekmēja signālu stiprumu datubāzes izmantošanas izvēli, ir tās darbības principu piemērotība dažādām tehnoloģijām (RFID, Bluetooth, Wi-Fi utt). Šos darbības principus neietekmē tehnoloģijas signālu izplatīšanas īpašības. Dotā metode tika izpētīta, un rezultātā tika izstrādāts prototips, kas detalizēti ir aprakstīts [Jek2011a, Jek2011b, Jek2010] publikācijās.