

**LATVIJAS UNIVERSITĀTE**

**AIVARS NIEDRĪTIS**

**Daudzu Nomnieku Adaptīvas  
Tīmekļa Informācijas Sistēmas  
arhitektūra**

Promocijas darbs  
Datorzinātņu doktora (Dr. sc. comp.) zinātniskā grāda iegūšanai

Nozare: Datorzinātne  
Apakšnozare: Datu apstrādes sistēmas un datortīkli

Zinātniskais vadītājs:  
profesors, Dr. sc. comp.,  
MĀRIS TREIMANIS

R ī g a - 2013

Darba vadītājs:

Māris Treimanis, Dr. sc. comp., Profesors, Latvijas Universitāte

Recenzenti:

- 1) Jānis Bičevskis, Dr.sc. comp., Profesors, Latvijas Universitāte;
- 2) Rimantas Butleris, Ph.D. datorzinātnēs, Kauņas Tehnoloģiju Univeristāte
- 3) Uldis Sukovskis, Dr.sc. ing., Profesors, Rīgas Tehniskā Universitāte

Darba aizstāvēšana notiks Datorzinātnes nozares promocijas padomes atklātajā sēdē 2013. gada 8.aprīlī, Raiņa bulvārī 29., 413.auditorijā.

Ar darbu un tā kopsavilkumu var iepazīties LU bibliotēkā (Raiņa bulvārī 19).

Padomes priekšsēdētājs

prof. Jānis Bārzdīņš

# SATURS

<b>1. IEVADS</b> .....	<b>6</b>
1.1. Darba mērķis un uzdevumi.....	6
1.2. Pētāmie jautājumi un aizstāvēšanai izvirzītās tēzes.....	6
1.3. Metožu raksturojums .....	7
1.4. Pētījuma motivācija un aktualitāte.....	7
1.5. Promocijas darba galvenie rezultāti.....	8
1.6. Rezultātu publikācijas, prezentācijas konferencēs un aprobācija .....	9
1.7. Promocijas darba struktūra .....	10
<b>2. TĪMEKĻA INFORMĀCIJAS SISTĒMAS</b> .....	<b>12</b>
2.1. Nodaļas nolūks .....	12
2.2. Tīmekļa sistēmas .....	12
2.2.1. Hiperteksta un hipervides sistēmas .....	12
2.2.2. Tīmekļa Informācijas Sistēmas .....	12
2.2.3. Adaptīvas hipervides sistēmas .....	14
2.2.4. Adaptīvas Tīmekļa informācijas sistēmas .....	16
2.2.5. Lietotāja modeļa vieta adaptācijas procesā .....	18
2.2.6. Adaptācijas efektu izpausmes veidi .....	20
2.2.7. TIS inženierija un adaptācija.....	21
2.2.8. Eksistējoši risinājumi adaptīvām TIS .....	22
2.3. TIS implementācijas mehānismi.....	24
2.3.1. TIS arhitektūras.....	24
2.3.1.1. Programmatūras produktu līnija .....	25
2.3.1.2. Programmatūra kā serviss.....	25
2.3.2. Daudzu nomnieku TIS eksistējoši risinājumi.....	27
2.4. Nodaļas secinājumi.....	28
<b>3. DAUDZU NOMNIEKU ADAPTĪVAS TĪMEKĻA INFORMĀCIJAS SISTĒMAS ARHITEKTŪRAS IZVEIDE</b> .....	<b>29</b>
3.1. Nodaļas nolūks .....	29
3.2. DNATIS arhitektūras vispārīgs apraksts.....	29
3.3. DNATIS arhitektūras implementācijas mehānismi.....	33
3.3.1. DNATIS arhitektūras datu slāņa profili .....	33
3.3.2. DNATIS Prezentācijas modelis.....	36
3.3.3. DNATIS arhitektūras komponenti.....	37

3.3.3.1.	Konteksta monitors.....	37
3.3.3.2.	Adaptācijas komponents.....	38
3.3.3.3.	Rupjās adaptācijas operācijas.....	38
3.3.3.4.	Detalizētās adaptācijas operācijas.....	42
3.3.3.5.	Piemērs adaptācijas komponenta darbībai.....	43
<b>3.4.</b>	<b>Lietotāja saskarnes veidošanas specifika DNATIS arhitektūrā.....</b>	<b>46</b>
3.4.1.	Integrētas lietotāja saskarnes izveidošana vairāku organizāciju gadījumā.....	46
3.4.2.	Adaptācijas procesa rezultāti no lietotāja saskarnes veidošanas viedokļa.....	47
3.4.3.	Lietotāja datu integrācija lietotāja datu apgabalā.....	50
3.4.4.	Lietotāja saskarnes adaptēšana lietotāja veiktajām darbībām.....	52
<b>3.5.</b>	<b>DNATIS arhitektūras aprobācija.....</b>	<b>54</b>
3.5.1.	DNATIS arhitektūras pielietojums Augstskolu informatīvajā sistēmā.....	54
3.5.2.	DNATIS arhitektūras pielietojums Transportlīdzekļu reģistrā.....	55
<b>3.6.</b>	<b>Nodaļas secinājumi.....</b>	<b>58</b>
<b>4.</b>	<b>LIETOTĀJA MODELIS DNATIS ARHITEKTŪRĀ.....</b>	<b>60</b>
<b>4.1.</b>	<b>Nodaļas nolūks.....</b>	<b>60</b>
<b>4.2.</b>	<b>Lietotāja profili un lietotāja modeļa nepieciešamība.....</b>	<b>60</b>
<b>4.3.</b>	<b>Lietotāju aprakstošā profila jēdziens.....</b>	<b>62</b>
<b>4.4.</b>	<b>Integrētais lietotāja modelis.....</b>	<b>63</b>
4.4.1.	Integrētā lietotāja modeļa īpašības.....	64
4.4.2.	Eksistējošu lietotāja profilu integrācijas process.....	65
<b>4.5.</b>	<b>Integrētā lietotāju modeļa izmantošana mērīšanas sistēmā.....</b>	<b>66</b>
4.5.1.	Datu noliktavu modeļu izveides pieejas.....	67
4.5.2.	Metadati mērīšanas sistēmas shēmas evolūcijas atbalstam.....	68
4.5.3.	Informācijas avoti vaicājumu lietošanas analīzei.....	69
4.5.4.	Vaicājumu vadīta metode datu noliktavas modeļa attīstīšanai pēc lietotāja aktivitātēm datu avotu sistēmās.....	70
4.5.5.	Metodes darbības piemērs.....	72
<b>4.6.</b>	<b>Nodaļas secinājumi.....</b>	<b>74</b>
<b>5.</b>	<b>VEIKTSPĒJAS MĒRĪŠANA DNATIS ARHITEKTŪRĀ.....</b>	<b>76</b>
<b>5.1.</b>	<b>Nodaļas nolūks.....</b>	<b>76</b>
<b>5.2.</b>	<b>Veiktspējas mērīšanas sistēmas vieta DNATIS arhitektūrā.....</b>	<b>76</b>
<b>5.3.</b>	<b>Jēdzienu definīcijas.....</b>	<b>77</b>
5.3.1.	Mērīšanas jēdzieni no mērījumu teorija viedokļa.....	78
5.3.2.	Mērīšanas jēdzieni no pārvaldības modeļu viedokļa.....	78
<b>5.4.</b>	<b>Datu noliktavas kā veiktspējas mērījumu sistēmas.....</b>	<b>79</b>
<b>5.5.</b>	<b>Indikatoru dzīves cikls un indikatoru aprakstīšanas aspekti.....</b>	<b>81</b>
5.5.1.	Indikatoru definēšanas formālais modelis.....	83

5.5.2.	Indikatoru pārformulēšanas principi .....	83
5.5.3.	Prasību šablonu aprakstīšana.....	85
<b>5.6.</b>	<b>Veiktspējas mērīšanas ietvars ar indikatoru dzīves cikla atbalstu .....</b>	<b>87</b>
5.6.1.	Indikatoru un to dzīves cikls kā pamats veiktspējas mērīšanas ietvaram.....	87
5.6.2.	Atskaišu rīka izmantošana mērījumu ietvarā.....	88
5.6.3.	Veiktspējas mērīšanas sistēmas komponenti .....	90
5.6.4.	Indikatoru dzīves cikla soļi veiktspējas mērīšanas ietvarā .....	92
5.6.5.	Integrācija ar datu noliktavas komponentiem.....	94
5.6.6.	Veiktspējas mērījumu sistēmas novērtējums.....	94
<b>5.7.</b>	<b>Biznesa procesu divu līmeņu mērīšanas pieeja .....</b>	<b>97</b>
5.7.1.	Divu līmeņu mērīšanas pieeja .....	97
5.7.2.	Procesu mērīšanas sistēmas implementācija.....	99
5.7.3.	Divu līmeņu mērīšanas pieejas aprobācija .....	100
<b>5.8.</b>	<b>Secinājumi.....</b>	<b>103</b>
<b>6.</b>	<b>NOBEIGUMS .....</b>	<b>105</b>
	<b>LITERATŪRAS SARAKSTS .....</b>	<b>107</b>
<b>1.</b>	<b>PIELIKUMS.....</b>	<b>117</b>
1.1.	DNATIS arhitektūras pielietojums Augstskolu informatīvajā sistēmā .....	117
1.2.	DNATIS arhitektūras pielietojums Transportlīdzekļu reģistrā.....	120
1.3.	Divu līmeņu mērīšanas pieejas pielietojums Latvijas Universitātē .....	123
1.3.1.	Pārvaldības Informācijas Sistēma LUIS .....	123
1.3.2.	E-Studiju sistēmas .....	124
1.3.3.	Reģistrācija uz semestri .....	124
1.3.4.	E-kursu lietošana.....	126
<b>2.</b>	<b>PIELIKUMS. SAISTĪTĀS PUBLIKĀCIJAS.</b>	<b>Atsevišķā sējumā</b>

## 1. Ievads

Programmatūra kā pakalpojums (Software as a service – SaaS) ir jauns programmatūras piegādes veids, kas balstīts uz Interneta tehnoloģijām. SaaS lietotne vienmēr ir pieejama, kad ir pieejams Internets. Atbilstoši savām vajadzībām lietotājs (vai organizācija) izvēlas SaaS lietotni un nomā šo programmatūras pakalpojumu.

SaaS lietotnes gadījumā tiek stingri nodalīts, ka programmatūras īpašnieks ir pakalpojuma sniedzējs, kas izvieto programmatūru uz saviem resursiem, bet nomnieks var šo programmatūru lietot, piekļūstot šim pakalpojumam ar Internetu (Laplante et al., 2008).

SaaS lietotnes gadījumā aktuāls kļūst jautājums, kā daudzu lietotāju gadījumā pielāgot vienu un to pašu lietotni kaut nedaudz, bet tomēr atšķirīgām lietotāju specifiskām vajadzībām, jo maza varbūtība, ka, piemēram, divu organizāciju biznesa procesi, kuru atbalstam paredzēts izmantot SaaS lietotni, būs identiski.

Aktuāli kļūst konfigurēšanas jautājumi, un šajā sakarā tiek pārskatīti arī programmatūras arhitektūras jautājumi, lai saprastu, kā aprakstīt konfigurējamu SaaS lietotni (Laplante et al., 2008).

SaaS lietotnei daudzu nomnieku gadījumā ir arī būtiski nodrošināt datu nošķirtību katram nomniekam. SaaS pieejai jāgarantē, ka dažādi nomnieki vienas fiziskas programmatūras instances gadījumā saņems viņu nolīgto, viņu biznesam piemērotu lietotni, kā arī tiks nodrošināta datu glabāšana uz pakalpojuma piegādātāja servera.

### 1.1. Darba mērķis un uzdevumi

Darba mērķis ir izstrādāt arhitektūru SaaS tipa tīmekļa informācijas sistēmas (TIS) izstrādei, kas nodrošinātu TIS adaptējamību individuālu lietotāju vajadzībām, izmantošanu vienlaicīgi vairākām organizācijām, uzturēšanas un attīstīšanas atbalstu.

Lai sasniegtu mērķi, ir jārisina sekojoši uzdevumi:

- veikt literatūras studijas un izpētīt eksistējošas pieejas tīmekļa informācijas sistēmu adaptēšanā,
- veikt literatūras analīzi, lai noskaidrotu eksistējošās pieejas SaaS lietotņu konfigurēšanā un adaptēšanā,
- veikt tīmekļa informācijas sistēmu (TIS) arhitektūras konceptuālo izstrādi, kas nodrošinātu TIS adaptējamību individuālu lietotāju vajadzībām,
- dotās arhitektūras ietvaros izveidot SaaS tipa tīmekļa informācijas sistēmu uzturēšanas un attīstīšanas atbalsta mehānismu,
- veikt arhitektūras aprobāciju.

### 1.2. Pētāmie jautājumi un aizstāvēšanai izvirzītās tēzes

Aizstāvēšanai ir izvirzīta sekojoša tēze: ir iespējama efektīva, uz inovatīvu arhitektūru balstīta tīmekļa informācijas sistēmu izstrāde, lai realizētu programmatūras kā pakalpojuma pieeju.

Promocijas darbā ir pētīti sekojoši jautājumi:

- arhitektūras uzbūves principi, lai nodrošinātu TIS adaptāciju gan programmatūras pakalpojuma nomniekiem, gan individuāliem lietotājiem,

- lietotāja modeļa vieta un uzbūve adaptācijas arhitektūrā, lai realizētu programmatūru kā pakalpojumu,
- pēc piedāvātās adaptācijas arhitektūras uzbūvētas TIS izmantošana uzturēšanas laikā - mērīšana, analizēšana, novērtēšana

### 1.3. Metožu raksturojums

Teorētiskie pētījumi balstīti uz literatūras avotu analīzi, apkopošanu, sistematizēšanu, lai noskaidrotu risinājumus adaptīvu tīmekļa informācijas sistēmu jomā un novērtētu iespējas izmantot adaptācijas pieejas speciāla tipa tīmekļa informācijas sistēmām – SaaS (programmatūra kā pakalpojums). Darbā izmantota modelēšana – gan piedāvātās adaptācijas arhitektūras lietotāju modeļa, gan arhitektūrā ietvertā mērīšanas komponenta indikatoru modeļa izstrādei, kas tiek aprakstīti ar UML diagrammu palīdzību. Izmantotas programmatūras inženierijas metodes, lai izstrādātu piedāvāto arhitektūru un to pielietotu divos projektos.

### 1.4. Pētījuma motivācija un aktualitāte

Autora pieredze tīmekļa informācijas sistēmu (TIS) izstrādē atspoguļota daudzos promocijas darba autora rakstos un uzstāšanās specializētās konferencēs par augstskolu informācijas sistēmām (Stonis et al., 2001), (Niedritis, Niedrite, 2000), (Niedritis, 1999), (Niedritis, Niedrite, 1998). Sākot jau no augstskolu IS rašanās brīža, kas sākumā tika veidota tikai Latvijas Universitātes vajadzībām (LUIS), šī IS tika veidota kā TIS. Vēlāk šī sistēma tika izstrādāta vairāku universitāšu vajadzībām - tika uzsākta Latvijas augstskolu informācijas sistēmas (LAIS) izstrāde uz LUIS bāzes. Pieredze izstrādājot LAIS, ļāva nonākt pie secinājumiem, kādas problēmas sastopamas šādu TIS sistēmu izstrādē un uzturēšanā (Niedrite, Niedritis, 2003). Ja augstskolu TIS izstrādei finansējums bija paredzēts, tad tālākai uzturēšanai un attīstīšanai resursi gan finansiālie, gan cilvēkresursi bija ierobežoti, bija nepieciešams gan izveidot jaunu arhitektūru, kas būtu izmantojama adaptējamās TIS izstrādei, gan pietiekoši viegli paplašināmai un uzturamai, gan adaptējamai individuālu lietotāju vajadzībām.

Literatūrā ir aprakstītas un praksē tiek plaši pielietotas dažādas informācijas sistēmu arhitektūras. Lai nodrošinātu lietotājus ar tiem piemērotu TIS, īpaša uzmanība tika pievērsta elastīgu un izmaksu efektīvu sistēmu izstrādei, tajā skaitā programmatūras kā pakalpojuma pieejai (*Software as a Service* - SaaS). Savukārt, lai nodrošinātu izstrādātās TIS efektīvu izmantošanu, novērtējot sistēmas atsevišķu daļu izmantošanu biznesa procesu atbalstam, bija nepieciešams atrast, novērtēt esošās biznesa procesu mērīšanas un veikspējas mērījumu metodes.

Promocijas darbā piedāvātās jaunās arhitektūras, kas balstās uz programmatūras kā pakalpojuma pieeju, novitāti nosaka:

1) atšķirībā no esošiem risinājumiem, kas uz SaaS pieeju balstītu informācijas sistēmu pielāgošanai izmanto konfigurēšanu, promocijas darbā piedāvātā arhitektūra tiek balstīta uz divu līmeņu adaptācijas pieeju, ko raksturo sekojošas jaunas īpašības:

- nodrošina atšķirīgas TIS instances katrai organizācijai, kas ir dotās TIS nomnieks, pielāgojot tās organizācijas funkcionalitātei,
- nodrošina adaptētu TIS instanci katram TIS individuālam lietotājam, balstoties uz lietotāja modeli, kas ļauj dinamiski darba procesā ģenerēt lietotāja darbībām atbilstošu saskarni,

- nodrošina integrētu, adaptētu TIS saskarni gadījumam, ja lietotājs ir pilnvarots strādāt vienlaicīgi ar vairāku organizāciju TIS instancēm, tādējādi tiek nodrošināta iespēja lietotājam strādāt ar visu TIS instanču viņa lietotāja modelim atbilstošiem datiem,
- lietotāja modeļa izveidei tiek piedāvāta metode, kura balstīta uz atsevišķo organizāciju TIS instanču lietotāju modeļu integrāciju, tādā veidā nodrošinot iespēju veidot integrēto TIS saskarni.

2) TIS adaptācijas arhitektūras vajadzībām tiek piedāvāts mērīšanas ietvars un metode, ko raksturo sekojošas jaunas īpašības:

- mērīšanas indikatoru noteikšanas un kontroles atbalstam tiek piedāvāts piecu soļu indikatoru dzīves cikls, kas atbalsta sistemātisku mērīšanas procesu,
- mērīšanas sistēma balstīta uz datu noliktavas komponentiem, pielāgojot tos uz SaaS pieejas balstītu sistēmu darbības analīzei, lai nodrošinātu daudzo TIS sistēmu uzturēšanu,
- piedāvātā divu līmeņu mērīšanas metode nodrošina sistēmu lietošanas paradumu noteikšanu, lai rezultātus varētu izmantot TIS attīstīšanai, kā arī izmaksu noteikšanu par sistēmas lietošanu.

Tīmekļa informācijas sistēmu izstrādes problēmu aktualitāti nosaka tēmai atbilstošu konferenču eksistence, piemēram, specializētās konferences WISE (*Web Information Systems Engineering*), kas 2012. gadā jau tiks organizēta 12. reizi, kas pilnībā veltīta dažādiem TIS izstrādes tehnoloģiju, metožu un pielietojumu jautājumiem, kā arī tēmas, kas tiek iekļautas citās konferencēs, piemēram, ICEIS (*International Conference on Enterprise Applications*), ISD (*Information Systems Development*) konference, kas citu tēmu vidū apskata arī Tīmekļa informācijas sistēmu izstrādes jautājumus.

Savukārt jautājumi par biznesa procesu mērīšanu un pārbūvi, indikatoru noteikšanu un formālu definēšanu ir viena no tēmām, kas atbilst BIR (*Business Informatics Research*) konferences tēmu klāstam.

## 1.5. Promocijas darba galvenie rezultāti

Izstrādāta tīmekļa informācijas sistēmas arhitektūra DNATIS (Daudzu Nomnieku Adaptīva Tīmekļa Informācijas Sistēma), kas balstās uz daudzu nomnieku vienas instances SaaS lietotnes arhitektūras ideju bāzes, bet promocijas darbā izstrādātajā arhitektūrā iekļautas jaunas īpašības:

- divu līmeņu adaptācijas iespēja – organizācijas līmenim un individuāla lietotāja līmenim,
- adaptācija balstās uz promocijas darba ietvaros izstrādātu vispusīgu lietotāju modeli, kas iekļauj dažādus lietotāju aprakstošus aspektus,
- ir nodrošināta dinamiska adaptētas tīmekļa informācijas sistēmas lietotāja saskarnes ģenerēšana atbilstoši lietotāja modelim,
- arhitektūra ietver arī veiktspējas mērīšanas komponentu, kas paredzēts pēc DNATIS arhitektūras principiem izstrādātas TIS uzturēšanai, darbības novērtēšanai, attīstīšanai,
- mērīšanas komponents balstās uz promocijas darbā piedāvātu veiktspējas mērīšanas ietvaru, kura pamatā ir piecu soļu indikatoru dzīves cikls, kas ļauj sistemātiski izmantot indikatorus procesu mērīšanai,
- veiktspējas mērīšanai tiek piedāvāta 2 līmeņu mērīšanas pieeja: sistēmas līmeņa mērīšana un biznesa procesu mērīšana.



## 1.6. Rezultātu publikācijas, prezentācijas konferencēs un aprobācija

Promocijas darbs balstās uz pētījumiem, kuru rezultāti ir atspoguļoti 10 publikācijās (publikācijas pievienotas atsevišķā pielikumā):

- (Niedrite et al., 2007a) Niedrite, L., Solodovnikova, D., Treimanis, M., Niedritis, A. The Development Method for Process-Oriented Data Warehouse, In: WSEAS Transactions on Computer Research, Issue 2, Vol. 2, pp. 183 – 190 (2007)
- (Medvedis et al., 2008) Medvedis, I., Niedrite, L., Niedritis, A., Treimanis, M., Voitkans, A. E-University Initiative: Approach, Solutions and Case Studies. In: Proceedings of the 8th International Baltic Conference on Databases and Information Systems, Estonia, pp. 141-152 (2008)
- (Medvedis et al., 2008b) Medvedis, I., Niedrite, L., Niedritis, A., Treimanis, M., Voitkans, A. Approach and Technical Solutions for e-University Initiative', Postconference proceedings of the 8th International Baltic Conference on Databases and Information Systems, Tallinn, Estonia, Frontiers in Artificial Intelligence and Applications by IOS Press, 2008.
- (Niedritis, Niedrite, 2011) Niedritis, A., Niedrite L. The Adaptation of a Web Information System: a Perspective of Organizations, In: Pokorny, J. Repa, V., Richta K. et al. (Eds) Proceedings of the 19th International Conference on Information Systems Development (ISD2010), Springer, pp. 539- 550, (2011).
- (Niedritis, 2011) Niedritis, A. Delivery of Consistent and Integrated User's Data within a Multi-Tenant Adaptive SaaS Application. In: Niedrite, L., Strazdina, R., Wangler B. (Eds.), Perspectives in Business Informatics Research. Local Proceedings of 10th International Conference, BIR2011 Associated Workshops and Doctoral Consortium, Riga Technical University, pp. 307-314, (2011)
- (Niedritis et al., 2011a) Niedritis, A., Niedrite, L., Kozmina, N. Performance Measurement Framework with Formal Indicator Definitions. In: Grabis, J., Kirikova M. (Eds), Perspectives in Business Informatics Research, Lecture Notes in Business Information Processing, Volume 90, Springer, p. 44-58. (2011)
- (Niedritis et al., 2011b) Niedritis, A., Niedrite, L., Kozmina, N. Integration Architecture of User Models. In: Niedrite, L., Strazdina, R., Wangler B. (Eds.), Perspectives in Business Informatics Research, Local Proceedings of 10th International Conference BIR2011 Associated Workshops and Doctoral Consortium, Riga Technical University, pp. 323 – 330. (2011)
- (Niedritis, Niedrite, 2012) Niedritis, A, Niedrite, L. Adaptation of the Presentation in a Multi-tenant Web Information System, In: Skersys, T., Butleris R., Butkiene R. (Eds.), Information and Software Technologies, CCIS , Vol. 319, Springer, pp. 176 – 186 (2012).
- (Niedritis et al., 2012) Niedritis, A., Niedrite, L., Zutens, J. Performance Measurement Framework with Indicator Life-cycle Support, In: A.Caplinskas, G.Dzemyda, A.Lupeikiene, O.Vasilecas (Eds.) Databases and Information Systems. Tenth International Baltic Conference on Databases and Information Systems. Local Proceedings, Materials of Doctoral Consortium. Vilnius, Žara, pp. 115-127, (2012).
- (Solodovnikova et al., 2012) Solodovnikova, D, Niedrite, L., and Niedritis, A. Query-Driven Method for Improvement of Data Warehouse Conceptual Model, Proceedings of the 21st International Conference on Information Systems Development (ISD2012), Prato, Italy, Springer (in print).

Autors prezentējis promocijas darba rezultātus zinātniskās konferencēs:

- Baltic DB&IS konferencē 2008. gadā (Medvedis et al., 2008) un 2012. gadā (Niedritis et al., 2012),
- ISD konferencē 2010. gadā (Niedritis, Niedrite, 2011) un 2012. gadā (Solodovnikova et al., 2012),
- 68. LU zinātniskā konferencē, Informācijas tehnoloģiju sekcijā, 2010. gada 19. februārī, prezentācija A. Niedritis, L. Niedrite, M. Treimanis „Biznesa procesu mērīšanas sistēmas arhitektūra”,
- BIR konferences UOII zinātniskajā darbnīcā 2011. gadā (Niedritis, 2011),
- ICIST konferencē 2012. gadā (Niedritis, Niedrite, 2012).

Par vienu no sistēmām, kurā aprobēti promocijas darba rezultāti, autors uzstājies specializētā konferencē EUNIS (European Universities Information Systems) ar sekojošiem ziņojumiem:

(Niedritis, Niedrite, 1998) A.Niedritis, L.Niedrite “Latvian University Information System”, Proceedings of the congress EUNIS98, Prague.

(Niedritis, 1999) A.Niedritis „Latvian University Information System security aspects in different application environments” Proceedings of the congress EUNIS99, Espoo

(Niedrite, Niedritis, 2003) L.Niedrite, A.Niedritis “The Development and Implementation of Unified Information System for Universities in Latvia – a Retrospective View and Conclusions”, Proceedings of the congress EUNIS2003, Amsterdam.

(Niedritis et al., 2007) Niedritis, A., Bumanis, A., Niedrite, L. Student and Employee Self Information/Registration Services. In Proceedings of the International Conference of European University Information Systems EUNIS 2007, Grenoble, (2007)

Promocijas darba rezultāti ir aprobēti Augstskolu informatīvajā sistēmā (skat. 3.5.1. punktu un 1.1. pielikumu) un Ceļu Satiksmes Drošības Direkcijas Transportlīdzekļu reģistrā (skat. 3.5.2. punktu un 1.2. pielikumu), Latvijas Universitātē (skat. 5.7.3. punktu un 1.3. pielikumu).

## 1.7. Promocijas darba struktūra

Promocijas darba pamattekstā ir 116 lappuses, pamattekstā ir 45 attēli un 6 tabulas. Promocijas darbs ir sakārtots 5 nodaļās.

Darba 1. nodaļā tiek definēta promocijas darba pētījuma sfēra un dots pamatojums tēmas izvēlei, kā arī tiek aprakstīti iegūtie promocijas darba rezultāti.

Darba 2. nodaļā tiek definēti ar tīmekļa informācijas sistēmām saistīti jēdzieni, kuri ir būtiski promocijas darbam. Nodaļas mērķis ir dot priekšstatu par tīmekļa informācijas sistēmu (TIS) dažādām eksistējošām arhitektūrām.

Darba 3. nodaļā aprakstīta Daudzu Nomnieku Adaptīvas Tīmekļa Informācijas Sistēmas arhitektūra DNATIS, definējot tās pamatprincipus, komponentus un to īpašības. Nodaļā dots vienas tīmekļa informācijas sistēmas piemērs, kas būvēts, balstoties uz piedāvāto DNATIS arhitektūru. Nodaļā aprakstīts arī, kā šādai TIS tiek veidota lietotāja saskarne, gadījumam, kad viens konkrēts lietotājs ir vairāku TIS instanču lietotājs.

Darba 4. nodaļā tiek piedāvāta metode, kas konstruē lietotāja modeli, balstoties uz lietotāju aprakstošo profilu jēdzienu, tiek izveidots integrēts, vispusīgs lietotāja modelis un metodes tā izveidei un datu uzkrāšanai. Integrētais, vispusīgais lietotāja modelis ir paredzēts izmantošanai DNATIS arhitektūrā kā lietotāja modelis. Tiek piedāvāta metode lietotāju

darbību analīzei datu avotos, kas izmantojama kā datu avots promocijas darba 5. nodaļā piedāvātās veikspējas mērīšanas sistēmas datu noliktavas modeļa evolūcijai.

Darba 5. nodaļā dotas biznesa procesu mērīšanas un veikspējas mērīšanas jēdzienu definīcijas. Apskatītas veikspējas mērīšanas sistēmas un analizēta datu noliktava kā viens no iespējamajiem veikspējas mērīšanas realizācijas veidiem. Nodaļā aprakstīts veikspējas mērīšanas ietvars ar indikatoru dzīves ciklu, tiek definēts indikatoru pierakstīšanas metamodelis, un aprakstīta metode, kā šo indikatoru formālo pierakstu izmantot veikspējas mērījumu ietvarā. Nodaļā aprakstīta divu līmeņu mērīšanas metode biznesa procesu mērīšanai. Risinājumi aprakstīti, izmantojot aprobācijai e-universitātes projektu un tajā iesaistītās informācijas sistēmas, metode piedāvāta kā atbalsts indikatoru izvēlei veikspējas pārvaldības ietvaram.

Rezultāti un secinājumi, kas iegūti, izstrādājot promocijas darbu, doti promocijas darba nobeigumā.

**Pateicības:** Pētījums veikts ar Eiropas Sociālā Fonda atbalstu, projekts No. 2009/0216/1DP/1.1.1.2.0/09/APIA/VIAA/044.

## 2. Tīmekļa informācijas sistēmas

### 2.1. Nodaļas nolūks

Šajā nodaļā sniegts apskats par tīmekļa informācijas sistēmām, raksturota to izstrādes specifika. Nodaļas mērķis ir sniegt priekšstatu par eksistējošām tīmekļa informācijas sistēmu arhitektūrām, detalizētāk aprakstot dažus kā, piemēram, programmatūra kā serviss, kura idejas tiek attīstītas promocijas darbā piedāvātajā arhitektūrā, kā arī tiek dots pārskats par adaptācijas jēdzieniem un tīmekļa informācijas sistēmu adaptācijas metodēm.

### 2.2. Tīmekļa sistēmas

Tīmeklis ir ne tikai informācijas avots, kas sastāv no pārsvarā statiskām tīmekļa lapām, bet ir arī kļūvis par lietojumprogrammu izstrādes platformu.

Ja apskata WWW un tā attīstību, var saskatīt, kā tas attīstījies no HTML bāzētas platformas, kas kombinē saturu ar informācijas prezentēšanu, kā arī var novērot tā hiperteksta dabu ar saitēm un navigāciju. (Houben, 2004).

#### 2.2.1. Hiperteksta un hipervides sistēmas

Hiperteksts ir “Datorizētas informācijas nelineāra pārlūkošanas un izguves sistēma, kas lietotājam ar īpaši izveidotu asociatīvu saišu (hipersaišu) palīdzību ļauj veikt informācijas izguvi no savstarpēji saistītiem dokumentiem neatkarīgi no to izvietojuma datora (datoru) atmiņā” (Letonika).

Hipervide ir “datu, teksta, grafikas un audio informācijas elementu integrācija hipertekstu sistēmā. Hipervidi veido hipersaites, kas atvieglo lietotājam pāreju no vienas informācijas formas uz citu” (Letonika). Hipervide būtībā ir hiperteksta vides teksta veida informācijas papildināšana ar cita tipa informāciju.

Pirmās zināmākās hiperteksta sistēmas (skat., piemēram, (Yankelovich et al. 1988)) bija ierobežotas ar lokālas vai dalītas failu sistēmas izmantošanu un paredzētas kādai noteiktai platformai. Kaut arī hiperteksta un hipervides sistēmas eksistēja jau iepriekš, šos jēdzienus plaši pazīstamus padarīja Tīmeklis (Berners-Lee et al., 1992).

Lai aprakstītu hiperteksta vai hipervides sistēmas nozīmīgas īpašības un jēdzienus, lai būtu pamats salīdzināt dažādās sistēmas un būtu iespējams izstrādāt standartus, tiek izmantoti hiperteksta vai hipervides references modeļi (Koch, 2000). Plašāk pazīstamais hipervides references modelis ir Dexter hiperteksta datu modelis (Halasz, Schwartz, 1990), kas būtībā piedāvāja hiperteksta sistēmu konceptuālo modeli, kurā tika iekļautas uz modeļa izveides brīdi eksistējošo hiperteksta sistēmu nozīmīgas abstrakcijas.

Kaut arī Dexter references modelis ir plaši izmantots, tas neapraksta visus iespējamus aspektus hipervides sistēmām, tajā skaitā nav adaptivitātes atbalsta. Tālāk tiks apskatīts tīmekļa informācijas sistēmas jēdziens un adaptīvas tīmekļa informācijas sistēmas jēdziens, kam ir izveidoti atbilstoši references modeļi, piemēram, AHAM references modelis (De Bra et al., 1999) un MRM jeb Munich references modelis (Koch, Wirsing, 2002).

#### 2.2.2. Tīmekļa Informācijas Sistēmas

Fraternali (1999) aprakstījis tīmekļa informācijas sistēmu (TIS) kā hibrīdu starp hiperteksta vidi un informācijas sistēmu. Tomēr jēdziens tika ieviests vēl gadu iepriekš. 1998.

gadā Tomas Isakowitz ar līdzautoriem radīja terminu tīmekļa informācijas sistēmas (TIS) - *Web Information Systems* (WIS)– kas apzīmē: “informācijas sistēmas, kas bāzētas tīmekļa tehnoloģijās” (Isakowitz et al., 1998).

Panhilla (2006) aprakstījis TIS kā sistēmas, kas ir izstrādātas ar tīmekļa tehnoloģijām, tās strādā ar pārlūkprogrammu, un ir fokusētas uz uzdevumu veikšanu, un liek uzsvāru uz nepārtrauktu lietotāja mijiedarbību ar sistēmu. Tīmekļa Informācijas Sistēma vispirms ir jāuztver kā informācijas sistēma, un tikai pēc tam kā tīmekļa sistēma. Līdzīgu domu pāudis Jazayeri (2007): „Tīmekļa lietotne ir lietotne, kas tiek darbināta ar tīmekļa pārlūku Internetā”.

Tīmekļa informācijas sistēmas izmanto tīmekļa tehnoloģijas, lai realizētu profesionālu informācijas sistēmu prasības (Frasincar, 2002). Pielietojot TIS biznesa kontekstā, tam seko vajadzība pielietot tieši vai pielāgot principus, kas tiek lietoti IS izstrādē, kas balstās uz datubāzēm.

Houbens (2004) raksta, ka TIS atšķiras no tradicionālām statiskām tīmekļa lapām jeb „virsmas tīmekļa”, kam ir manuāli radīts saturs. Viena no būtiskām atšķirībām ir t.s. „dziļais tīmeklis” ar datu bāzē glabātu saturu, TIS ir nepieciešams rīkoties ar datiem, kas ir dinamiski, tiek aplūkoti shēmas līmenī, un tie tiek lietoti lielos apjomos.

Daudzām TIS ir datu-vadīta izcelsme, tādēļ tām ir nepieciešams automātiski ģenerēt datu izvades prezentāciju hipervidē (Frasincar, 2002), t.i. TIS ģenerē tīmekļa lapas, balstoties uz datiem, kas tiek iegūti no datu avota.

TIS pēc sarežģītības var būt dažādas, sākot no funkcionāli nelielām, līdz ar to ātri un vienkārši izstrādājamām lietotnēm, un beidzot ar uzņēmuma līmeņa sistēmām. Tīmekļa lietotņu pielietojums var būt plašs, piemēram, darba plūsmu tipa sistēmas (noliktavas pārvaldības u.c. sistēmas), transakciju sistēmas (internetveikali, internetbankas, biļešu rezervēšanas sistēmas u.c.), sadarbības rīki (rīki sadarbībai starp grupas dalībniekiem).

Var runāt par trim pakāpēm TIS attīstībā. Sākotnējās TIS tika manuāli radītas kā hiperteksta sistēmas. Nākošais etaps TIS attīstībā ir ģenerētas, dinamiskas TIS, kur noteikta veida šabloni tiek aizpildīti ar datiem pēc pieprasījuma. Tālāk TIS attīstās virzienā, kas nodrošina datu saprotamību automātiskai apstrādei, par pamatu ņemot semantiskā tīmekļa tehnoloģijas.

TIS atšķiras ne tikai no „virsmas tīmekļa” sistēmām, bet arī no tradicionālām informācijas sistēmām, jo tām ir “potenciāls sasniegt plašāku auditoriju, izmantojot dažādas platformas” (Isakowitz et al., 1998). Pastāv pieaugoša nepieciešamība integrēt datus no dažādiem datu avotiem, kas izplatīti tīmeklī un ir heterogēni. Tāpat līdz ar plašāku iespējamo auditoriju rodas nepieciešamība pēc šo sistēmu pielāgošanas dažādu lietotāju vajadzībām.

Fraternali ir definējis prasības tīmekļa informācijas sistēmām (Fraternali, 1999):

- TIS nodrošina gan strukturētu datu (piemēram, datu bāzes ierakstu), gan nestukturētu datu apstrādi,
- piekļuve informācijai tīmekļa informācijas sistēmās notiek ar navigācijas tipa saskarnes palīdzību,
- tiek nodrošināta augsta līmeņa grafiskā kvalitāte,
- pielāgošanas iespēja un iespēja dinamiski adaptēt satura struktūru, navigācijas primitīvus un prezentācijas stilus,
- atbalsts rekomendācijām un filtrēšanai.

Pēc šī pētījuma ir pagājis laiks un pētījumu tālākā attīstība saistīta ar Web 2.0 jēdzienu. Salīdzinot ar tradicionālām tīmekļa lietotnēm (Web 1.0), jauna tipa lietotnes (Web 2.0) ir dinamiskas, tās par būtisku sastāvdaļu pieņem lietotāja iesaistīšanos satura radīšanā (piemēram, blogi, wikipēdija) un reaģēšanas ziņā uz lietotāja pieprasījumiem ir līdzvērtīgas lokāli lietojamām programmām.

Saskaņā ar Fraternali (1999) prasībām, var runāt par sekojošām TIS īpašībām:

- TIS ir datu ietilpīgas, jo TIS jānodrošina gan nestrukturētu, gan strukturētu datu apstrādi. Tai pat laikā TIS integrē datus no daudziem heterogēniem datu avotiem,
- TIS ir visur izplatītas, kas šajā gadījumā nozīmē, ka tīmekļa tehnoloģijas, valodas un metodoloģijas rada iespēju izstrādāt dinamiskas lietotnes, kas atspoguļo jaunu lietošanas un sadarbības modeļi starp ļoti daudziem lietotājiem (Jazayeri, 2007). Nākotnes tīmekļa lietotnes būs saistītas ar jauniem risinājumiem tīmekļa pārlūkiem, interneta infrastruktūrai, protokolu standartiem, kā arī ar programminženierijas metožu attīstību un jauna tipa lietotņu rašanos,
- TIS ir pielāgojamas, tām ir jāņem vērā lietotāja vēlmes. Pielāgojamība izpaužas veidā, kā sistēma mijiedarbojas ar katru konkrēto lietotāju. Šīs lietotnes atbalsta gan individuālus lietotājus un mazos uzņēmumus, gan lietotāju apvienības un lielu uzņēmumu darbību. Izstrādājot TIS jāņem vērā liels lietotāju skaits, kas nosaka dažādu lietotāju izmantoto pārlūkprogrammu veidus un dažādu lietotāju atšķirīgu datorprasmju līmeni, un līdz ar to atsevišķi jādomā par prezentācijas modelēšanu un izstrādi. Lielais lietotāju skaits izvirza īpašas prasības pret informācijas sistēmas veiktspēju, drošumu un kvalitāti. Jaunās paaudzes lietotnes piedāvā lietotājiem ne tikai komunikācijas un sadarbības iespējas, bet arī iespēju ietekmēt un mainīt lietotnes īpašības.

### 2.2.3. *Adaptīvas hipervides sistēmas*

Tīmeklis ir radījis jaunas iespējas piekļuvei datiem. Houbens (2004) atzīst, ka līdz ar Tīmekļa izmantošanu IS izstrādei, satura loma informācijas sistēmās ir mainījies. IS izstrādes metodoloģijas bieži ir palikušas nemainīgas, tai pat laikā nepieciešams izmantot jaunās iespējas, ko piedāvā tīmeklis – metadatus, mijiedarbību, adaptāciju.

Brusilowsky (1996) definēja adaptīvu hipervides sistēmu: „Ar adaptīvu hipervides sistēmu saprot visas hiperteksta un hipervides sistēmas, kas atspoguļo atsevišķas lietotāja īpašības lietotāja modelī un izmanto šo modeli, lai adaptētu dažādus redzamus sistēmas aspektus atbilstoši lietotājam. Citiem vārdiem sakot, sistēmai jāatbilst trim kritērijiem: tai jābūt hiperteksta vai hipervides sistēmai, tai ir jābūt lietotāja modelim, un tai jāvar adaptēt hipervidi atbilstoši šim modelim”.

Adaptīva sistēma pielāgo mijiedarbību, balstoties uz lietotāju un viņa vajadzībām. Tiek tomēr atšķirti divi jēdzieni – adaptīvas un adaptējamās sistēmas (Brusilowsky, Maybury, 2002). Abos gadījumos būtiska ir lietotāja ietekme uz sistēmu un tiek izveidota lietotāja vajadzībām atbilstoša sistēma, atšķiras tikai veids, kā sistēma tiek adaptēta (Koch, 2000). Līdz ar to abu līdzīgo jēdzienu definīcijas ņem vērā lietotāja modeļa ietekmi uz sistēmas adaptāciju (Brusilowsky, Maybury, 2002), (Koch, 2000):

Adaptējama (*adaptable*) sistēma prasa, lai lietotājs pats specificē, kā sistēmas darbībai ir jāatšķiras. Lietotājs var konfigurēt dažus parametrus un sistēma tam atbilstoši maina savu darbību. Ārēja sistēma vai pats lietotājs pieņem lēmumu, kad un kā ir jāmaina lietotāja modelis. Konfigurēšana nozīmē šajā gadījumā iestatījumu (preferenču) vērtību maiņu.

Adaptīvas (*adaptive*) sistēmas automātiski pielāgojas lietotājam, ja ir dots lietotāja modelis. Sistēma monitorē lietotāja darbības, reģistrē šīs darbības un dinamiski adaptē sistēmu atbilstoši aktuālajam lietotāja modelim. Lai veidotu lietotāja modeli, sistēma ņem vērā gan pārlūkošanas darbības, gan lietotāja sniegtas tiešas atbildes uz jautājumiem, piemēram, sākotnējās informācijas veidā, ko lietotājs sniedz, pirmo reizi uzsākot savu darbu ar sistēmu.

Pētniecības jomā hiperteksta vidē ilgstoši viens no nozīmīgiem virzieniem bija adaptācijas problēmas, kas ļauj ņemt vērā tur gūtās atziņas adaptīvām TIS (Houben, 2004). Tika pētīta gan iekārtu adaptācija, gan lietotāju modelēšana, gan datu integrācija, gan

mijiedarbība un dinamika, kā arī kvalitāte. Adaptīva hiperteksta vide nozīmē gan satura adaptāciju, ko nodrošina adaptīva prezentācija, gan saišu jeb navigācijas adaptāciju. Tika radītas dažādas Adaptīvas Hipervides Sistēmas (AHS).

Arī Houbens (2004) izšķir divus jēdzienus: pielāgošanās spēju (*adaptability*) un pielāgojamību (*adaptivity*). Pielāgošanās spēja balstās uz statistisku lietotāja profilu, TIS piegādā lietotājam informāciju saskaņā ar lietotāja profila nosacījumiem, lietotāja informācija var tikt šai gadījumā savākta tiešas reģistrācijas ceļā vai atsevišķas īpašības izsecinot no līdzīgu lietotāju grupām. Pielāgojamība balstās uz dinamisku lietotāja profilu, kur lietotāja modelis tiek veidots, analizējot lietotāja pārlūkošanas paradumus, piemēram, kā to varētu darīt e-studiju vidēs.

Brusilovskis savos darbos (Brusilovsky et al., 1998), (Brusilowsky et al., 2000) un (Brusilowsky, Maybury, 2002) izdala trīs paaudzes adaptīvām sistēmām:

- pirmās paaudzes (“pirms- Tīmekļa”) adaptīvās hipervides sistēmas pamatā pievērsās adaptīvai prezentācijai un adaptīvam navigācijas atbalstam un koncentrējās uz lietotāja zināšanu un mērķu modelēšanu,
- otrās paaudzes (Tīmekļa) adaptīvās sistēmas paplašināja adaptīvās hipervides sistēmas ar adaptīvu satura atlasī un rekomendācijām, balstoties uz lietotāju interešu modelēšanu,
- trešās paaudzes (mobilās) adaptīvās sistēmas paplašina adaptāciju, pievienojot konteksta modeļus (vietu, laiku, platformu) klasiskajiem lietotāju modeļiem, un veic pielāgošanu ne tikai lietotājam, bet arī kontekstam, kādā lietotājs strādā ar sistēmu.

Adaptācijas specififikācijas sākotnējās AHS bija visai vienkāršas, galvenais bija precīzi aprakstīt lietotāju un domēnu. Dažādās adaptīvās sistēmās adaptācijas efekti var tikt realizēti dažādos veidos. Adaptīvas hipervides un tīmekļa sistēmas pamatā ir saistītu informācijas vienumu kolekcijas, kas ļauj lietotājam navigēt no viena vienuma uz otru un meklēt interesējošu informāciju. Adaptācija šai gadījumā ir ierobežota ar trim adaptācijas veidiem: adaptīva satura izvēle, adaptīvs navigācijas atbalsts, adaptīva prezentācija (Brusilowsky, Maybury, 2002).

Hipervides sistēmām, eksistēja references modeļi (būtībā konceptuālie modeļi), ko izmantot par pamatu, lai specifificētu jaunas sistēmas. AHS konceptuālie modeļi nereti tika veidoti, balstoties uz Dexter references modeli, tādēļ tas tiks apskatīts tuvāk.

Dexter references modelis hiperteksta sistēmām paredzēja trīs galvenos slāņus (Halasz, Schwartz, 1990): Izpildes slāni (*Run-Time*), komponenta slāni (*Within-Component*) un glabātuves slāni (*Storage*). Saistību starp slāņiem nodrošina divi saskarnes slāņi: prezentācijas specififikācijas (*Presentation Specifications*) un enkuru (*Anchoring*) slānis. Glabātuves slānis modelē galveno mezglu un saišu struktūru. Komponenta slānis apraksta konkrētu komponentu (no glabātuves slāņa) saturu un struktūru (piemēram, lapas fragmentus), bet enkuru slānis ļauj vērsties pie konkrētām vietām komponentē, kas definēta glabātuves slānī. Izpildes slānis attēlo glabātuves slāņa komponentus lietotājam. Viens no svarīgiem jēdzieniem šajā slānī ir komponenta instances veidošana, kad komponents tiek prezentēts lietotājam. Saistību starp izpildes slāni un glabātuves slāni definē prezentācijas specififikācijas slānis.

Adaptīvas hipervides piemērs ir AHAM (De Bra et al., 1999). AHAM balstās uz Dexter modeli hipervidēm (Halasz, Schwartz, 1990). AHAM definē arhitektūru adaptīvai e-apmācības videi, balstās uz lietotāja, apmācīšanas un domēna jeb konceptuālo modeli, un adaptācijas specififikāciju, kas nosaka, kā saturs prezentējams lietotājam. Tiek ieviests papildināms lietotāja modelis, kas tiek aktualizēts ar katru nākošo lietotāja sesiju darbā ar sistēmu.

De Bra et al., (1999) uzsver principu adaptīvām hipervides sistēmām, tai skaitā arī AHAM modelī, kas tālāk tiek izmantots arī adaptīvām TIS, ka jāatdala sekojoši vienumi:

- 1) domēna modelis, kas apraksta kā informācija ir strukturēta un savstarpēji saistīta,

2) lietotāja modelis, kas apraksta, kāda informācija par lietotāju glabā adaptīvā hipervides sistēma, ieskaitot zināšanu reprezentāciju, ko lietotājs iegūst, kā arī navigācijas vēsturi,

3) pedagoģiskais modelis – likumu bāze, kas apraksta, kā domēna modelis un lietotāja modelis tiek kombinēti adaptācijas nolūkos,

4) adaptācijas modulis, kas realizē adaptāciju, kas adaptē (jeb dinamiski ģenerē) saturu un izveido saišu kopu, kas paredzēta individuāla lietotāja vajadzībām navigācijai.

Dexter un AHAM modeļu idejas tālāk tiek attīstītas MRM modelī (*Munich Reference Model*) adaptīvām hipervidēm (Koch, Wirsing, 2002), kur pamatā ir objektorientēta pieeja, arhitektūras aprakstam tiek izmantots UML.

Adaptīvas hipervides sistēmas bieži tiek ņemtas par pamatu, lai izstrādātu TIS arhitektūru. Kad pāriet no TIS uz ATIS, jāsaprot, ko nozīmē adaptācija TIS eksistējošai navigācijas un pārlūkošanas paradigmai. Navigācijas adaptācija ir būtiska, lai padarītu vajadzīgo saturu pieejamu. Atšķirībā no satura, praktiski nav modeļu navigācijas aprakstam un kā risinājumu Houbens (2004) iesaka izmantot Lietotnes modeli (*Application model*) no Hera (Frasincar et al., 2002) metodoloģijas. Adaptācijas jēdzieni ATIS tiks skatīti tālākās apakšnodaļās.

#### **2.2.4. Adaptīvas Tīmekļa informācijas sistēmas**

Tīmekļa informācijas sistēmas raksturo datu vadīta izcelsme, tām piemīt arī tradicionālo informācijas sistēmu īpašības, kas realizē biznesa loģiku.

TIS kontekstā mēdz runāt par TIS inženieriju. TIS Inženierijā (*Web Information Systems Engineering - WISE*) tiek uzskatīts, ka būtiskākā TIS īpatnība ir datu bāzē glabāta satura izmantošana. TIS ir jārikojas ar datiem, kas ir dinamiski, tiek aplūkoti shēmas līmenī, un lietoti lielos apjomos. Ja izmanto TIS biznesa kontekstā, rodas vajadzība pielāgot principus, kas tiek lietoti datubāzu IS inženierijā (Houben, 2004).

Tādi aspekti, kā datu integrācija un adaptācija, kas ir tipiski tīmekļa informācijas sistēmām, padara sarežģītāku izstrādes procesu. Frasincar et al. (2002) pamatoja nepieciešamību TIS izstrādes ietvariem, norādot, ka „lai atbalstītu šādu datu-vadītu TIS inženieriju, ir nepieciešami izstrādes ietvari, kas ļautu izstrādātājam specificēt TIS piemērotā abstrakcijas līmenī, atbilstoši dažādām TIS inženierijas projekta fāzēm”. Nepieciešams ņemt vērā gan datu modelēšanu, gan navigācijas modelēšanu, kā arī, ja tiek veidota adaptīva TIS, nepieciešams specificēt adaptāciju.

Adaptācija (un personalizācija) ir izstrādes aspekts, kas ir ieguvis plašu uzmanību, tomēr dažādās metodoloģijās TIS inženierijai adaptācijas aspekts ir joprojām par maz atbalstīts (Houben, 2004). Praksē adaptāciju ir grūti specificēt, sarežģīti ir kombinēt adaptāciju ar citiem TIS izstrādes aspektiem, piemēram, veikspēju. Adaptācijas specificēšanai dažādas metodoloģijas ļauj definēt likumus, kā informācija attēlojama lietotājam.

Runājot par vienu no metodoloģijām TIS izstrādei – Hera - Frasincar et al. (2002) precizē, ko šīs metodes kontekstā autori (Frasincar et al., 2002) saprot ar adaptāciju. Tiek izdalītas divas iespējas. Pirmajā gadījumā tiek adaptēta ģenerēšana. Ģenerēšanas rezultāts – prezentācija – ir atkarīga no lietotāja modeļa, kas ietver gan platformas aprakstu, gan lietotāja iestatījumus u.c. Otrajā gadījumā ģenerētā prezentācija pati var būt adaptīva, t.i. var mainīties, kad lietotājs to pārlūko.

Pirmo gadījumu var uzskatīt par personalizāciju, bet Hera autori atzīmē, ka šis jēdziens nav īsti precīzs, dažkārt tā vietā tiek lietots jēdziens „adaptējama” (kā pretstats jēdzienam



„adaptīva”), bet Hera metodoloģijas ietvaros netiek ņemta vērā „šī smalkā, neskaidrā atšķirība” (Frasincar et al., 2002). Jēdziena „adaptējama” skaidrojumā autori izmanto kā sinonīmu arī jēdzienu „konfigurējama”, rakstot, ka „ģenerēšanas process tiek konfigurēts, izmantojot informāciju, kas apraksta situāciju, kādā lietotājs izmantos ģenerēto sistēmu”.

Lai sistēma, kas balstīta uz tīmekļa tehnoloģijām, piegādātu informāciju, kas atbilstu gan lietotāja vajadzībām, gan apstākļiem, kādos lietotājs informāciju ir pieprasījis, sistēmai ir nepieciešams spēt adaptēt un personalizēt informācijas piegādi. Lai aprakstītu saturu (informāciju) TIS izmanto metadatus, bieži šim nolūkam tiek izmantotas ontoloģijas.

Rakstā (De Virgilio, Torlone, 2005) ir definēts, ka tīmekļa-bāzēta informācijas sistēma ir adaptīva, ja tā ir spējīga mainīt un personalizēt satura un servisu piegādi atbilstoši klienta kontekstam. Personalizācija pieprasa lietotņu adaptāciju tik lielā mērā, cik vien iespējams, atbilstoši lietotāja iestatījumiem un lietotāja kontekstam (Abbar et al., 2008).

Adaptīvām TIS būtiski jēdzieni, kas tiek izmantoti, lai aprakstītu, kā notiek adaptācijas process, ir konteksts, profils un konfigurācija. Dažādi autori šai jomā definē šos jēdzienus atšķirīgi. Daudzas arhitektūras un pieejas (De Virgilio, Torlone, 2005), (Valeriano et al., 2006), (Abbar et al., 2008), (De Bra et al., 1999), (Tvarožek et al., 2007), (Jablonski et al., 2004), kas ir piedāvātas TIS adaptācijai, tiek bāzētas uz dažādu izpratni par kontekstu, profilu un konfigurāciju, kā arī katrā no piedāvātajiem risinājumiem tiek aplūkotas dažādas šo jēdzienu īpašības.

Konteksta jēdziens ieviests, lai apzīmētu “atribūtu kopu, kas raksturo piekļuves mehānisma iespējas, konfigurāciju, lietotāja iestatījumus, un vides nosacījumus” (W3C), kādā ir jāpiegādā tīmekļa lapa.

Kontekstu (Abbar et al., 2008) var uzskatīt par īpašību kopu, kas apraksta vidi, kurā lietotājs mijiedarbojas ar TIS, piemēram, Abbar et al. (2008) kontekstu ir definējis ar laika, vietas, iekārtas, lietotāja un vides dimensijām. Katrai dimensijai, savukārt, var būt savi aprakstoši atribūti (De Virgilio, Torlone, 2005). Var kontekstu definēt arī kā visu profilu kopu (De Virgilio, Torlone, 2005).

Profila jēdzienu izmanto, lai reprezentētu dažādas konteksta dimensijas dažādās detalizācijas pakāpēs (De Virgilio, Torlone, 2005).

Profilam jāietekmē satura prezentāciju. Piemēram, tipiski profili TIS gadījumā ir iekārtas, vietas u.c. profili. Citi autori (Abbar et al., 2008) definē profilu kā īpašību kopumu, kas raksturo lietotāju vajadzības pēc datiem. Īpaši apskatāms jēdziens „lietotāja profils”, par ko detalizētāk skaidrots punktā 2.2.5.

Konfigurācija specificē, kā jāģenerē tīmekļa lapas: kā informācija ir jāpiegādā lietotājam, ņemot vērā lietotāja pieprasījumu un viņa kontekstu, atbilstoši viņa profilā aprakstītām adaptācijas prasībām. (De Virgilio, Torlone, 2005). Konfigurācijas var tikt kombinētas, lai integrētu dažādas adaptācijas prasības. Konfigurācija ir piekārtota noteiktam profilam.

Personalizācijas paradigma paredz adaptēt lietotnes cik vien iespējams atbilstoši lietotāja iestatījumiem un lietotāja kontekstam (Abbar et al., 2008). Lietotnes adaptācija šai gadījumā var mainīt sistēmas konfigurāciju, datu avotu izvēli, datu attēlojumu, vaicājumu pārformulēšanu, lietotāju reakcijas uzkrāšanu, un citus aspektus. Ja tuvāk apskata tieši datus, tad datu personalizācija attiecas uz tehnikām, kas atļauj nodrošināt lietotāju ar viņam visatbilstošākajiem datiem, atkarībā no interešu sfēras, datu kvalitātes prasībām, atrašanās vietas un laika vaicājuma izpildes brīdī, datu attēlošanas ierīces u.c.

Personalizācija kā jēdziens tiek lietots dažādās jomās, kā, piemēram, informācijas izguves, datu bāzes jomās un tiek aprakstīts tieši saistībā ar saturu. Būtiski ir saprast, kā notiek personalizācijas process. Lai piedāvātu personalizētas rekomendācijas lietotājam, ir nepieciešams modelēt lietotāja īpašības un vajadzības. Šī informācija literatūrā ir pazīstama ar lietotāja modeļa jēdzienu (Kobsa, 2001).

Informācijas izguves sistēmās personalizācija ir mašīnmācīšanās process (Abbar et al., 2008), kas balstīts uz lietotāja atbildes reakciju uz piedāvātajiem datiem, kā rezultātā sākotnējais lietotāja profils tiek papildināts atbilstoši viņa darbību analīzei. Rezultātā informācijas atlase tiek pielietota atbilstoši arvien precīzētiem lietotāja modeļiem.

Datu bāzu sistēmās personalizācija ietekmē lietotāju pieprasījumus datiem. Zināmākie veidi ir vaicājumu valodas paplašināšana ar lietotāju preferenču datiem (Preference SQL) (Kiesling, 2002), un lietotāju datu bāzu vaicājumu izvēršana (*query expansion*), izmantojot lietotāju profilu datus (Koutrika, Ioannidis, 2004).

Lai arī Tīmekļa informācijas sistēmās, ir ļoti plašas iespējas personalizācijai (atkarībā no konteksta un profiliem), parasti tiek apskatīti tikai daži aspekti, pie tam piedāvātie risinājumi ir iebūvēti kādā konkrētā sistēmā. Nav darbu, kas apskatītu personalizācijas servisu, kas būtu pietiekami vispārīgi un pielietojami daudzās lietotnēs (Abbar et al., 2008).

Personalizācijas process ir atkarīgs no lietotnes domēna un no lietotajām tehnoloģijām (Abbar et al., 2008)

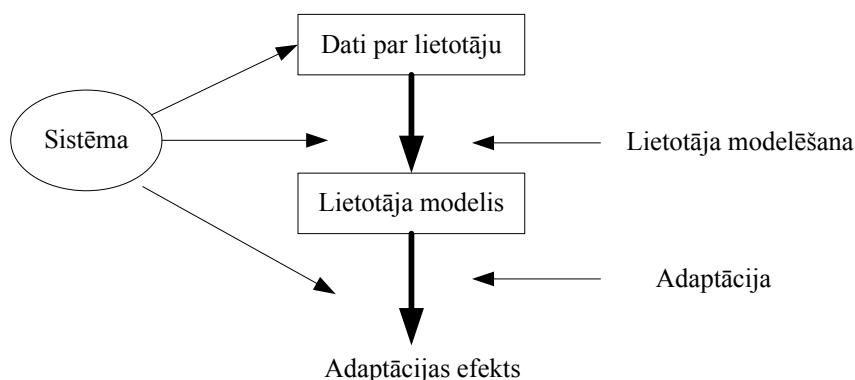
### 2.2.5. Lietotāja modeļa vieta adaptācijas procesā

Adaptīvām sistēmām raksturīgs ir skaidri definēts lietotāja modelis, kas reprezentē lietotāja zināšanas, mērķus, intereses un citas īpašības, kas sistēmai ļauj atšķirt dažādus lietotājus.

Adaptīva sistēma savāc datus lietotāja modelim no dažādiem avotiem, kas var ietvert gan netiešu lietotāja novērošanu, kad lietotājs mijiedarbojas ar sistēmu, gan tieši pieprasot no lietotāja datu ievadi par sevi. (Brusilowsky, Maybury, 2002). Lietotāja modeļa vieta sistēmas mijiedarbībā ar lietotāju un sistēmas adaptācijā redzama 1. attēlā.

**Adaptācijas efekts** ir sistēmas atšķirīga darbība dažādiem lietotājiem - sistēmas darbība tiek pielāgota dažādiem lietotājiem. (Brusilowsky, Maybury, 2002).

Lai panāktu adaptācijas efektu, tiek izmantots sistēmas rīcībā esošs lietotāja modelis.



1. attēls. Lietotāja modeļa vieta sistēmas adaptācijā (pēc Brusilowsky, Maybury, 2002)

**Lietotāja modelis** ir lietotāja abstrakta reprezentācija. Abstrakta reprezentācija ir tādēļ, ka tikai noteiktas, sistēmai būtiskas īpašības tiek iekļautas modelī. Lietotāja modelī lietotājs

tiek reprezentēts ar datu kolekciju. Būtībā tiek radīts sistēmas priekšstats par lietotāju. (Koch, 2000).

Lietotāja modelis tiek izveidots atbilstoši viņa mijiedarbībai ar sistēmas lietotāja saskarni. Lietotāja modelis tiek bāzēts tikai uz šo informāciju un ierobežoti atspoguļo reālo lietotāju. Tomēr lietotāja modelim jāatspoguļo lietotāja īpašības attiecībā pret lietotnes kontekstu.

**Lietotāja modelēšana** ir process, kas pārklāj visu lietotāja modeļa dzīves ciklu, ieskaitot zināšanu iegūšanu par lietotāju, modeļa izveidi, aktualizēšanu, uzturēšanu un izmantošanu. (Koch, 2000)

Saistīts jēdziens ir lietotāja profils. Jēdzienus “lietotāja modelis” un “lietotāja profils” bieži lieto kā sinonīmus. Dažkārt tomēr tie tiek lietoti ar atšķirīgu nozīmi.

Viena no iespējamām atšķirību interpretācijām balstās uz dažādu informācijas uzkrāšanas veidu lietotāja modelim un lietotāja profilam. Lietotāja profils šajā gadījumā satur lietotāja tieši norādītu informāciju, pretstatā informācijai lietotāja modelim, kas var tikt uzkrāta, novērojot sistēmas lietotāju mijiedarbības laikā.

Cita definīcija, kuru izmanto arī Koch, pieņem, ka „**lietotāja profili** ir vienkāršoti lietotāja modeļi” (Koch, 2000). Tie tiek lietoti, lai piemēram, reprezentētu kādu atsevišķu lietotāja aspektu (vai dažus aspektus) – izziņas īpatnības, intelektuālās spējas, mērķus u.c. Tātad var teikt, ka lietotāja profils un lietotāja modelis ir ar atšķirīgu sarežģītības pakāpi.

Promocijas darba ietvaros arī tiek izmantota šī lietotāja profila definīcija.

Brusilowsky (1996) ir definējis, kādu lietotāja īpašību reprezentācija ir adaptīvas hipervides sistēmas lietotāja modelis. Informācija tiek uzkrāta par sekojošām īpašībām:

- iestatījumi (preferences),
- uzdevumi, mērķi, intereses,
- zināšanas par domēnu,
- pieredze un priekšzināšanas.

Šo sarakstu var papildināt gan ar personiskām īpašībām, kas var ietekmēt lietotāja darbību sistēmā, piemēram, mācīšanās stils u.c., kā arī konteksta informāciju. Tāpat no sistēmas veida un pielietojuma sfēras lietotāja modelis var tikt papildināts ar citu īpašību reprezentāciju. Tātad lietotāja modelis apraksta lietotāju no dažādiem aspektiem, tajā skaitā, mijiedarbības, ģeogrāfiskās vietas, demogrāfiskiem u.c. aspektiem, kas ir būtiski dotās sistēmas kontekstā.

Lietotāja profila dzīves cikls sastāv no trim fāzēm: (i) informācijas vākšanas par lietotāju, (ii) integrēta skatījuma par lietotāju veidošana (integrēta profila izveidošana) un (iii) lietošanas, kas piedāvā personalizētu servisu lietotājam, balstoties uz lietotāja profilu (Gauch et al., 2007).

Houbens (2004) lietotāja modelēšanas jomā piemin vairākas problēmas, tajā skaitā, nav skaidrs, ko tieši glabāt lietotāja modelī, jo dažādu autoru viedokļi atšķiras, pie tam katrai TIS var būt specifiskas prasības. Daudzos gadījumos nav sākotnējo datu par lietotāju, un salīdzināšana ar citiem lietotājiem nav izmantojama, kad sistēma tikai uzsāk darbu – t.s. *Cold Start* problēma. Tāpat aktuāli ir apmainīties ar informāciju par lietotāju ar citām TIS, nepārkāpjot privātumu. Būtiska ir arī identificēšanas problēma, tai skaitā, ko no informācijas ļaut labot pašam lietotājam.

Lietotāju modeļi tiek izmantoti, lai varētu realizēt sistēmu adaptāciju (Brusilowsky, 2001), lai veidotu personalizētas atbildes uz lietotāju jautājumiem (Koutrika, Ioannidis, 2004), tie tiek izmantoti dažādu tipu sistēmās, piemēram, rekomendāciju sistēmās, meklēšanas dzīņos (search engines), u.c. Arī informācijas integrācijas jomā, piemēram, datu noliktavās,

tiek pētīti personalizācijas aspekti (Jerbi et al., 2009), (Golfarelli, Rizzi, 2009) un lietotāju modeļi tiek izmantoti personalizācijas nolūkos.

Dažas pieejas izmanto ontoloģiju, lai veidotu lietotāja modeli (Razmerita et al., 2003), (Zhang et al., 2007). Citi autori piedāvā arhitektūras, kā konstruēt un pielietot lietotāja modeli (De Bra et al., 2004).

Personalizācijas kvalitāte ir atkarīga no lietotāja modeļa un tajā uzkrātās informācijas par lietotāju, ir noderīgi papildināt lietotāja modeli ar importēšanas un integrēšanas iespējām, no dažādiem citiem lietotāja modeļiem, kas būvēti citās personalizācijas sistēmās (Berkovsky et al., 2009). Berkovsky et al. (2008) piedāvāja starpniekplatformu, kas veiktu lietotāja modeļa datu piegādi sistēmai, kam nepieciešams veikt personalizāciju, savācot daļējus lietotāja modeļus no citām iesaistītajām sistēmām.

Niederee et al. (2004) ierosina izmantot vienu un to pašu lietotāja modeli dažādās sistēmās, bet papildinātu un uzlabotu. Autori piedāvā izmantot universālu lietotāja konteksta modeli, kas savāc daļējus lietotāja modeļa datus no individuālām personalizācijas sistēmām. Apkopoto informāciju pēc tam katra iesaistītā sistēma var saņemt atpakaļ, lai papildinātu lietotāja modeļa datus, kas ir izmantojami, kad nepieciešams veikt personalizāciju. Līdzīgu sistēmu ar centralizētu lietotāja modeli, ko izveido, apkopojot individuālos lietotāja modeļus, un aktualizē katra iesaistītā personalizācijas sistēma, piedāvā (Kay et al., 2003).

#### **2.2.6. Adaptācijas efektu izpausmes veidi**

Jau iepriekš tika minēts, ka dažādās adaptīvās sistēmās adaptācijas efekti var tikt realizēti dažādos veidos. De Bra (1999) savā AHAM modelī runā par satura adaptāciju un saišu adaptāciju, bet Brusilowsky (1996) šos adaptācijas veidus sauc par adaptīvu prezentāciju un adaptīvu navigāciju.

Adaptācija adaptīvas hipervides sistēmas gadījumā ir ierobežota ar trim adaptācijas veidiem: adaptīva satura izvēle, adaptīvs navigācijas atbalsts, adaptīva prezentācija (Brusilowsky, Maybury, 2002).

Satura adaptācijas paņēmieni, kas minēti dažādu autoru darbos (De Bra et al., 1999), (Brusilowsky, 1996) ir sekojoši:

- papildus skaidrojumi, priekšnoteikumu skaidrojumi un salīdzinoši paskaidrojumi,
- skaidrojumu varianti,
- satura (fragmentu) kārtošana,
- teksts ar nosacījumu,
- fragmentu varianti,
- lapu varianti,
- teksta fragmentu spilgtuma maiņa,
- paplašināms teksts (stretchtext), kas nozīmē, ka dažādiem lietotājiem var tikt parādīts teksts atšķirīgā detalizācijas pakāpē, atkarībā no priekšzināšanām.

Daudzi no minētajiem satura adaptācijas paņēmieniem, kas sākotnēji bija domāti tieši teksta veida satura adaptācijai, ir attiecināmi arī uz cita veida satura adaptāciju, piemēram, attēlu iekļaušana/neiekļaušana saturā.

Saišu adaptācijas paņēmieni, kas minēti dažādu autoru darbos (De Bra et al., 1999), (Brusilowsky, 1996) ir sekojoši:

- vadība, gan lokāla, gan globāla,

- globālas vai lokālas orientēšanās atbalsts,
- personalizēta skatījuma izveide uz saišu struktūru,
- tieša vadība, piemēram, ar atbilstošām pogām, lietotājam piedāvā pēc sistēmas skatījuma vienu piemērotāko nākošo darbību,
- saišu kārtošana, kur lietotājam saites tiek parādītas sakārtotas pēc atbilstības lietotājam,
- saišu apslēpšana, ja tās nav piemērotas (saite tiek parādīta kā teksts),
- saišu iekrāsošana dažādās krāsās, atbilstoši piemērotībai,
- nepiemērotu saišu padarīšana par neizmantojamām (saiti parāda kā saiti, bet tā nedarbojas kā saite),
- saišu izņemšana no satura (neatbilstošas saites saturā neparādās vispār),
- lapas kartes personalizācija.

### **2.2.7. TIS inženierija un adaptācija**

Lai TIS adaptētu saskaņā ar lietotāja zināšanām, vajadzībām, lietošanas mērķiem, ir nepieciešams modelēt lietotāju, izveidojot lietotāja īpašības aprakstošu lietotāja modeli.

Vēl no pētniecības problēmām Houbens (2004) nosauc arī datu integrācijas problēmas, kur būtisks novērojums ir, ka tiek ekspluatētas idejas no dalītu informāciju sistēmu un datu bāzu jomas, kas rada risku, ka integrācija tiek attiecināta uz saturu, bet jāatceras, ka TIS lietotņu priekšrocības balstās uz to hipervides dabu informācijas prezentācijā, un ir būtiski, ka integrācija apskata arī iesaistītās navigācijas struktūras.

Tāpat aktuālas ir mijiedarbības problēmas, jo TIS ļauj lietotājam piedalīties datu apmaiņā, pievienot jaunu informāciju, līdz ar to radot jaunus izejas datus adaptācijai – pielāgojamībai, radot nepieciešamību specificēt dinamiskus aspektus adaptācijai.

TIS izstrādes īpatnības var tikt apskatītas gan no lietotāju viedokļa, gan arī attiecībā pret izstrādes tehnoloģijām un metodēm. Tālāk tiks minēti tie TIS izstrādes faktori, kas ir būtiski adaptīvu TIS izstrādei (Suh, 2005):

- no neierobežota lietotāju skaita seko dažādas lietotāju vajadzības sistēmā. Jādomā arī par apmācības atbalstu darbā ar sistēmu, jo klātienē tas nav realizējams, tāpēc būtiska ir intuitīva, pat adaptēta navigācija, lietotājam jāpiegādā tikai viņam atbilstoša informācija,
- aktuāli ir arī pilnvarošanas jautājumi, sistēmai jābūt pieejas tiesību kontroles mehānismam informācijas pievienošanai, rediģēšanai un dzēšanai,
- izstrādes vides izmaiņas – ātri mainās gan tehnoloģijas, gan biznesa modeļi, rodas jaunas valodas, standarti, rīki un metodes,
- prasību izmaiņas, jo attīstās gan izstrādātā sistēma, gan biznesa prasības, prasības attīstās izstrādes gaitā, kā arī lietojot sistēmu, jo attīstās lietotāju priekšstats par sistēmu.

Tīmekļa vides īpašības nosaka tīmekļa informācijas sistēmu izstrādes specifiku, pielietojamās metodes un tehnoloģijas. Tā kā TIS ir specifiskas IS, tad tīmekļa sistēmu izstrāde arī ir specifiska. Līdz ar to var pieņemt, ka tās metodes, kas tiek izmantotas tradicionālo sistēmu izstrādē, var atšķirties no tīmekļa projektiem piemērotām metodēm, vai arī eksistējošās pieejas ir jāpielāgo.

TIS satur informāciju, kas dinamiski mainās, tādēļ tiek nodalīta pati informācija un tās prezentācija, savukārt, lai specificētu TIS funkcionalitāti, galvenā uzmanība tiek pievērsta informācijas piekļuves kontrolei un tās attēlošanas kontrolei (Houben, 2004).

TIS modelēšanai populāri ir lietot uz UML valodas bāzes veidotas tīmekļa modelēšanas valodas, piemēram, UML UWE [UWE] metodi. Modelējot tīmekļa informācijas sistēmas, ir atsevišķi jāapraksta 1) saturs, kas TIS gadījumā būtu informācija, 2) TIS iebūvētā biznesa loģika (navigācija), kas TIS gadījumā paredz satura sadalīšanu noteiktos mezglos un saitēs starp šiem mezglēm, un 3) prezentācija.

TIS funkcionalitātes un lietotāju grupu mijiedarbības ar funkcionalitāti modelēšanai lieto UML lietojumu diagrammas. Navigācija UWE [UWE] tiek modelēta lietojumu diagrammā, lietojot stereotipus. Satura modelēšanai tiek izmantotas klašu diagrammas, kā arī stāvokļu diagrammas, lai parādītu objekta stāvokļa izmaiņas. Modelējot navigāciju, jāapraksta tīmekļa lapas un saites. UWE modelēšanas metode izmanto UML stereotipus.

Prezentācijas modelēšanai arī eksistē dažādas pieejas, piemēram, elementus var aprakstīt trīs hierarhijas līmeņos (Schwinger, Koch, 2006): 1) Prezentācijas lapa, ar ko tiek saprasta lietotāja saskarnes vizualizācijas vienība, 2) Prezentācijas vienība, kas ir grupēšanas elements lietotāja saskarnes elementiem, un ko reprezentē kā loģisku lapas fragmentu, 3) Prezentācijas elements, jeb prezentācijas pamatbloks (teksts, attēls, u.c.). Lai attēlotu šos elementus, izmanto atbilstošus UML stereotipus [UWE]. Tomēr UWE nav sākotnēji veidots adaptīvām TIS, kaut arī atsevišķus aspektus var attēlot.

Dažkārt ar esošo arhitektūru un tehnoloģiju, kaut arī pārdomātu izvēli, var nepietikt, lai uzlabotu efektivitāti, navigāciju un satura piegādi tīmekļa lietojumprogrammās. Pieejas, ko var izmantot, ir analizētas (Facca et al., 2005), kurā tiek piedāvātas sekojošas pieejas:

- profilu bāzēta personalizācija,
- kontekstu - zinošas (*context - aware*) jeb pielāgojamas (*adaptive*) tīmekļa lietotnes,
- darba plūsmu (*workflow*) bāzētas tīmekļa lietotnes,
- lietojamības pētījumi un tīmekļa žurnālfailu analīze.

Nākošajā promocijas darba punktā tiks aprakstīti piemēri eksistējošiem risinājumiem (arhitektūrām, modeļiem, metodēm) adaptīvām tīmekļa informācijas sistēmām.

### 2.2.8. Eksistējoši risinājumi adaptīvām TIS

Apskatīsim dažus eksistējošus risinājumus adaptīvām tīmekļa informācijas sistēmām.

#### Arhitektūra un metodoloģija kontekstu-zinošai datu piekļuvei

Metodes autori rakstā (De Virgilio, Torlone, 2005) piedāvā vispārīgu **arhitektūru** adaptīvai sistēmai, galvenā īpašība un atšķirība pēc autoru domām ir šīs sistēmas paplašināmība, lai atbilstu iepriekš nezināmām jaunām adaptācijas prasībām. Uz šo arhitektūru balstās arī rakstā piedāvātā **metodoloģija** satura adaptācijai, kuras galvenā ideja ir konfigurāciju ģenerēšana un pārvaldība, tiek izmantoti gan dažādi profili, gan konteksts. Metode izstrādāta **datu intensīvām TIS**, kas nodrošina tīmekļa pieeju lieliem strukturētu datu apjomiem.

Tā kā autori izmanto definīciju „TIS ir adaptīva, ja tā ir spējīga modificēt un personalizēt satura piegādi un servisu atbilstoši lietotāja kontekstam”, tad autori, atbilstoši šai adaptācijas definīcijai, galvenokārt pievēršas satura adaptācijai, kaut arī autori raksta, ka adaptācijā atsevišķi jāapskata trīs komponentes: saturs, prezentācija, un navigācija. Adaptācija šajā gadījumā nozīmē lietotāja interesēm piemērotākā satura izvēli, lapas izskata pielāgošanu (autori ar to rakstā saprot iekārtu spēju attēlot datus) un hiperteksta struktūras būvēšanu, sadalot lielus datu apjomus vairākās lapās.

Autoru piedāvātā metode nodrošina piegādātā satura prezentācijas adaptēšanu atbilstoši kontekstam un profiliem. Tā kā katram profilam atbilst noteikta konfigurācija, tad jābūt

skaidrībai, ko un kādā veidā šī konfigurācija apraksta. Prezentācijas līmenī autori apskata tīmekļa lapu, kas sastāv no noteikta tipa tīmekļa objektiem, piemēram, teksts, attēls, forma. Katram objekta tipam var būt savi prezentācijas atribūti ar iepriekš definētām iespējamām vērtībām.

Adaptācijas process tiek veikts ar sekojošiem soļiem: lietotājam mijiedarbības brīdī ar sistēmu tiek noteikts konteksts, katra konteksta dimensija tiek reprezentēta ar profilu. Tālāk no konfigurāciju repozitorija tiek meklētas profiliem atbilstošas konfigurācijas. Ja tādas tieši netiek atrastas, tad tiek meklētas tuvākās konfigurācijas, kas ietver vajadzīgās īpašības. Adaptācijas koordinators veic atrasto konfigurāciju apvienošanu. No apvienotās konfigurācijas tiek ģenerētas adaptācijas operācijas, kas var attiekties gan uz vaicājumiem satura slānī, gan arī uz hiperteksta manipulācijām navigācijas slānī un stilu prezentācijas slānī. Adaptācijas operācijas tiek izpildītas, dažkārt pat ārpus sistēmas.

#### Personalizēts piekļuves modelis (*Personalized Access Model* - PAM)

Raksta (Abbar et al., 2008) autori piedāvā metamodeli, kurā ir definēti profils un konteksts, kā arī tiek piedāvāta servisu kopa, lai definētu personalizētu piekļuves modeli PAM. PAM definīcija atļauj dažādus personalizācijas scenārijus, ieskaitot iestatījumu bāzētas rekomendācijas, kontekstam atbilstošu satura piegādi un arī personalizētu piekļuvi saturam.

Autori definē PAM ar atbilstošiem profila un konteksta metamodeļiem un definētu servisu kopu, lai personalizētu lietotnes. Profilu definē ar 5 dimensijām: personas dati (identitāte, demogrāfiskie rādītāji u.c.), interešu sfēra (lietotāja kvalifikācija un saturs, kas lietotājam interesē), datu kvalitāte (prasības datu aktualitātei, piegādāto ierakstu skaits, formāts, un izkārtojums u.c.), drošība un privātums (drošības likumi un ierobežojumi vaicājumiem vai vaicājumu rezultātam kā sensitīvai informācijai). Kontekstu definē ar 5 dimensijām – Laiks, Vieta, Iekārta, Lietotāja stāvoklis, Vide.

Autori uzskata profilu par iestatījumu kopu, kur katru iestatījumu apraksta ar predikātu un skaitli, kas apraksta tā svarīgumu lietotājam.

Personalizācijas servisi: PAM nodrošina sekojošas adaptācijas darbības (būtībā var runāt tikai par satura adaptāciju): 1) Profila kontekstualizācija: tiek atrastas lietotāju profilu daļas, kas atbilst kontekstam, tātad atlasīti tie iestatījumi, kas atbilst kontekstam, 2) Vaicājuma pārformulēšana, kas adaptē lietotāja vaicājumu atbilstoši viņa profilam, izmantojot divas tehnikas – vaicājumu pārrakstīšanu un vaicājumu paplašināšanu.

Katra lietotne glabā savus profilus un kontekstus un nodrošina servisu izsaukumus. PAM satur profilu un kontekstu metamodeļus lietotāju profiliem un kontekstiem, kas ir pamats servisu un lietotņu mijiedarbībai.

#### Arhitektūra ar personalizētas prezentācijas slāni

Rakstā (Tvarožek et al., 2007) ir piedāvāta arhitektūra personalizētam TIS prezentācijas slānim, kas definē tīmekļa bāzētus komponentus, kas implementēti kā atsevišķi programmatūras rīki prezentācijai, personalizācijai un lietotāju modelēšanai, lai varētu nodrošināt katram lietotājam personalizētu navigāciju un atšķirīgus datu skatījumus. Prezentācijas rīki izmanto domēna un lietotāja modeļus.

Portāla līmenī tiek nodrošināta kopēja vide visu sistēmas atsevišķo rīku izmantošanai: navigāciju, pilnvarošanu, lietotāju pārvaldīšanu, ieskaitot globāla līmeņa elementu izskatu izvēlnēm, saitēm, kā arī vietnes navigācijas struktūru. Individuālo rīku slānī tiek nodrošināta pieeja sistēmas funkcionalitātei, šajā slānī prezentācijas līdzekļiem nosaka dažādu elementu izskatu: teksta apgabaliem, attēliem utt., bet citiem rīkiem, kā, piemēram, biznesa loģikai – funkcionalitāti.

Autori piedāvā šīs pieejas realizāciju - TIS, kas sastāv no kombinējamiem lietotāja saskarnes komponentiem – portletiem, kas reprezentē noteiktu sistēmas funkcionalitāti, nodrošinot, ka viena un tā pati funkcionalitāte dažādiem lietotājiem var tikt piegādāta dažādi, atkarībā no portletu izvēles. Portāla līmenī vēl iespējams katram lietotājam izvēlēties portletu izkārtojumu.

Adaptācija tiek realizēta ar adaptācijas rīku palīdzību un balstās uz datiem, kas tiek glabāti lietotāja modelī.

### Likumu bāzēta adaptīva TIS

Valeriano et al. (2006) piedāvā TIS adaptācijas tehniku, kas balstās uz likumiem, kas definē, kā generēt dotajam kontekstam atbilstošas adaptācijas specifikācijas.

Likumi balstās uz vispārīga profila un abstraktas konfigurācijas jēdzieniem. Profila P jēdziens atbilst (De Virgilio, Torlone, 2005) lietotājam (konteksta atsevišķa aspekta reprezentācija), ko apraksta ar dimensijām un dimensiju atribūtiem. Abstrakta konfigurācija tiek definēta kā  $C = (q, h, s)$ , kur  $q$  ir vaicājums datubāzē, kas izteikts ar relāciju algebras palīdzību, bet  $h$  ir hiperteksta abstrakta definīcija ar WebML (Ceri et al., 2002) līdzekļiem, kas apraksta, kā organizētas tīmekļa lapas; un  $s$  ir prezentācijas specifikācija, kas katram tīmekļa objektu tipam, kas ir iepriekš nedefinēts, apraksta atbilstošā objekta īpašības, piemēram, novietojumu, krāsu utt.

Autori tad adaptācijas likumu definē kā attiecību starp profilu P un konfigurāciju C, un semantika šādam likumam ir sekojoša: ja klientam ir profils P un nosacījumu kopas C nosacījumi izpildās (profila atribūtu salīdzinājums ar konstantēm vai citiem atribūtiem), tad tiek ģenerēta TIS konfigurācija C.

### Hera

Hera modelis (Frasincar et al., 2002), (Frasincar et al., 2006) apskata vairāku slāņu arhitektūru. Semantiskajā slānī autori apskata konceptuālo modeli, kas apraksta domēnu, vēl šai Semantiskajā slānī ir citi, piemēram, integrācijas modelis, u.c. Lietotnes slānī Hera apskata Lietotnes modeli, kas apraksta navigācijas struktūru pa hipervides prezentācijas elementiem. Autori piedāvā modelēšanas primitīvus – satura konteinerus (*slices*) un attiecības starp tiem navigācijas reprezentēšanai (*navigation between slices*). Šajā slānī tiek specificēta arī adaptācija.

## **2.3. TIS implementācijas mehānismi**

### **2.3.1. TIS arhitektūras**

Lai varētu piedāvāt jaunu arhitektūru, jādefinē, kas tiek saprasts ar šo jēdzienu.

Sākumā apskatīsim informācijas sistēmas arhitektūras dažādas definīcijas, kas sastopamas literatūrā. Tā piemēram, Garlan et al. (1995) raksta, ka informācijas sistēmas arhitektūra risina jautājumus, kā reprezentēt IS komponentu struktūru, to saistību, darbības principus un vadlīnijas.

Oreizy et al. (1998) definē arhitektūru sekojoši: “programmatūras arhitektūra reprezentē programmatūras sistēmas struktūru augstā abstrakcijas līmenī un formā, kas padara to analizējamu, papildināmu, un citādi noderīgu dažādiem ar inženieriju saistītiem uzdevumiem.”



Goepp et al. (2006) raksta, ka IS arhitektūra sastāv no IS komponentu definīcijām, ar to kopsakarību aprakstiem un to mijiedarbību laikā (sistēmas dinamikas apraksts).

Arhitektūru nozīme tika atkārtoti pārvērtēta 2007. gada ICSE konferences sekcijā “*Future of Software Engineering*” (Programminženierijas nākotne), sniedzot definīciju, ka programmatūras arhitektūra ir “principiālu projektēšanas lēmumu kopa, kas ir pamatā sistēmai”, tā iesaista plašu spektru ar projektēšanas aktivitātēm (Taylor, Van der Hoek, 2007).

Kā redzams no iepriekš minētā, gan arī Gorton (2011) savā grāmatā par programmatūras arhitektūru, raksta, ka ir ļoti daudz definīciju programmatūras arhitektūras jēdzienam. Gorton (2011) piemin divas pamatdefinīcijas.

**Programmatūras arhitektūra** tiek definēta kā ieteicamā prakse sistēmas struktūrai, ko iemieso tās komponenti, to saistība vienam ar otru un vidi, un principi, kas nosaka tās izveidi un evolūciju” (ANSI/IEEE Std 1471-2000).

Tāpat citiem vārdiem sakot, arhitektūra apraksta, no kādiem komponentiem sastāv programmatūras sistēma un kā tie mijiedarbojas. Tiek arī definēti visas sistēmas līmenī projektēšanas principi, kā arī tiek noteikts, kā sistēma var tikt mainīta.

Gorton (2011) apskata arī definīciju, ko devis kāds no atzītiem nozares speciālistiem: “Programmatūras vai datorsistēmas arhitektūra ir sistēmas struktūra (vai struktūras), kas ietver programmatūras elementus, ārēji redzamas šo elementu īpašības, un saistības starp tiem” (Bass et al., 2003).

Literatūrā ir aprakstītas un praksē tiek plaši pielietotas dažādas programmatūras arhitektūras, piemēram, pakalpojumu orientēta arhitektūra, komponentbāzēta arhitektūra un citas.

Tajā pašā laikā, lai nodrošinātu lietotājus ar tiem piemērotu TIS, ir jāapskata arī pieejas elastīgu un izmaksu efektīvu sistēmu izstrādei, tālāk tiks apskatītas divas – programmatūras produktu līnija (*Software Product Line*) un programmatūra kā pakalpojums. (*Software as a Service* - SaaS).

### 2.3.1.1. Programmatūras produktu līnija

Programmatūras produktu līnijas (*Software product line* - SPL) (Clements, Northrop, 2002) pieeja ir bāzēta uz iepriekš definētu arhitektūru un precīzi definētām, labi zināmām pamatfunkcijām. Visas atkārtojami izmantojamās komponentes ir definētas jau pašā sākumā.

Lai iegūtu atšķirīgu kādas programmatūras instanci tradicionālās arhitektūrās parasti tiek mainīts lietojumprogrammas kods. SPL pieeja piedāvā realizēt variācijas dažādos veidos (Bass et al., 2003), piemēram, elementu iekļaušanu vai izslēgšanu, dažādu elementu dažādu versiju kombinēšanu. Dažādas var būt, piemēram, kvalitātes un reakcijas īpašības. SPL pieejai ir būtiska arhitektūra un konfigurācijas pārvaldība, jo katrs jauns programmatūras produkts no vienas līnijas sastāv no neskaitāmiem (bet komplekts var būt atšķirīgs) pamatkomponentiem un tam pievienotām variācijām.

### 2.3.1.2. Programmatūra kā serviss

Programmatūra kā serviss (*Software as a service* - SaaS) (MS06) tiek definēts kā “programmatūra, kas tiek piegādāta kā internetā izvietots pakalpojums un kurai piekļūst, izmantojot Internetu”. SaaS var būt pakalpojumi, kas piegādāti dažāda lieluma organizācijām, lai atbalstītu biznesa procesus, kas līdzīgas visām šīm organizācijām, piemēram, lietotne, kas nodrošina klientu attiecību pārvaldību. Organizācijas piesakās un reģistrējas lietot šo SaaS programmatūru un maksā par lietošanu. Pakalpojuma piegādātājs izvieto šo SaaS

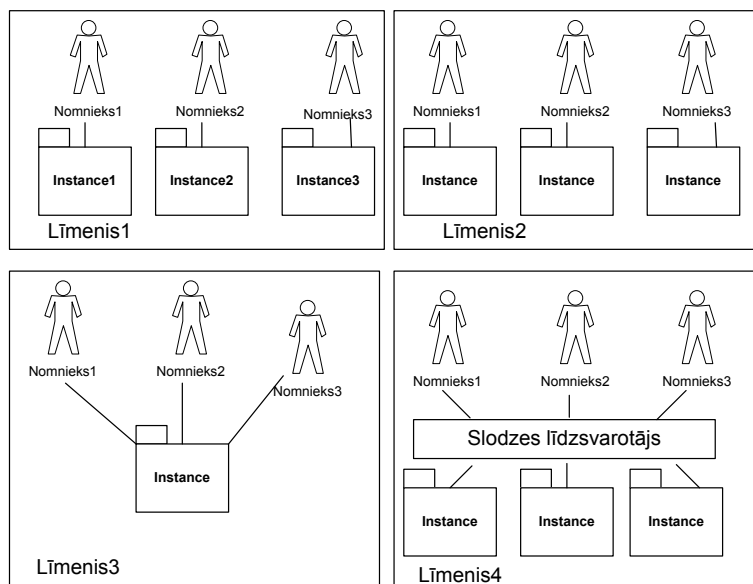
programmatūru uz saviem serveriem un uztur programmatūru un infrastruktūru. Šis bieži ir izmaksu efektīvs risinājums maziem un vidējiem uzņēmumiem.

SaaS lietotnes var tikt piedāvātas 4 dažādos līmeņos (2. attēls. SaaS programmatūras 4 līmeņi), sākot ar pielāgotas lietotnes instances izveidošanu individuāli katram klientam uz SaaS lietotnes piegādātāja servera.

### 1. līmenis - klienta

Pirmais līmenis ir līdzīgs tradicionālam lietojumu pakalpojumu piegādes modelim programmatūrai. Šajā līmenī katram klientam ir sava pielāgota instance izvietotajai programmatūrai uz pakalpojuma sniedzēja servera. No arhitektūras viedokļa šāds līmenis ir ļoti līdzīgs tradicionālām komerciālām programmatūrām, kur daudzi vienas organizācijas lietotāji strādā ar vienu programmatūras instanci, kas ir pilnīgi neatkarīga no citas tās pašas programmatūras instances, kas tiek darbināta uz tā paša servera.

Klienta servera lietotnes parasti tiek pārveidotas tieši par šī līmeņa SaaS lietotnēm. Šāda pāreja ir pietiekoši viegla, ļauj pakalpojuma sniedzējam ietaupīt uz administrēšanas un serveru izmaksu rēķina.



2. attēls. SaaS programmatūras 4 līmeņi

### 2. līmenis: konfigurējams

Otrajā līmenī piegādātājs izvieto atsevišķi katram klientam (nomniekam) savu programmatūras instanci uz pakalpojuma sniedzēja servera. Šajā līmenī visi klienti izmanto vienādu programmatūras kodu, bet piedāvā klientam plašas programmatūras konfigurēšanas iespējas, lai pielāgotu programmatūru klienta vajadzībām. Katra instance ir pilnīgi neatkarīga no citām. Šāds risinājums ievērojami samazina uzturēšanas izmaksas, jo izdarot izmaiņas kodā, to var viegli piegādāt visiem klientiem.

Šim risinājumam arhitektūrā jāparedz konfigurācijas datu glabāšana metadatu repozitorijā. Lielākās izmaksas paredzamas serveru jaudu nodrošināšanai, lai varētu izpildīt daudzas paralēlas instances.

### 3. līmenis: konfigurējams, daudzu nomnieku, efektīvs

Trešajā līmenī programmatūras piegādātājs uztur vienu programmatūras instanci, kas apkalpo katru klientu, ar konfigurējamiem metadatiem nodrošinot katram lietotājam savu īpašību kopu.

No lietotāja viedokļa viņš strādā it kā ar neatkarīgu programmatūru, dati tiek nodalīti no citiem klientiem.

#### 4. līmenis: mērogojams, konfigurējams, daudzu nomnieku, efektīvs

Pēdējā, 4. līmenī programmatūras pakalpojuma sniedzējs izvieto daudzus klientus līdzsvarotas slodzes identisku instanču apvienībā, ar atsevišķi glabātiem katra klienta datiem, ar konfigurējamiem metadatiem

Sarežģītāks variants ir izmantot konfigurējamu, daudzu nomnieku vienas instances risinājumu. Šajā gadījumā pielāgošana tiek veikta balstoties uz metadatu bāzētu SaaS lietotnes konfigurāciju, lai varētu no vienas un tās pašas programmatūras katram pasūtītājam izveidot viņa vajadzībām pielāgotu SaaS lietotni. Daudzu-nomnieku risinājums nozīmē, ka dažādas organizācijas lieto vienu un to pašu lietotnes instanci, bet dati tiek nodalīti starp dažādajiem lietotnes nomniekiem. Nodalījums iespējams gan fiziskā līmenī, ko praktizē biežāk, gan arī loģiskā līmenī. Fiziskā nodalīšana parasti prasa papildus resursus uzturēšanai, jo jāuztur vairākas realizācijas. Darba apjoms īpaši pieaug gadījumā, kas funkcionalitātē nepieciešamas specifiskas nianse katram lietotnes nomniekam. Šādos gadījumos var samazināt uzturēšanai nepieciešamos resursus veidojot vienu sistēmu, datus dalīt loģiskā līmenī, bet katras organizācijas nianse realizējot kā papildus funkcionalitāti.

Lai varētu efektīvi risināt mērogojamības, konfigurācijas un daudzu-nomnieku esamības jautājumus, SaaS lietotnēm nepieciešamas šim nolūkam īpaši izstrādātas arhitektūras. Šīm arhitektūrām jāatrisina arī problēmas, kas saistītas ar daudzu nomnieku vajadzībām atbilstošu lietotnes pielāgošanu.

Piedāvātā arhitektūra ir paredzēta TIS lietošanas kontekstam, kur daudzas līdzīgas organizācijas izmanto vienu un to pašu TIS, bet katra no tām saņem adaptētu sistēmas instanci. Tiek piedāvāta divu līmeņu adaptācija. Ne tikai organizācija, bet arī lietotājs saņem savu adaptētu, personalizētu lietotnes instanci.

#### **2.3.2. Daudzu nomnieku TIS eksistējoši risinājumi**

Viena no eksistējošām metodēm, kas ļauj padarīt daudzu nomnieku TIS pielāgojamu, izmanto programmatūras arhitektūru kā līdzekli, lai koordinētu darbību starp sistēmas funkcionāliem komponentiem un konfigurēšanas komponentiem (Wang, Zheng, 2010).

Daudzu nomnieku gadījumā konfigurēšana pamatā ietver funkciju un lietotāja saskarnes konfigurēšanu. Wang un Zheng (2010) apskata atšķirības konfigurēšanā starp tradicionālo pieeju (produkta bāzētu pieeju) un situāciju daudzu nomnieku gadījumā. Daudzu nomnieku gadījumā konfigurācija var tikt veikta izpildes laikā (nevis tikai pirms darba uzsākšanas), var tikt izveidota viena atsevišķa konfigurācija katram nomniekam (nevis tikai viena visai sistēmai kopumā), to uzstāda katra nomnieka administrators un ir spēkā tikai nomnieka datiem, dinamiski izmanto izpildes laikā (nevis sistēmas sākuma fāzē).

Ja apskata SaaS tipa programmatūru, tad visticamāk katram nomniekam ir sava specifika, kā izpildās biznesa funkcijas, un lai SaaS lietotne būtu piemērota katram no nomniekiem, ir vajadzīgs to pielāgot katra nomnieka vajadzībām. Vēl nozīmīgs moments ir tas, ka nomniekam ir jāmaksā par sistēmas nomu, līdz ar to nomnieks ir ieinteresēts maksāt tikai par sev vajadzīgām funkcijām, šo funkciju komplektiem jeb moduļiem. Tādēļ funkciju konfigurēšana SaaS lietotnēs atrisina problēmu prasību atšķirībās, lai tās varētu tikt adaptētas dažādiem nomniekiem no dažādām biznesa jomām.

Lietotāja saskarnei jāattēlo tikai nomātās funkcijas. Lietotāja saskarnes konfigurēšana savukārt attiecas uz izvēlņu konfigurēšanu un lapu satura konfigurēšanu.

SaaS lietotnēm, lai realizētu funkciju konfigurējamību, jāveic funkciju dekompozīcija, lai noteiktu atomāras funkcijas ar mērķi sadalīt sistēmu neatkarīgās funkciju grupās, pēc tam atomāras funkcijas jāapvieno Funkciju pakotnēs. Wang un Zheng (2010) piedāvā modeli, kas apraksta katra nomnieka gan funkciju, gan izvēlņu konfigurāciju un ļauj glabāt datus par katru šo konfigurāciju, lai lietošanas brīdī sistēma tiktu pielāgota nomnieka vajadzībām. Wang un Zheng (2010) piedāvā programmatūras arhitektūras elementu – konektoru, lai realizētu šo saistību starp komponentiem.

Autori izmanto aspektu orientēto programmēšanu, lai realizētu šo arhitektūras vadīto pieeju.

Citas pieejas realizē konfigurēšanu, izmantojot ārējus modeļus, izmanto formālas metodes, lai aprakstītu konfigurējamību (Wei et al., 2008), (Kwok et al., 2008).

## **2.4. Nodaļas secinājumi**

TIS adaptācijai eksistē daudzas, plaši pazīstamas pieejas, kas tika apskatītas nodaļā, tās tipiski balstās uz hipervides adaptīvām sistēmām jau izstrādātiem references modeļiem, kaut arī eksistē neatkarīgi izveidotas pieejas TIS adaptācijai.

Gan adaptīvas hipervides sistēmām, gan adaptīvām TIS, to autoriem ir dažādas izpratnes, ko nozīmē adaptīva sistēma, ar ko tā atšķiras no adaptējamās sistēmas. Dažkārt šie jēdzieni netiek šķiroti, savukārt adaptējama sistēma (ja tāds jēdziens tiek lietots) tiek saukta arī par konfigurējamu sistēmu. Tiesa, visos gadījumos ir būtiska lietotāja loma adaptācijas procesā, var runāt par atšķirīgiem lietotāja modeļiem un veidu, kā tas tiek pielietots, kā tiek savākti dati par lietotāju

Līdz ar jauna tipa TIS sistēmu rašanos, aktualizējas arī jautājumi par šo sistēmu adaptēšanas iespējām, jo tās jebkurā gadījumā paliek TIS, kam raksturīgs liels lietotāju skaits un to specifiskās vajadzības.

SaaS tipa TIS ar daudzu nomnieku pieeju, arī tiek analizētas adaptācijas iespējas, kas šai gadījumā ir vēl sarežģītāk, jo pati daudzu nomnieku sistēmas pieeja jau rada papildus sarežģītību, paredzot nodalīt katra lietotāja datus. Šajās sistēmās sākotnēji aktuālāki bija konfigurācijas jautājumi, tās tika veidotas kā adaptējamās (ne adaptīvas) sistēmas, lai nodrošinātu katram nomniekam viņa vajadzībām pielāgotu instanci. Par to liecina eksistējošie pētījumi, tai skaitā iepriekšējā apakšnodaļā aprakstītie. Tomēr pastāv arī vajadzība pēc jau smalkāka mehānisma, kā piedāvāt SaaS tipa lietotni pielāgotu jau konkrētam lietotājam, mainīgu ne tikai no sākotnējiem uzstādījumiem, bet arī atbilstoši veiktajām lietotāja darbībām un sistēmas priekšstatam par lietotāju.

### **3. Daudzu Nomnieku Adaptīvas Tīmekļa Informācijas Sistēmas arhitektūras izveide**

#### **3.1. Nodaļas nolūks**

Nodaļa paredzēta, lai aprakstītu vienu no promocijas darba rezultātiem – Daudzu Nomnieku Adaptīvas Tīmekļa Informācijas Sistēmas (DNATIS) arhitektūru, definējot tās pamatprincipus, komponentus un to īpašības. Nodaļā dots tīmekļa informācijas sistēmas piemērs, kas būvēts, balstoties uz piedāvāto DNATIS arhitektūru. Nodaļā aprakstīts arī, kā šādai TIS tiek veidota lietotāja saskarne, gadījumam, kad viens konkrēts lietotājs strādā vienlaicīgi ar vairākām adaptētām TIS instancēm. Nodaļā aprakstītie rezultāti ir publicēti rakstos: (Niedritis, Niedrite, 2011), (Niedritis, 2011) un (Niedritis, Niedrite, 2012).

#### **3.2. DNATIS arhitektūras vispārīgs apraksts**

Tīmekļa informācijas sistēmas arhitektūra, kas tiek piedāvāta promocijas darbā balstās uz sekojošiem principiem:

- Tiek izmantota programmatūras kā servisa (SaaS) pieeja, precīzāk, tieši daudzu nomnieku vienas instances variants. Līdz ar to tiek arī pieņemts, ka TIS vai tā daļu izmantos organizācijas ar līdzīgiem biznesa procesiem, kuru atbalstam dotā TIS ir pielāgojama.
- Tiek apskatītas TIS, kas glabā datus datu bāzē un kurām ir noteikta biznesa loģika. Tās paredzētas konkrētu biznesa procesu izpildei. Liela lietotāju un izpildāmo procesu skaita gadījumā ir nepieciešams katram lietotājam nodrošināt viņa kontekstam atbilstošu sistēmas funkcionēšanu.
- Šī pieeja ir papildināta ar divu līmeņu adaptācijas iespēju, adaptējot sistēmu organizācijas līmenī un katra individuāla lietotāja līmenī.
- Adaptācija balstīta uz lietotāju modeļa izveidi un lietotāju darbību analīzi lietotāju modeļa datu papildināšanai. Lietotāja dati var tikt mainīti biznesa procesu izpildes rezultātā, vai nu lietotājam pašam mainot savus datus biznesa procesu kontekstā, vai citiem lietotājiem mainot datus par konkrēto lietotāju.
- Adaptācijas arhitektūra kā būtisku komponenti ietver arī veiktspējas mērīšanas sistēmu, lai uzturētu un pilnveidotu pēc piedāvātās adaptācijas arhitektūras būvēto daudzu nomnieku TIS.
- Sistēmas navigācijas adaptāciju vēl nosaka lietotāja darba gaitā veiktās izmaiņas ar datiem, funkciju pieejamība lietotājam ir atkarīga no objektu stāvokļa, ko maina lietotājs.
- Sistēmas darbību nosaka arī lietotāja mijiedarbības konteksts ar sistēmu, piemēram, atkarībā no mijiedarbības laika noteiktas funkcijas var nebūt pieejamas lietotājam.

Promocijas darbā tiek piedāvāta Tīmekļa informācijas sistēmas arhitektūra DNATIS (skat. 3. attēls), kas paredzēta, lai varētu nodrošināt tīmekļa informācijas sistēmas izmantošanu vairākās vienas biznesa jomas organizācijās ar līdzīgiem biznesa procesiem, adaptējot to katras atsevišķas organizācijas vajadzībām (Niedritis, Niedrite, 2011).

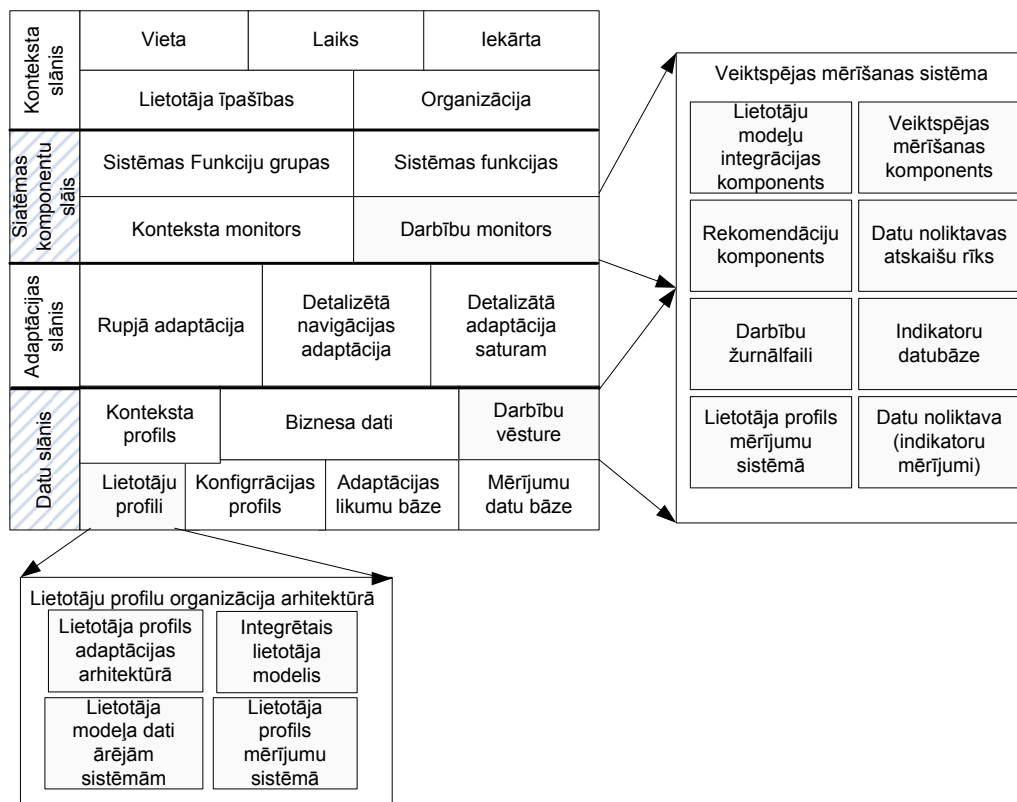
DNATIS arhitektūrai ir četri slāņi:

- 1) konteksta slānis definē lietotāja mijiedarbību ar tīmekļa informācijas sistēmu, kas būvēta, balstoties uz DNATIS arhitektūru. Kontekstu šajā mijiedarbībā nosaka sistēmas lietotājs, mijiedarbības vieta un laiks, iekārta, kas tiek izmantota mijiedarbībai, kā arī organizācija, kuras TIS instanci izmanto lietotājs,
- 2) sistēmas komponentu slānis definē pašu adaptējamo sistēmu, tās uzbūvi, sastāvdaļas, tajā skaitā sistēmas funkciju grupas, funkcijas. Šajā slānī ietilpst arī tie komponenti, kas vajadzīgi, lai atbalstītu adaptētās sistēmas izmantošanu, tajā skaitā, piemēram, veikspējas mērīšanas sistēma,
- 3) adaptācijas slānis sastāv no DNATIS komponentiem, kas nodrošina tīmekļa informācijas sistēmas adaptācijas nodrošināšanai nepieciešamos komponentus,
- 4) datu slānis glabā gan TIS biznesa datus, gan TIS adaptācijas realizācijai nepieciešamos datus, gan veikspējas mērīšanas datus.

Tālāk promocijas darbā DNATIS arhitektūras apraksts tiks organizēts sekojoši: promocijas darba 3. nodaļas tālāk sekojošās apakšnodaļas aprakstīs DNATIS arhitektūru un tās komponentus no adaptācijas realizācijas perspektīvas.

Promocijas darba 4. nodaļā aprakstīta lietotāja modeļa loma un vieta DNATIS arhitektūrā gan adaptācijas, gan mērīšanas atbalstam.

Promocijas darba 5. nodaļa veltīta veikspējas mērīšanas jautājumiem DNATIS arhitektūrā, ko kopējā shēmā raksturo veikspējas mērīšanas sistēmas sadaļa (skat. 3. attēlu), kas ietver gan komponentus no DNATIS arhitektūras sistēmas komponentu slāņa, gan datu slāņa.

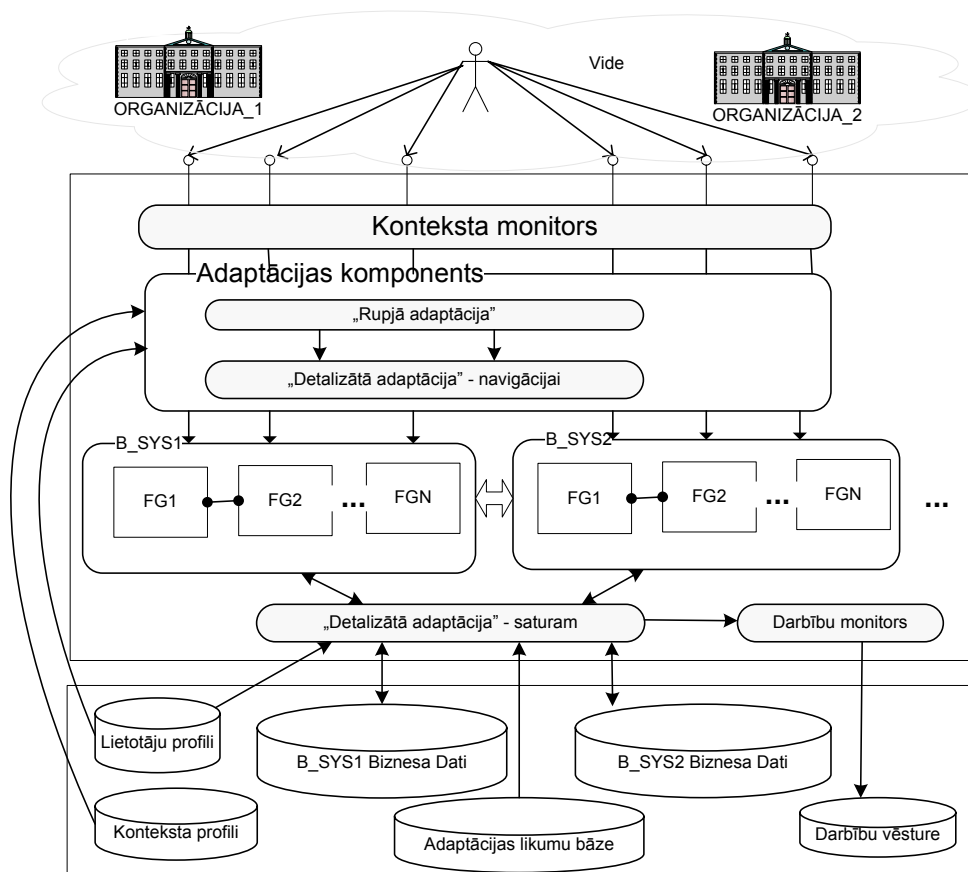


3. attēls. DNATIS arhitektūras slāņi un komponenti

Tālāk DNATIS arhitektūra šajā nodaļā tiks aprakstīta no adaptācijas realizācijas perspektīvas, un tiks balstīta uz 4. attēlā doto shēmu, kas satur tos DNATIS arhitektūras komponentus no visiem arhitektūras slāņiem, kas nepieciešami adaptācijai, kā arī atspoguļo šo komponentu savstarpējo saistību.

DNATIS arhitektūra sastāv no sekojošiem komponentiem, kas nepieciešami tieši adaptācijas realizācijai (skat. 4. attēlu):

1. Konteksta monitors (skat. apakšpunktu 3.3.3.1), kas brīdī, kad lietotājs pieslēdzas sistēmai, nosaka sistēmas lietošanas konteksta īpašības. Konteksta monitors noskaidrotās konteksta īpašības salīdzina ar DNATIS arhitektūras profilu atbilstošo elementu vērtībām, lai noteiktu vēl citas profilu atribūtu vērtības, kas raksturo kontekstu. Tālāk konteksta īpašības tiek nodotas DNATIS arhitektūras adaptācijas komponentam.



4. attēls. Daudzu Nomnieku Adaptīvas Tīmekļa Informācijas Sistēmas arhitektūra

2. Adaptācijas komponents (skat. apakšpunktu 3.3.3.2), kas noskaidro ar DNATIS arhitektūras līdzekļiem iepriekš nedefinēto TIS konfigurāciju un nodrošina TIS divu līmeņu adaptāciju:

- rupjo adaptāciju (skat. apakšpunktu 3.3.3.3.) – kas veic TIS lietotājam pieejamo funkciju grupu noteikšanu atbilstoši sistēmas lietošanas kontekstam (4. attēlā apzīmēts ar FG1, FG2 utt.). Kontekstu šajā gadījumā raksturo, piemēram, organizācija, kuru pieslēgšanās brīdī pārstāv lietotājs, pieslēguma punkts u.c. (skat. apakšpunktu 3.3.3.1). Atbilstoši DNATIS arhitektūrā definētajiem konfigurācijas un organizācijas profiliem, kas apraksta, piemēram, organizācijai pieejamās funkciju grupas, noformējumu u.c. (skat. punktu 3.3.1), tiek izveidota adaptēta TIS organizācijas līmeņa instance.

Organizācija, ar kuru ir saistīts lietotājs, kas mijiedarbojas ar TIS, DNATIS arhitektūras izpratnē tiek uzskatīta par nomnieku. Rupjās adaptācijas rezultātā tiek izveidota nomnieka

prasībām atbilstoša TIS konfigurācija, adaptēts noformējums, u.c. īpašības, atbilstoši profilos glabātajai informācijai par nomnieku. DNATIS arhitektūra nodrošina arī situāciju, kad viens lietotājs var būt saistīts ar vairākām organizācijām, kā rezultātā tiek izveidotas vairāku organizāciju adaptētas TIS instances. Šai gadījumā lietotāja ērtībai DNATIS arhitektūra apvieno vairākas TIS instances vienotā TIS saskarnē, atbilstoši konfigurācijas profilā definētajām atļautajām pārejām.

TIS organizācijas līmeņa instances 4. attēlā apzīmētas ar B\_SYS1, B\_SYS2 utt., zīmējumā attēlota situācija, kad viens lietotājs ir saistīts ar divām organizācijām, kas ir nomnieki dotajai TIS.

- detalizētu adaptāciju (skat. apakšpunktu 3.3.3.4), ko, savukārt, nosaka DNATIS arhitektūras lietotāja profils un kam ir divas darbības:
  - “detalizēta adaptācija – navigācijai”. Atbilstoši lietotāja profilam, kas apraksta lietotājam pieejamās funkcijas, rupjās adaptācijas līmenī izveidotās TIS organizācijas instances ietvaros tiek izveidota lietotāja vajadzībām adaptēta TIS navigācija starp lietotājam pieejamām funkcijām;
  - “detalizēta adaptācija – saturam”. Atbilstoši lietotāja profilam, kas apraksta lietotājam pieejamos datus un šo datu saistību ar iepriekšējā solī noteiktajām lietotāja funkcijām, tiek izveidota lietotāja TIS instance, iekļaujot tajā arī lietotājam atļautos datus.
- 3. Darbību monitors, kas tiek izmantots, lai uzkrātu datus par izmantotajām TIS funkcijām, lai šo informāciju izmantotu TIS darbības operatīvai analīzei un vēlākai šo datu izmantošanai tālākai TIS attīstīšanas stratēģijas izstrādei, kā arī, lai analizētu daudzu nomnieku gadījumā katra nomnieka TIS izmantošanas apjomu un veidu, lai izmantotu šo informāciju gan atbilstošas samaksas noteikšanai par TIS uzturēšanu un izmantošanu. Darbību monitors DNATIS tiek veidots saskaņā ar promocijas darbā piedāvāto organizācijas veiktspējas pārvaldības ietvaru (5.6. apakšnodaļa) un divu līmeņu mērīšanas metodi (5.7. apakšnodaļa), kas ļauj novērtēt TIS katru instanci kā atsevišķu informācijas sistēmu, gan var tikt izmantota atsevišķi vienas organizācijas ietvaros, vērtējot organizācijas TIS instanci salīdzinājumā ar organizācijas citām informācijas sistēmām.
- 4. TIS datu slānis, kas satur profilus (skat. punktu 3.3.1.), kas tiek izmantoti abu līmeņu adaptācijā un TIS atbilstošos biznesa datus, kuru virtuālas instances tiek izveidotas rupjās adaptācijas laikā, nodalot TIS organizācijas instancei atbilstošās organizācijas datus. 4. attēlā vienas organizācijas dati apzīmēti, piemēram, ar “B\_SYS1 Biznesa dati”. Lietotājam pieejamie dati tiek noteikti TIS detalizētās adaptācijas laikā, lai izveidotu TIS saturu (skat. punktu 3.4.3.). TIS datu slānī tiek glabāta arī informācija par TIS funkciju darbību vēsturi, ko savāc darbību monitors. Adaptācijas likumu bāze glabā adaptācijas darbību implementācijas (procedūras), kas izmantojamas realizējot kādu no adaptācijas darbībām (skat. punktus 3.3.3.3. un 3.3.3.4), kad tiek veidotas organizācijas līmeņa vai pēc tam lietotāja līmeņa adaptētas TIS instances.

TIS komponenti tālāk sekojošās promocijas darba apakšnodaļās tiks apskatīti detalizētāk.

DNATIS arhitektūrai ir vairāki ierobežojumi. Piedāvātā arhitektūra orientēta uz TIS funkcionalitātes un datu piekļuves adaptāciju TIS ietvaros atbilstoši lietošanas kontekstam, tomēr, jāatzīmē, ka minimāli tiek nodrošināta TIS prezentācijas adaptācija. Netiek apskatīta TIS adaptācija mobilām iekārtām, vizuālā noformējuma adaptācija ierobežota ar dažiem atsevišķiem atribūtiem (fona krāsu, logo). Prezentācijas adaptācija ir sekas navigācijas un satura adaptācijai, un tiek realizēta atbilstoši DNATIS prezentācijas modelim, lai lietotājam



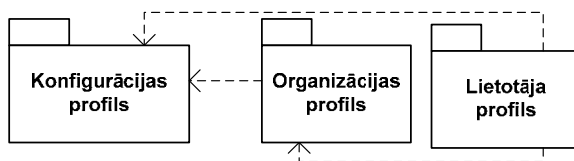
attēlotu viņam pieejamās funkcijas un datus. Tātad promocijas darbā piedāvātā DNATIS arhitektūra izmantojama tāda tipa TIS, kam būtiski panākt liela skaita dažādu adaptētu TIS instanču izveidi, kas nodrošina biznesa funkcionalitāti, un ne tik būtiski ir noformējuma aspekti.

### 3.3. DNATIS arhitektūras implementācijas mehānismi

#### 3.3.1. DNATIS arhitektūras datu slāņa profili

DNATIS arhitektūra izmanto profilus, kas definē metadatus, kas ir pamatā adaptācijas operāciju izpildei. Katrs atsevišķais profils apraksta kādu no būtiskiem TIS adaptācijas aspektiem. Konfigurācijas profils apraksta adaptējamās TIS uzbūvi, no kādām funkcijām tā sastāv, kā funkcijas grupējamas. Organizācijas profils apraksta organizāciju, kas izmanto savu biznesa procesu atbalstam adaptējamo TIS, tās saistību ar konfigurācijas profilu, ieskaitot, kādā apjomā (kuras tieši) TIS eksistējošas funkcijas un funkciju grupas ir ieviestas un tiek izmantotas, un līdz ar to ir iesaistītas adaptācijas procesā. Lietotāja profils apraksta TIS lietotāju, piemēram, lietotāja saistību ar organizāciju, kurā tiek izmantota TIS un lietotāja tiesības darbā ar TIS, lai varētu izveidot lietotāja vajadzībām atbilstošu adaptētu TIS instanci.

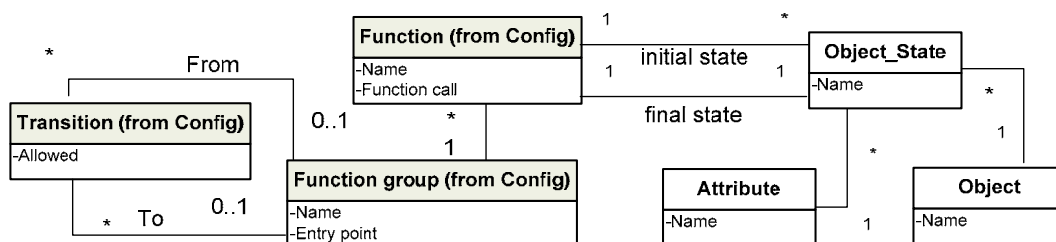
DNATIS arhitektūrā izmantotie profili ir definēti ar UML klašu diagrammām. Lai veidotu grupas no loģiski saistītām klasēm, izmantotas pakotnes. Katra pakotne apraksta vienu profilu. Pakotņu savstarpējās atkarības attēlotas ar pakotņu diagrammu (5. attēls). Klašu diagrammās, kas apraksta atsevišķus profilus, pelēkā krāsā iezīmētas klases, kas iekļautas no citām pakotnēm.



5. attēls. DNATIS profilu pakotņu diagramma

DNATIS konfigurācijas profils apraksta TIS uzbūvi no tajā realizēto funkciju viedokļa - no kādām funkcijām sistēma sastāv un kā šīs funkcijas grupējamas pēc to realizēšanas tehnoloģijas. Grupēšanas mērķis ir aprakstīt situācijas, kādā gadījumā iespējams realizēt vairākām funkciju grupām vienotu TIS saskarni. Vajadzību definēt šādas funkciju grupas ilustrē gadījums, kad dažādām funkciju grupām izmantots atšķirīgs autentifikācijas mehānisms. Atkarībā no autentifikācijas veidiem šo funkciju grupu apvienošana vienotā saskarnē var vai arī nevar tikt nodrošināta.

DNATIS konfigurācijas profils ir aprakstīts ar modeli, kas dots 6. attēlā.



6. attēls. DNATIS paplašinātais konfigurācijas profila modelis

Konfigurācijas profilam ir divas daļas – pamatprofils un paplašinājums. Pamatprofils apraksta TIS uzbūvi no realizēto funkciju viedokļa, bet paplašinājums definē, kā funkcijas mijiedarbojas ar sistēmā definētiem objektiem – pie kādiem objektu stāvokļiem funkcijas ir izpildāmas un uz kādu citu stāvokli drīkst mainīt objekta sākuma stāvokli.

Konfigurācijas pamatprofila elementi ir Funkciju grupa (*Function group*), Funkcija (*Function*) un Pāreja (*Transition*). Funkciju grupa sastāv no daudzām TIS realizētām organizācijas biznesa funkcijām. Funkciju grupai ir definētas pārejas starp dažādām funkciju grupām. Pārejai ir definēts virziens – no kuras uz kuru funkciju grupu pāreja ir atļauta.

Funkciju grupa ir TIS realizēto funkciju grupēšanas līdzeklis, ko definē kopējs ieejas punkts. Piemēram, grupēšana varētu būt nepieciešama gadījumā, kad sistēmas dažādām daļām tiek lietotas atšķirīgas autentificēšanās metodes, kā datu bāzes pilnvarošanas mehānisms vai LDAP autentifikācija. Lietotājs, pieslēdzoties sistēmai, izvēlas vienu no „login” formām, ko var uzskatīt par funkciju grupas ieejas punkta realizāciju. Rezultātā lietotājam tiek nodrošināta piekļuve atbilstoši TIS funkciju grupai, ko šis ieejas punkts definē. Atbilstoši DNATIS arhitektūrai, dažādas TIS funkciju grupas var tikt apvienotas, lai lietotājs vienā saskarnē varētu strādāt ar dažādu TIS instanču dažādām funkciju grupām (bez papildus pieslēgšanās, autentificēšanās), ja konfigurācijas profilā ir definētas atbilstošas pārejas (*Transition*) starp šīm funkciju grupām, kā arī atbilstoši lietotājam nedefinētajām piekļuves tiesībām.

Funkciju grupu apvienošana tiek veikta, balstoties uz DNATIS arhitektūras konfigurācijas profilā definētajām pārejām. Pārejai ir virziens no kuras funkciju grupas, uz kuru funkciju grupu pāreja tiek definēta. Modelī virzienu apraksta ar attiecībām No un Uz. Ne vienmēr visas teorētiski iespējamās pārejas starp funkciju grupām ir atļautas. To norāda klases „*Transition*” atribūts „*Allowed*”.

Piemēram, ja apzīmē ar FG1 un FG2 divas funkciju grupas, tad iespējams definēt divas pārejas: FG1 → FG2 un pretējā virzienā FG2 → FG1. Atkarībā no atribūta „Atļauts” vērtībām, TIS var būt realizēta un konfigurācijas profilā aprakstīta funkciju grupu pāreja FG1 → FG2, bet var nebūt FG2 → FG1.

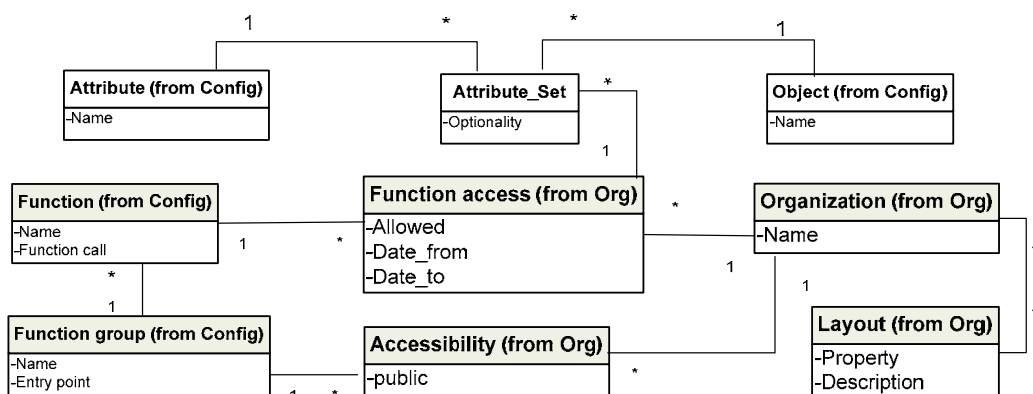
Ieejas punktu TIS nosaka kortežs (piekļuves\_vieta, piekļuves\_veids). Piekļuves\_vieta šai gadījumā var būt lietotāja ar saiti no pārlūkprogrammas izsaukta procedūra, vai no TIS sākuma lapā ietvertām daudzām saitēm lietotāja izvēlēta viena konkrēta saite, kas realizē noteikta veida „login formu”. Piekļuves vieta savukārt ļauj kontrolēt piekļuves veidu. Piemēram, atšķirīgos veidos var tikt kontrolēta pieeja pilnvarotajai TIS daļai (datu bāzes autentifikācijas mehānisms) vai organizācijas darbinieku pašapkalpošanās funkcijām (LDAP autentifikācija).

Konfigurācijas profila paplašinājumā ietilpst klase *Object State* (Objekta stāvoklis), kas reprezentē situāciju, ka funkcijas var būt atkarīgas no objektu stāvokļiem. Funkcija var būt pieejama un līdz ar to tikt izpildīta dažos stāvokļos, bet citos nē. Stāvokli definē objekta (klase *Object*) atribūts (klase *Attribute*) vai atribūtu kopa. Ja funkcijai, atbilstoši profilam, definēts pieļaujama objekta sākuma stāvoklis (ar attiecību *initial state*), tad funkcija var veikt manipulācijas ar šī objekta datiem. Kad funkcija ir izpildīta, tā maina objekta stāvokli uz iepriekš definētu beigu stāvokli (to apraksta attiecība *final state*).

DNATIS organizācijas līmeņa instances konfigurācijas profils ir aprakstīts ar modeli, kas dots 7. attēlā. Šis profils pēc būtības definē katrai organizācijai, kāda būs šīs organizācijas TIS konfigurācija un noformējums. Organizācijas līmeņa konfigurācijas profils sastāv no pamatprofila un paplašinājuma.

Organizācijas līmeņa konfigurācijas pamatprofils apraksta DNATIS arhitektūras konfigurācijas pamatprofila speciālgadījumu konkrētai organizācijai. Modeļa elementi, kas definē šo pamatprofilu ir klases Organizācija (*Organization*), Funkciju pieejamība (*Function access*), Pieejamība (*Accessibility*), Noformējums (*Layout*), kā arī tiek izmantota saistība ar

DNATIS konfigurācijas pamatprofila elementiem - klasēm Funkcija (*Function*) un Funkciju grupa (*Function Group*).

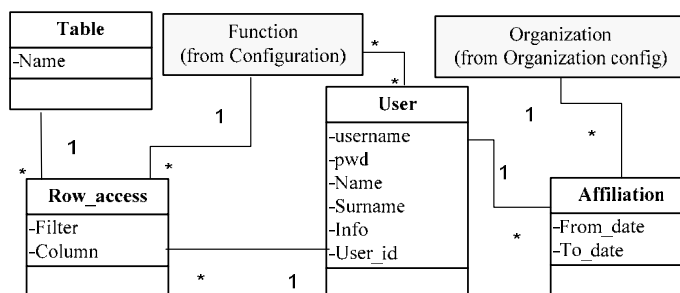


7. attēls. DNATIS organizācijas līmeņa instances konfigurācijas profila modelis

*Organizācija* reprezentē tās biznesa organizācijas, kas izmanto konkrēto TIS. *Pieejamība* definē TIS organizācijas līmeņa instanci, aprakstot, kuras no Funkciju grupām ir pieejamas konkrētai organizācijai, kas izmanto TIS, jo dažādas organizācijas var izmantot dažādas funkciju grupas. Funkciju grupu iespējams definēt arī kā publiski pieejamu, izmantojot atribūtu „Public”. *Funkciju pieejamība* definē, kuras no funkcijām un kādā laika intervālā ir pieejamas organizācijai, ar nosacījumu, ka atbilstošā funkciju grupa, kurā ietilpst funkcija, ir pieejama organizācijai. *Noformējums* apraksta atsevišķu vizuālo elementu īpašības, ja ir nepieciešams organizācijai specifisks TIS izskats, piemēram, tādiem elementiem, kā stils, organizācijas logo, fons, noklusētais fonts utt.

TIS organizācijas līmeņa instancei organizācijas profila paplašinājums atļauj definēt atribūtu kopu (*Attribute Set*), ko izmanto funkcija, un kas būtu atšķirīga no tās atribūtu kopas, kas definēta sākotnējā TIS. Atribūtu kopai (*Attribute Set*) ir asociācijas ar klasēm *Attributes* un *Objects* no DNATIS arhitektūras konfigurācijas profila. Atribūtu kopas īpašība “*optional*” apraksta, vai atribūts, kas iekļauts atribūtu kopā, ir obligāts funkcijas izpildes loģikai.

Sistēmas lietotāja profils DNATIS arhitektūrā ir aprakstīts ar modeli, kas dots 8. attēlā. Sistēmas lietotāja profils apraksta lietotāja piederību noteiktā laika periodā kādai no TIS izmantojošām organizācijām, līdz ar to nosakot TIS organizācijas instances pieejamību un konfigurāciju, kā arī lietotāja tiesības uz funkcijām un ierobežojumus uz datiem TIS datu bāzē. Modeļa elementi ir Lietotājs (*User*), Piederība (*Affiliation*), Tabula (*Table*), Ierakstu Piederība (*Row access*), kā arī tiek izmantota saistība ar TIS konfigurācijas profila elementiem Funkcija un Funkciju grupa un TIS organizācijas līmeņa instances konfigurācijas profila elementu Organizācija.

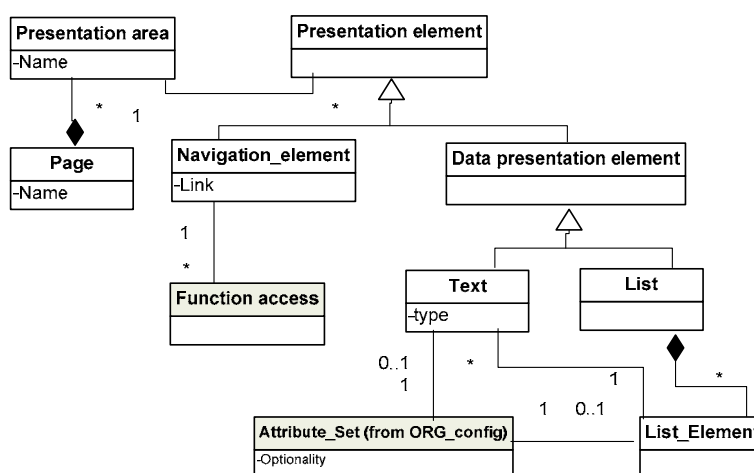


8. attēls. Sistēmas lietotāja profils DNATIS arhitektūrā

Klase *Lietotājs* apraksta sistēmas lietotāja personas datus, pieslēgšanās datus un identifikatoru, kas raksturo lietotāja datu piederību TIS datu bāzē. Klase *Piederība* apraksta lietotāja piederību organizācijai noteiktā laika periodā. Lietotājam ir tiesības strādāt ar noteiktām funkcijām organizācijas TIS instances ietvaros, ko definē TIS konfigurācijas profils. Lietotājs ar sev atļautajām funkcijām var piekļūt datiem TIS datu bāzē, bet šo piekļuvi apraksta kolonnu filtri, kas definēti *Ierakstu\_pieejā* no *Tabulām*.

### 3.3.2. DNATIS Prezētācijas modelis

Lai aprakstītu, kā TIS instance tiek attēlota lietotājam organizācijas līmenī, DNATIS arhitektūrā tiek izmantots prezētācijas modelis (9. attēls), kura elementi apraksta TIS lietotāja saskarnes uzbūvi. Kā jau tika minēts nodaļas sākumā DNATIS arhitektūras ierobežojumos, tad TIS prezētācijas adaptācija tiek nodrošināta tikai tādā apjomā, lai lietotājam attēlotu viņam pieejamās funkcijas un datus, līdz ar to šis pieņēmums arī nosaka prezētācijas modeli.



9. attēls. DNATIS prezētācijas modelis

Lapa (*Page*) ir galvenais prezētācijas modeļa elements, kas ietver visus pārējos lietotāja saskarnes elementus. Lapa paredzēta, lai attēlotu Funkciju grupu no TIS pamatkonfigurācijas, ko definē konfigurācijas profils.

Lapa sastāv no vismaz viena prezētācijas apgabala (*Presentation area*), kas paredzēts Prezētācijas elementu (*Presentation elements*) grupēšanai. Prezētācijas elements var būt navigācijas elements (*Navigation element*) vai datu prezētācijas elements (*Data presentation element*).

Navigācijas elements tiek attēlots lietotāja saskarnē kā saite, kas savukārt reprezentē funkciju izsaukumus. DNATIS arhitektūrā navigācijas elements nav tiešā veidā saistīts ar TIS funkciju, kas definēta konfigurācijas profilā, bet ir saistīts ar klasi Funkciju pieejamība (*Function access*), kas attēlo lietotājam tikai viņam atļautās funkcijas no lietotājam atbilstošās TIS organizācijas līmeņa instances.

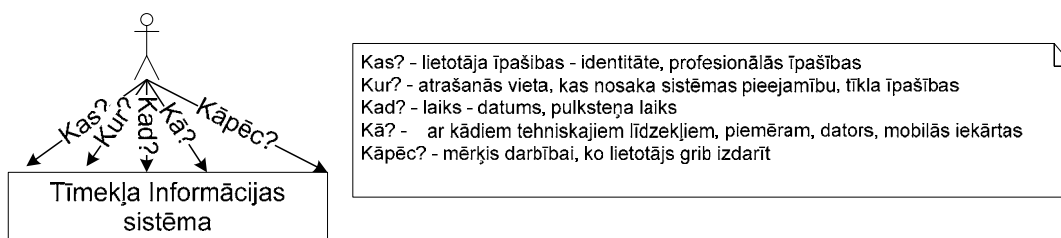
Datu prezētācijas elements var būt teksts (*Text*) vai saraksts (*List*), kas abi dažādos veidos attēlo objekta atribūtus. DNATIS arhitektūrā datu prezētācijas elementi nav tiešā veidā saistīti ar klasi *Atribūts*, bet gan ar klasi *Atribūtu kopa*, lai nodrošinātu, ka lietotāja saskarnē tiktu attēloti tikai viņam atļautie dati no viņam atbilstošās TIS organizācijas līmeņa instances.

Saraksts sastāv no *Saraksta elementiem*, kas arī paredzēti atribūtu attēlošanai no *Atribūtu kopas*. Mērķis *Saraksta elementa* ieviešanai ir nodrošināt vairāku vērtību attēlošanu lietotāju saskarnē atribūtiem no *Atribūtu kopas*. *Saraksta elementi* var būt saistīti arī ar *Teksta elementu*, kas šai gadījumā lietotāja saskarnē var attēlot papildus informāciju par katru no saraksta elementiem. Īpašība „Type” Teksta veida Datu prezentācijas elementam apraksta, vai dotais datu prezentācijas elements konkrētajai TIS instancei var tikai attēlot datus, vai arī ir atļauta datu labošana.

### 3.3.3. DNATIS arhitektūras komponenti

#### 3.3.3.1. Konteksta monitors

Sistēmas lietošanas konteksts ir apkārtējo vidi raksturojošu īpašību kopa, kas tajā brīdī, kad notiek lietotāja mijiedarbība ar TIS, var to noteiktā veidā ietekmēt. Kontekstu veido lietotāja, vietas, laika, tehnisko līdzekļu īpašības, kā arī mērķis vai nodoms, ko lietotājs grib veikt, pieslēdzoties sistēmai, un shēmā tas redzams 10. attēlā.



10. attēls. Sistēmas lietošanas konteksts

DNATIS arhitektūras sistēmas konteksta monitora uzdevums ir noteikt visas šīs pieminētās īpašības un nodot tās adaptācijas komponentam, lai veiktu TIS adaptāciju. Konteksta monitors identificē kontekstu raksturojošās īpašības un salīdzina ar DNATIS arhitektūras atbilstošā profila atribūtu vērtībām, nosaka kontekstam atbilstošās citas profila atribūtu vērtības, kas raksturo kontekstu, vai arī atsevišķas konteksta īpašību vērtības uzreiz bez salīdzināšanas tiek nodotas adaptācijas komponentam. Konteksta īpašību atbilstību profila elementiem (vai tiešu izmantošanu), kā arī noteikšanas veidu var redzēt 1. Tabulā.

1. Tabula. Saistība starp kontekstu un profila elementiem

Konteksta veids	Konteksta īpašība un noteikšanas veids	Profils	Atbilstošie profila elementi
Lietotājs	Identitāte (identificē monitors)	TIS Lietotāja profils	User (Username, pwd)
	Profesionālās īpašības, piemēram, organizācija, kurā strādā (nolasa no profila)	TIS Lietotāja profils, TIS Organizācijas profils	Organization, Affiliation (From_date, To_date)
Vieta	Piekļuves vieta, kas nosaka sistēmas pieejamību (identificē monitors)	TIS Konfigurācijas profils	Function Group (name, entry point)
Laiks	Datums (identificē monitors)	Sistēmas lietotāja profils, TIS Organizācijas profils	Affiliation (From_date, To_date) Function_access (date_from, date to)
Tehniskie līdzekļi	Līdzekļu veids, piemēram, mobilās iekārtas (identificē monitors)	Neapraksta profilā	Neapraksta profilā
Mērķis	Darbība	TIS Konfigurācijas profils	Function Group (name, entry point)

Noteikšanas veids var būt gan konteksta monitora darbības rezultāts (t.i. īpašību identificē monitors kā rezultātu lietotāja mijiedarbībai ar TIS), vai arī konteksta monitors konteksta īpašību identificē, nolasot no atbilstošā profila pēc kādas citas, konteksta monitora jau iepriekš identificētas konteksta īpašības, piemēram, lietotāja identitāte ļauj noteikt viņam atbilstošo organizāciju, kas tālāk jau tiek izmantota, lai veidotu atbilstošo TIS organizācijas līmeņa instanci.

### 3.3.3.2. *Adaptācijas komponents*

DNATIS arhitektūras adaptācijas komponents veic divu līmeņu adaptāciju: rupjo adaptāciju un detalizēto adaptāciju.

Adaptācijas komponenta darbības rezultāts ir tīmekļa informācijas sistēmas instance, kas adaptēta divos līmeņos:

- rupjās adaptācijas rezultāts - organizācijas līmenī ir izveidota konkrētajai organizācijai adaptēta sistēmas konfigurācija; veikta Funkciju grupu apvienošana no vairākām dažādu organizāciju tīmekļa informācijas sistēmas instancēm, pie nosacījuma, ja lietotājs ir piederīgs vairākām organizācijām vienlaikus, un šī piederība aprakstīta arī lietotāja profilā,
- detalizētas adaptācijas rezultāts - individuāla lietotāja līmenī ir izveidota individuāla lietotāja sistēmas instance, ko raksturo:
  - adaptēta tīmekļa lietotnes navigācija,
  - adaptēts tīmekļa lietotnes saturs.

### 3.3.3.3. *Rupjās adaptācijas operācijas*

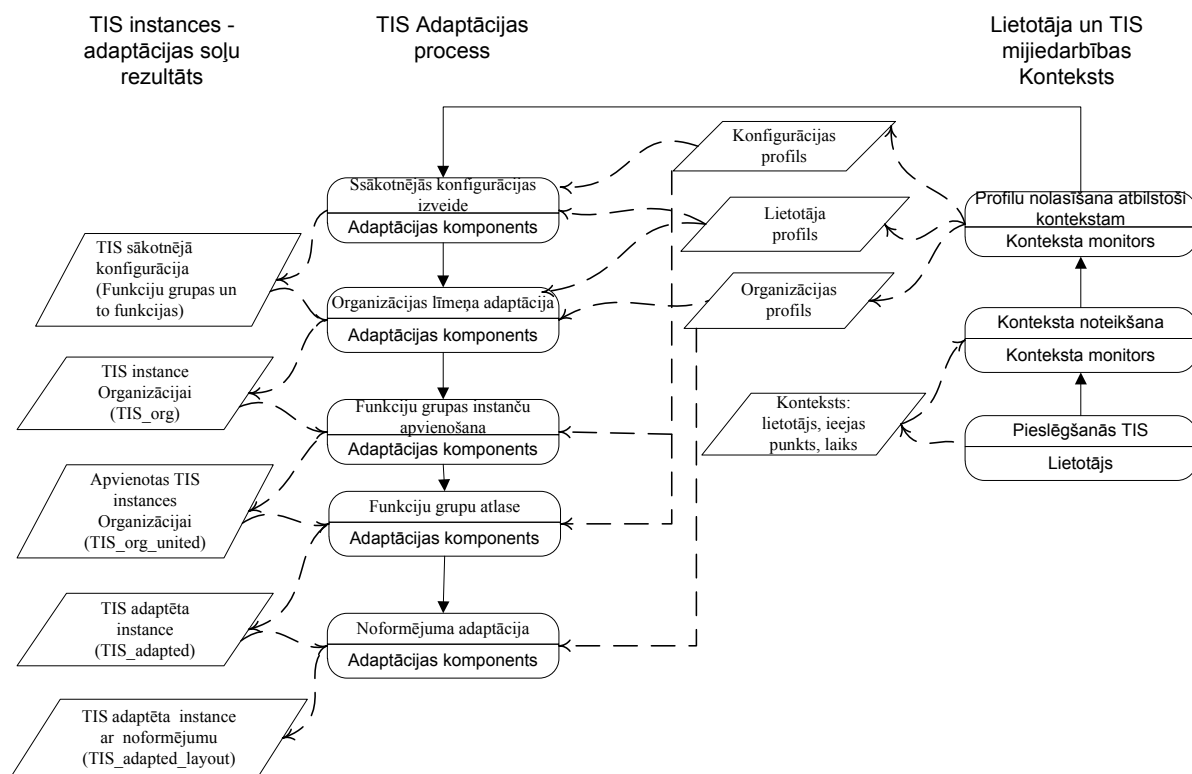
DNATIS adaptācijas komponents no konteksta monitora saņem informāciju par sistēmas lietošanas kontekstu – lietotāju, pieslēguma punktu, laiku, kā arī pārējos konteksta datus. Lai veiktu adaptāciju, adaptācijas komponents no DNATIS arhitektūras datu slāņa nolasa kontekstam atbilstošo TIS konfigurācijas profilu, TIS organizācijas līmeņa instances konfigurācijas profilu un lietotāja profilu.

Rupjās adaptācijas ietvaros jāveic sekojošas operācijas atbilstoši kontekstam:

1. sākotnējās konfigurācijas izveide,
2. organizācijas līmeņa adaptācija,
3. funkciju grupas instanču apvienošana,
4. funkciju grupu atlase,
5. noformējuma adaptācija.

Operāciju izpildes secību TIS adaptācijas procesa ietvaros, profilu izmantošanu katrā no operācijām, konteksta noteikšanu brīdī, kad lietotājs mijiedarbojas ar TIS, kā arī katra adaptācijas soļa rezultātu – kādas izmaiņas notikušas TIS instancēs – var redzēt tālāk dotajā shēmā (11. attēls).

Tālāk nodaļā doti algoritmu apraksti adaptācijas operācijām. Lai pierakstu vienkāršotu, tālāk dotajos algoritmu aprakstos netiek precizēts no kāda profila tiek izmantota klase, izmantotos profilus var noteikt pēc shēmas (11. attēls).



11. attēls. Adaptācijas darbību, dažādas adaptācijas pakāpes TIS instanču, konteksta un profilu savstarpējo sakarību shēma

Tālāk doti apzīmējumi, kas izmantoti, lai aprakstītu algoritmus, kas atbilst katrai adaptācijas operācijai.

- **For each ... do** apzīmē ciklā izpildītas darbības,
- **return** ir algoritma darbības rezultāts,
- **If** nosacījums **Then** darbības **<Else darbības>** apzīmē izvēli, atbilstoši nosacījuma vērtībai,
- **Build** **<class>** (parametri) – apzīmē implementāciju atbilstošas klases elementiem, parametru saraksts atkarīgs no atribūtiem, kas definēti dotajai klasei,
- **Build\_copy** (jaunās TIS instances nosaukums, dublējamās TIS nosaukums),
- **el from** **<class>** viena dotās klases elementa izvēle,
- **Show** (el from **<class>**, TIS\_instance) apzīmē atbilstošās klases elementa izvēli tajā TIS instancē, kas tiek nodota kā parametrs),
- **<class 1>.<class 2>** apraksta izvēli tiem klases **<class 2>** elementiem, kas saistīti ar atbilstošo **<class 1>** elementu,
- **el.<attribute>** apraksta klases elementa atribūtu.

#### TIS sākotnējās konfigurācijas izveide.

*Neformālais apraksts.* Apzīmēsim TIS sākotnējās konfigurācijas izveides operāciju ar `start_konfig (user)`. Ja lietotājs ir reģistrēts TIS sistēmas lietotājs, tad tiek veidota TIS sākotnējā konfigurācija, pretējā gadījumā var būt atšķirīgi risinājumi atkarībā no TIS realizācijas. Gan tad, ja atrod, gan ja neatrod reģistrētu lietotāju tiek piedāvāta TIS publisko funkciju grupa, ja tāda ir paredzēta TIS konfigurācijā. Sākotnējā TIS konfigurācija

TIS\_konfig tiek izveidota atbilstoši TIS konfigurācijas profilam, iekļaujot TIS visas profilā aprakstītās funkciju grupas un tām piederošās funkcijas.

*Algoritms.*

```
start_konfig (user).
If user is in User Then
  For each el from Function_group do
    Build_Function_group(el.name, el.entrypoint)
    For each el from Function_group.Function do
      Build_f (el.name, el.function call)
  Else Build_fg ('public')
    For each el from 'public'.Function do
      Build_function (el.name, el.function call)
Return (TIS_konfig)
```

### TIS organizācijas līmeņa adaptācija.

*Neformālais apraksts.* Apzīmēsim TIS organizācijas līmeņa adaptācijas operāciju ar *adapt\_org*(TIS\_konfig, user, time).

Operācijas darbības mērķis ir sākotnējo sistēmas konfigurāciju, ko iegūst ar *start\_konfig* operāciju, adaptēt atbilstoši konkrētās organizācijas vajadzībām, kas ir aprakstītas TIS organizācijas līmeņa instances konfigurācijas profilā. Operācija *adapt\_org* nosaka konkrēto organizāciju (vai organizācijas) atkarībā no lietotāja (*user*), kas pieslēdzies sistēmai, un no tā, kāda informācija par lietotāju un viņa piederību organizācijām iegūstama no lietotāja profila. Pēc tam pārbauda sistēmas konfigurācijā ietilpstošo Funkciju grupu un Funkciju pieejamību iepriekš noteiktajai organizācijai atkarībā no pieslēguma laika (*time*). Tiek izveidota adaptēta TIS instance TIS\_org.

Atbilstoši sistēmas lietotāja profilam, lietotājs var noteiktā laika periodā būt piederīgs vairākām organizācijām. Šajā gadījumā katrai organizācijai tiek izveidota sava adaptēta TIS instance (gadījumā, ja organizācijas ir divas, tad izveido TIS\_org1 un TIS\_org2), atbilstoši organizācijas līmeņa instances konfigurācijas profilā aprakstītajām īpašībām. Katrai organizācijai iepriekš aprakstītās *adapt\_org* darbības veic atsevišķi.

*Pieņēmumi.* Tālāk dotā algoritma darbībai no datu iepriekšējās sagatavošanas viedokļa ir jābūt spēkā:

1. organizācijas profilā, izmantojot klasi *Accessibility* ir definētas tikai tās TIS funkciju grupas, kas realizētas un ieviestas konkrētā organizācijā, ko reprezentē klase *Organization*.
2. organizācijas profilā, izmantojot klasi *Function\_access* ir definētas tikai tās funkcijas, kam ir spēkā ierobežojumi atkarībā no laika.

*Algoritms.*

```
adapt_org (TIS_konfig, 'user', 'time')
For each el from 'user'.Organization do
  Build_copy (TIS_el, TIS_konfig)
  For each fg from Function_group do
    If fg is in el.Accessibility then
      Show (el.Accessibility.Function_group, TIS_el)
      For each el. Function_access do
        If 'time' is between (el. Function_access.date_from,
                              el. Function_access.date_to) then
          Show (el.Function_access.Function, TIS_el)
Return (TIS_el)
// katrai dotā lietotāja organizācijai ir izveidota sava
// TIS organizācijas līmeņa instance
```



## TIS Funkciju grupu instanču apvienošana.

*Neformālais apraksts.* Sekojošā adaptācijas operācija vajadzīga gadījumā, ja lietotājs pieder vairākām organizācijām, un organizācijas līmeņa adaptācijas rezultātā ir izveidotas vairākas TIS instances. Apzīmēsim TIS Funkciju grupu instanču apvienošanas operāciju ar  $FG\_union(TIS\_1, TIS\_2)$ .

Apzīmēsim ar  $FGj\_1$  un  $FGj\_2$  vienas un tās pašas funkciju grupas  $FGj$  divas instances. Pieņemsim, ka katra no tām ietilpst kādā no TIS instancēm –  $TIS\_1$  un  $TIS\_2$ .

Funkciju grupas instanču apvienošanas operācija  $FG\_union$  no divām sistēmas instancēm apvieno tikai vienādas funkciju grupas abām instancēm, pie nosacījuma, ja lietotāja profilā dotajam lietotājam ir definēta piederība abām organizācijām, kam izveidotas šīs sistēmas instances. Apvienotā instance  $TIS\_1\_2$  sastāv gan no šādām apvienotām funkciju grupām, gan no  $TIS\_1$  un  $TIS\_2$  funkciju grupām, kam TIS organizācijas līmeņa konfigurācijas profilā bija norādīta piederība tikai vienai no organizācijām, kam tiek veikta TIS adaptācija.

*Pieņēmumi.* Tālāk dotais algoritms strādā ar izveidotajām TIS instancēm, tiek izmantots tikai DNATIS arhitektūras konfigurācijas profils, lai zinātu, kādas vispār TIS ir funkciju grupas un tām atbilstošās funkcijas. Tās funkciju grupas, kas neietilpst kādas atsevišķas instances sastāvā, kā tas definēts konfigurācijas profilā (tas pats attiecas uz funkcijām), iepriekšējā solī *adapt\_org* darbības rezultātā iegūtajās TIS organizācijas līmeņa instancēs nav pieejamas.

⊗ operācija algoritmā nozīmē divu funkciju grupu apvienošanu.

### *Algoritms.*

```
FG_union (TIS_1, TIS_2).  
If TIS_2 is not null then  
For each el from Function_group do  
  If Function_group (el.name) is in TIS_1 then  
    Function_Group_united(el.name) := Function_Group_united (el.name) ⊗  
                                     Function_group (el.name) from TIS_1  
    For each f from el.Function do  
      If f.name is in TIS_1 then  
        Function_Group_united (el.name).Functions :=  
          Function_Group_united(el.name).Functions ⊗  
          Function_Group(el.name).Functions from TIS_1  
  If Function_group (el.name) is in TIS_2 then  
    Function_Group_united(el.name) := Function_Group_united (el.name) ⊗  
                                     Function_group (el.name) from TIS_2  
    For each f from el.Functions do  
      If f.name is in TIS_2 then  
        Function_Group_united(el.name).Functions :=  
          Function_Group_united(el.name).Functions ⊗  
          Function_group(el.name).Functions from TIS_2  
Return (TIS_org_united)
```

## Funkciju grupu atlase.

*Neformālais apraksts.* Operācija vajadzīga gadījumam, ja lietotājs pieder vairākām organizācijām, un ir veiktas *adapt\_org* un *FG\_union* operācijas. Tālāk vajadzīga Funkciju grupu atlase, atkarībā no ieejas punkta. Apzīmēsim ar *FG\_Select* (ieejas\_punkts) funkciju grupu atlases operāciju.

Pieņemsim, ka parametrs 'ieejas\_punkts' ir funkciju grupai FGk. Apzīmēsim ar FGi patvaļīgu funkciju grupu, ar Pij pāreju no FGi uz FGj. Iespējams definēt pāreju virknes PVk, ko veido secīgas pārejas Pij starp funkciju grupām, sākot no funkciju grupas FGk. Iespējamās dažādās PVk nosaka DNATIS arhitektūras konfigurācijas profila dati.

FG\_Select(ieejas\_punkts) darbības rezultāts ir adaptēta apvienotā TIS instance, ko veido visas funkciju grupas FGi, kas sasniedzamas no FGk ar pāreju virknēm PVk.

*Pieņēmumi.* TIS konfigurācijas profilā ir iepriekš definētas iespējamās pārejas starp funkciju grupām, kas tehniski ir realizējamas, atkarībā no tā, ko lietotājs izvēlēties par ieejas punktu sistēmā, lai varētu veidot vienotu TIS no apvienotām organizācijas līmeņa TIS instancēm. Pieņemot, ka ir definētas arī pārejas starp vienāda nosaukuma funkciju grupām no dažādām TIS instancēm, kas tika iekļautas apvienotajā instancē TIS\_united; šāda pāreja starp vienādām grupām var būt un var arī nebūt atļauta, atkarībā no funkciju grupu realizācijas tehnoloģijas.

*Algoritms.*

```
FG_Select (ieejas_punkts)
  FGe:= Function_Group(entry_point).name // izmanto profilu
For each FG in Function_Group_united (FGe) do // apstrādā TIS instanci
  If Function_group(FG).Transition(From).Allowed = 1 then
    Show(Function_group(FG).Transition(From).Transition(To).
          Function_Group,TIS_united)
Return (TIS_adapted)
```

### TIS noformējuma adaptācija.

*Neformālais apraksts.* Apzīmēsim ar adapt\_noform operāciju, kas veic TIS noformējuma adaptāciju organizācijas vajadzībām. Vienkāršākajā gadījumā, ja lietotājs pieder vienai organizācijai, tad tiek noskaidroti noformējuma elementi un to īpašības no DNATIS arhitektūras organizācijas līmeņa konfigurācijas profila un tiek pielietoti TIS. Ja lietotājs pieder vairākām organizācijām, tad tiek izmantoti noformējuma elementi, kas atbilst TIS organizācijas līmeņa instancei, kuras ieejas punktu lietotājs ir izmantojis.

*Algoritms.*

```
Adapt_noform(ieejas_punkts)
For each Layout.Property in
  Function_Group(ieejas_punkts).Accessibility.Organization.Layout do
  Apply (Layout.Property, TIS_adapted)
Return (TIS_adapted_layout)
```

#### 3.3.3.4. *Detalizētās adaptācijas operācijas*

Detalizētās adaptācijas pamatā ir informācija no DNATIS arhitektūras lietotāja profila. Tiek veikta rupjās adaptācijas rezultātā iegūtās TIS instances tālāka adaptācija. Detalizētajai adaptācijai ir divas darbības:

- 1) “detalizēta adaptācija – navigācijai”,
- 2) “detalizēta adaptācija – saturam”.

Tiek veiktas divas operācijas: *adapt\_navig(user)* un *adapt\_data(user)*. Operācijas tiek pielietotas rupjās adaptācijas rezultātā izveidotajai vienas vai vairāku organizāciju līmenī

adaptētajai un apvienotajai TIS instancei. Operāciju izpildei vēl tiek izmantota arī informācija no TIS lietotāja profila.

### Operācija adapt\_navig (user)

*Neformālais apraksts.* Operācija atbilstoši operācijas parametram ‘user’ noskaidro lietotāja tiesības uz funkcijām no lietotāja profila. Rupjās adaptācijas līmenī izveidotās TIS organizācijas instances ietvaros tiek izveidota lietotāja tiesībām atbilstoša adaptēta TIS navigācija. Tiek izveidota lietotāja TIS instance *TIS\_user*.

*Pieņēmumi.* Realizācijā TIS funkcijām kā navigācijas elements ir izmantotas saites, kas nodrošina piekļuvi atbilstoši funkcijai. Detalizētāk par TIS prezentācijas elementu izmantošanu piedāvātajā TIS adaptācijas arhitektūrā atrodams apakšnodaļā par lietotāja saskarnes adaptāciju (skat. apakšnodaļu 3.4)

#### *Algoritms.*

```
Adapt_navig (user)
For each el from Function (user). Function do
    If el is in User(user). Function then
        Show (Function(el), TIS_adapted_layout)
Return (TIS_user)
```

### Operācija adapt\_data (user)

*Neformālais apraksts.* Operācija atbilstoši operācijas parametram ‘user’ noskaidro, kādi ir lietotāja ierobežojumi uz datiem, kas definēti, aprakstot tabulas, kolonnas un tām pielietotos filtrus. TIS lietotāja instance *TIS\_user* tiek papildināta ar saturu, ko vispārīgā gadījumā definē funkciju izmantotie dati (attiecība Tabulas – Funkcijas lietotāja profilā), bet tiek veikta adaptācija atbilstoši lietotāja ierobežojumiem uz datiem. Šādā veidā ar kolonnu filtriem ir iespējams definēt vienkāršākus adaptācijas likumus. Sarežģītākos gadījumos adaptācijas likumi var tikt definēti Adaptācijas likumu bāzē, piemēram, ar OCL (*Object Constraint Language*) [OCL] teikumu palīdzību un būtībā šie likumi tāpat apraksta lietotāja ierobežojumus datiem.

Detalizētāk par TIS datu elementu attēlošanu lietotāja saskarnē un datu glabāšanas risinājumiem TIS adaptācijas arhitektūrā atrodams punktā par lietotāja datu integrāciju saskarnē (skat. punktu 3.4.3).

#### *Algoritms.*

```
Adapt_data (user)
For each el in User(user).Row_access do
    F:= CreateFilter (el.Table.name, el.column, el.Filter)
    ApplyFilter (F, el.Function, TIS_user)
Return (TIS_user_data)
```

#### 3.3.3.5. Piemērs adaptācijas komponenta darbībai

Apskatīsim Augstskolu informācijas sistēmu, kas realizēta atbilstoši DNATIS arhitektūrai (skat. 4. attēlu). Arhitektūras komponentus apzīmējot ar atbilstošiem Augstskolu IS komponentu nosaukumiem (attēlotas ir divas no visām iesaistītajām augstskolām - LU un LiepU un trīs dažādas funkciju grupas), arhitektūras shēma dota 12. attēlā.

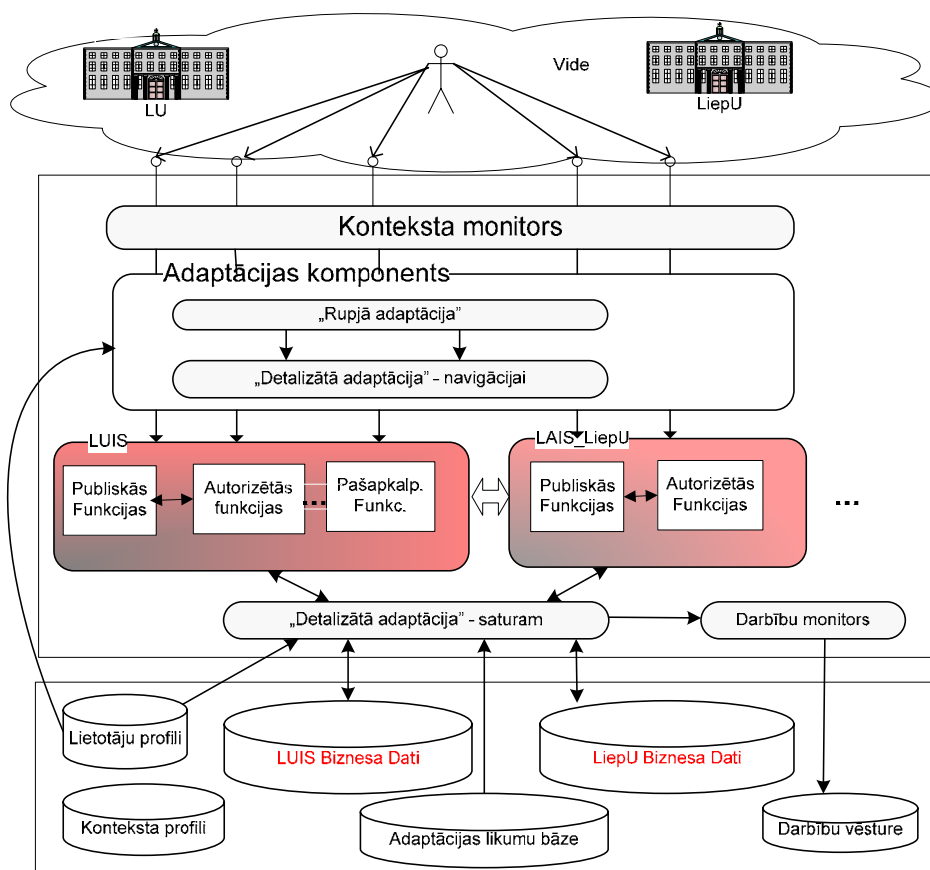
Augstskolu informācijas sistēmas konfigurācijas profilā ir definētas 3 funkciju grupas:

1) Apzīmēsim ar FGa – pilnvaroto funkciju grupa. Par pilnvarotām funkcijām šajā gadījumā uzskatīsim tās funkcijas, kas paredzētas dažādu augstskolas biznesa funkciju veikšanai. Šīs funkcijas veic darbības ar studentu vai darbinieku datiem, piemēram, atzīmju ievads, rīkojumi par studentu statusa maiņu, darbinieku rīkojumi u.c. Galvenā pazīme – veicot šīs funkcijas, lietotājam tiek atļauts strādāt ar citu cilvēku datiem savu darba pienākumu ietvaros un ir definēts ierobežojums uz pieejamo informāciju.

2) Ar FGs apzīmēsim pašapkalpošanās servisa funkciju grupu. Pašapkalpošanās funkciju grupu veido funkcijas, ko lietotājs var izmantot tikai savu datu vai ar tiem saistītu datu stāvokļa mainīšanai. Piemēram, lietotājs var pierēģistrēties uz semestri, pieteikties uz studiju kursiem, veikt maksājumu par sevi, apskatīties datus par savu datu stāvokli uz doto brīdi – personas datiem, rīkojumiem, atzīmēm u.c.

3) Ar FGp apzīmēsim publisko funkciju grupu. Tās ir pieejamas jebkuram publiskam lietotājam bez lietotājvārda un paroles ievada un nodrošina pieeju dažādām publiskām atskaitēm, kas ietver publiskus datus.

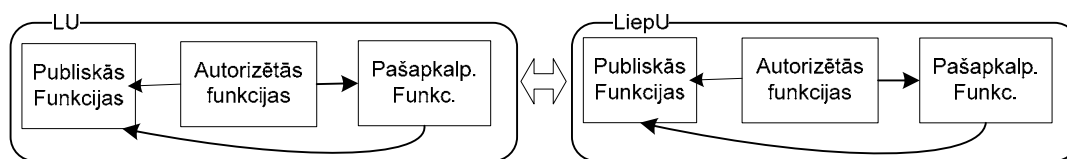
Konfigurācijas profilā ir arī informācija par stilu, augstskolas logo, un citiem noformējuma elementiem.



12. attēls. DNATIS arhitektūras shēma aprobācijas projektā augstskolu IS (saskaņā ar 4. attēlu)

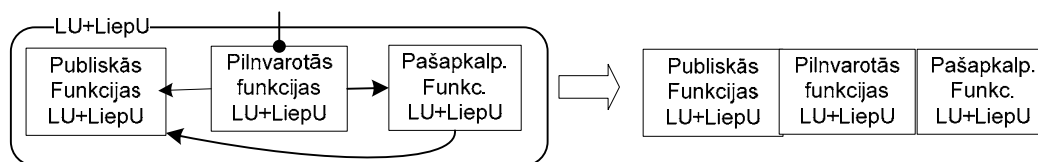
Apskatīsim gadījumu, kad lietotājs pieslēdzas augstskolu informācijas sistēmai, un lietotājs ir darbinieks Liepājas Universitātē, kā arī studē Latvijas Universitātē doktorantūrā. Tā kā sistēmai ir vairāki ieejas punkti – atkarībā no tā, cik augstskolas piedalās projektā, tad konteksta monitors identificē pieslēguma punktu, lietotāju, pieslēguma laiku un vietu. Rupjās adaptācijas komponents nosaka, ka jāizveido 13. attēlā redzamās TIS instances ar LU un LiepU konfigurācijā definētām funkciju grupām – abu augstskolu sistēmu instancēm ir

definētas visas trīs funkciju grupas. Konfigurācijā ir definētas arī iespējamās pārejas starp funkciju grupām, ko zīmējumā attēlo bultiņu virzieni (nav atkarīgs no augstskolas un nav adaptējams organizācijai, šai gadījumā augstskolai).



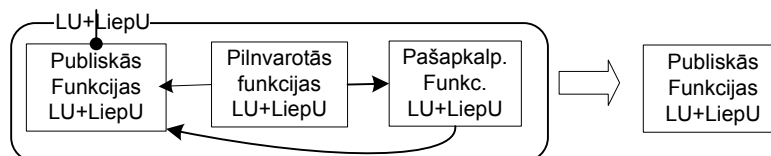
13. attēls. Augstskolu TIS instances funkciju grupas un to pārejas

Adaptācijas komponents nosaka pēc lietotāja profila datiem, kuras funkciju grupas ir pieejamas individuālajam lietotājam katrā no sistēmas instancēm un izveido apvienoto sistēmas instanci lietotājam. Atbilstoši piemēra situācijai (LiepU darbinieks un LU students), ja lietotājs ir pieslēdzies, izmantojot pieeju LiepU pilnvarotajām funkcijām, tad sistēmas organizācijas līmeņa instance būs tāda, kāda dota 14.attēlā.



14. attēls. Augstskolu TIS organizācijas līmeņa instance atkarībā no ieejas punkta (1. gadījums)

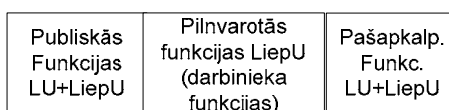
Ja lietotājs pieslēgsies, izmantojot pieeju LU publiskajām funkcijām, tad atbilstoši nedefinētajām funkciju grupu pārejām, sistēmas organizācijas līmeņa instance būs tāda, kāda dota 15.attēlā.



15. attēls. Augstskolu TIS organizācijas līmeņa instance atkarībā no ieejas punkta (2. gadījums)

Detalizētās adaptācijas rezultātā TIS organizācijas līmeņa instance tiek adaptēta individuālam lietotājam, kas piemēra gadījumā ir LiepU darbinieks un LU students. Viņam lietotāja konfigurācijā ir definētas tiesības uz LiepU pilnvarotajām funkcijām, un uz pašapkalpošanās funkcijām – gan darbiniekiem pieejamām funkcijām, gan studentiem pieejamām funkcijām. Publiskās funkcijas redzamas visas.

Zīmējumos redzamajos piemēros 1. gadījumā (skat. 14. attēlu) tiktu adaptēta navigācija, pilnvarotās funkcijas paliktu tikai atbilstoši lietotāja darba pienākumiem LiepU un rezultāts izskatītos atbilstoši 16. attēlam.



16. attēls. Detalizēta adaptācija navigācijai individuāla lietotāja gadījumā

Pašapkalpošanās funkciju grupai ir redzams abu TIS instanču pilns funkciju komplekts navigācijas ziņā – gan darbinieka funkcijām, gan lietotāja funkcijām. Vēl tiek veikta detalizētā adaptācija saturam. Tādēļ, lietojot darbinieka pašapkalpošanās funkcijas, dotais lietotājs redz datus par sevi kā LiepU darbinieku, bet lietojot studenta pašapkalpošanās funkcijas dotais lietotājs redz datus par sevi kā LU studentu.

Pilnvarotās funkcijas arī tiek detalizētāk adaptētas satura ziņā – tiek pielietoti lietotājam definētie datu ierobežojumi, līdz ar to apskatītā piemēra gadījumā dotais lietotājs strādās ar LiepU datiem vai to apakškopu, atkarībā no ierobežojumu definīcijām.

### 3.4. Lietotāja saskarnes veidošanas specifika DNATIS arhitektūrā

Lietotāja saskarnes konstruēšana balstās uz iepriekš aprakstīto prezentācijas modeli (skat. apakšnodaļu 3.3.2) un izmanto profilus no DNATIS arhitektūras (skat. apakšnodaļu 3.3.1.).

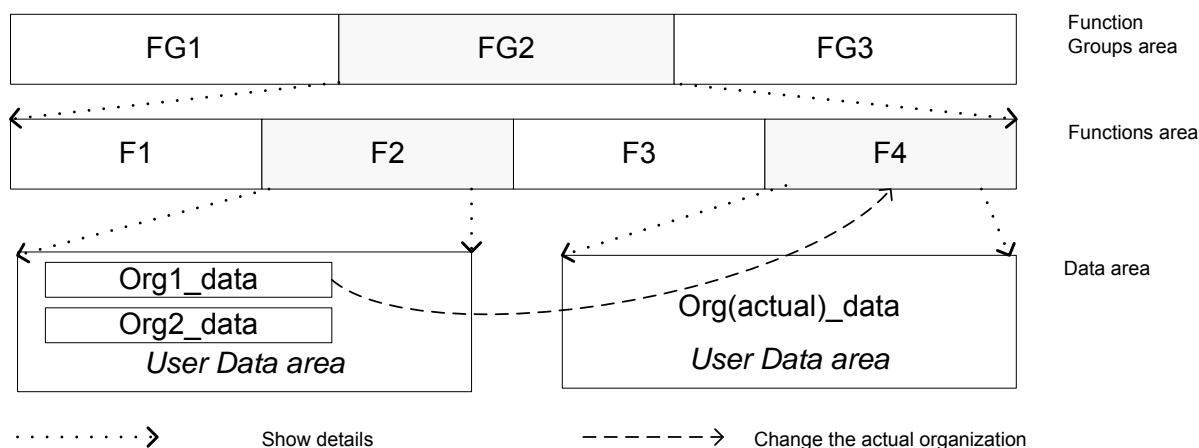
Katrā konkrētā TIS projektā var būt nedaudz atšķirīga lietotāja saskarnes uzbūve, bet atbilstoša DNATIS prezentācijas modelim. Saskarnes struktūra tāpat atkarīga no biznesa prasībām katrai TIS un DNATIS arhitektūras prezentācijas modeļa. Tālāk tiks apskatītas divas dažādas TIS saskarnes uzbūves, katra no atšķirīgas TIS, abas saskarnes tiks izmantotas divu atšķirīgu adaptācijas risinājumu ilustrēšanai.

3.4.1. punktā 17. attēlā dota saskarnes uzbūve TIS, kas izmantota Augstskolu TIS, lai aprakstītu integrētas lietotāja saskarnes izveidi, bet 3.4.4 punktā 20. attēlā redzama saskarnes uzbūve TIS, kas izmantota otrajā TIS arhitektūras aprobācijas projektā – Transportlīdzekļu reģistrā, lai ilustrētu dinamisku tīmekļa saskarnes ģenerēšanu atkarībā no lietotāja darbībām.

#### 3.4.1. Integrētas lietotāja saskarnes izveidošana vairāku organizāciju gadījumā

Apskatīsim saskarnes struktūru, kas tipiska vienam no TIS aprobācijas projektiem – Augstskolu TIS.

Apskatīsim gadījumu, kad lietotājam ir pieeja divu organizāciju TIS, kas būtībā ir vienas un tās pašas TIS divas organizācijas līmenī adaptētas instances. Tomēr lietotājs, lai strādātu, saņem savā rīcībā vienu pielāgotu, integrētu lietotāja saskarni. Abu TIS instanču apvienotās saskarnes piemērs, kuras saturs tiek izveidots, izmantojot abus adaptācijas līmeņus redzams 17. attēlā. Ar saturu šai gadījumā tiek saprastas lietotājam pieejamās funkcijas un dati.



17. attēls. TIS integrētās lietotāja saskarnes uzbūves piemērs aprobācijas projektā

Lietotāja saskarne apskatāmajai TIS sastāv no vairākiem prezentācijas apgabaliem: Funkciju grupu prezentācijas apgabala (*Function Groups Area*), Funkciju prezentācijas apgabala (*Functions Area*) un Datu apgabala (*Data Area*).

### 3.4.2. *Adaptācijas procesa rezultāti no lietotāja saskarnes veidošanas viedokļa*

TIS lietotāja saskarnes elementi tiek iegūti TIS adaptācijas procesa rezultātā, šajā punktā tiks skatīts adaptācijas process un adaptācijas operācijas no lietotāja saskarnes izveides, nevis TIS instances izveides viedokļa, kas bija aprakstīts punktā par adaptācijas komponenta darbību (skat. punktu 3.3.3.2).

Var runāt par adaptācijas procesa rezultātiem no lietotāja saskarnes viedokļa, rezultāti tiks apskatīti atsevišķi pa saskarnes elementu veidiem.

- Funkciju grupu prezentācijas apgabals satur navigācijas tipa prezentācijas elementus, kas attēlo funkciju grupas (rezultāts organizācijas līmeņa adaptācijai). Aplūkojamā piemērā 17. attēlā ir trīs funkciju grupas.

#### **1. rezultāts.** Saskarnē attēlojamo funkciju grupu noteikšana

Atbilstoši 3.4.1. punkta pieņēmumiem, TIS organizācijas līmeņa adaptācijas procesā tiek konstatēts, ka lietotājs pieder vairāk nekā vienai organizācijai. No TIS konfigurācijas profila tiek konstatēts, kuras funkciju grupas ir atļautas katrai organizācijai. Organizācijas līmenī adaptētās TIS instances izveido adaptācijas operācija `Adapt_org` (`TIS_konfig`, `'user'`, `'time'`).

Iespējamās divas situācijas, kad funkciju grupa ir definēta vairākām vai vienai organizācijai.

Ja Funkciju grupa ir definēta vairākām organizācijām, tad tiek izveidota apvienota funkciju grupa `FGi_orgk`, kur `orgk` apzīmē vienu no `k` konkrētajam lietotājam atbilstošajām organizācijām (divu organizāciju gadījumā `FGi_org1_org2`). TIS instanču apvienošanu veic adaptācijas operācija `FG_union(TIS_1, TIS_2)`. Apvienotā funkciju grupa tiek pievienota funkciju grupu komplektam, ko lietotājam attēlot Funkciju grupu prezentācijas apgabalā, izveidojot atbilstošu navigācijas prezentācijas elementu.

Funkciju grupa var būt definēta arī tikai vienai no organizācijām, ko nosaka atbilstoši visu TIS instanču konfigurācijas profiliem. Šai gadījumā funkciju grupa, bez nekādas papildus apstrādes (bez Funkciju grupu apvienošanas) tiek pievienota funkciju grupu komplektam, ko lietotājam attēlo Funkciju grupu prezentācijas apgabalā, izveidojot atbilstošu navigācijas prezentācijas elementu.

Lietotāja saskarnē būs pieejamas gan apvienotās, gan katrai organizācijai specifiskās funkciju grupas (atbilstošie navigācijas elementi).

Adaptācijas procesa operācija `FG_select` var ietekmēt Funkciju grupu prezentācijas apgabalu atkarībā no tā, kā lietotājs pieslēdzas TIS (t.i. no ieejas punkta). No konfigurācijas profila tiek noskaidrots, kādām funkciju grupām nav nodefinētas pārejas, līdz ar to lietotājam tās nebūs pieejamas un lietotāja saskarnē neparādīsies.

Piemērs. Aplūkotā piemēra gadījumā ar 2 organizācijām, kur strādā lietotājs, pieņemsim, ka pēc organizāciju profilu analīzes ir konstatēts, ka funkciju grupa `FG2` pieder abu organizāciju TIS konfigurācijai. Tādā gadījumā tiek izveidota integrēta funkciju grupa `FG2_org1_org2`, bet no lietotāja saskarnes viedokļa tiek izveidots apvienotās funkciju grupas `FG2_org1_org2` navigācijas prezentācijas elements. Pieņemsim, ka konfigurācijas profilā ir definēts, ka lietotājam pieejamā funkciju grupa `FG3` viņam ir pieejama `TIS_2`

instancē, bet ne TIS\_1 instancē. Tādā gadījumā lietotājam netiek veidota apvienota funkciju grupa, bet FG3 saturēs tikai tās funkcijas, kas lietotājam pieejamas TIS\_2 instances ietvaros komplektā ar tajā esošajiem datu ierobežojumiem, par ko aprakstīts tālāk (skat. punktu 3.4.3.).

**2. rezultāts.** Funkciju grupu attēlošana integrētajā saskarnē atkarībā no TIS lietošanas laika.

Var būt divu veidu funkciju grupas: (i) laika atkarīgas funkciju grupas, kas satur laika atkarīgas funkcijas un (ii) statiskas (laika neatkarīgas) funkciju grupas.

Konfigurācijas profilā dažas funkcijas var būt definētas ar sākuma laiku (*actual\_date\_start*) un beigu laiku (*actual\_date\_end*), kas norāda laika intervālu, kad funkcija ir pieejama TIS lietotājiem (tas var būt noteikts atkarībā no organizācijas darbības nosacījumiem). Funkciju grupa, kurā eksistē vismaz viena funkcija, kurai definēts spēkā būšanas periods, tad tiek sauktas par laika atkarīgām funkciju grupām.

Abu veidu funkciju grupas tiek izveidotas TIS organizācijas līmeņa adaptācijas procesā ar operāciju **Adapt\_org** (TIS\_konfig, 'user', 'time'), atbilstoši tam, kas norādīts TIS konfigurācijas profilā. Operācija **Adapt\_org** no TIS instances laika atkarīgām funkciju grupām izslēdz visas tās funkcijas, kas atbilstoši TIS konfigurācijas profilam nav spēkā adaptācijas brīdī.

Ja visas funkcijas kādā funkciju grupā ir laika atkarīgas un nav pieejamas uz TIS lietošanas brīdi, tad lietotāja saskarnē Funkciju grupai navigācijas prezentācijas elements netiek veidots.

**3. rezultāts.** Funkciju grupu īpašas izcelšanas iespējas lietotāja saskarnē

DNATIS arhitektūrā netika iekļauts mehānisms precīzai saskarnes izkārtojuma definēšanai (piemēram, visu funkciju savstarpējam izkārtojumam), vienīgi pēc nosaukuma var izdalīt, kuru Funkciju grupu rādīt pirmo.

Tā, piemēram, viena no funkciju grupām var tikt nosaukta "Aktuāli", kas satur uz adaptācijas momentu aktuālās funkcijas katrā organizācijā, šo funkciju grupu rāda lietotāja saskarnē kā pirmo, aktuālāko (apskatāmajā piemērā funkciju grupa FG1, skat. 17. attēlu). Lietotāja aktuālo funkciju grupa tiek dinamiski izveidota no aktuālām funkcijām visās organizācijās.

Šis rezultāts tiek sasniegts pēc **Adapt\_org** (**FG\_union** netiek pielietota) un **FG\_select** pielietošanas, Funkciju grupā 'Aktuāli' iekļaujot tikai tās funkcijas, kas pēc konfigurācijas profila ietilpst kādā no TIS\_adapted instances funkciju grupām un ir atzīmēta organizācijas profilā kā aktuāla.

Funkciju grupai 'Aktuāli' tiek izveidots navigācijas prezentācijas elements.

- Funkciju apgabals - tiek veidots, izmantojot "adaptāciju navigācijai" – **Adapt\_navig**(user) .

**4. rezultāts.** Saskarnē attēlojamo funkciju noteikšana

**Adapt\_navig**(user) operācija organizācijas līmeņa adaptācijas rezultātā iegūtai instancei TIS\_adapted atstāj tikai lietotāja tiesībās nedefinētās funkcijas, atbilstoši lietotāja profilam.

No attēlošanas viedokļa lietotāja saskarnē, visas funkcijas netiek attēlotas uzreiz, kad lietotājs pieslēdzas sistēmai. Funkcijas tiek attēlotas kā prezentācijas navigācijas elementi.

Lietotājs, strādājot ar integrēto, adaptēto saskarni, var izvēlēties kādu no funkciju grupām, ar ko strādāt. Adaptācijas procesa ietvaros no izvēlētai funkciju grupai atbilstošās



TIS instances (vai TIS instancēm, ja ir apvienotā funkciju grupa) konfigurācijas profila tiek noskaidrots, kuras funkcijas ir definētas izvēlētai funkciju grupai. Funkcijas tiek attēlotas saskarnes funkciju prezentācijas apgabalā.

Apskatāmā piemēra gadījumā funkciju apgabala saturu, t.i. attēlojamās funkcijas nosaka, kuru funkciju grupu lietotājs izvēlas- FG1 (atbilst Funkciju grupai ‘Aktuāli’), FG2(atbilst apvienotai funkciju grupai `FG2_org1_org2`) vai FG3 (funkciju grupa, kas lietotājam pieejama tikai vienā instancē: `TIS_2`) . Apskatītā piemēra gadījumā (skat. 17. attēlu) funkciju prezentācijas apgabals satur četrām funkcijām atbilstošos navigācijas prezentācijas elementus. Lietotāja saskarnē funkciju apgabalā tiek parādīts rezultāts funkciju grupas FG2 izvēlei, ko veicis lietotājs navigācijas elementa izvēles rezultātā.

## 5. rezultāts. Divu tipu funkciju attēlošanas iespēja

Arhitektūras ietvaros tiek apskatītas divu veidu funkcijas: nosakošās un atkarīgās. Nosakošās funkcijas manipulē ar datiem, kuru izvēle nosaka atkarīgās funkcijas darbību.

Apskatīsim piemēru, kas ilustrē nosakošās un atkarīgās funkcijas un to darbību. Pieņemsim, ka nosakošā funkcija ir “Parādīt lietotāja studiju datus”, kas satur tādus parametrus kā: studiju programma, studiju statuss, sākuma datums, u.c. Atkarīgā funkcija šajā gadījumā varētu būt “Parādīt lietotāja atzīmes”, kas nodrošina ar informāciju par lietotāja atzīmēm tikai izvēlētajā studiju programmā, ko iepriekš izvēlējis lietotājs nosakošās funkcijas darbības ietvaros.

17. attēlā redzamajā lietotāja saskarnē funkcija F2 ir nosakoša attiecībā pret F4. Situāciju, kad lietotājs izvēlas uzreiz F4, aprakstīta pie lietotāja datu attēlošanas datu prezentācijas apgabalā lietotāja saskarnē.

- Lietotāja datu prezentācijas apgabals - rezultāts TIS detalizētajai adaptācijai, tiek izmantota adaptācijas operācija `Adapt_data(user)`, dati tiek attēloti kā datu prezentācijas elementi, atbilstoši prezentācijas modelim.

## 6. rezultāts. Nosakošo un atkarīgo funkciju datu attēlošanas iespēja

Ir jāapskata atsevišķi divas situācijas: nosakošajām un atkarīgajām funkcijām. Nosakošās funkcijas gadījumā lietotājs var izvēlēties datus, kas tiek izmantoti kā parametri atkarīgām funkcijām.

Apskatāmā piemēra gadījumā nosakošā funkcija, ko izvēlējis lietotājs, ir funkcija F2 (skat. 17. attēlu). Tālāk lietotājs var izvēlēties datus, kas tiek izmantoti kā parametri atkarīgai funkcijai – apskatāmā piemēra gadījumā tie ir apzīmēti ar `Org1_data` un `Org2_data`.

Var runāt par aktuālo datu elementu, kas ir laika ziņā pēdējais izveidotais TIS datu elements, bet lietotāja saskarnē tiek attēlots kā pirmais. `Org1_data` ir apskatāmā piemēra aktuālais datu elements. Dažos gadījumos aktuālie datu elementi var būt sarežģītāki, par ko tiks pieminēts vēlāk, definējot nosakošo organizāciju.

Šie datu elementi vienmēr ir saistīti ar lietotāja datiem, piemēram, lietotāja studiju programmas divās universitātēs.

Atkarīgās funkcijas gadījumā, ja to sākotnēji (bez nosakošās funkcijas izvēles) ir izvēlējis lietotājs, lietotāja datu apgabals ir atkarīgs no aktuālā datu elementa.

Apskatāmajam piemēram (skat. 17. attēlu), pieņem, ka atkarīgā funkcija ir F4. Lietotāja datu apgabals un datu prezentācijas elementi būs atkarīgi no aktuālā datu elementa, kas šajā gadījumā ir `Org1_data` (ja vien iepriekš tas nav nomainīts uz citu datu elementu nosakošās funkcijas datu apgabalā). Rezultātā dati tiek iegūti no TIS datu bāzes un attēloti lietotājam datu apgabalā saskaņā ar funkcijas F4 loģiku un tā parametru `Org1_data`.

### 3.4.3. *Lietotāja datu integrācija lietotāja datu apgabalā*

Iepriekš aprakstītajā piemērā (skat. 17. attēlu), saskaņā ar piedāvāto DNATIS arhitektūru veidotā TIS lietotnē, viena no funkciju grupām, kas tiek piedāvāta, ir “Pašapkalpošanās funkciju grupa” un funkcijas varētu būt “parādīt personas datus”, „parādīt nodarbošanās datus” u.c.

Arhitektūras ietvaros var runāt par diviem datu veidiem, kas tiek izmantoti funkcijās, un kurus ir nepieciešams integrēt un parādīt vienotā, atbilstošā datu apgabalā lietotāja saskarnē (Niedritis, 2011).

Viens no datu veidiem ir lietotāja dati, kas raksturo personu (identifikācijas informācija, adrese, izglītība). Šos datus var ievadīt, piemēram, pilnvaroti lietotāji vienā vai citā organizācijā vai persona pati, bet šie dati nav saistīti ar organizācijas darbību atbalstošām funkcijām.

Otrs datu veids ir dati, ko rada pats lietotājs, vai kāds cits par šo lietotāju organizācijas biznesa procesu izpildes laikā. Piemēram, students pats reģistrējas uz studiju kursiem vai pasniedzējs ievada konkrētā studenta iegūtos rezultātus studiju kursā.

Integrācijas problēmas rodas pirmā datu veida gadījumā. Iemesls tam ir datu piederība, kad lietotājs ir kādas organizācijas klients (piemēram, students var būt piederīgs vairākām universitātēm, lielākā laika periodā vai arī vienlaicīgi).

Lai paskaidrotu šo situāciju, var apskatīt sekojošu piemēru.

Students studē Universitātē1. Studenta personas datus ievada sistēmā pilnvaroti lietotāji. Šajā gadījumā var pieņemt, ka studenta personas dati pieder Universitātei1 (iespēja ievadīt un labot savus datus atbilstošajai personai tiks apskatīta vēlāk). Administratīvais personāls ir saņēmis dokumentus, kas ir bijis pamats datu ievadam, diploma kopijas, pieteikuma formas, ko parakstījusi persona, u.c.

Pēc kāda laika students var iestāties Universitātē2. Ja dokumentos, kas tiek nodoti administratīvajam personālam Universitātē2 (piemēram, pieteikuma formā), ietvertie dati satur informāciju, kurus ievadot sistēmā, tie ir konfliktējoši un dublējoši ar sistēmā jau esošo informāciju, šis integrācijas konflikts ir jāatrisina.

Problēma ir sekojoša, ka viens datu elements satur atšķirīgas vērtības dažādās organizācijās, tie attiecas uz vienu personu, bet ir iegūti no dažādiem avotiem, t.i. dažādi dokumenti par vienu personu. TIS ieviešana universitātēs parāda arī, ka iespēja ļaut uzturēt personas informāciju personai pašai neapmierina administratīvo personālu, jo ne visi lietotāji attiecas vienādi apzinīgi pret šo datu korektumu un aktualitāti. Citi savukārt var izveidot datus, kas neatbilst nevienam no dokumentiem, kas nosūtīti augstskolām.

Tiek izmantoti sekojoši risinājumi, lai glabātu, labotu, uzturētu aktuālus, saskaņotus un integrētus lietotāju datus:

1) Tiek lietoti no biznesa loģikas atkarīgi statusi, lai noteiktu datu piederību organizācijai, gadījumā, ja dati ir kopīgi visām organizācijām.

Šis risinājums lietotāja datu glabāšanai un mainīšanai tiek lietots tikai dažiem lietotāja atribūtiem un tiek izvēlēts pirms TIS izmantošanas un tiek lietots lietotāja identifikācijai, neskatoties uz viņa aktuālo organizāciju. Atribūtu kopai jābūt minimālai, bet pietiekamai, lai identificētu piederību organizācijai. Jāņem vērā, ka šim nolūkam tiek rekomendēti atribūti, kas reti mainās.

2. Tabulā doti divi piemēri ar ierakstiem, kas satur četrus personu aprakstošus atribūtus. Atribūts "Īpašnieks" glabā informāciju par organizāciju, kas ierakstu izveidoja. Atribūts "Statuss" tiek iegūts no organizācijas biznesa likumiem, vienkāršākajā gadījumā statusa vērtība tiek pārbaudīta, salīdzinot ar kāda sistēmā esoša atribūta vai to komplekta vērtībām un tiek izmantots kontrolei.

Piemēram, ja persona studē Universitātē U1 (pirmais ieraksts 2. tabulā), tad statuss ir "aktīvs". Šajā gadījumā, ja persona vienlaicīgi sāk strādāt citā Universitātē U2, tad gadījumā, ja U2 administratīvais personāls grib izmainīt personas identifikācijas informāciju, tad jākontaktējas ar datu īpašnieku (piemēram, pa telefonu, izmantojot TIS ziņojumu nosūtīšanas funkciju, u.c.)

Ja persona beigusi studijas Universitātē U2 (otrais ieraksts 2. tabulā), atbilstošais ieraksta statuss ir "neaktīvs". Ja citas augstskolas administratīvais personāls, kad students tajā iestājies, grib mainīt personas atribūtus, to var izdarīt bez kontaktēšanās ar iepriekšējo datu īpašnieku.

### 2. Tabula. Datu piederības noteikšanas piemēri

Vārds	Uzvārds	Pers kods	Dz datums	Īpašnieks	Statuss	Kontakti
Linda	Liepiņa	12345	16.08.1990	U1	aktīvs	Tel.: 292929
Andris	Bērziņš	67899	01.01.1989	U2	neaktīvs	Tel.: 353535

2) Atsevišķi dažādos ierakstos glabāta noteiktu atribūtu kopa - katrai organizācijai savi ieraksti par vieniem un tiem pašiem atribūtiem.

Atribūti, kam atkarībā no organizācijas var būt dažādas vērtības būs organizācijas atkarīgi atribūti. Ieraksta atribūts „Īpašnieks” norāda, kas ierakstu ir izveidojis, var to labot un izmantot.

Piemēram, 3. tabulā ir attēloti trīs ieraksti, kas attiecas uz vienas personas izglītību (personu identificējošie dati ir aprakstīti 2. tabulā). 2. un 3. ieraksts ir par vienu un to pašu izglītības līmeni, kas iegūts organizācijā U2 (iet runa par pēc būtības vienu un to pašu iegūto izglītību) - maģistra, bet abas organizācijas (ieraksta īpašnieki U2 un U1) atšķirīgi saglabā *doc\_date* vērtību.

Ja apskata pirmo ierakstu par bakalaura līmeņa izglītību, tad var secināt, ka šī informācija ir bijusi vajadzīga tikai organizācijai U2, kas šo ierakstu ir izveidojusi, bet otrs ieraksts par bakalaura izglītību nav izveidots, jo šī informācija nav bijusi vajadzīga organizācijai U1.

### 3. Tabula. Datu piederības piemēri organizāciju atkarīgiem atribūtiem

N	Līmenis	Izgl. iestāde	Dok. Nr.	Doc_Date	Owner
1	Bachelor	U1	11111	20.06.2008	U2
2	Master	U2	22222	10.06.2010	U2
3	Master	U2	22222	2010	U1

Viena no organizācijām TIS arhitektūrā ir definēta kā noteicošā. Tas atkarīgs no ieejas punkta sistēmā (atkarīgs no ieejas adreses un lietotāja informācijas).

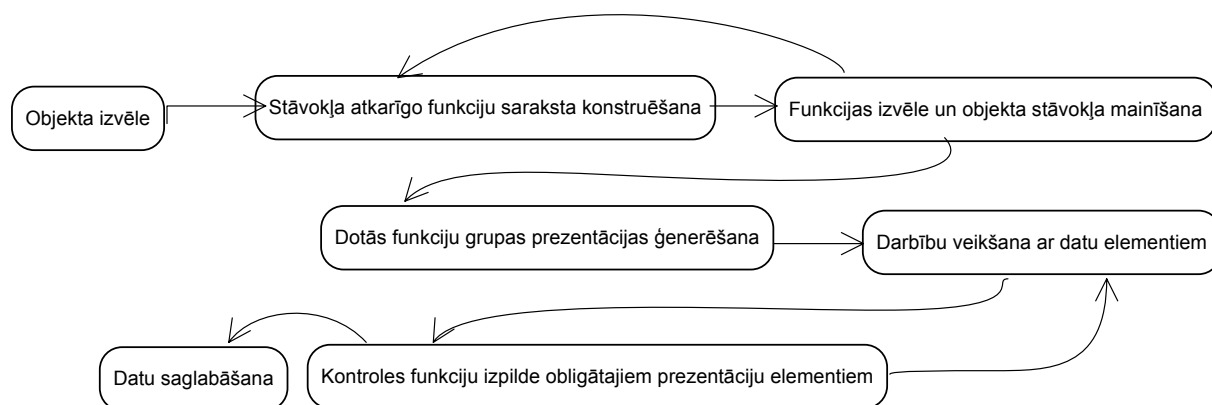
Noteicošā organizācija tiek izmantota, lai izmainītu aktuālo datu elementu noteicošās funkcijas datu apgabālā. Pēc noklusēšanas aktuālais datu elements tiek uzstādīts uz jaunāko. Dažreiz personas identificējošie dati norāda, ka organizācija vairs nav noteicošā (statuss – "neaktīvs", skat. 2. tabulu). Neraugoties uz to, ka noteicošās funkcijas datu elements ir

jaunākais, aktuālais datu elements šai gadījumā tiek uzstādīts kā jaunākais datu elements, kas pieder noteicošajai organizācijai.

#### 3.4.4. Lietotāja saskarnes adaptēšana lietotāja veiktajām darbībām

Tālāk aprakstītā darbu plūsma (skat. 18.attēlu) veic lietotāja saskarnes konstruēšanu, balstoties uz iepriekš aprakstīto prezentācijas modeli (skat. apakšnodaļu 3.3.2) un izmanto konfigurācijas un organizāciju profilu paplašinājumus no DNATIS arhitektūras. Lietotāja saskarne tiek ģenerēta ar lietotāja iesaistīšanos procesā. Lietotāja izdarītās izvēles un tai sekojošā saskarnes ģenerēšana ir atkarīga no kāda objekta stāvokļa (statusa), kas ir veikto darbību fokuss biznesa procesu veikšanas laikā. Tīmekļa bāzētas biznesa informācijas sistēmas lietotājs šādā veidā iegūst TIS saskarni, kas ir pielāgota specifiskai biznesa situācijai (Niedritis, Niedrite, 2012).

Tālāk tiks aplūkots katrs saskarnes konstruēšanas procesa solis, lai aprakstītu, kā tiek izmantoti profili un adaptācijas operācijas. Jāņem vērā, ka tiek veikta organizācijas līmeņa adaptācija. Detalizētā jeb lietotāja līmeņa adaptācija, kas balstās pamatā uz lietotāja profilu, šeit netiek pielietota.

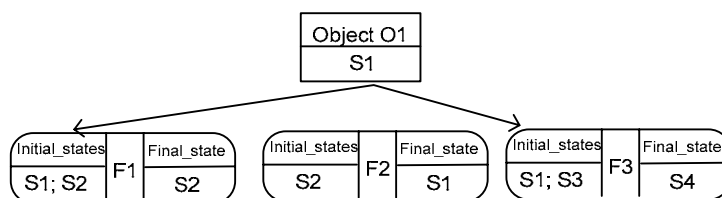


18. attēls. Lietotāja darbību ietekme uz TIS adaptāciju

**Objekta izvēle.** Lietotājs meklē un izvēlas objektu, kas būs tālāk izpildāmo biznesa procesu fokuss. Piemēram, lietotājs atrod TIS iepriekš reģistrēta transportlīdzekļa ierakstu, lai pēc tam veiktu ar šo ierakstu virkni operāciju.

**Stāvokļa atkarīgo funkciju saraksta konstruēšana.** Objekta sākuma stāvoklis nosaka pieejamo operāciju kopu, kas var tikt pielietoti objektam. Operāciju saraksts (vai funkcijas TIS kontekstā), ko atļauts pielietot objektam, ir balstīts uz DNATIS arhitektūras konfigurācijas profilu. Funkciju saraksts tiek attēlots saskarnē kā navigācijas tipa prezentācijas elementi.

Piemērā, kas redzams attēlā (skat. 19. attēlu), dota situācija ar izvēlētu objektu O1 ar sākuma stāvokli S1, un dots saraksts ar objektam atļautām funkcijām F1, F2 un F3. Tās ir funkcijas, kam ir definēti sākuma un beigu stāvokļi. Lietotājs var šajā gadījumā izvēlēties funkciju F1 vai F3, kuriem atribūta 'Initial state' vērtības starp visām pieļaujamām vērtībām satur arī S1, kas atbilst objekta O1 sākuma stāvoklim.



19. attēls. Piemērs objekta stāvokļiem un funkcijām

**Funkcijas izvēle un objekta stāvokļa mainīšana.** Atbilstoši biznesa vajadzībām, no pieļaujamo darbību saraksta, kas attēlotas saskarnē, lietotājs secīgi izvēlas visas nepieciešamās darbības, kuras nepieciešams izpildīt ar objektu. Pēc katras izvēles tiek fiksēts potenciālais objekta stāvoklis, ja vēlāk operācija tiks izpildīta, un tiek ģenerēts jauns pieļaujamo funkciju saraksts, kas atbilst jaunajam objekta stāvoklim.

Lietotāja izvēlētās funkcijas tiek attēlotas atsevišķā prezentācijas apgabalā kā navigācijas tipa prezentācijas elementi.

Iepriekšējā piemēra gadījumā (skat. 19. attēlu) objekta statuss tiek mainīts atbilstoši tam, kādu funkciju lietotājs izvēlēsies, un kāda būs šai funkcijai definētā atribūta ‘*Final\_state*’ vērtība. Šajā gadījumā visi iespējamie varianti darba plūsmām būs sekojoši: 1) F1→End 2) F1→F2→End; 3)F1→F2→F3→End; 4) F3→End. Lietotāja izvēles beigās apzīmē ar ‘End’.

**Dotās funkciju grupas prezentācijas ģenerēšana.** Lietotājs pabeidz funkciju izvēli un ģenerē prezentācijas apgabalu, kas satur datu prezentācijas elementus. Izvēlēto funkciju saraksts un organizācijas profils, kas atbilst tai organizācijai, kurā strādā lietotājs, nosaka atribūtu sarakstu konkrētajai TIS instancei. Prezentācijas modelis arī ietekmē veidu, kā izskatīsies ģenerētās saskarnes prezentācijas apgabals. Piemēram, viens un tas pats atribūts var tikt attēlots kā teksta prezentācijas elements ar tipu ‘Input’(pieļauj datu ievadi) vai ar tipu ‘Display’ (pieļauj tikai datu attēlošanu), kas atkarīgs no dažādu lietotāju funkciju izvēles dažādās TIS instancēs.

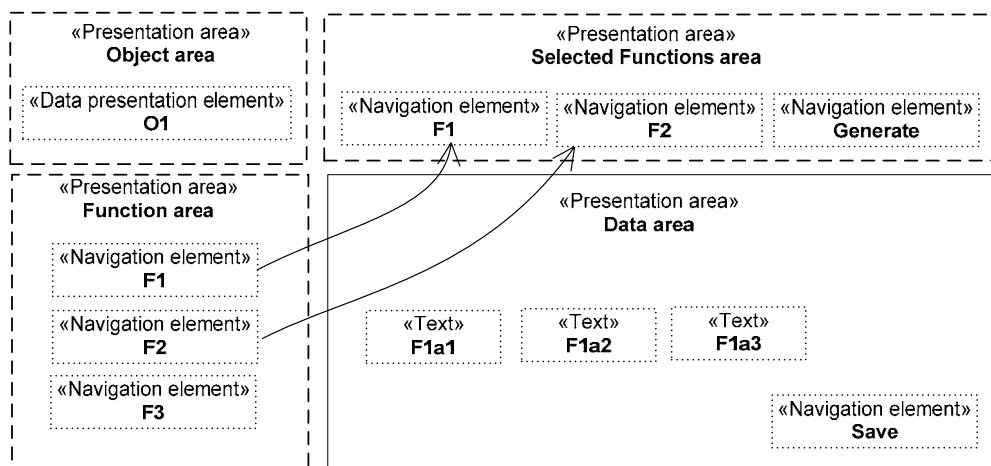
Iepriekšējam piemēram (skat. 19. attēlu) prezentācija TIS instancei ir dota 20. attēlā, kas parāda lietotāja izvēles un stāvokļa-atkarīgu atļautu funkciju piegādi lietotājam, ja viņš ir sekojis darba plūsmas scenārijam 2) F1→F2→End, kas tika aprakstīta iepriekš.

Objektu prezentācijas apgabals (*Object area*) satur datu prezentācijas elementu O1, Funkciju prezentācijas apgabals (*Function area*) satur visas atļautās funkcijas izvēlētajam objektam O1, atbilstoši šī objekta stāvoklim. Izvēlēto funkciju apgabals (*Selected Functions area*) satur lietotāja izvēlētās funkcijas. Datu prezentācijas apgabals tiek izveidots pēc tam, kad lietotājs ir pabeidzis funkciju izvēli atbilstoši kādam konkrētam darbību scenārijam biznesā (darba plūsmai). Datu prezentācijas apgabals (*Data area*) satur tos atribūtus, kas ir saistīti ar izvēlētajām funkcijām atbilstoši tam, kā ir definēts konfigurācijas un organizācijas profils, lai varētu izveidot organizācijai atbilstošu adaptētu TIS instanci. Visi 20. attēlā redzami elementi ir stereotipi, kas atbilst prezentācijas modelim (skat. 9. attēlu).

**Darbību veikšana ar datu elementiem.** Šis solis tieši neietekmē lietotāja saskarni, tas var tikai ietekmēt objekta atribūtu vērtības, tajā skaitā var tikt mainīts arī atribūts (vai atribūti), kas nosaka objekta stāvokli. Ja mainās objekta stāvoklis, tas var izraisīt izmaiņas saskarnē, ja kļūst pieejamas jaunas funkcijas, kuras tiek attēlotas saskarnē.

**Kontroles funkciju izpilde obligātajiem prezentāciju elementiem.** Kontroles funkciju izpilde veic pārbaudi, vai visi obligātie prezentācijas elementi ir ieguvuši vērtības, un pirms sākt jaunu darbību no lietotāja izvēlētām funkcijām, arī tiek vēl pārbaudīts objekta stāvoklis, lai nodrošinātu, ka lietotājs ir veicis visas darbības no darba plūsmas, ko lietotājs

pats ir definējis, un nav nevienu aizmirsis izpildīt, tā radot situāciju, kas ir pretrunā ar objekta stāvokli.



20. attēls. Lietotāja saskarne dotajam lietotāja darbību scenārijam

**Datu saglabāšana.** Ja kontroles funkciju izpilde ir veiksmīga, tad lietotājs var saglabāt datu elementu vērtības un var sākt jaunas darba plūsmas definēšanu, vispirms izvēloties jaunu objektu, kam pielietot izvēlētās funkcijas.

Soļi, kas tika aprakstīti iepriekš, kā arī DNATIS arhitektūra ir pielietota CSDD Transportlīdzekļu reģistrā. Objekts, kam tiek pielietotas lietotāja izvēlētās funkcijas, šajā TIS ir transporta līdzeklis. Transporta līdzekļa stāvoklis var būt, piemēram, 'Uzskaitē', 'Noņemts no uzskaites', 'Pagaidu uzskaitē', 'Izslēgts', utt. Funkcijas, ko atļauts izpildīt transporta līdzekļiem statusā 'Uzskaitē' ir, piemēram, 'Pārbūve', 'Numura maiņa', 'Izslēgt no reģistra' un daudzas citas.

### 3.5. DNATIS arhitektūras aprobācija

TIS adaptācijas arhitektūra izmantota 2 sistēmu realizācijā – Augstskolu informatīvajā sistēmā un Transportlīdzekļu reģistrācijas sistēmā. Realizācijas specifika katrā no tām, kā arī detalizētāks apraksts dots tālāk sekojošās apakšnodaļās.

#### 3.5.1. DNATIS arhitektūras pielietojums Augstskolu informatīvajā sistēmā

Augstskolu informatīvā sistēma realizēta, izmantojot DNATIS arhitektūru. 2012. gada novembrī šo sistēmu izmanto 13 nomnieki (augstskolas un viena Izglītības un Zinātnes ministrijas struktūrvienība). Dati par DNATIS arhitektūras pielietojumu Augstskolu informatīvajā sistēmā atrodami arī pielikumā 1.1.

Augstskolu Informatīvajā sistēmā realizētas trīs funkciju grupas 1) pamatfunkcijas, kas atbalsta galvenos biznesa procesus, 2) pašapkalpošanās funkcijas un 3) publiskās funkcijas.

Nozīmīgākie sistēmas lietotāji, kas uztur galveno sistēmas informāciju ir augstskolu darbinieki, kas strādā ar pamatfunkcijām. Katram lietotājam ir ierobežota pieeja sistēmai, saskaņā ar viņu darba pienākumiem un viņiem sistēmā piešķirtām pilnvarām. Tāpat šo lietotāju tiesības ir definētas lietotāju profilā DNATIS arhitektūras ietvaros.

Otra nozīmīga Augstskolu informatīvās sistēmas funkciju daļa ir pašapkalpošanās funkcijas, kas nodrošina pieeju personīgai informācijai studentiem un darbiniekiem. Tā ir sistēmas daļa, kas paredzēta pašapkalpošanās servisa vajadzībām (Niedritis et al., 2007). Šajā

gadījumā studenti un darbinieki izmanto LDAP (*Lightweight Directory Access Protocol*) kontus, lai pieslēgtos sistēmai. Līdzīgi kā pamatfunkciju gadījumā, šo lietotāju tiesības ir definētas, izmantojot lietotāju profilu DNATIS arhitektūras ietvaros.

Trešā funkciju grupa ir publiskās funkcijas, kas, salīdzinot ar abām pirmajām grupām, ir neliela daļa. Funkcijas ir pieejamas visiem lietotājiem.

Funkciju skaitu pašapkalpošanās funkciju grupā pa gadiem un pa augstskolām, kuras dažādos gados ir ieviesušas šo funkciju grupu, var redzēt pielikumā dotā tabulā (skat. 7. tabulu). Kaut arī Augstskolu informatīvā sistēma tika uzsākta pirms 2000. gada, promocijas darbā aprakstītā DNATIS arhitektūra tika izstrādāta laika gaitā, un par nozīmīgu ieviešanu pašapkalpošanās funkciju grupai var runāt no 2006. gada. Funkciju skaita samazinājums pārsvarā gadījumu nozīmē jaunas versijas realizāciju TIS, kur apvienotas vairākas funkcijas. Var pieņemt, ka visas realizētās funkcijas tiek izmantotas LU, un uz doto brīdi ir 101 funkcija pašapkalpošanās funkciju grupā.

Pamatfunkciju grupas funkciju un lietotāju skaits pa gadiem un augstskolām dots pielikumā (skat. 8. tabulu).

Augstskolu IS, kas veidota atbilstoši DNATIS adaptācijas arhitektūrai, aprobācijas projektā realizēto LU un LiepU adaptēto instanču saskarņu piemēri redzami pielikumā 46. attēlā (LUIS sistēmas lietotāja saskarne) un 47. attēlā. (LiepU sistēmas lietotāja saskarne).

#### Realizācijas specifika Augstskolu informatīvajā sistēmā

Augstskolu informatīvajā sistēmā piekļuve tiek realizēta ar lietotāju tiesību mehānismu, definējot katru tiesību kā funkcijas un datu ierobežojumu virknes apvienojumu. Šim nolūkam tiek veikta denormalizācija, tiek izmantoti dati no lietotāja un konfigurācijas profiliem (klases *User*, *Row\_access*, *Table* un *Function*, klases *Row\_access* atribūts 'Filter'). Tiesību definīcijas LUIS realizācijā tiek glabāta tabulā *User* atribūtā *Filter* kā tiesību virknes, kas savukārt katra satur denormalizēto informāciju par funkciju un datu ierobežojumiem. Denormalizācija izvēlēta sistēmas ātrdarbības nodrošināšanai, lai nodrošinātu ātrdarbību adaptācijas darbību izpildei, kad tiek ģenerēta TIS adaptēta instance ar integrētu, adaptētu lietotāja saskarni. Filtra ierobežojums tiek realizēts kā papildus nosacījums, kas vaicājumam vai funkcijai atļauj atrast tikai to informāciju, kas atbilst filtram.

Tiesību pieraksta sintakse ir sekojoša: R\*fun f1% f2% ...#, kur fun ir funkcijas nosaukums, f1, f2 utt.– datu ierobežojumi (var būt vairāki), kas glabā vai nu pilnu vērtību kādam datu atribūtam (piemēram, studiju programmas kodu) vai sākuma simbolus, tad tiek izmantota % zīme, lai varētu ar vienu ierobežojumu aprakstīt piekļuvi hierarhiski pakļautiem datiem. Piemēram, 2% nozīmē piekļuvi visām mācību prorektoram pakļautām studiju programmām. Tiesības definīcijas beigas norāda #.

Atkarībā no konkrētās funkcijas, filtros izmantotie datu atribūti var būt dažādi.

Filtrā augstskolu sistēmā tiek lietoti: 1.) studiju programmu kodi: 22301 210% - ierobežojums pieejai norādīto programmu studentiem 2.) struktūrvienību kodi: LU210% - ierobežojums pieejai norādīto struktūrvienību darbiniekiem 3.) augstskolas kodi: L, R u.c. – ierobežojums uz augstskolas specifiskām izvēlnēm 4.) nozares kodi studiju kursiem: 2DAT% 2MED% -ierobežojums šādā pierakstā nosaka piekļuvi tikai datorzinātņu un medicīnas nozares kursiem 5.) finansējuma kodi: S%-100 – ierobežojums uz norādītajiem finansējumiem u.c. Ierobežojumi tiek realizēti kā papildus nosacījums dinamiskajā SQL, kas tiek pievienoti SQL izpildes brīdī, bet sarežģītākajos gadījumos kā funkcijas izsaukums. Ierobežojums automātiski tiek pievienots izpildāmā SQL operatora WHERE daļā, piemēram, (and programma\_pkods like '210%') vai (and chk(programma\_pkods,'22301 210%')>0).

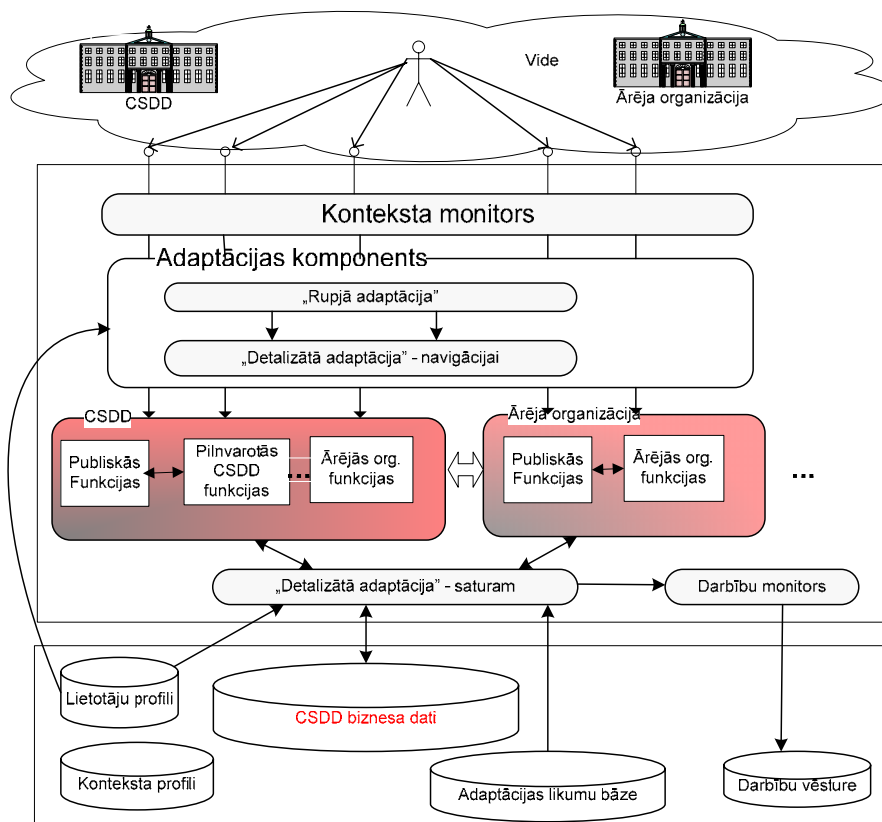
#### **3.5.2. DNATIS arhitektūras pielietojums Transportlīdzekļu reģistrā**

Ceļu satiksmes drošības direkcijas (CSDD) transportlīdzekļu reģistrācijas IS ir vēl viena TIS, kur aprobēta promocijas darbā piedāvātā DNATIS adaptācijas arhitektūra. Dati par DNATIS arhitektūras pielietojumu Ceļu satiksmes drošības direkcijas sistēmā atrodami arī pielikumā 1.2.

Sistēmas izveides sākuma posmā tā tika veidota kā viena sistēma ar vienu datubāzi, bet pieeja datiem tika nodalīta pa 26 organizācijas nodaļām, kur katrai nodaļai bija pieeja tikai savas nodaļas datiem. Nodalījums sākotnēji (līdz 1998. gadam) bija tādēļ, ka CSDD rajona nodaļām nebija iespējama tiešsaistes pieeja. Sākotnēji rajonu nodaļas funkcionēja neatkarīgi ar datu apmaiņu starp rajoniem un centrālo sistēmu. Tādēļ tika izveidota centrālā sistēma ar loģisku dalījumu pa rajoniem, kas līdzīga augstskolu fakultāšu gadījumam, ar ierobežojumiem vienas organizācijas iekšienē. Vēlāk šī sadalīšana līdz ar pāreju uz centrālās sistēmas pilnīgu ieviešanu vairs nebija tik būtiska, bet atsevišķās daļās šis dalījums vēl ir saglabājies arī šobrīd, kā, piemēram, veikto reģistrācijas darbību dokumentācijas plūsma rajona ietvaros u.c.

Funkcionalitātei paplašinoties, daudzas funkcijas tiek piedāvātas arī ārējām organizācijām, atsevišķu funkciju veikšanai tās izmanto tās pašas iespējas, ko organizācijas iekšējie lietotāji, bet visai bieži ar atšķirīgiem ierobežojumiem uz datiem, pieejamiem ievada vai izvada laukiem darbību izpildē un arī niansēm funkcionalitātē. Tādējādi pieaugot atšķirībām vienas un tās pašas funkcijas izpildē un dažādi pieejai datiem radās nepieciešamība nodrošināt ērtāku un efektīvāku funkciju realizēšanu un uzturēšanu. Līdzīgi kā daudzu augstskolu gadījumā tiek izmantota DNATIS arhitektūra. Lietotāja specifika tiek aprakstīta ar tiesību piešķiršanas mehānismu lietotāja profilā, kur lietotāja tiesību apraksts nosaka pieejamo funkcionalitāti, pieejamo datu kopumu un funkcijas loģikas nianses.

Ilustrācijai dota shēma (skat. 21. attēls), kas atspoguļo TIS arhitektūras komponentus, nodalot divu tipu iesaistītās organizācijas – CSDD un ārējās organizācijas. Ir definētas trīs funkciju grupas: pilnvaroto CSDD funkciju grupa, ārējo organizāciju funkciju grupa, kā arī publisko funkciju grupa.



21. attēls. DNATIS arhitektūras pielietojuma piemērs Transportlīdzekļu reģistrā (saskaņā ar 4. attēlu)



Transportlīdzekļu reģistrācijas sistēmas funkcijas pieejamas vairāk kā 9000 lietotājiem no 36 dažādām ārējām organizācijām, kā arī no 24 rajonu nodaļām. Informāciju par nomnieku skaitu, lietotāju skaitu un izmantoto funkciju skaitu katram nomniekam var redzēt pielikumā 9. tabulā.

Līdzīgi kā Augstskolu informatīvajā sistēmā, profilu realizācijā Transporta līdzekļu reģistrā izmantota denormalizācija. Piekļuves ierobežošanai datiem, lietotāja aprakstā tiek izmantoti: 1) organizācijas kods, piemēram, 40003330004 – ierobežojums pieejai norādītās organizācijas transportlīdzekļiem 2) sertifikāta sērija: piemēram, RS – var izdot norādītās sērijas sertifikātus, numuriem veidojoties automātiski 3) reģistrācijas rajons – pēc darbības veikšanas dokumenti pieejami norādītajā rajonā 4) darbības izpildes IP adrese – veicot darbību no IP adrešu segmenta dokumenti pieejami norādītajā rajonā u.c. Ierobežojumi realizēti kā papildus nosacījumi dinamiskajā SQL, bet sarežģītākajos gadījumos kā speciālas funkcijas izsaukumi.

Transportlīdzekļu reģistrācijā iespējamas daudzas funkcijas: pirmā reģistrācija, īpašnieka maiņa, informācijas maiņa par īpašnieku, numura zīmes maiņa, transportlīdzekļa pārbūve, noņemšana no uzskaites un citas. Veicot transportlīdzekļa reģistrāciju, var tikt veiktas vairākās secīgas darbības. To atļautās kombinācijas arī ir definētas profilu līmenī, katrai funkcijai norādot, kādas citas funkcijas tai var sekot, atkarībā no objekta stāvokļa, ar ko strādā funkcijas. Tā kā šīs sakarības definētas profilos, un tās netiek realizētas kā fiksēta funkcionalitāte, tādēļ izmaiņas veicamas, nemainot programmatūru.

Lietotājam, atkarībā no izvēlētās darbības, piemēram, ir iespējams ievadīt tikai definētos veicamo funkciju pamatojuma dokumentus, kā arī lietotājs, piemēram, nevar veikt transportlīdzekļa reģistrāciju, ja trūkst obligāti nepieciešamie dokumenti.

Lai šādā gadījumā, kad katra darbība nav realizēta kā atsevišķa sistēmas funkcija, aizsargātos no lietotāju kļūdām, tiek kontrolēta arī ievadlauku pieejamība lietotāju saskarnē. Atkarībā no izvēlētajām darbībām un to secības, lietotājam ievadam ir pieejami katrai darbībai atļautie lauki. Piemēram, veicot pārbūvi var mainīt šasijas numuru, veicot īpašnieka maiņu – izvēlēties jaunu īpašnieku, veicot informācijas maiņu par īpašnieku – mainīt īpašnieka adresi. Katras darbības veikšanai nepieciešami atbilstoši pamatojuma dokumenti, tie var būt obligāti vai neobligāti.

Pielikumā doti piemēri, kas raksturo, kādas ir adaptētās lietotāju saskarnes, divām dažādām organizācijām - Transportlīdzekļu reģistrācijas TIS sistēmas nomniecēm (skat. 48. attēlu un 49. attēlu).

Sistēmas izmantošanā tiek kontrolēta un analizēta tās lietošana ar mērķi atrast šaurās vietas, kam pievērt pastiprinātu uzmanību, lai veiktu optimizāciju. Datubāzu sistēmās ir iebūvēti līdzekļi, piemēram, SQL optimizācijai, bet ir gadījumi, kad ar tiem nepietiek.

Visas lietotāju darbības tiek saglabātas apkopotā veidā ar dienas precizitāti, cik reizes kurš lietotājs no kuras IP adreses konkrēto funkciju izmanto un cik resursi (procesora laiks) uz to tiek patērēti. Katra darbība parasti aizņem nelielu laika sprīdi, bet daudzlietotāju sistēmā, atrodot lielākos resursu izmantotājus, grupējot gan pēc lietotāja, gan veicamās funkcijas var atrast sistēmas daļas, kam jāpievērš pastiprināta uzmanība. Optimizējot biežāk izmantotās un vairāk procesora laiku izmantojušās sistēmas daļas, tiek panākta sistēmas efektivitāte. Šī daļa tiek realizēta kā iebūvēta funkcionalitāte, tādēļ sistēmas darbības laikā var meklēt šaurās vietas ar mērķi tās uzlabot.

Iepriekšminētais mehānisms ļauj kontrolēt arī lietotāju aktivitāti un tiek izmantots ārējo organizāciju norēķiniem par sistēmas izmantošanu gadījumos, ja ārējiem lietotājiem piedāvātās darbības ir piedāvātas par maksu.

Realizēta datu apmaiņa arī ar citām ārējām sistēmām, no kurām datu apjoma un apmaiņas intensitātes ziņā lielākā ir apdrošināšanas datu saņemšana un transportlīdzekļu datu nodošana apdrošināšanas kompānijām. Arī datu apmaiņas mehānismā ir iebūvēti iepriekšminētie ierobežojumi uz pieejamo informāciju, atšķirīgā funkcionalitāte atkarībā no organizācijas un nianse darbības izpildes loģikā saskaņā ar promocijas darbā aprakstītajiem DNATIS adaptācijas arhitektūras principiem.

### 3.6. Nodaļas secinājumi

Promocijas darbā piedāvātā DNATIS arhitektūra izmanto divus adaptācijas līmeņus – rupjo jeb organizācijas līmeņa un detalizēto jeb lietotāja līmeņa adaptāciju. Organizācijas līmeņa adaptācija izveido TIS adaptētu instanci, ņemot vērā profilā aprakstīto, kādas funkcijas no TIS un kādā veidā tiek izmantotas biznesā, bet lietotāja līmeņa adaptācija papildus ievieš nosacījumus, kā TIS pielāgojama individuālam darbam ar sistēmu, balstoties uz lietotāja profilu.

DNATIS arhitektūra arī atbalsta situāciju, kad lietotāji var strādāt ar vienas TIS sistēmas vairākām instancēm. Adaptētās TIS instances organizācijas līmenī tiek integrētas vienā konkrētam lietotājam adaptētā TIS instancē, ko lietotājam piegādā, izmantojot integrēto lietotāja saskarni.

Daudzo nomnieku un vienas fiziskas instances ideja, kas tika izmantota šajā arhitektūrā, izraisīja datu piederības problēmas. Tika piedāvāti risinājumi, kā šos datu konfliktus risināt, promocijas darbā ilustrējot šos risinājumus ar piemēriem no TIS implementācijām, kas balstītas uz aprakstīto DNATIS adaptācijas arhitektūru. Piedāvātie risinājumi datu glabāšanai, labošanai un integrētai un saskaņotai datu piegādei lietotājam pielietoti TIS implementācijās.

Aprakstītā adaptācijas arhitektūra ir apbēta divās TIS implementācijās. Viena TIS – Augstskolu informatīvā sistēma – ir realizēta projektā, kurā uz doto brīdi 13 augstskolas izmanto vienas TIS 13 adaptētas instances. Otra TIS – Ceļu Satiksmes Drošības Direkcijas Transportlīdzekļu reģistrācijas IS – ir izmantota CSDD un ar to saistītās citās ārējās organizācijās.

Prakse rāda, ka šo arhitektūru var izmantot dažādās biznesa jomās. Šādas arhitektūras pielietošana var tikt apsvērta sekojošās situācijās: 1) pastāv daļa lietotāju, kas izmanto vairāk nekā vienu TIS instanci vienā laika periodā vai arī secīgi, tā piekļūstot visiem saviem datiem; 2) daudziem lietotājiem jāveic viena un tā pati funkcija, bet katram atšķirīgā veidā.

Arhitektūru var reducēt arī uz apakšgadījumu, kad lietotāji ir stingri katrs savas organizācijas TIS instances lietotāji. Šajā gadījumā tiek izmantotas TIS adaptācijas arhitektūras divu līmeņu adaptācijas pieeja un arhitektūras komponenti, lai adaptāciju nodrošinātu, bet netiks izmantota integrētās lietotāja saskarnes veidošana, jo tā nebūs nepieciešama.

Kaut arī piedāvātā DNATIS arhitektūra ļauj pielāgot tīmekļa informācijas sistēmu organizācijas biznesa vajadzībām, pielāgošanas iespēju robežas nosaka profilu modeļi, kā arī prezentācijas modelis un adaptācijas darbības, kas izveido adaptētu TIS instanci un atbilstošu prezentāciju.

Dinamiskas adaptācijas iespējas piedāvātajā arhitektūrā ir saistītas ar objektu stāvokļa jēdziena izmantošanu, kas atkarībā no lietotāja veiktajām darbībām ļauj veidot darbībām adaptētu lietotāja saskarni.

Dotā arhitektūra sekmīgi pierādījusi savu dzīvotspēju vairāku gadu garumā divos atšķirīgos projektos. Tomēr jāsaprot, ka ir sastopami pieprasījumi kādu no jau esošām funkcijām realizēt specifiski kādai no iesaistītajām organizācijām. Tas atsevišķos gadījumos ir realizēts, kad ar esošām DNATIS arhitektūras iespējām nevar panākt vajadzīgo funkcijas

darbību (t.i. ar adaptāciju nepietiek), tomēr vairāku gadu pieredze liecina, ka šādu atsevišķu funkciju realizācija nav vēlama, jo pēc tam visa TIS ir sarežģīti uzturama, it īpaši, ja jāveic TIS pamatfunkciju pārbūve, piemēram, likumdošanas izmaiņu rezultātā. Tiesa, šādu individuālu pieprasījumu skaitu ierobežo par TIS izmantošanu noslēgtie līgumi, kur papildfunktionalitātes programmēšana paredz papildus izmaksas.

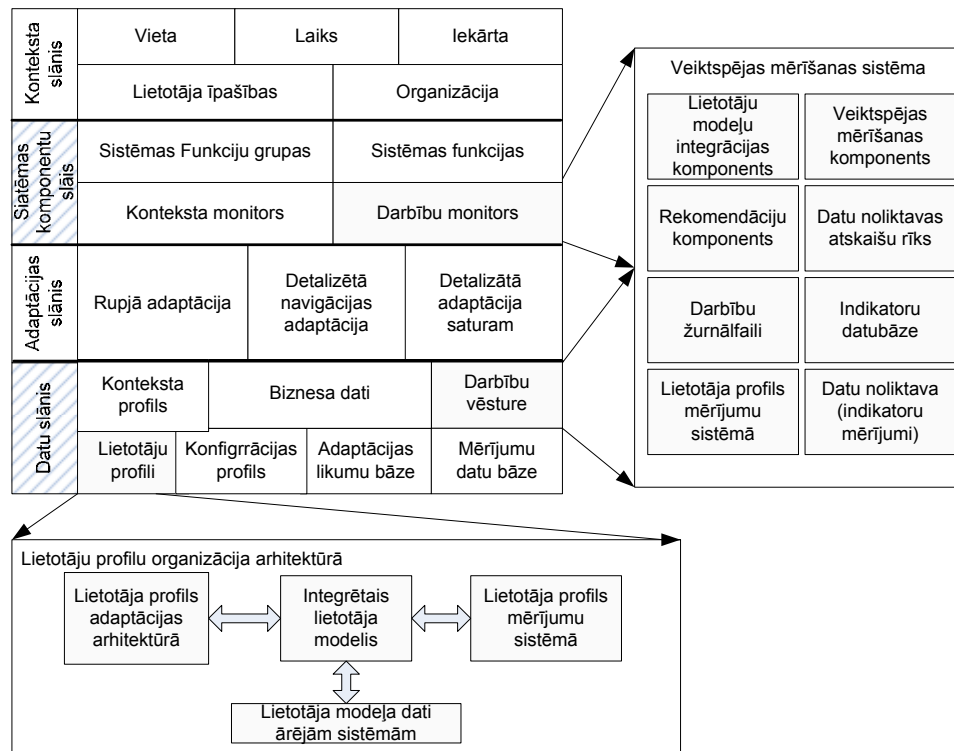
## 4. Lietotāja modelis DNATIS arhitektūrā

### 4.1. Nodaļas nolūks

Nodaļas nolūks ir aprakstīt metodi, kas konstruē lietotāja modeli adaptācijas arhitektūras vajadzībām, balstoties uz lietotāju aprakstošo profilu jēdzienu. Tiek izveidots integrēts lietotāja modelis un aprakstīta metode tā izveidei un datu uzkrāšanai. Integrētais lietotāja modelis ir paredzēts izmantot DNATIS arhitektūras adaptācijas komponentā, kā arī DNATIS arhitektūras ietvaros izveidotajā mērījumu sistēmā, kas balstīta uz datu noliktavu tehnoloģijām. Piedāvātā metode un rezultāti aprakstīti divos rakstos: (Niedritis et al., 2011b) un (Solodovnikova et al., 2012).

### 4.2. Lietotāja profili un lietotāja modeļa nepieciešamība

Promocijas darbā iepriekš tika aprakstīta DNATIS arhitektūra (skat. 4. attēlu). Arhitektūras shēma, kas 1) detalizētāk parāda komponentus lietotāju darbību monitoram un darbību vēstures datu bāzei un 2) precizē lietotāju profilu un lietotāja modeļa vietu un savstarpējo saistību DNATIS arhitektūrā, dota 22. attēlā. Šajā nodaļā tiks aprakstīts tieši 2) punkts no iepriekš minētā – lietotāju profili un nepieciešamība ieviest integrētu lietotāju modeli, kā arī tiks apskatīta metode šī integrētā lietotāja modeļa izveidei. 1) punkts – arhitektūras apraksts, saistībā ar lietotāju darbību monitoru un darbību vēstures datu bāzi, dots 5. nodaļā par veikspējas mērīšanas ietvaru.



22. attēls. Lietotāju profilu un lietotāja modeļa vieta DNATIS arhitektūrā

Mērīšanas jēdzieni, veikspējas mērīšanas ietvars un sistēma tiks detalizēti skatīti 5. nodaļā. DNATIS arhitektūrā kā komponenti ietilpst arī lietotāju darbību monitors un darbību vēstures datu bāze, kas kopā veido veikspējas mērīšanas sistēmu, kuras komponenti redzami 22. attēlā. Tā balstīta uz datu noliktavas datu bāzi mērījumu vērtību glabāšanai, tā izmanto gatavu datu noliktavu atskaišu rīku, un lietotāju aprakstīšanai tai ir savs lietotāja profils.

Uz datu noliktavu tehnoloģijām balstīta procesu mērīšanas sistēma tiek veidota gadījumā, ja organizācijā jau eksistē iepriekš datu noliktava, kas izveidota organizācijas rezultātu, piemēram, finansiālu rādītāju analīzei, bet vajadzība mērīt procesu aspektus tiek apzināta vēlāk. Datu noliktavās tiek integrēti dati no dažādām datu avotu sistēmām organizācijā. Datu noliktavām, lai aprakstītu lietotāju, tajā skaitā lietotāja identitāti, kā arī lai definētu lietotāju tiesības piekļūt dažādiem datiem, arī tiek izmantots lietotāja profils.

Lietotāja profils tika aprakstīts arī nodaļā par DNATIS arhitektūru (skat. 3. nodaļu), un šajā gadījumā TIS lietotāja profils tika izmantots TIS adaptācijai.

Lietotāji parasti mijiedarbojas ar dažādām sistēmām: gan Internetā pieejamām, gan iekšējām organizācijas sistēmām. Gan ārējas, gan organizācijas iekšējas sistēmas glabā informāciju par viena un tā paša lietotāja dažādām īpašībām, informācijas uzkrāšanas mērķis var būt dažāds – identificēšana, kontrole, personalizācija, rekomendācijas un citi mērķi.

Katras sistēmas lietotāja profils apraksta lietotāju no dažādiem aspektiem, piemēram, mijiedarbības, ģeogrāfiskās vietas, demogrāfiskiem u.c. aspektiem, kas ir būtiski dotās sistēmas kontekstā. Sistēmu glabātie informācijas vienumi par lietotāju var tikt izmantoti, lai iegūtu visaptverošu skatījumu par lietotāju. Tātad būtībā dažādi lietotāju profili var tikt izmantoti, lai vāktu informāciju par lietotāju, t.i. var tikt uzskatīti par informācijas avotu, un var tikt izmantoti, lai varētu izveidot visaptverošu, vispārīgu, integrētu lietotāja modeli.

2. nodaļā tika definēti gan lietotāja profila, gan modeļa jēdzieni, un, atkārtojot, jāuzsver, ka lietotāja profili ir vienkāršoti lietotāja modeļi, bet lietotāja modelis ir lietotāja abstrakta reprezentācija, jeb sistēmas priekšstats par lietotāja būtiskām īpašībām.

Eksistē projekti, kuros pētīta publiski pieejamu dažādās sistēmās uzkrātu lietotāju datu integrēšana, piemēram, sociālajos tīklos (Abel et al., 2010). Dati tiek vākti, lai kļūstot par kādas citas TIS lietotāju, varētu tikt izmantots jau eksistējošs lietotāja profils.

Atšķirībā no eksistējošiem risinājumiem, kas orientēti uz izmantošanu sociālajos tīklos, šajā nodaļā tiek aplūkotas uzņēmuma iekšējas tīmekļa bāzētas informācijas sistēmas, un datu noliktava kā viena no sistēmām.

Promocijas darbā tiek piedāvāts integrēts lietotāja modelis un tā izveides metode, kas izmantojams DNATIS arhitektūras ietvaros gan tajā izveidotajai datu noliktavai, gan TIS adaptācijai, kā arī tiek piedāvāta metode, kā datu noliktavas modelis var tikt attīstīts, izmantojot lietotāju datu lietošanas paradumu analīzi kā avotu rekomendācijām datu noliktavas modeļa evolūcijai.

Lietotāja profila (un arī modeļa) dzīves cikls sastāv no trim fāzēm: (i) informācijas vākšanas par lietotāju, (ii) integrēta skatījuma par lietotāju veidošana (integrēta profila izveidošana) un (iii) lietošanas, kas piedāvā personalizētu servisu lietotājam, balstoties uz lietotāja profilu (Gauch et al., 2007). Promocijas darbā galvenā uzmanība pievērsta diviem pirmajiem no minētajiem soļiem, bet lietošanas iespējas ir saistītas ar lietotāja modeļa izveides mērķiem.

Galvenie mērķi, lai izveidotu integrētu lietotāja modeli DNATIS arhitektūrā:

- pēc iespējas pilnīgāka informācija par lietotāju, lai veiktu TIS adaptāciju,
- datu noliktavas modeļa uzlabošana un papildināšana, balstoties uz lietotāju darbību izpēti datu avotos (izstrādātāja perspektīva),

Informācijas par lietotāju, tai skaitā, lietotāja aktivitātes datu savākšana datu avotu sistēmās ir noderīga, lai analizētu katra lietotāja izpildītās funkcijas un iegūtos vaicājumu rezultātus. Lietotāji var tikt grupēti dažādos veidos – piemēram, pēc viņus interesējošiem datu elementiem, pēc veikto vaicājumu rezultātu detalizācijas pakāpes, pēc iemaņu līmeņa, pēc pieejas tiesībām, pēc lietotāja lomas, u.c. Saprotot kopējās tendences dažādām lietotāju grupām, ir iespējams uzlabot datu noliktavas modeli, pievienojot jaunus atribūtus – gan

faktus, gan dimensijas, kas tiku veidoti no datu avotu komponentiem, ko visaktīvāk izmanto atšķirīgās lietotāju grupas (Solodovnikova et al., 2012). Promocijas darbā tiek piedāvāta metode datu noliktavas modeļa attīstīšanai (skat. 4.5. apakšnodaļu).

- datu noliktavas atskaišu rīku personalizācija (lietotāja perspektīva).

Dažreiz skatot datu noliktavas atskaišu rezultātus, lietotājam nav priekšstata, ko var atrast šajās atskaitēs. Lietotājs var nebūt informēts par derīgu atskaiti, kas nesen izveidota un kas nav tikusi iepriekš izmantota. Ir būtiski iesaistīt personalizāciju datu noliktavas atskaišu rīkā, lai lietotājam sniegtu rekomendācijas par citām atskaitēm, kas var būt derīgas un atbilst lietotāja iestatījumiem. Rezultātā lietotājs var orientēties datu noliktavas atskaišu klāstā, ietaupot laiku un pūles. Šādā veidā profila izmantošanu plānots izmantot datu noliktavas atskaišu rīkā, papildinot jau realizētās atskaišu rīka rekomendāciju sniegšanas metodes (Kozmina, Solodovnikova, 2011a). Promocijas darbā personalizācijas jautājumi netiek skatīti.

### 4.3. Lietotāju aprakstošā profila jēdziens

Promocijas darbā aprakstītais integrētais lietotāja modelis un tā izveides metode izstrādāta projekta ietvaros par datu noliktavu pētījumiem, un balstās uz atsevišķiem citu projekta dalībnieku rezultātiem, piemēram, iepriekšējos darbos tika jau ieviests jēdziens - lietotāju aprakstošais profils datu noliktavām (Kozmina, Niedrite, 2010). Šie lietotāju aprakstošie profili ir paredzēti izmantošanai datu noliktavu personalizācijas nolūkiem. Promocijas darbā lietotāju aprakstošais profils tiek attīstīts, izveidojot integrētu lietotāja modeli.

Lietotāju aprakstošs profils ir profilu kopa (Kozmina, Niedrite, 2010), kas raksturo lietotāja aktivitātes datu noliktavu sistēmā no dažādiem skatu punktiem. Definīcija izmanto Zahmana ietvara koncepciju (Zachman, 2003), kur tiek uzdoti sekojoši jautājumi: kas, ko, kā, kad, kur, kāpēc. Promocijas darbā tika definēts konteksta jēdziens, un šie jautājumi tiek uzdoti, lai aprakstītu lietotāja mijiedarbības kontekstu ar TIS (skat. punktu 3.3.3.1) Atbilstoši šiem uzdotajiem jautājumiem, lietotāju aprakstošais profils sastāv no sešiem atsevišķiem dažādiem profiliem:

- Iestatījumu (*Preferential*) profils raksturo lietotāja vēlmes – norādot, *ko* interesē lietotājam noskaidrot,
- Lietotāja (*User*) profils raksturo lietotāju identificējošu informāciju, personas datus, apraksta, *kas* ir lietotājs,
- Vietas (*Spatial*) profils raksturo lietotāja fizisko atrašanās vietu – *kur* atrodas lietotājs,
- Laika (*Temporal*) profils raksturo laiku, *kad* lietotājs izmanto sistēmu,
- Mijiedarbības (*Interaction*) profils raksturo mijiedarbības veidu, *kā* tā notiek, izmantojot konkrētu tehnisku ierīci un programmas noteiktu versiju,
- Rekomendāciju (*Recommendation*) profils raksturo mijiedarbības mērķi, kāpēc lietotājs ir ieinteresēts konkrētajā sistēmā. Šis profils tiek ģenerēts, pamatojoties uz citiem profiliem, lai izveidotu ieteikumus lietotājam.

Piedāvātie profili apraksta ne tikai pašu lietotāju, bet arī visu mijiedarbības vidi starp lietotāju un datu noliktavu sistēmu. Lietotāja, vietas, laika, mijiedarbības un rekomendāciju profils kopā veido lietotāja aprakstu.

Iepriekš minēto profilu atribūtu saraksti (Kozmina, Niedrite, 2010) tika veidoti literatūras studiju, kā arī projekta dalībnieku un ekspertu praktiskās pieredzes dokumentēšanas rezultātā. Tika apzināti vairāku tipu potenciālie atribūtu avoti (tālāk doti daži piemēri): (i) grāmatas par dažādu sistēmu veidiem, piemēram, datu noliktavu sistēmām (Imhoff et al., 2003), (ii) atbilstošās vides standarti un apraksti, piemēram, datu noliktavu jomas standarts

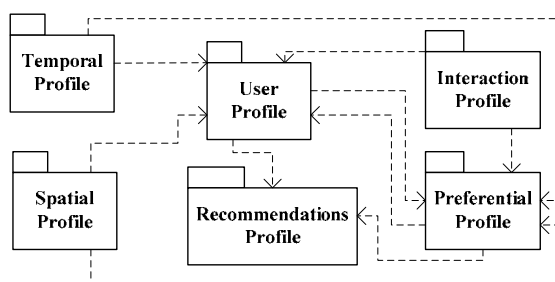
CWM (Poole et al., 2003), tīmekļa servisu izmantošanas dokumenti (IP), (iii) zinātniski raksti un tehniska dokumentācija, (iv) ekspertu praktiskā pieredze darbā ar izstrādes rīkiem un rīku metamodeļi. Katra profila atribūtu kopa tika loģiski sadalīta klasēs, izveidojot klašu diagrammu katram profilam. Sīkāks lietotāju aprakstošā profila klašu apraksts atrodams rakstā (Kozmina, Niedrite, 2010).

Tomēr, lai labāk saprastu piedāvāto metodi un lietotāja modeli, tālāk tiek dots īss profilu apraksts – nosaucot katra profila klases. Katra klase nepieciešamības gadījumā var tikt papildināta ar citiem atribūtiem.

- Lietotāja profila klases: *Role* atribūti apraksta lietotāja sistēmas lomu. *Personal* satur personas informāciju. *Work* un *Education* apraksta lietotāja darbu un izglītību. *Session* apraksta lietotāja sesijas raksturlielumus. *Activity* atribūti raksturo lietotāja aktivitāti tīmekļa lapās noteiktā laika periodā. *Rights* satur atribūtus, kas apraksta lietotāja tiesības uz sistēmas objektiem.
- Laika profila klases: *Standard Calendar* un *Fiscal Calendar* ir divi kalendāru veidi. *Time* satur atribūtus, kas raksturo datumu kā skaitli, *Time Status* satur jā/nē veida atribūtus, *Domain Specific* satur domēnam specifiskus atribūtus (piemēram, semestris izglītības jomā, uc), *Special Period* satur atribūtus, kas raksturo kādu plānotu, globālu vai lokālu notikuma intervālu.
- Vietas profils satur divas klases: *Physical Location* (atribūti, kas raksturo personas fizisko adresi) un *Location By IP* (atribūti, kas iegūstami no lietotāja IP adreses ar tīmekļa servisu palīdzību).
- Mijiedarbības profila klases ir *Web Access*, kura atribūti raksturo operāciju sistēmas, pārlūkprogrammas un interneta pieslēguma īpašības, *Functional*, kura atribūti apraksta pārlūkprogrammas funkcionālās īpašības un atbalstītās lietojumprogrammas un *Visual Layout*, kura atribūti apraksta vizuālā izkārtojuma īpašības pārlūkprogrammā.

Iestatījumu (*Preferential*) un rekomendāciju (*Recommendations*) profila izveidošanai – klašu un atribūtu noteikšanai - tiek izmantota cita metode. Iestatījumu profils ir atribūtu kopa no citiem lietotājam izveidotajiem profiliem, kur lietotājs izsaka savas vēlmes. Ieteikumu profils satur iestatījumu kopu, kas atbilst citiem dažādiem lietotājiem.

Lietotāju aprakstošā profila saites ir aprakstītas ar pakotņu diagrammu (skat. 23. attēlu).



23. attēls. Saistība starp lietotāju aprakstošiem profiliem (Kozmina, Niedrite, 2010)

Vienam lietotājam var būt vairāk kā viens vietas, laika, mijiedarbības, iestatījumu un rekomendāciju profils, ja viņš darbojas vairākās vidēs. Iestatījumu profils ir saistīts ar visiem citiem profiliem, bet rekomendāciju profils tiek izmantots, lai parādītu iespējas no citu lietotāju iestatījumiem, un kādas ir bijušas lietotāju izvēles.

#### 4.4. Integrētais lietotāja modelis

Pamatojoties uz lietotāju aprakstošajiem profiliem, kas aprakstīti iepriekšējā nodaļā, ir izveidots lietotāja modelis. Lietotājus aprakstošo profilu jēdziens paplašināts, lai tas

pārstāvētu vairāku sistēmu vajadzības, kas satur savus lietotāja profilus. Tiek uzkrāti dati katrā no avotiem par lietotāju, pēc tam integrētais lietotāja modelis var tikt izmantots gan pašās sistēmās, gan arī datu noliktavā dažādiem mērķiem.

Analizēta tiks tikai viena lietotāju aprakstošo profilu komponente – tieši lietotāja profils, jo tradicionāli sistēmās vietas, laika u.c. profilus neglabā datu bāzē, līdz ar to nav izejas datu atbilstoša integrēta profila izveidei.

Apskatīsim gadījumu, kad lietotājs darbojas ar vairākām sistēmām organizācijā, kur viena no tām ir datu noliktavas sistēma, jo integrēta profila informācija var būt noderīga pamatā tieši datu noliktavai, jo glabā integrētu saturu no vairākām sistēmām, kā arī var tikt izmantota profilu informācijas papildināšanai – apmaiņai ar citām sistēmām ar datiem, kas katrā sistēmā atšķirīgi apraksta vienu un to pašu lietotāju. Šajā gadījumā mērķis varētu būt informācijas piegādes personalizācijas iespēju paplašināšana.

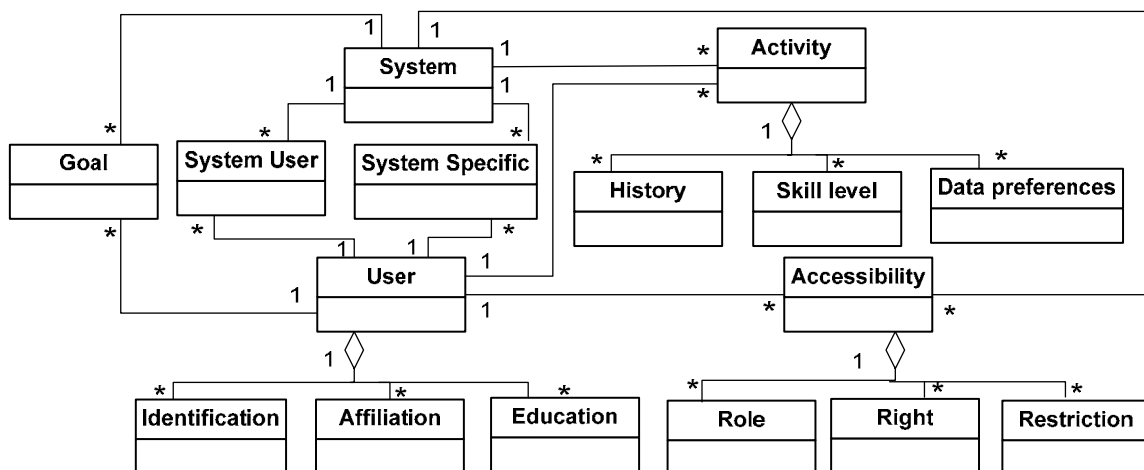
#### 4.4.1. Integrētā lietotāja modeļa īpašības

Lai izveidotu integrētu lietotāja modeli, tiek izmantotas struktūras, kas tika aprakstītas 4.3. apakšnodaļā par lietotāju aprakstošo profilu, izmantojot sekojošus principus:

- pievienotas datu struktūras no “izglītojamā profila” (IMS), jo pētījuma un arī promocijas darba rezultāti aprobēti universitātē, tādēļ tika izvēlēts izglītības jomas sistēmu lietotājus aprakstošs standarts, citos gadījumos būtu jāizmanto konkrētai sfērai atbilstošu sistēmu lietotājus aprakstošs standarts, ja tāds eksistē, vai vismaz izmantotās sistēmas lietotāja atribūti,
- modelis vispārināts daudzu sistēmu gadījumam, jo metode paredz integrēt datus no daudzu sistēmu lietotāju profiliem,
- modelim jānodrošina paplašināmība, ļaujot pievienot modelim jaunus, sistēmai specifiskus atribūtus, ja tiek ieviesta jauna informācijas sistēma, un integrētajā lietotāja modelī tiek integrēti dati arī no jaunās sistēmas profila.

Tālāk dots īss apraksts katrai klasei no integrētā lietotāja modeļa (skat. 24. attēls).

Identifikācija (*Identification*) ietver lietotāja personīgo informāciju (piemēram, vārds, uzvārds, dzimšanas datums), demogrāfisko informāciju (dzimums, vecuma grupa), kontaktinformāciju (telefons, e-pasts, adrese) u.c. Nodarbošanās (*Affiliation*) satur lietotāja profesionālos atribūtus (amats, kompānija, darba pieredze gados). Izglītība (*Education*) apraksta lietotāja izglītību (vai patreiz studē, izglītības iestāde, izglītības ieguves gads, diploma numurs, apbalvojumi u.c.).



24. attēls. Integrētā lietotāja modeļa klašu diagramma



Pieejība (*Accessibility*) apraksta lietotāja spēju mijiedarboties ar datu avotu vai datu noliktavas vidi. Struktūra satur tādu informāciju, kā lietotāja loma (*Role*), lietotāja tiesības (*Rights*) uz katru datu avotu, un ierobežojumi (*Restrictions*) (piemēram, invaliditāte vai valodas izvēle). Mijiedarbības sistēmas puse (pārlūkprogrammas funkcionālas īpašības) ir aprakstītas mijiedarbības profilā.

Aktivitāte (*Activity*) tiek izmantota, lai aprakstītu lietotāja darbības, tai skaitā darbību vēsturi (*History*), lai izsekotu detalizētām vai apkopotām lietotāja darbībām un sistēmu komponentu lietošanas intensitātei, ko raksturo, piemēram, klikšķu skaits (Interneta bāzētām lietotnēm) un kādas komponentes izmantošanas laiks noteiktā laika periodā, kopējais sesiju skaits, pēdējās sesijas sākums/beigas, lietotāja IP adrese, kopējais izmantoto sistēmas komponentu skaits, izmantoto komponentu skaits pret lietotājam pieejamo komponentu skaitu. Prasmju līmeņa (*Skill level*) atribūti definē lietotāja pieredzi darbā ar sistēmām, kas kalpo par datu avotiem datu noliktavai, un darbā ar datu noliktavas atskaišu līdzekļiem. Piemēram, ir iespējams izsekot, vai lietotājs izmanto datu avotus un kā tos lieto, piemēram, cik intensīvi tos izmanto (laiks, biežums). Datu iestatījumi (*Data Preferences*) iekļauj lietotāja iestatījumus attiecībā uz datiem datu avotu sistēmās un datu noliktavas atskaišu rīkos, iestatījumus uz atskaišu struktūru u.c.

Mērķa (*Goals*) atribūti ņem vērā lietotāja lomu katrā sistēmā. Sekojot līdzī komponentu kopai, ko izmanto lietotājs, un to izmantošanas biežumam, var noteikt sistēmas izmantošanas mērķi – darba vajadzībām, personīgām vajadzībām, apmācībai, analīzei, datu ievadam u.c.

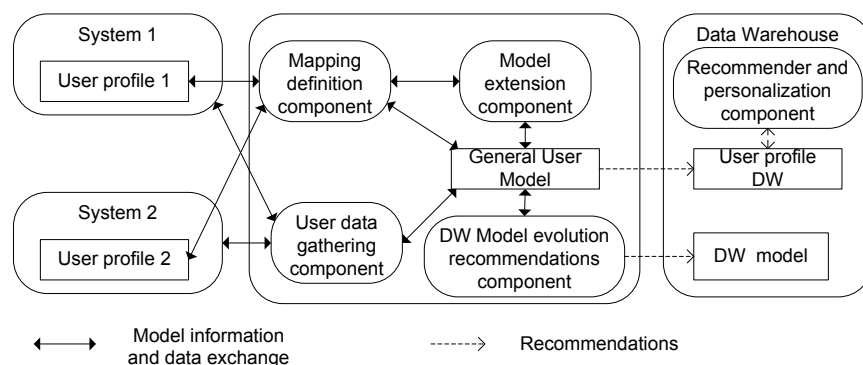
Klase Sistēma (*System*) reprezentē sistēmas, kas tiek izmantotas organizācijā. Sistēmas specifiskie (*System Specific*) atribūti tiek izmantoti vispārīgā modeļa paplašināšanai nepieciešamības gadījumā, ja jauna sistēma ar būtiskiem, lietotāja modeli papildinošiem atribūtiem tiek ieviesta organizācijā.

Sistēmas lietotāja (*System user*) atribūti apraksta lietotāja piekļuves informāciju katrai atsevišķajai sistēmai. Klase Lietotājs (*User*) identificē un apraksta lietotāja personas datus.

#### 4.4.2. Eksistējošu lietotāja profilu integrācijas process

Metode ir piedāvāta situācijai organizācijā ar vairākām sistēmām, kur viena no tām ir datu noliktava, tādēļ iespējams izmantot eksistējošu infrastruktūru un tipiskus paņēmienus datu noliktavu jomai – kartējumu definēšanu, datu integrāciju un glabāšanu datu noliktavā.

Tālāk metodes izklāstam apskatīsim gadījumu, kad viena fiziska persona darbojas ar divām atšķirīgām sistēmām, kas kalpo par datu noliktavas avota sistēmām. Katra no tām satur savus, atšķirīgus lietotāja profilus. Attēlā (skat. 25. attēlu) parādīts integrācijas process diviem profiliem turpmākai integrētā lietotāja modeļa izmantošanai saskaņā ar mērķiem, kas aprakstīti 4.2. apakšnodaļā. Shēma definē komponentus, sistēmas datu apmaiņas un lietotāju modeļu datu apmaiņas virzienus starp komponentiem (*Model information and data exchange*), rekomendāciju avotus un izmantošanas mērķus (*Recommendations*).



25. attēls. Integrētā lietotāja modeļa datu ieguves un izmantošana

Galvenie soļi lietotāja profilu integrācijas procesam ir sekojoši:

- Atbilstību definēšanas komponents (*Mapping definition component*) – atbilstība tiek definēta katram atsevišķajam lietotāja modelim pret integrēto lietotāja modeli (*General user model*). Trīs atšķirīgas situācijas ir iespējamās: (i) eksistē automātiski nosakāma atribūtu identifikācija ar identiskiem atribūtu vārdiem, (ii) semantiski vienādi atribūti ar atšķirīgiem atribūtu vārdiem, (iii) atbilstošie atribūti nav atrodami. Pēdējie divi gadījumi jāapskata manuāli, bet no tiem trešajam gadījumam (iii) vēl tiek izpildīts nākošais solis. Manuālai apstrādei atbilstību definēšanai starp atribūtiem izmantojamas tradicionālas datu noliktavas tehnikas – ETL procesu definēšana, atribūtu kartēšana.
- Lietotāju modeļa paplašināšanas aktivitāte (*Model extension component*) tiek izmantota, lai pieņemtu lēmumu, kurus lokālā lietotāju modeļa atribūtus nepieciešams pievienot integrētajam lietotāja modelim.
- Lietotāju datu savākšana (*User data gathering component*) un apkopošana notiek pēc tam, kad visi lokālie lietotāja profili ir apstrādāti, atbilstības definētas un, ja nepieciešams, pamata modelis paplašināts. Atkarībā no atbilstības informācijas, dati tiek savākti un integrēti vispārīgā lietotāju modelī. Šeit lietojami no ETL procedūru definīcijām (atribūtu kartējumiem) ģenerēti skripti datu izguvei un pārvietošanai no datu avota uz datu noliktavu (integrēto lietotāju modeli - tā realizāciju - var uzskatīt par organizācijas datu noliktavas vienu specifisku datuvi).
- Rekomendāciju komponente tiek izmantota, lai analizētu savāktu informāciju (lai identificētu pretrunīgus lietotāja profila datus vienam un tam pašam vispārīgā lietotāja modeļa atribūtam, piemēram, atšķirīgas pieejas tiesības uz vienu un to pašu objektu dažādās sistēmās). Tālāk tiek veiktas sekojošas divas darbības:
  1. Rekomendācijas datu noliktavas lietotāja modelim tiek sagatavotas, piemēram, pieejas tiesību definēšanā, ja izveidotas jaunas datu noliktavas atskaites, vai tiek iesaistīti jauni datu noliktavas lietotāji. Integrētā profila informācija izmantojama arī datu noliktavas sistēmas personalizācijai (*Recommender and personalization component*), lai nodrošinātu atbilstošu informāciju lietotājam.
  2. Tiek sagatavotas rekomendācijas tālākai datu noliktavas izstrādei (*DW evolution recommendation component*), piemēram, kādi jauni datu kubi varētu būt noderīgi lietotājiem, balstoties uz savāktu lietotāju aktivitātes informācijas datu avotu sistēmā.

Datu noliktavu personalizācijas aspekti promocijas darbā netiek apskatīti, bet 4.5. nodaļā tiek piedāvāta metode, kā integrētais lietotāja modelis izmantojams veikspējas mērīšanas sistēmas datu noliktavas modeļa attīstīšanā.

#### **4.5. Integrētā lietotāju modeļa izmantošana mērīšanas sistēmā**

Integrētais lietotāja modelis var tikt izmantots arī veikspējas mērīšanas sistēmā, kuras pamatā ir datu noliktava. Integrētais lietotāja modelis pielietots 4.4.2. punktā aprakstītajā otrajā variantā (rekomendāciju sagatavošana datu noliktavas modeļa attīstīšanai). Izmantotas sekojošas iesaistītās sistēmas un komponenti sekojošā konfigurācijā:

- izmantota datu noliktavas datu avota sistēma ar eksistējošu lietotāja modeli,
- eksistējošais lietotāja modelis papildināts ar klasēm no integrētā lietotāja modeļa, lai aprakstītu lietotāja aktivitātes.

Tālāk aprakstīta metode datu vākšanai un analīzei par lietotāja aktivitātēm datu avotā, lai varētu sagatavot rekomendācijas datu noliktavas modeļa attīstīšanai. Lietotāja aktivitātes tiek uzskatītas par integrētā lietotāja modeļa sastāvdaļu.

Tiek piedāvāta vaicājumu-bāzēta metode, kas nosaka lietotāju informācijas vajadzības no eksistējošiem lietotāju vaicājumiem un to lietošanas statistikas. Metode pieņem, ka vaicājumi attiecībā pret avota datubāzi atspoguļo lietotāju analīzes vajadzības. Metode rekomendē izmaiņas eksistējošas datu noliktavas shēmā, metode izmanto datu noliktavu versiju atbalstu, lai atspoguļotu visas izmaiņas.

#### **4.5.1. Datu noliktavu modeļu izveides pieejas**

Datu noliktavas ir bāzētas uz daudzdimensiju modeļiem, kas satur faktus (analīzes mērķi), mērījumus (kvantitatīvus datus), dimensijas (klasificējošus datus) un dimensiju atribūtus. Dimensiju atribūti veido klasifikācijas hierarhijas.

Izstrādātās, uzturētās un lietotās datu noliktavas pēc kāda laika perioda ir nepieciešams mainīt, jo rodas jaunas informācijas vajadzības, mainās datu avotu datu struktūras. Eksistē dažādas metodes, kā noskaidrot prasības datu noliktavai, lai tiktu izveidots piemērotākais konceptuālais modelis. Tai pat laikā eksistē daudzas pieejas, kā uzturēt izmaiņas datu noliktavās.

Datu noliktavu konceptuālo modeļu izstrādes metodes tiek klasificētas kā piedāvājuma vadītas (jeb datu vadītas), vai kā pieprasījuma vadītas metodes. Pieprasījuma vadītas metodes vēl sīkāk iedala atkarībā no veida, kā noskaidro prasības, piemēram, lietotāju vadītas (Westerman, 2001), procesu vadītas (Kaldeich, Oliveira, 2004), vai mērķu vadītas (Giorgini et al., 2005). Bieži tiek lietota ātrākā, un zināmā mērā vienkāršākā metožu grupa – datu vadītas metodes (Golfarelli et al., 1998), kas transformē datu avotu modeļus par datu noliktavas zvaigznes shēmām. Tomēr tiek piedāvātas arī jauna tipa metodes, piemēram, šablonu vadītas (Jones, Song, 2005), lai novērstu eksistējošu metožu trūkumus, starp kuriem jāmin sekojošie: datu vadītās metodes mehāniski izveido modeli, kas rezultātā var neatspoguļot lietotāju patiesās vajadzības pēc datiem, bet pieprasījumu vadītām metodēm rezultāts (modelis) ir atkarīgs no intervēto lietotāju zināšanām, turklāt netiek pienācīgi ņemti vērā pieejamie dati datu avotos.

Datu noliktavu konceptuālie modeļi laika gaitā attīstās, jo mainās gan datu avotu modeļi, gan lietotāju prasības. Lai atspoguļotu šīs izmaiņas, ir iespējams izmainīt eksistējošas datu noliktavu shēmas un datu savākšanas un transformācijas (*Extract, Transform, Load*, turpmāk ETL) procesus. Šo izmaiņu rezultātā var pazaudēt izmaiņu vēsturi, tādēļ labāk ir izmantot datu noliktavas shēmu versijas. Atbilstoši (Golfarelli et al., 2006) ‘shēmas versija ir shēma, kas atbilst biznesa prasībām noteiktā laika intervālā (derīguma intervālā), kas atbilst laikam starp dotās shēmas izveidi un nākošās shēmas izveidi’

Atbilstoši *Design Science* pieejai (Kuechler, Vaishnavi, 2008), problēma, kas tiek risināta ar šo metodi, ir kā saskaņot jaunās informācijas prasības ar eksistējošu datu noliktavas konceptuālo modeli. Lai problēmu atrisinātu, tiek ieteikta jauna tipa metode – vaicājumu bāzēta metode, kas noskaidro prasības no datu avotos eksistējošiem vaicājumiem un to lietošanas statistikas, tādejādi kombinējot lietotāju vajadzības un eksistējošu datu piedāvājumu. Metode tiek lietota, lai rekomendētu izmaiņas datu noliktavas konceptuālajā shēmā.

Metode ir paredzēta gadījumam, kad datu noliktavas evolūcijas atbalsts ir implementēts ar shēmu versijām, kas aprakstītas metadatos (skat. punktu 4.5.2). Metodes soļi ir aprakstīti punktā 4.5.4.

Piedāvātā metode lietotāju informācijas prasību noteikšanai balstās uz pieņēmumu, ka lietotāja vaicājumi datu avotu datu bāzēs atspoguļo lietotāja vajadzības pēc datiem. Daži citi autori izmanto līdzīgu ideju, bet nedaudz atšķirīgā kontekstā.

Datu kubu izveidi, balstoties uz lietotāju vaicājumiem eksistējošā datu noliktavā, piedāvā Niemi et al. (2001). Autoru mērķis ir uzlabot vaicājumu ātrdarbību, rekomendējot piemērotu apakškubu izveidi. Autori citam pētījumam (Pang et al., 2004) uzskata, ka informācijas sistēmās eksistējoši vaicājumi atspoguļo lietotāju vajadzības pēc datiem, bet viņi uzskata visus vaicājumus par līdzvērtīgiem lietotāju vajadzību reprezentēšanā, tiek analizēta vaicājumu struktūra, netiek ņemts vērā, vai vaicājums vispār tiek lietots, un cik bieži. Nair et al. (2007) piedāvā vaicājumu vadītu izstrādes ietvaru. Ietvars lieto paredzamos lietotāju vaicājumus, reālu vaicājumu vietā. Lietotāju vaicājumi tiek izteikti dabiskā valodā. Vēl citi autori pievēršas datizrades metodēm, kā noskaidrot tipiskus lietotāju vaicājumus (Zhang et al., 2005).

No iepriekš minētām, vislīdzīgākā promocijas darbā piedāvātajai metodei ir Pang et al. (2004) piedāvātā, tomēr promocijas darbā piedāvātajā tiek ņemta vērā arī vaicājumu lietošanas statistika, kā arī metode ir paredzēta noteiktam mērķim, lai rekomendētu nepieciešamās izmaiņas datu noliktavā.

Literatūrā ir aprakstītas daudzas pieejas datu noliktavas attīstības atbalstam, kas izmanto versiju pieeju. Galvenā ideja Golfarelli et al. (2006) ir glabāt paplašinātās shēmas kopā ar shēmu versijām, lai nodrošinātu vaicājumu izpildi uz vairākām shēmas versijām. Cita metode (Body et al., 2002) piedāvā datu un struktūru versijas dimensijām, kas ļauj saglabāt vēsturi, nodrošina datu atspoguļošanu noteiktā struktūras versijā, kas atbilst vaicājumam atbilstošiem laika parametriem. (Wrembel, Bebel, 2005) piedāvā meta datu pārvaldības modeli daudzversiju datu noliktavai.

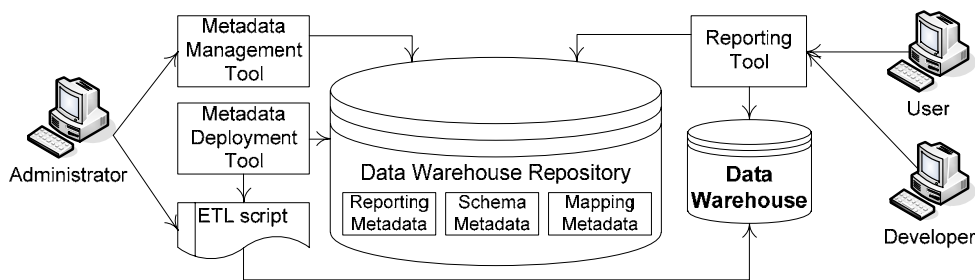
Iepriekš apskatītie pētījumi neapskata tiešā veidā, kā datu noliktavu pielāgot datu avotu izmaiņām, atšķirībā no citiem pētījumiem (Velegrakis et al., 2003). Šīs pieejas balstās uz transformācijām no vienas shēmas uz otru. Šāda tipa metodes definē, kā pēc izmaiņām datu avota shēmā iegūt datu noliktavas shēmu, bet uztur tikai vienu – aktuālo datu noliktavas shēmu.

Promocijas darbā piedāvātajā metodē tiek izmantota versiju pieejas priekšrocības, lai atspoguļotu visas izmaiņas, kas notiek analizējamā procesā, tāpat tiek analizēta lietotāju aktivitāte avotu sistēmā, bet netiek skatītas avota shēmas izmaiņas, lai ieteiktu izmaiņas datu noliktavas shēmā.

#### ***4.5.2. Metadati mērīšanas sistēmas shēmas evolūcijas atbalstam***

Piedāvātā metode izmanto eksistējošu datu noliktavas shēmu, attēlojumu no tās elementiem uz datu avota shēmu, kā arī eksistējošas atskaites, kas definētas uz datu noliktavas shēmu.

Tiek izmantots datu noliktavas evolūcijas ietvars, kas aprakstīts rakstā (Solodovnikova, 2007). Visu rīku darbība un ietvars balstās uz metadatu repozitoriju, kas apraksta datu noliktavas shēmas versijas. Datu noliktavas shēma ir aprakstīta ar loģiskā līmeņa metadatiem, kas apraksta daudzdimensiju elementus un shēmas versijas, un fiziskā līmeņa metadatiem, kas apraksta shēmas glabāšanu relāciju datu bāzē (Solodovnikova, 2010). Repozitorijs satur arī attēlojuma metadatus, kas apraksta ETL procesu loģiku- avota datu elementus (tabulas un kolonnas relāciju datubāzē) un veidu, kā šie dati tiek attēloti par datu noliktavas elementiem, tabulām un kolonnām relāciju datubāzē (Solodovnikova, Niedrite, 2006). Datu noliktavas evolūcijas ietvara metodei būtiskās daļas redzamas attēlā (skat. 26. attēlu).



26. attēls. Datu noliktavas evolūcijas ietvars (daļa)

Evolūcijas ietvarā esošais atskaišu rīks atļauj definēt atskaites uz vairākām datu noliktavas shēmas versijām. Datu noliktavas metadatu repozitorijs tiek administrēts ar metadatu pārvaldības rīku, metadatu izveides rīks ģenerē ETL skriptus, balstoties uz shēmu attēlojumu metadatiem no repozitorija, pēc tam skripti tiek izpildīti, lai aizpildītu datu noliktavas shēmu ar datiem.

#### 4.5.3. Informācijas avoti vaicājumu lietošanas analīzei

Sistēmās tiek izmantoti dažādi žurnālfaili operatīviem mērķiem, piemēram, auditēšanai, optimizēšana u.c. Datu noliktavu jomā žurnālfailus datu avotu sistēmā bieži lieto, lai noteiktu izmaiņas datu avotu datos. Lai noteiktu izmaiņas datu avotu datos tiek izmantoti programmatūras līdzekļi, datubāzes trigeri, transakciju žurnālfaili, salīdzinot lauku vērtības divās dažādās datubāzes kopijās. Dažas no šīm tehnikām var tikt pielietotas, lai analizētu vaicājumu lietošanas statistiku. Metode izmantos dažāda tipa žurnālfailu eksistenci sistēmā, kas uzkrāj darbības un izmaiņas datus.

Funkciju lietošanas statistika tiek uzkrāta ar iebūvētu īpašību lietotnē, kas tiek izsaukta katru reizi, kad funkcija tiek izpildīta. Žurnālfails apraksta funkcijas nosaukumu, lietotāju, un datumu formātā: *Function\_name, user, date*. Šis žurnālfails ļauj noteikt visbiežāk lietotās funkcijas. Nepieciešama ir papildus informācija par datu manipulācijas teikumiem šajās funkcijās.

Datu izmaiņas saglabā trigeri. Katrai tabulai ir izveidots savs trigeris, kas saglabā informāciju par Insert, Update un Delete datu manipulācijas teikumiem. Visi trigeri saglabā izmaiņas vienā žurnālfailā ar sekojošu formātu: *Table\_name, user, date, change\_type, change\_text* (Tabulas nosaukums, lietotājs, datums, izmaiņas tips, mainītā vērtība). Statistika par katru tabulu un atribūtu ļauj noteikt biežāk lietotās.

Speciāla tipa funkcijas ir atskaites ar daudziem parametriem, ko izmanto, lai iegūtu specifisku atskaiti, atfiltrējot tikai interesējošos datus. Var teikt, ka vaicājumi atskaitēs reprezentē lietotāju informācijas vajadzības, tādēļ informācijas saglabāšana par atskaišu izmantošanu ir īpaši nozīmīga izpētei. Šāda žurnālfaila formāts ir sekojošs:

*Function\_name, user, date, parameter\_text*, kur  
*parameter\_text* satur parametru vērtības formā:  $p1=<value1> \dots pN=<valueN>$ , kur  
 $p1 \dots pN$  ir formā:  $<table\_name.column\_name>$ ).

Papildus žurnālfailu datiem, būtiska vēl ir sistēmā realizēto funkciju datu manipulācijas teikumu struktūras analīze. Šie dati kopā ar funkciju lietošanas statistiku dod precīzāku informāciju par lietotāja darbībām. Var tikt sagatavoti divu tipu informācijas avoti – viens par *Insert* un *Update* teikumiem, otrs par *Select* teikumiem.

Pirmā avota (*Insert* un *Update*) struktūra ir sekojoša:

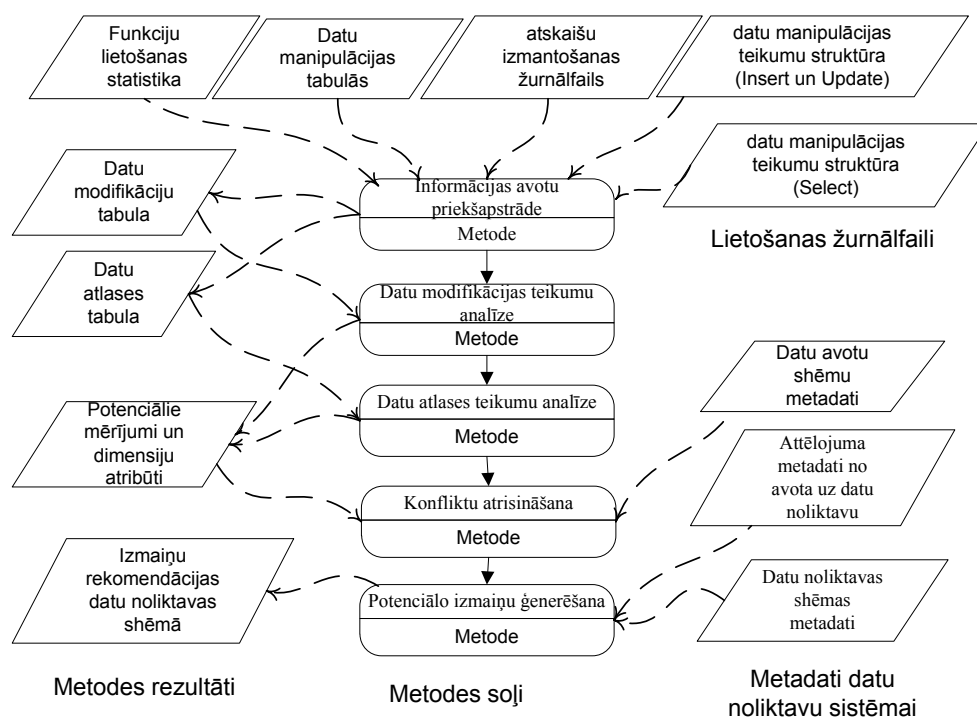
*Function\_name*, *Action\_Type*, *List\_of\_columns*, *Where\_condition*, kur:  
*Action\_Type* vērtības ir Update vai Insert  
*List\_of\_columns* un *Where\_condition* satur kolonnu nosaukumus formā:  
 <table\_name.column\_name>

Otrā avota (*Select*) struktūra ir sekojoša:

*Function\_name*, *Joins*, *Where\_condition*, *Group\_By*, *Columns\_Select*, *Aggregation*, kur:  
*Joins*, *Where\_condition*, *Columns\_Select*, un *Group\_By* satur kolonnu nosaukumus formā: <table\_name.column\_name>  
*Aggregation* satur kolonnu nosaukumus formā:  
 <table\_name.column\_name> un agregācijas funkciju, piemēram, *SUM*, *COUNT*, u.c..

#### 4.5.4. Vaicājumu vadīta metode datu noliktavas modeļa attīstīšanai pēc lietotāja aktivitātēm datu avotu sistēmās

Piedāvātā vaicājumu vadītā metode datu noliktavu shēmas izmaiņu rekomendēšanai un jaunu atskaišu rekomendēšanai sastāv no 5 soļiem (27. attēls).



27. attēls. Vaicājumu vadītas metodes shēma

##### 1. solis. Informācijas avotu priekšapstrāde

Pirmajā solī notiek iepriekš aprakstīto informācijas avotu priekšapstrāde. Rezultātā tiek izveidotas divas lietošanas tabulas.

Pirmā lietošanas tabula ir Datu modifikāciju tabula, kas satur datus par *Insert* un *Update* teikumu izpildes biežumu tabulās. Pirmās lietošanas tabulas formāts ir sekojošs:

*Function\_name, List\_of\_columns, Where\_condition, count.*

Tā kā kolonnas *Date* precizitāte lietošanas žurnāltabulā un *Insert, Update* un *Delete* datu manipulācijas teikumu žurnāltabulā ir ne mazāk kā 1 sekunde, tad tiek pieņemts, ka vienā sekundē viens lietotājs nevar veikt datu manipulācijas vienā tabulā ar divām dažādām funkcijām.

Turklāt ir pieejama informācija par datu manipulācijas teikumu struktūru, kas izmantoti funkcijās. Tas rada iespēju sasaistīt doto funkciju, ko izpildīja lietotājs ar konkrētu datu manipulācijas teikumu, kas reģistrēts žurnāltabulā ar trigeru palīdzību. Datu manipulācijas teikumu skaits vienai tabulai ar vienu un to pašu kolonnu komplektu tiek summēts analizējamā periodā par visiem lietotājiem.

Otrā lietošanas tabula ir Datu atlasas tabula, kas satur datus par *Select* teikumu biežumu, ko funkcijas pielieto tabulām. Datu atlasas tabulas formāts ir sekojošs:

*Function\_name, Joins, Where\_condition, Group\_By, Columns\_Select, Aggregation, count.*

Lai izveidotu ierakstu Datu atlasas tabulās, dati no atskaišu lietošanas žurnāltabulas tiek kombinēti ar datiem par *Select* teikumu uzbūvi atskaitēm atbilstošajās funkcijās. *Select* teikumu skaits tiek summēts analīzes perioda ietvaros par visiem lietotājiem.

Divas aprakstītās lietošanas tabulas, kas iegūtas priekšapstrādes rezultātā, ierobežo tabulu skaitu, ļaujot ņemt vērā tikai tās, ko lieto analizējamie biznesa procesi, kas ir paredzamās mērīšanas fokuss, un kuru mērīšanai jāveic datu analīze datu noliktavā.

## **2. solis. Datu modifikācijas teikumu analīze**

Pēc abu lietošanas tabulu sagatavošanas, dati šajās tabulās tiek analizēti, lai iegūtu potenciālos datu noliktavu shēmas elementus.

Vispirms tiek analizēta Datu modifikāciju tabula. Katrai kolonnai tiek aprēķināts skaits, cik reižu tā parādās *Insert* vai *Update* teikumos (skaits cik reizes atbilstošā kolonna parādās *List\_of\_columns* kolonnā). Pēc šī aprēķinātā skaita tiek izvēlētas Top N kolonnas (N vērtība jāizvēlas katrā analīzes gadījumā, par pamatu ņemot konkrētajā gadījumā aprēķinātās skaita vērtības). Šīs Top N kolonnas var tikt izvēlētas par potenciāliem datu noliktavas mērījumiem. Vairāki vienai tabulai piederoši mērījumi tiek apvienoti vienā potenciālā faktu tabulā.

Tad tiek identificēti potenciālie dimensiju atribūti no Top N kolonnām ar lielāko izmantošanas skaitu *Where\_condition* kolonnā datu modifikāciju tabulā. Tiek pieņemts, ka potenciālie faktu atribūti var tikt analizēti kopā ar potenciāliem dimensiju atribūtiem, ja tie tiek lietoti kopā vienā datu modifikāciju teikumā. Tāpēc, ja potenciālais mērījums ir iekļauts *List\_of\_columns* un potenciālais dimensijas atribūts ir lietots *Where\_condition* daļā vienā datu modifikācijas teikumā, tad potenciālais dimensijas atribūts tiek piesaistīts potenciālai faktu tabulai. Vairāki vienai avota tabulai piederoši šādi potenciālie dimensiju atribūti tiek apvienoti vienā datu noliktavas dimensiju tabulā.

## **3. solis. Datu atlasas teikumu analīze**

Informācija datu atlasas tabulā tiek analizēta, lai atrastu potenciālos dimensiju atribūtus, mērījumus un saites starp faktu un dimensiju tabulām.

Analīzes laikā apskata Top N atribūtus pēc lietošanas biežuma *Where\_condition* vai *Group\_By* kolonnās, šie atribūti tiek uzskatīti par potenciāliem dimensiju atribūtiem. Top N atribūti ar biežāko pārādīšanās skaitu kolonnā *Aggregation* tiek uzskatīti par potenciāliem mērījumiem.

Tiek analizēti arī atribūti, kas iekļauti kolonnā *Columns\_Select*. Ja atribūts ir izmantots vaicājumā, kas satur agregācijas funkciju, kas atrodama *Aggregation* kolonnā, tad atribūtu

uzskata arī par potenciālu dimensijas atribūtu. Atribūti, kas izmantoti *Columns\_Select* sadaļā, ja vaicājumā nav *Aggregation* daļas, ir uzskatāmi par potenciāliem mērījumiem tikai tad, ja atbilstošais atribūts ir skaitlisks.

Vairāki potenciālie mērījumi, kas datu avotā pieder vienai avota tabulai tiek apvienoti vienā potenciālā faktu tabulā, un vairāki vienai avota tabulai piederoši potenciālie dimensiju atribūti tiek apvienoti vienā datu noliktavas dimensiju tabulā.

Avota atribūti, kas atrodami *Joins* kolonnā, parasti ir primārās vai ārējās atslēgas avota tabulā, kas var uzrādīt saistību arī starp potenciālām dimensiju un faktu tabulām vai arī saistību starp divām potenciālām dimensiju tabulām. Pēdējā gadījumā šī saistība var norādīt uz potenciālu hierarhisku sakarību starp vienas hierarhijas līmeņiem un nepieciešamību apvienot divas potenciālās dimensiju tabulas vienā.

#### **4. solis. Konflikta atrisināšana**

Analizējot lietošanas datus, ir iespējams, ka tā pati avota kolonna var tikt identificēta kā potenciāls fakta mērījums un kā potenciāls dimensijas atribūts. Šādā gadījumā tiek analizēts kolonnas datu tips. Ja datu tips ir skaitlisks, tad šādu kolonnu uzskata par mērījumu, bet ja datu tips ir teksts vai datums, tad to uzskata par dimensijas atribūtu, kaut arī teorētiski šāda atribūta vērtību bieža maiņa varētu interesēt arī kā mērījums.

#### **5. solis. Potenciālo izmaiņu ģenerēšana**

Pēdējā metodes solī potenciālie datu noliktavas elementi tiek salīdzināti ar eksistējošo datu noliktavas shēmu, kam tiek izmantoti datu noliktavas evolūcijas ietvarā esošie metadati – attēlojums starp avota shēmas elementiem un eksistējošiem datu noliktavas elementiem. Ja potenciālais datu noliktavas elements ir jau ietverts eksistējošā datu noliktavas shēmā, tad šo elementu ignorē. Pārējos gadījumos tiek ģenerētas rekomendācijas kā izmaiņu darbības, kas tiek atbalstītas datu noliktavas evolūcijas ietvarā. Tādu izmaiņu darbību piemēri ir jauna dimensijas atribūta pievienošana, jaunas dimensijas tabulas vai faktu tabulas pievienošana, jauna mērījuma pievienošana eksistējošai faktu tabulai, jaunas hierarhijas vai hierarhijas līmeņa pievienošana. Ja datu noliktavas administrators akceptē šādas izmaiņas, tad tās tiek izpildītas atbilstoši noteiktai iepriekš definētai izmaiņu apstrādes procedūrai (Solodovnikova, 2010), kas veic nepieciešamās izmaiņas datu struktūrās datu noliktavā, kā arī loģiskajos, fiziskajos un attēlojuma metadatos. Šo modifikāciju rezultātā tiek izveidota jauna datu noliktavas versija.

Atsevišķi tiek veidotas rekomendācijas jaunu atskaišu izveidei, par pamatu ņemot biežāk lietoto *Select* teikumu statistiku, kas tiek noskaidrota no *Datu atlas* tabulas.

#### ***4.5.5. Metodes darbības piemērs***

Lai ilustrētu, kā metode darbosies, tālāk tiks apskatīts piemērs.

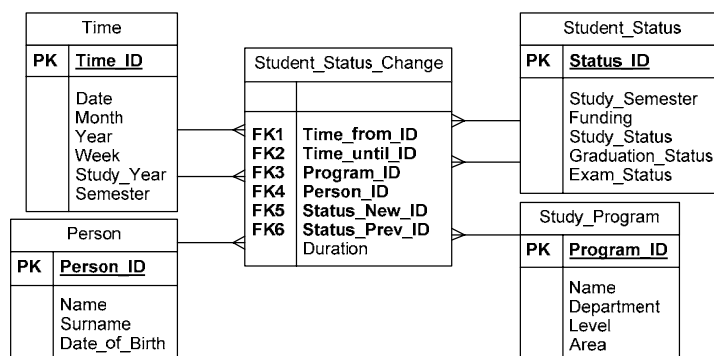
##### **Iesaistīto sistēmu datu modeļi**

Šajā piemērā datu noliktavā tiek glabāta informācija par studentu statusu maiņu universitātē. Zvaigznes shēma attēlota 28. attēlā. Vienkāršības dēļ shēma satur tikai apakškopu no dimensiju atribūtiem, kas ir būtiski metodes ilustrēšanai. Datu noliktava satur faktu tabulu „Studiju statusa izmaiņas” (*Student\_Status\_Change*), kas uzglabā personas studiju statusa izmaiņas laika gaitā, visu studiju programmu studentiem.

Dimensija „Studentu Statusi”, (*Student\_Status*) glabā statusus, ko raksturo atribūti: semestris (*Study\_Semester*), finansējums (*Funding*) – budžets vai maksa, statuss (*Study\_Status*) - studē, pārtraukumā, bijušais students u.c., beigšanas statuss



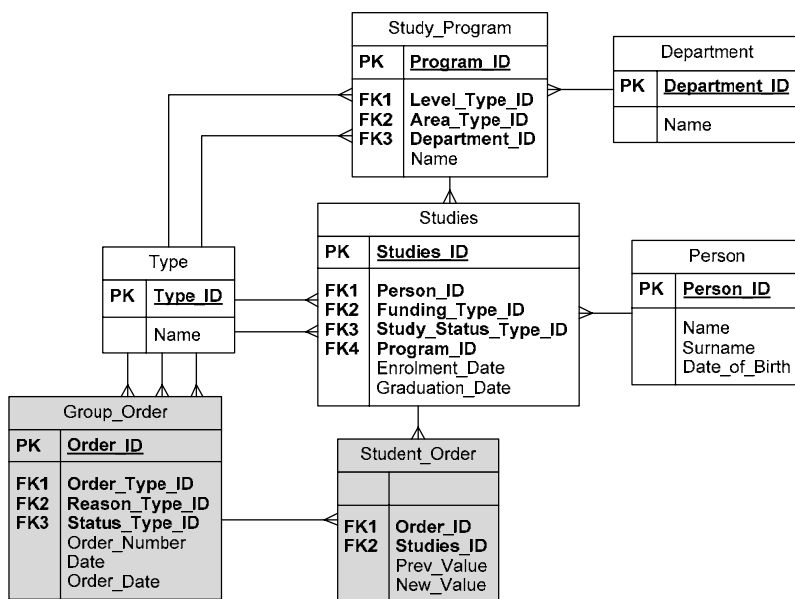
(*Graduation\_Status*) – beidzis ar grādu, beidzis bez grāda, u.c.) un eksāmena statuss (*Exam\_Status*) – gala pārbaudījums, gala pārbaudījumu nokārtojais u.c.



28. attēls. Piemēra datu noliktava

Datu noliktavai dati tie iegūti no atbilstošas informatīvās sistēmas, kuras datu modeļa fragments ir redzams 29. attēlā. Tabulas, kas tiek izmantotas kā datu avots datu noliktavas ETL procesiem ir sekojošas:

- Persona (*Person*)– dati par visiem studentiem,
- Struktūrvienības (*Department*) – universitātes struktūrvienības – fakultātes, departamenti u.c.,
- Studiju programmas (*Study\_Program*) – studiju programmas, kurās ir reģistrēti studenti,
- Studijas (*Studies*) – informācija par studentu reģistrāciju studiju programmās, ieskaitot datus par finansējumu, studiju statusu, kas raksturo senestri, beigšanas informāciju, ja students eksmatrikulēts, eksāmenu informāciju, ja studentam tādi jākārtu u.c.,
- Tipi (*Types*) – dažādu tipu nosaukumi, piemēram, studiju programmu līmeņi, nozares, finansējuma veidi, statusu nosaukumi u.c.



29. attēls. Datu noliktavas datu avota sistēmas datu modelis apskatāmajam piemēram

Izmaiņas, kas notikušas laika gaitā ar datu avota sistēmu ir saistītas ar pāreju uz atbilstošu rīkojumu sastādīšanu par izmaiņām studenta studiju statusā, līdz ar to mainās kārtība, kā dati tiek glabāti avota sistēmā –netiek glabāts tabulā Studijas tikai pēdējais, aktuālais statuss, bet tiek glabāti rīkojumi par maiņu, līdz ar to tiek glabāta visu studiju statusu maiņu vēsture. Izveidotas jaunas tabulas:

- Grupu rīkojumi (*Group\_Order*) - rīkojuma dati, ļauj veidot rīkojumus vairākiem studentiem: numurs, rīkojuma tips (ieskaitīšana, semestra maiņa, atskaitīšana u.c.), rīkojuma tipa iemesls (piemēram, atskaitīšanai – līguma nosacījumu nepildīšana), rīkojums statuss (sastādīts, atcelts u.c.), rīkojuma sastādīšanas un reģistrēšanas datumi;
- Studentu rīkojumi (*Student\_Order*) – individuāla studenta statusa maiņas informācijas glabāšanai – atribūti *Prev\_Value* un *New\_Value* paredzēti attiecīgi iepriekšējās un jaunās mainītās statusa vērtības glabāšanai.

Tā kā mainījās biznesa process, darbības, kādas veica lietotāji datu avota sistēmā arī mainījās. Datu noliktavas shēma tajā pašā laikā atbilst vecajam datu avota modelim un vecajam biznesa procesam. Piedāvātā metode palīdz noteikt nepieciešamās izmaiņas datu noliktavas shēmā, balstoties uz lietotāju darbībām datu avotā.

#### Rekomendācijas shēmas izmaiņām dotajam piemēram

Piedāvātā metode tika pielietota studentu datu shēmai avota sistēmā un datu noliktavas zvaigznes shēmai, atbilstoši iepriekš dotajam shēmas aprakstam. Pēc lietošanas žurnālfailu analīzes avota sistēmā, pielietojot iepriekš aprakstīto metodi, kā biežāk lietotie atribūti tika noteikti *Where\_Condition*, *Group\_By* and *Columns\_Select* kolonnās Datu atlasēs tabulā:

- *Name* atribūts tabulai *Type*;
- *Order\_Date* un *Order\_Number* atribūti tabulai *Group\_Order*;
- *Prev\_Value* un *New\_Value* atribūti tabulai *Student\_Order*.

Šīs kolonnas tika uzskatītas par potenciāliem dimensiju atribūtiem. Atbilstoši metodei, tika rekomendēts esošo datu noliktavas shēmu (skat. 28. attēlu) papildināt ar dimensijām *Group\_Order*, *Student\_Order* un *Type*. Pēc rekomendāciju caurskatīšanas, tika nolemts, apvienot šīs dimensijas vienā dimensijā *Order* ar atribūtiem *Order\_Date*, *Order\_Number*, *Prev\_Value*, *New\_Value*. Tajā pašā laikā atribūts *Name* tabulai *Type* tika pievienoti dimensijai *Order* kā 3 atsevišķi atribūti: *Order\_Type*, *Reason\_Type* un *Status\_Type*.

Rezultātā, izpildot jaunu dimensiju atribūtu un jaunu dimensiju pievienošanas darbības, tika iegūta jaunā datu noliktavas shēma, kurā bez esošajām ir vēl viena jauna dimensija *Order*.

## 4.6. Nodaļas secinājumi

Nodaļā aprakstīts integrēts lietotāja modelis, detalizētāk pievēršoties vienai no tā komponentēm – lietotāja profilam, kas ir būtiskākā visa lietotāja modeļa sastāvdaļa.

Tiek piedāvāta integrētā lietotāja modeļa izveides metode, kas balstās uz lietotāja profilu integrāciju. Aprakstīta situācija ar divām datu avota sistēmām, kuru lietotāju modeļi kalpo par datu avotu integrētajam modelim, un datu noliktavu, kas izmanto integrēto lietotāju modeli.

Iesaistīto sistēmu skaits metodei nav ierobežots, bet lielāka skaita sistēmu gadījumā, manuālas iejaukšanās gadījumu skaits integrācijas procesā var būt biežāk nepieciešams, lai

atrisinātu integrācijas konfliktus. Metodē nav arī aprakstīti atribūtu vērtību konflikti, kur risinājumi varētu atšķirties atkarībā no iesaistīto atribūtu semantikas.

Eksperimentu vide, kura izmantota vispārīgā lietotāju modeļa izveidei un lietotāju profilu integrēšanai, ir universitātes informācijas sistēma, e-studiju sistēma, un universitātes datu noliktava. Rekomendācijas iegūtas par datu noliktavas datu modeļa attīstīšanu promocijas darbā apskatītā piemēra gadījumā. Datu noliktavas lietotāja modeli paredzēts izmantot datu noliktavas personalizācijas vajadzībām, datu noliktavas personalizācijas jautājumi promocijas darbā netiek analizēti.

Piedāvātā datu noliktavas shēmas attīstīšanas metode ļauj sekot līdzī reālajai datu izmantošanai datu noliktavas datu avotu sistēmās un pielāgot datu noliktavas shēmu atbilstoši biežāk izmantotajiem datu elementiem lietotāju vaicājumos un datu manipulācijas teikumos.

Metode tiek izmantota, lai veidotu rekomendācijas jaunai datu noliktavas versijai. Tajā pašā laikā metode var tikt izmantota viena pati, bez versiju atbalsta ietvara, šajā gadījumā rekomendēto jauno shēmas elementu novērtēšana jāveic izstrādātājam. Eksistējošas datu noliktavas gadījumā tās shēma kalpo par pamatu salīdzināšanai, tai skaitā, lai atfiltrētu elementus, kas nav saistīti ar eksistējošu shēmu, bet var tikt izmantoti tālākai manuālai apstrādei. Eksistējošā datu noliktavas shēma reprezentē esošās datu analīzes vajadzības, ļauj papildināt esošās analīzes iespējas ar jaunām analīzes iespējām, pievienojot jaunus datu noliktavas shēmas elementus (pēc akceptēšanas). Tāpēc var uzskatīt, ka datu avota sistēmas ar datu analīzi nekādi nesaistīto operatīvo atskaišu ietekme uz izmaiņām datu noliktavas shēmā tiek pēc iespējas minimizēta.

Analizējot apskatīto piemēru un metodes pielietojumu tajā, tika iegūtas rekomendācijas datu noliktavas shēmas izmaiņām, kas pēc izvērtēšanas un diskusijām, tika ieviestas jaunā shēmas versijā. Metodes realizācijā nav paredzēta dinamisku SQL apstrāde, kas ierobežoja metodes pārbaudi. Metode var tikt uzlabota, papildinot ar informāciju par atskaišu lietošanas paradumiem pašā datu noliktavā.

## 5. Veiktspējas mērīšana DNATIS arhitektūrā

### 5.1. Nodaļas nolūks

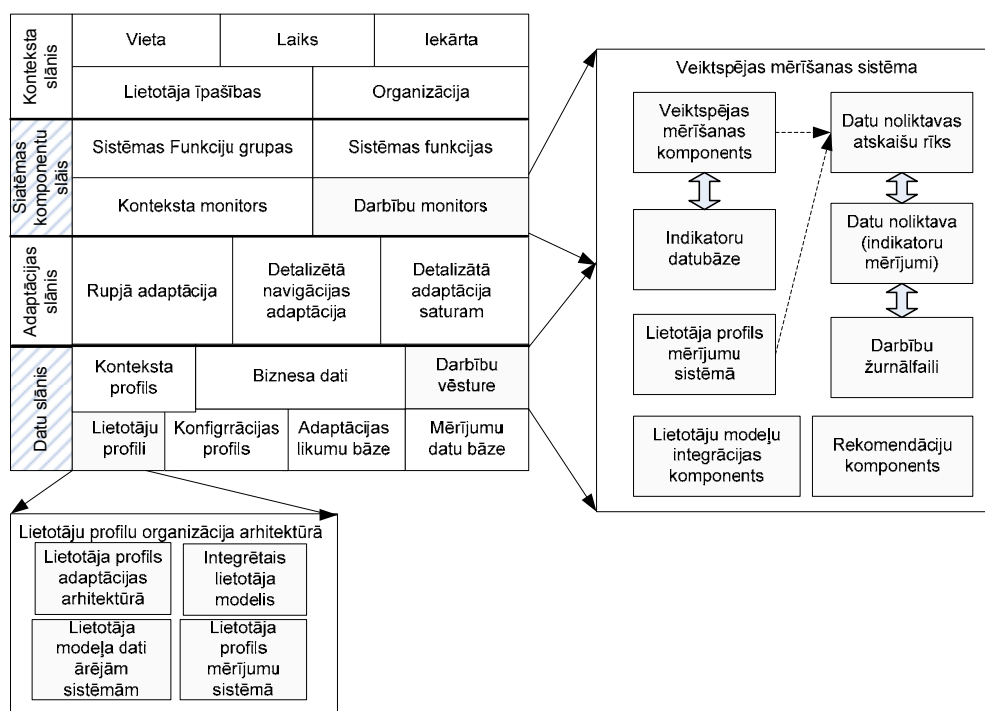
Šajā nodaļā dotas gan biznesa procesu mērīšanas jēdzienu definīcijas, gan arī veiktspējas mērīšanas jēdzieni. Nodaļā analizēta datu noliktava kā iespējamais realizācijas veids veiktspējas mērīšanas sistēmai. Nodaļa apraksta indikatoru dzīves ciklu, tiek definēts indikatoru pierakstīšanas modelis. Nodaļā definētais indikatoru dzīves cikls tiek izmantotas par pamatu, piedāvājot jaunu organizācijas veiktspējas mērīšanas ietvaru. Piedāvāta arī divu līmeņu mērīšanas metode biznesa procesu mērīšanai, kas izmantojama kopā ar piedāvāto veiktspējas mērīšanas ietvaru. Nodaļas rezultāti publicēti 2 rakstos par veiktspējas mērījumu ietvaru (Niedritis et al., 2012) un (Niedritis et al., 2011a) un 3 rakstos par mērīšanas pieeju (Medvedis et al., 2008), (Medvedis et al., 2008b), (Niedrite et al., 2007a).

### 5.2. Veiktspējas mērīšanas sistēmas vieta DNATIS arhitektūrā

Promocijas darbā tika aprakstīta DNATIS arhitektūra (skat. 4. attēlu). Arhitektūras shēma, kas 1) detalizētāk parāda komponentus lietotāju darbību monitoram un darbību vēstures datu bāzei un 2) precīzē lietotāju profilu un lietotāja modeļa vietu DNATIS arhitektūrā, dota 30. attēlā.

DNATIS arhitektūrā kā komponenti ietilpst arī lietotāju darbību monitors un darbību vēstures datu bāze, kas kopā veido veiktspējas mērīšanas sistēmu, kuras komponenti redzami 30. attēlā. Tā balstīta uz datu noliktavas datu bāzi mērījumu vērtību glabāšanai, tā izmanto gatavu datu noliktavu atskaišu rīku, un lietotāju aprakstīšanai tai ir savs lietotāja profils.

Šajā nodaļā tiks piedāvāts veiktspējas mērīšanas ietvars, saskaņā ar ko DNATIS arhitektūrā paredzēta veiktspējas mērīšanas sistēma.



30. attēls. Veiktspējas mērīšanas sistēmas vieta DNATIS arhitektūrā

DNATIS arhitektūrā procesu mērīšanas sistēma ir iekļauta tāpēc, ka uzturot sarežģītu sistēmu, kurai tiek veidotas daudzas adaptētas instances dažādu organizāciju un dažādu šo organizāciju darbinieku vajadzībām, nepieciešams mērīt un analizēt dažādus aspektus:

- sistēmas lietošanas raksturojums organizācijā – ko tieši katrs lieto, aprakstot kvantitatīvi organizācijas līmenī izmantotās funkcijas, izmantošanas biežumu, lietotāju skaitu,
- sistēmas lietošanas paradumu analīze lietotājiem- analizējot detalizēti, tieši kādi dati tiek izmantoti biznesa funkciju veikšanai.

Mērīšanas mērķi organizācijā var būt dažādi. Parasti mēra biznesa procesu rezultātus, iepriekš nedefinējot veikspējas rādītājus, lai varētu sekot līdzi biznesa attīstībai vai noskaidrot nepieciešamību pēc biznesa procesu pārbūves. DNATIS arhitektūras ietvaros, mērīšanas mērķi ir sekojoši:

- finansējuma noteikšanai TIS izmantošanai,
- TIS attīstīšanas prioritāšu noteikšanai,
- TIS tehnisko parametru optimizēšanai,
- personalizācijas un rekomendāciju vajadzībām.

Uz datu noliktavu tehnoloģijām balstīta procesu mērīšanas sistēma tiek veidota gadījumā, ja organizācijā jau eksistē iepriekš datu noliktava, kas izveidota organizācijas rezultātu, piemēram, finansiālu rādītāju analīzei, bet vajadzība mērīt procesu aspektus tiek apzināta vēlāk.

### 5.3. Jēdzienu definīcijas

Ilgākā laika posmā ir nepieciešams saprast ne tikai esošo situāciju organizācijā un pārbūvēt biznesa procesu, bet arī turpināt uzlabot biznesa procesu, balstoties uz vispusīgu organizācijas veikspējas mērīšanu.

Efektīva biznesa procesu organizēšana nodrošina iestādes mērķu sasniegšanu. Veikspējas mērīšanas laikā mērījumu rezultāti jāsalīdzina ar mērķa vērtībām, lai varētu pieņemt lēmumus, vai mērķi tiek sasniegti. Organizācijas izmanto veikspējas mērījumus, lai salīdzinātu ikdienas aktivitātes ar stratēģiskajiem mērķiem (Parmenter, 2010). Piemērotu mērījumu loma ir nenovērtējama. Tā, piemēram, Haringtons (1991) apgalvoja, ka “Mērījumi ir galvenais. Ja nav iespējams mērīt, tad nav iespējams kontrolēt. Ja nav iespējams kontrolēt, tad nav iespējams pārvaldīt. Ja nevar pārvaldīt, tad nevar uzlabot”.

Nozīmīgs aspekts tātad ir, kā izvēlēties piemērotus mērījumus, un kā definēt piemērotu mērījumu ietvaru. Veikspējas mērījumi ir jāveic no dažādām perspektīvām.

Uzņēmumiem ir jāizmanto veikspējas mērījumu sistēmas, ja tās grib būt veiksmīgas konkurences cīņā ar citām kompānijām. Mērījumu sistēmas ir jāizveido atbilstoši kompānijas stratēģijām un atbilstoši kādam no pārvaldības modeļiem, piemēram, *Balanced Scorecard* (Kaplan, Norton, 1996). Kā realizācijas veids veikspējas mērījumu sistēmai var tikt izmantotas arī datu noliktavas, tālāk tiks apskatīti arī esošie pētījumi šajā jomā.

Veikspējas indikatoru definēšana jābalsta uz uzņēmuma stratēģiju un šim nolūkam jāizmanto piemērota metode. Indikatori var tikt definēti dažādās formalizācijas pakāpēs, daži piedāvājumi indikatoru formālam pierakstam eksistē un ir jau aprakstīti (Frank et al., 2008), (Popova, Treur, 2007).

Viens no piedāvājumiem (Frank et al., 2008) satur formālu valodu mērķu modelēšanai, kas balstīta uz veikspējas indikatoriem, lai varētu tikt kontrolēta mērķu sasniegšana un tikt novērtēta organizācijas veikspēja. Savukārt cita raksta autori (Popova, Treur, 2007) piedāvā indikatoru definēšanas formālu valodu, ieviešot indikatoru kategorijas, predikātus un funkcijas, var tikt definētas arī saites starp indikatoriem.

Var lūkoties uz veikspējas mērījumiem un indikatoriem gan no mērījumu teorijas viedokļa, gan no pārvaldības modeļu viedokļa, lai noskaidrotu piemērotas definīcijas.

### **5.3.1. Mērīšanas jēdzieni no mērījumu teorija viedokļa**

Mērīšana saskaņā ar Fenton un Whitty (1995) ir: “Mērīšana ir process, kurā skaitliskas vērtības vai simboli tiek piešķirtas entītijai atribūtiem reālajā pasaulē tādā veidā, kas raksturo šos atribūtus pēc skaidri definētiem likumiem.”

Eksistē dažādu autoru izstrādātas ontoloģijas un metodes (Munson, 2003), (McGarry et al., 2001), (Garcia et al., 2006) indikatoru definēšanai atsevišķās industrijas nozarēs, piemēram, programmatūras procesu mērīšanā. Biznesa procesu mērīšanā jēdziens „mērīšana” tiek skaidrots tā pielietojuma kontekstā, piemēram, biznesa procesu pārbūvē vai veikspējas mērīšanā, tomēr biznesa procesu mērīšanas kontekstā pats mērīšanas jēdziens parasti netiek apspriests.

Tālāk dotas definīcijas galvenajiem ar mērīšanu saistītajiem jēdzieniem, lai parādītu, kā indikatoru tiek iegūti atbilstoši iepriekš minētajām mērījumu teorijām (McGarry et al., 2001).

Pirms mērīšanas, ir jāeksistē informācijas vajadzībai, kas saistīta ar lēmumu pieņemšanu. Mērīšanā var aplūkot sekojošu hierarhisku struktūru Atribūts → Bāzes mērījums → Atvasināts mērījums → Indikators → Informācija. Šajā hierarhiskajā struktūrā jēdzieni tiek skaidroti sekojoši:

- *Atribūts* ir nosakāma entītijas īpašība,
- *Bāzes mērījums* ir viena atribūta mērījums. Bāzes mērījuma vērtība tiek noteikta ar mērījumu metodes palīdzību,
- *Atvasināts mērījums* tiek definēts kā funkcija (algoritms vai aprēķini) no diviem vai vairākiem bāzes mērījumiem,
- *Indikator* ir mērījums, kas nodrošina novērtējumu noteiktiem atribūtiem, kas iegūti atbilstoši analīzes modelim. Analīzes modelis ir algoritms vai aprēķini, kas saista divus vai vairākus bāzes mērījumus un/vai atvasinātus mērījumus ar atbilstošiem lēmumu kritērijiem. Indikatori kalpo par pamatu mērījumu analīzei un lēmumu pieņemšanai,
- *Lēmumu kritēriji* ir skaitliski mērķi, ko lieto, lai novērtētu, vai nepieciešams veikt kādas darbības, vai varbūt nepieciešama tālāka procesu izpēte. Tie var būt balstīti uz pieredzi un izpratni par gaidāmo rezultātu, var būt balstīti uz vērtībām, kas izrēķinātas no vēsturiskiem datiem.

### **5.3.2. Mērīšanas jēdzieni no pārvaldības modeļu viedokļa**

Piemērotu veikspējas mērījumu definēšana ir jāveic sistēmiskā veidā, balstītu uz kādu no metodēm. Reālās pasaules pieredze liecina, ka uzņēmumi lieto nepareizus mērījumus (Parmenter, 2010), daudzi no kuriem tiek nepareizi uzskatīti par Galvenajiem Veikspējas Indikatoriem (*Key Performance Indicators* - KPI). Bieži vien nav skaidrs, kas ir un kas nav KPI, kā veiksmes faktori ir saistīti ar KPI un uzņēmuma stratēģiju.

Pirms uzsākt diskusiju par formalizāciju un kā izvēlēties piemērotus veikspējas mērījumus, jādefinē jēdzieni, kas saistīti ar veikspējas mērīšanu.

Veiksmes faktori (*Success factors*-SF) (Parmenter, 2010) parasti ir apmēram 30 aspekti uzņēmuma veikspējai, kas ir svarīgi, lai uzņēmums labi darbotos jebkurā tam atbilstošā sektorā vai industrijā. Kritiskie veiksmes faktori (*Critical success factors* - CSFs) (Parmenter, 2010) ir uzņēmuma veikspējas aspekti, kas nosaka notiekošā veselīgumu, vitalitāti un labklājību. Parasti CSF sarakstā vienam uzņēmumam tiek iekļauti 5-8 CSF.

Veiktspējas mērījumi (Parmenter, 2010) attiecas uz indikatoriem, ko izmanto vadība, lai mērītu, ziņotu un uzlabotu uzņēmuma veiktspēju. Veiktspējas mērījumi tiek klasificēti kā galvenie rezultātu indikatori, rezultātu indikatori, veiktspējas indikatori, vai kā galvenie veiktspējas indikatori (Parmenter, 2010):

- galvenie rezultātu indikatori (*Key result indicators* - KRIs) reprezentē kopsavilkumus par daudzām aktivitātēm saistībā ar kādu no uzņēmuma CSF, bet tie nepalīdz saprast, kas ir jāuzlabo uzņēmumā. KRIs var būt finansiāli un nefinansiāli,
- rezultātu indikatori (*Result indicators* - RIs) reprezentē dažas aktivitātes saistībā ar kādu no CSF vai SF, parasti ir vairāk nekā vienas aktivitātes rezultāts, bet līdzīgi kā KRIs, tie nedod informāciju par to, ko un kā vajadzētu uzlabot. Visi finansiālie veiktspējas rādītāji ir RIs,
- veiktspējas indikatori (*Performance indicators* - PIs) „saka, ko vajag uzlabot” (Parmenter, 2010), jo PIs mērījumi attiecas uz kādu atsevišķu aktivitāti, PIs ir nefinansiāli,
- galvenie veiktspējas rādītāji (*Key Performance Indicators* - KPIs) „saka, ko vajag darīt, lai veiktspēju uzlabotu ievērojami”, tie reprezentē mērījumu kopu, kas fokusējas uz uzņēmuma veiktspējas aspektiem, kas ir viskritiskākie uzņēmuma esošai un nākotnes veiksmi.

Izvēlēto mērījumu kopu parasti ietekmē pārvaldības modeļi uzņēmumā, piemēram, *Balanced Scorecard* (BSC) (Kaplan, Norton, 1996), un mērījumu perspektīvas, kas definētas šī modeļa ietvaros. BSC modelī ir definētas 4 perspektīvas – Finanšu, Klienta, Iekšējo procesu un Apmācības un izaugsmes perspektīvas. Parmenters (2010) piedāvā vēl lielāku perspektīvu skaitu, piemēram, BSC definētajām perspektīvām, pievienojot vēl divas – Vide/Sabiedrība un Darbinieku apmierinātība.

#### 5.4. Datu noliktavas kā veiktspējas mērījumu sistēmas

Uzņēmumi lieto mērījumu sistēmas, lai novērtētu savu veiktspēju. Pārvaldības modeļi, piemēram, *Balanced Scorecard*, ir jāizmanto, lai šādas sistēmas izstrādātu, lai varētu izvēlēties un precīzi definēt piemērotus mērījumus. Atbilstoši pārvaldības modelim, uzņēmumi var tikt novērtēti attiecībā pret dažādām perspektīvām, piemēram, finanšu, klientu, iekšējo procesu un citām. Jāmodelē un jādokumentē arī veiktspējas indikatoru dažādi aspekti, piemēram, saistība ar veiksmes faktoriem, atskaite, atbildība un citi.

Priekšrocība, ka datu noliktava tiek izmantota veiktspējas mērījumu sistēmas implementācijai, ir iespēja izmantot infrastruktūru uzņēmumā jau esošai datu noliktavai. Datu noliktavas tradicionāli atspoguļo klienta un finanšu perspektīvai atbilstošus indikatorus uzņēmumā. Parasti tradicionālas datu noliktavas gadījumā pārvaldības teorijas netiek izmantotas. Līdz ar to parasti iekšējo biznesa procesu, apmācības un izaugsmes, un citas (piemēram, vides/sabiedrības) perspektīvas netiek pārklātas.

Atsevišķos projektos datu noliktavās tiek ietverti arī iekšējie biznesa procesi (List, Machaczek, 2004), pieredze šādos projektos ir aprakstīta arī vēl citos rakstos (Jarke et al., 2000), (Kueng et al., 2001).

Datu noliktavas kā risinājums biznesa datu glabāšanai un analīzei ir aprakstīts virknē pētījumu: (Muehlen, 2001), (Kueng et al., 2001). Tālāk dots kopsavilkums specifiskām datu noliktavas īpašībām, ja tās tiek izmantotas veiktspējas mērījumu sistēmu implementācijai:

- atšķirīgi un jauni datu avoti. Darba plūsmu izpildes žurnāli tiek integrēti ar citiem datu avotiem. Atsevišķos gadījumos apskata datu noliktavas tikai šo žurnālfailu datu glabāšanai un analīzei (Bonifati et al., 2001). Tiek apskatīti trīs veidu dati, kas saistīti ar darba plūsmas sistēmām: lietojumprogrammu dati, darba plūsmu specifiski dati, un

iekšēji dati darba plūsmu sistēmās. Darba plūsmu specifiskie dati nosaka, piemēram, viena vai otra ceļa izvēli darba plūsmas izpildes laikā, bieži tie pat netiek glabāti žurnālfailā. Darba plūsmu sistēmu iekšējie dati ir dati, kas glabājas žurnālfailā un raksturo darba plūsmu izpildi (piemēram, izpildes laiks, izpildītājs u.c.),

- specifiskas datu analīzes pieejas, kas izriet no jauniem datu avotiem un jauniem analīzes mērķiem,
- atšķirīgs datu analīzes mērķis, līmenis, izpildītājs. Analīzei, piemēram, var būt tehniski mērķi – testēt darba plūsmu izpildes sistēmas ieviešanas laikā, var mērīt, piemēram, sistēmas reakcijas laiku vai noslodzi. Datu analīzi var veikt dažādos līmeņos un to var veikt dažādi lietotāji; datu analīzē jānodrošina gan atsevišķu konkrētu lietotāju personalizētas vajadzības, gan arī visas kompānijas kopējās vajadzības. Datu analīzes laikā individuālie lietotāji var būt dažādās lomās: procesu īpašnieki, procesu izpildītāji, vadītāji, atbalsta personāls, datu analītiķi,
- atšķirīgs datu analīzes laiks. Datu analīzei var būt dažāds laika periods – piemēram, īstermiņa datu analīze procesiem, kas nozīmē procesu monitorēšanu izpildes laikā, vai ilgtermiņa analīzi (vai procesa kontrole), kas nozīmē analīzi pēc procesa izpildes beigām,
- piemērots datu noliktavas modelis.

Šīs īpašības var tikt kombinētas, piemēram, procesu monitorēšana tehniskiem mērķiem var tikt lietota lai novērtētu lietotāju skaitu un paredzamo noslodzi. Datu analīzei arī var būt dažādas analīzes perspektīvas atkarībā no veikspējas mērījumu ietvara, kas tiek izmantots, piemēram, klienta, finansu u.c.

Iepriekš minētās īpašības var noteikt datu elementus, kas jāietver datu noliktavas modelī, kas paredzēts veikspējas mērījumu datu glabāšanai. Būtiski aspekti, kas jāizvērtē ir kādi ir veikspējas mērījumi, kas būs jāglabā datu noliktavā, to prioritātes, kā arī nepieciešamie datu avoti, lai šos mērījumus aprēķinātu.

Veikspējas mērījumu sistēmu implementācijas, izmantojot kā līdzekli datu noliktavu, ir aprakstītas vairākos darbos, ir piedāvāti dažādi specializēti risinājumi problēmām, kas rodas, kad datu noliktavas izstrādā veikspējas mērījumiem, piemēram, dažādi specializēti daudzdimensiju modeļi.

Procesu datu noliktava ir definēta (Jarke et al., 2000) „kā datu noliktavu, kas glabā datus par ražošanas procesu izpildi un rezultātiem pieredzes iegūšanai un atkārtotai izmantošanai”.

Veikspējas pārvaldības sistēmas jēdziens ir definēts (Kueng et al., 2001) kā „sistēma, kas centralizēti glabā un pārvalda visus procesu izpildes datus, gan finansiālos, gan nefinansiālos”. Tiek piedāvāta metode, lai varētu uzbūvēt veikspējas pārvaldības sistēmas modeli. Veikspējas indikatoru definēšanai vispirms tiek veikta organizācijas un tās mērķu analīze, nosaka ieinteresētās personas, biznesa procesus un biznesa procesu mērķus. Šo informāciju izmanto, lai noskaidrotu atbilstošus izpildes rādītājus. Veikspējas mērījumu sistēmas satur mērījumu vērtības un papildus informāciju par kompānijas struktūru, biznesa procesiem, mērķiem.

Pētījuma (List, Machaczek, 2004) autori piedāvā korporatīvo izpildes mērīšanas sistēmu (*Corporate Performance Measurement System - CPMS*), tajā procesu izpildes dati tiek integrēti uzņēmuma datu noliktavā. Darba plūsmas sistēmas žurnāla faili tiek izmantoti kā datu avoti. CPMS datu noliktavas projektējuma izveidei tiek piedāvāta īpaša metode: biznesa procesu mērķi tiek iegūti no uzņēmuma mērķiem, tiek definēti jautājumi par mērķu sasniegšanu, tiek dokumentēti atbilstoši indikatori un datu avoti šo indikatoru aprēķināšanai. CPMS tiek izstrādāta kā eksistējošas datu noliktavas modeļa sastāvdaļa.



Var izdalīt atsevišķu kategoriju datu noliktavām, kas paredzētas veikspējas mērīšanai, kur galvenais mērķis ir glabāt un analizēt procesu izpildes modeļus. Šāda tipa datu noliktavas tad var saukt par Procesu datu noliktavām, tomēr dažādi autori šo jēdzienu interpretē atšķirīgi.

Darba plūsmas datu noliktava (Bonifati et al., 2001) reprezentē universālu modeli, kas orientēts tieši darba plūsmu sistēmu datu glabāšanai. Darba plūsmas datu noliktavas autori pamato, vai un kad datu noliktava ir piemērots risinājums procesu izpildes žurnālfailu datu glabāšanai un analīzei. Eksistējošie analīzes rīki piedāvā ierobežotu funkcionalitāti, tipiski mērījumi ir izpildīto procesu instanču skaits laika periodā, vidējais izpildes ilgums. Savukārt piedāvātais vispārīgais datu noliktavas modelis paredzēts savstarpēji saistītu faktu analīzei, piemēram, aktivitāte var tikt izpildīta noteiktā darba plūsmā kā noteikta ceļa sastāvdaļa. Modelim ir arī dimensija, kas definē tipiskus šablonus darba plūsmas izpildei pagātnē, lai varētu klasificēt analizējamo darba plūsmu attiecībā pret šiem šabloniem. Vēl būtisks ir faktors, ka piedāvātais daudzdimensiju modelis ietver arī papildus faktu tabulu „Procesu dati”, kas reprezentē biznesa datus, kas apstrādāti biznesa procesa izpildes laikā.

No vienas puses nav pietiekami analizēt atsevišķi tikai vienu no aspektiem, piemēram, procesu izpildi, izmantojot darba plūsmu sistēmu žurnālfailus. Tas nepalīdz izprast patieso stāvokli uzņēmumā, jo, piemēram, ja neko nezina par procesu mainītajiem biznesa datiem, tad nevar novērtēt procesa rezultātu. No otras puses, kad tiek analizēti tikai biznesa specifiskie dati, ir iespējams iegūt vērtējumu stāvoklim biznesā, bet nav iespējams noskaidrot cēloņus, kāpēc stāvoklis ir tāds, kā to rāda indikatori datu analīzē. Darba plūsmu datu analīze var palīdzēt izprast darba plūsmu izpildes tipiskos ceļus, un vajadzības gadījumā, kad to rāda indikatori, iespējams šo darba plūsmu pārbūvēt.

Bieži datu analīzei netiek izmantoti visi iespējamie datu avoti, piemēram, dažādi žurnālfaili, jo to integrēšana ar citiem datu avotiem ir sarežģīta, kā arī šos avotus bieži vien nenovērtē kā noderīgus datu avotus. Ierobežots datu avotu skaits ierobežo brīvu indikatoru definēšanu.

Datu noliktava kā procesu mērīšanas sistēma var tikt realizēta un var būt piemērota, galvenokārt, ja ir paredzēta ilgtermiņa analīze, kad ir nepieciešama visu datu avotu integrācija un ja daļa no nepieciešamajiem datiem jau tiek uzkrāta uzņēmumā eksistējošā datu noliktavā. Datu noliktava ir piemērots risinājums arī gadījumā, kad analīze ir nepieciešama uzņēmuma līmenī.

## 5.5. Indikatoru dzīves cikls un indikatoru aprakstīšanas aspekti

Iepriekš apskatītie jēdzieni par veikspējas mērīšanu, un indikatoru īpašības jāņem vērā, lai maksimāli vispusīgi varētu noskaidrot un aprakstīt indikatoru būtību.

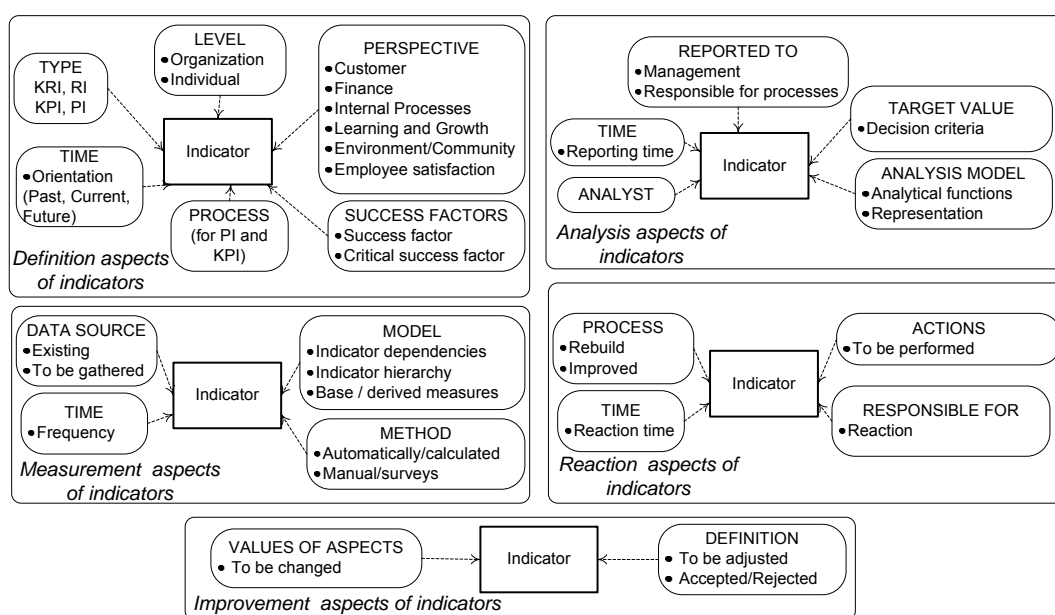
Promocijas darbā tiek piedāvāts indikatoru dzīves cikls, kas sastāv no 5 soļiem – indikatoru definēšanas, mērīšanas, analīzes, reakcijas un uzlabošanas soļi:

- **1. solis.** Indikatoru definēšanas solis apraksta galvenokārt dažādas indikatoru īpašības, kas palīdz saprast, kāpēc viens vai otrs indikators tiek ieviests,
- **2. solis.** Mērīšanas solis reprezentē procesu, kad indikatori iegūst savas vērtības,
- **3. solis.** Analīzes solis reprezentē procesu, kad indikatoru vērtības tiek izmantotas, lai pieņemtu lēmumus,
- **4. solis.** Reakcijas solis reprezentē procesu, kad lēmumi, kas tiek pieņemti iepriekšējā solī, tiek īstenoti,
- **5. solis.** Uzlabošanas solis atbalsta indikatoru definīciju novērtēšanu un veikspējas mērīšanas dažādu aspektu vērtību pārskatīšanu un mainīšanu indikatoriem (vajadzības gadījumā).

Atbilstoši piecu soļu indikatoru dzīves ciklam, katrs solis var tikt reprezentēts ar dažādiem aspektiem (skat. 31. attēlu). Indikatoru aspekti būtībā ir indikatoru īpašības, kas izriet no iepriekš aprakstītās veikspējas mērīšanas specifikas.

Tālāk doti indikatoru aspekti, kas iegūti analizējot rakstus (Parmenter, 2010), (Muehlen, 2001):

- Perspektīva. Šis aspekts jau iepriekš tika apskatīts, ir jāpiezīmē, ka ne visi indikatoru tipi pārklāj visas perspektīvas, piemēram, finansu perspektīva ir saistīta ar KRIs un RIs, bet ne ar PIs,
- Laiks. Jāņem vērā arī laika aspekta apakštīpi – mērījumu periodi, kad tiek noteiktas indikatoru vērtības, vai arī, piemēram, atskaišu veidošanas periodi, kas nosaka vēsturisko datu apjomu, kas jāietver atskaitēs. Kaut arī, KRIs tipiski tiek mērīti reizi mēnesī, atskaites periodi var ietvert garākus laika periodus, piemēram, gadu. KPIs tiek mērīti biežāk, piemēram, katru dienu vai nedēļu,
- Atbildība. Personas, kas atbildīgas par dažādiem veikspējas mērījumu tiem var būt dažādos līmeņos, sākot ar uzņēmuma vadību, līdz par individuālam darbiniekam, kas, gadījumā, kad tiek mērīti PIs, zina visas vajadzīgās darbības, kas jāveic, lai kaut ko uzlabotu, un uzreiz to arī ir spējīgs izdarīt,
- Aktivitātes. RIs nevar būt piesaistīts tikai vienai atsevišķai aktivitātei, savukārt, PIs ir cieši saistīti ar vienu konkrētu aktivitāti,
- Veiksmes faktori. Veikspējas mērījumi ietekmē vienu vai vairākus CSF vai SF atkarībā no veikspējas mērījumu tipa, piemēram, KPIs ietekmē vairāk nekā vienu CSF /SF,
- Atskaites. Rezultāti dažādiem veikspējas mērījumu tiem var tikt ziņoti dažādos līmeņos - sākot ar uzņēmuma vadību, līdz par individuālam darbiniekam. Piemēram, KPIs gadījumā rezultāti tiek ziņoti augstākā līmeņa vadībai, tie palīdz saprast, kādas ir nepieciešamās veicamās darbības, un tad atbildība var tikt piešķirta individuālā līmenī. Tādējādi atskaites un atbildības aspekti ne vienmēr ir ar vienu un to pašu nozīmi.



31. attēls. Piecas indikatoru aspektu grupas atbilstoši indikatoru dzīves ciklam

Promocijas darbā piedāvātajā indikatoru dzīves ciklā indikatoru aspekti tiek grupēti atbilstoši šiem pieciem soļiem. Šajā modelī ir ieviesti vēl papildus aspekti, piemēram, Līmenis (*Level*). Šis aspekts nepieciešams, lai varētu definēt, vai indikators vajadzīgs organizācijas (vai citas struktūrvienības, vai grupas) līmeņa vai individuāla līmeņa mērījumiem. Ja runā par procesa aspektu, tad jāpiemin, ka reakcijas aprakstīšanai, ir nepieciešams noskaidrot arī mērīšanas mērķus.

Promocijas darbā piedāvātais mērīšanas ietvars ar indikatoriem rada divus tālākus jautājumus: (i) kāda ir vispiemērotākā implementācija, lai atbalstītu aprakstīto dzīves ciklu, un (ii) kā indikatorus var formalizēt, lai maksimāli skaidri aprakstītu, kas, kāpēc un kā tiek mērīts.

Iepriekš aprakstīto veikspējas mērījumu sistēmu analīze parāda, ka datu noliktavu tehnoloģijas var tikt izvēlētas kā piedāvātā mērījumu ietvara realizācijas līdzekļi (skat. 5.4 apakšnodaļu). Tālāk šajā nodaļā tiks apskatīta indikatoru definīciju formalizācija.

### **5.5.1. Indikatoru definēšanas formālais modelis**

Programmatūras prasības parasti tiek aprakstītas ar dažāda līmeņa formalizācijas pakāpi (Videira, Silva, 2005).

Tā kā datu noliktavu tehnoloģijas izvēlētas kā piedāvātā mērījumu ietvara realizācijas līdzeklis, tad indikatoru formalizācijai tika izvēlēta un tālāk attīstīta prasību definēšanas metode datu noliktavu sistēmām (Kozmina et al., 2008). Pieeja balstās uz nepieciešamās informācijas iegūšanu no biznesa prasībām, kas definētas dabiskā valodā, tā padarot biznesa prasību definīcijas precīzākas un formālākas (Kozmina et al., 2008).

Šī pieeja promocijas darbā adaptēta indikatoru definēšanai mērījumu ietvara vajadzībām.

Izstrādājamās informācijas sistēmas veids zināmā mērā ietekmē veidu, kā tiek formulēti teikumi, kas izsaka prasības. Sākotnēji tika pieņemts, ka prasības datu noliktavai un īpaši informācijas prasības ir ar līdzīgu struktūru vai šablonu. Šis pieņēmums tika balstīts uz novērojumiem praktisku datu noliktavu sistēmu izstrādē. Arī tipiskais veids, kā vēlāk datu noliktavas dati tiek analizēti, izmantojot tipiskas OLAP (*On-Line Analytical Processing*) operācijas ar tām atbilstošiem datu noliktavām tipiskiem SELECT teikumiem, apstiprina šo pieņēmumu.

Prasības ir reprezentētas ar teikumiem (dabiskās valodas vārdu apakškopu), kas tiek izmantoti, lai aprakstītu kāda darbība(-as) sistēmai jāveic un kādu objektu(-us) tā ietekmēs. Jautājums ir, vai darbības(-u) un objekta(-u) apraksts var būt formalizācijas priekšmets, t.i. kā teikuma sastāvdaļas ir strukturētas un vai ir saskatāmi kopīgi šabloni. Kozmina et al. (2008) piedāvāja metamodeli, kas apraksta kopīgu struktūru datu noliktavas informācijas prasībām. Veikspējas mērījumu gadījumā promocijas darbā tiek piedāvāts pārskatīts un papildināts (Kozmina et al., 2008) metamodelis, lai varētu izteikt indikatorus.

Šo līdzīgo pieeju var izmantot vairāku iemeslu dēļ. No vienas puses, indikatori ir datu analīzes centrā mērījumu procesā, saskaņā ar mērījumu jēdzieniem. No otras puses, datu noliktavu modeļi tiek būvēti, lai atbilstu datu noliktavas sistēmas informācijas vajadzībām. Tādēļ indikatoru formalizācija var tikt balstīta uz datu noliktavas daudzdimensiju modeļa elementu semantiku, piemēram, uz atšķirībām starp Kvalificējošiem datiem un Kvantitatīviem datiem.

### **5.5.2. Indikatoru pārformulēšanas principi**

Piedāvātais modelis tika balstīts uz struktūras novērtēšanu teikumiem, kas formulē veikspējas indikatorus no veikspējas mērījumu indikatoru datubāzes, kas atrodams

Parmenter (2010) grāmatas pielikumā. Pēc apmēram 330 dažādu gatavu indikatoru izpētes, kas attiecas gan uz klienta fokusu (*customer focus* - CF), gan vidi un sabiedrību (*environment & community* - EC), gan darbinieku apmierinātību (*employee satisfaction* - ES), gan finansēm (*finance* - F), gan iekšējiem procesiem (*internal process* – IP), gan apmācību un izaugsmi (*learning & growth* - LG), promocijas darba ietvaros tika formulēti principi, lai pārformulētu dabīgā valodā izteiktas prasības pēc vienota šablona.

- Indikatora komponentu, ko ir paredzēts mērīt, var uzlūkot par kopskaitu visiem šī komponenta notikumiem (occurrences) - kad komponents ir noticis, bijis spēkā.

Piemēram: gadījumā, ja paredzēts mērīt zvanu skaitu, tad indikatora komponents „zvani” (*calls*) tiek pārformulēts par “*count (call occurrence)*”, kur „*count*” ir indikatora semantikai atbilstoša agregātfunkcija - skaitīt.

- Ja kādu indikatora komponentu ir paredzēts parādīt detalizētāk, tad atbilstošā prasībā var izmantot funkciju, kas apzīmē attēlošanas darbību „show”.

Piemēram: gadījumā, kad interesē apskatīt vairāk informāciju par konkrētu darbinieku, tad indikatora komponents „darbinieki” tiek pārformulēts par “show (darbinieki)”.

- Ja indikators satur komponentu, kā, piemēram, „saraksts” vai „instances”(“*listing of*”, “*list of*”, „instances of”), tad pārformulējot atbilstošo prasību tiek izmantota funkcija, kas apzīmē attēlošanas darbību - „show” (parādīt).

Piemēram, gadījumā, kad interesē apskatīt klientu sarakstu, tad indikatora komponents „klientu saraksts” tiek pārformulēts par “show (klienti)”.

- Ja indikators satur tādu komponentu kā skaits („*number of*”), tad pārformulējot atbilstošo prasību tiek izmantota agregātfunkcija *count*.

Piemēram, indikatora komponents „vizīšu skaits”(“*number of visits*) tiek pārformulēts par „*count (visit occurrence)*”

- Ja indikators satur tādas komponentus kā “cost of” (cena), “value of” (vērtība), “expense” (izmaksas), “total expense”(kopējās izmaksas), “income” (ienākumi), “total income” (kopējie ienākumi), “revenue” (peļņa), “investment” (ieguldījumi), utt., vai valūtas nosaukumu pašā indikatora sākumā, tad pārformulējot prasību, tiek izmantota agregātfunkcija Sum.

Piemēram, „Dolāru noguldījums” var tikt pārformulēts par Sum (dolāri), „kopējie ienākumi” – par Sum (ienākumi).

- Ja indikators satur tādu komponentu kā „average” (vidējais), tad atbilstošā prasība tiek pārformulēta, un tiek izmantota atbilstošā agregātfunkcija Avg.

Piemēram, „Vidējais reakcijas laiks” - tad prasība tiek pārformulēta, aizstājot atbilstoši ar “avg (reakcijas laiks)”.

- Ja indikators satur tādas komponentus kā “%”, “percent”, “percentage” (procenti) vai “ratio” (pakāpe), tad procenti tiek aizstāti ar dalīšanu (konkrēto lielumu daļa ar kopējo lielumu)

Piemēram, indikators „IT izmaksas kā % no kopējām administratīvām izmaksām” var tikt pārformulēts par sum (IT izmaksas) / sum (izmaksas)

Protams, minētie principi ir paredzēti, lai tiktu izmantoti, ņemot vērā katra indikatora kontekstu. Indikatori ir jāanalizē, lai saprastu, vai jāpielieto agregātfunkcijas vai nē, un jāizvēlas kontekstam piemērotu agregātfunkciju, ja nepieciešams. Daži piemēri šādiem indikatoriem – 1) pabeigtās iniciatīvas, 2) kandidāti, 3) patērētā enerģija. Potenciāli lietojamās funkcijas pārformulējot – 1) show vai count – atkarīgs no indikatora konteksta, 2) show vai count, 3) Sum.

### 5.5.3. Prasību šablonu aprakstīšana

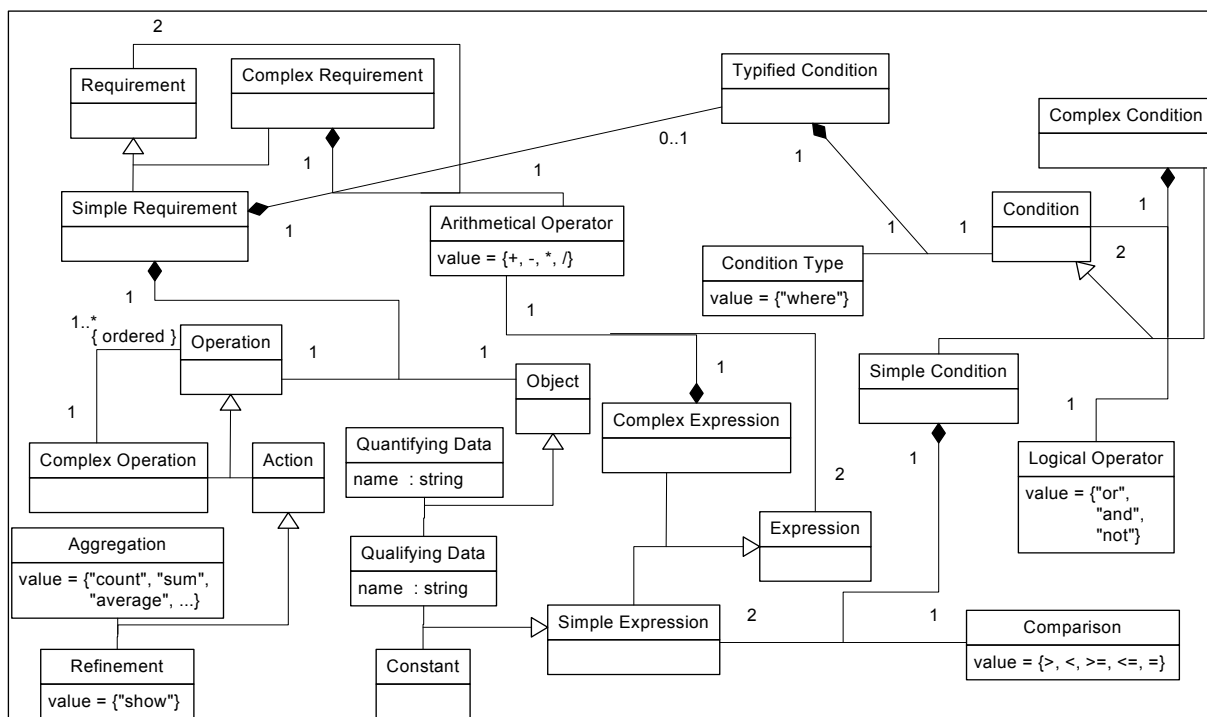
Visiem indikatoriem ir līdzīga struktūra, šim nolūkam ir iespējams noteikt šablonu, lai indikatorus varētu pārrakstīt formāli. Pieeja, kas tiek lietota indikatoru formālai pārrakstīšanai izmanto modeli, kas aprakstīta ar UML 2.0. klašu diagrammu (skat. 32. attēlu).

Biznesa prasība ir aprakstīta ar klasi *Requirement* (prasība), kas tālāk iedalās *Simple* (vienkārša) un *Complex Requirement* (kompleksa prasība). Kompleksa prasība sastāv no divām vai vairākām vienkāršām prasībām, kas saistītas ar aritmētisku operāciju (*Arithmetical Operator*). Vienkārša prasība sastāv no darbības vārda, kas reprezentē komandu - Operāciju (*Operation*), kas jāveic attiecībā pret objektu (*Object*) un nulles vai viena nosacījuma ar tipu (*Typified Condition*).

Datu noliktavā ir divu veidu dati Kvantitatīvie dati (*Quantifying Data*), kas reprezentē mērījumus un Kvalificējošie dati (*Qualifying Data*), kas reprezentē mērījumus raksturojošas īpašības. Objekts ir vai nu Kvantitatīvo datu vai Kvalificējošo datu instance, atkarībā no konkrētās modelējamās prasības.

Jēdziens operācija apraksta Darbības (*Action*) veidu, kas jāveic. Ja kāda veida darbība (vai vairāku veidu darbības) ir jāveic vairāk kā vienu reizi, tad to sauc par Saliktu operāciju (*Complex Operation*). Modelī tiek piedāvātas divu veidu darbības: Agregācija (*Aggregation*), kas tiek izmantota, lai veiktu tādus aprēķinus kā summēšana, skaitīšana u.c., un Precizēšana (*Refinement*), darbība, kas tiek izmantota informācijas atlasīšanai un detalizācijai (analogiska ar OLAP operāciju „drill-down”), t.i. Precizēšana ir pretēja darbība Agregācijai. Informācijas precizēšana ir vai nu detalizētas informācijas rādīšana (izvēlas viena vai vairāku objektu īpašības ) vai atlase ar ierobežošanu, t.i. parādot īpašības atbilstoši noteiktiem ierobežojumiem- nosacījumam ar tipu (*Typified Condition*)

Ja prasībā ir ietverts arī kāds ierobežojums, tad tas tiek reprezentēts ar nosacījumu ar tipu. Var būt divu veidu nosacījumi: Vienkāršs nosacījums un Komplekss nosacījums.



32. attēls. Prasību metamodelis

Komplekss nosacījums apvieno divus vai vairākus vienkāršus nosacījumus ar loģiskiem operatoriem (kā “and”, “or”, “not”). Vienkāršs nosacījums sastāv no salīdzinājuma (*Comparison*) starp divām izteiksmēm (*Expressions*), piemēram, „laiks ir lielāks nekā pēdējais pieslēgšanās laiks -1 sekunde”. Izteiksme pati var būt vai nu vienkārša izteiksme (*Simple Expression*) vai Kompleksa izteiksme (*Complex Expression*). Kompleksa izteiksme satur divas vai vairākas vienkāršas izteiksmes, kas saistītas ar aritmētisku operatoru starp tām. Vienkārša izteiksme ir vai nu kvalificējoši dati (piemēram, „pēdējais piekļuves laiks”) vai konstantes (*Constant*), piemēram, „1 sekunde”.

Piemēri, kas doti tālāk šajā apakšnodaļā, ilustrē piedāvātā formālā modeļa pielietošanu, definējot formāli indikatorus. Piemēri indikatoru definīcijām dabīgā valodā ņemti no veikspējas mērījumu datubāzes (Parmenter, 2010). Šajā datubāzē atrodams izsmelošs saraksts veikspējas mērījumu indikatoriem no 6 dažādām mērīšanas perspektīvām, piemēram, klientu, finanšu utt. Piemēriem izvēlēti indikatoru no dažādām perspektīvām, pie tam ar dažādu indikatora teikuma uzbūvi dabiskā valodā.

show	Refinement	Action	Operation		Simple Requirement
month	Qualifying Data	Object			
AVG	Aggregation	Action	Operation		
contacts occurrence	Quantifying Data	Object			
where	Condition Type				Typified Condition
customer type	Qualifying Data	Simple Expression	Simple Condition		
=	Comparison				
'key customer'	Constant	Simple Expression			

33. attēls. Prasību formalizācijas piemērs indikatoram ar tipu CF

Pieņemsim, ka nepieciešams noskaidrot informāciju par kontaktu skaitu ar galvenajiem klientiem mēnesī („*average number of contacts made with key customers per month*”). Šis indikators reprezentē indikatorus no klientu fokusa grupas. Šo teikumu var pārformulēt, izmantojot piedāvāto prasību šablonu modeli (32. attēls).

Teikums, kas ir pārformulēts pēc iepriekš minētajiem principiem un ir izveidots no sākotnējā indikatora formulējumā ir „parādīt mēnesi, vidējo kontaktu notikumu skaitu, kur klientu tips ir „galvenie” (“*show month, average (contact occurrence) where customer type is 'key customer'*”). 33. attēlā redzams, kā prasību šablons tiek pielietots iepriekš apskatītajam biznesa prasību piemēram. Tabulā kolonna kreisajā pusē ir aizpildīta ar teikuma sastāvdaļām, bet pārējās kolonnas virzienā no kreisās puses uz labo satur metamodeļa elementus virzienā no specializētākā līdz vispārīgākajam.

Tālāk tiks apskatīts cits piemērs no indikatoru saraksta. Pieņemsim, ka ir nepieciešams noskaidrot informāciju par novēlotām piegādēm klientiem (*late deliveries*). Šis indikators reprezentē indikatorus, kas attiecas uz klienta fokusu un iekšējiem procesiem. Teikums, kas ir iegūstams atbilstoši metamodelim (32. attēls), ir „parādīt piegādes, kam piegādes tips ir ‘vēlu’” (“*show deliveries where delivery type is 'late'*”) Vienkāršojot pierakstu tabulas formā (atšķirībā no iepriekšējā piemēra), 34. (a) attēlā redzami tikai detalizētākā līmeņa elementi no piedāvātā prasību šablonu metamodeļa – 2. kolonnā un atbilstošās teikuma daļas - 1. kolonnā.

Apskatīsim vēl vienu piemēru, kad nepieciešams iegūt informāciju par darbinieku kustību pēc tipa (piemēram, atlūgums, līguma beigas, pagaidu darbs, līguma pārtraukšana). Šis indikators reprezentē indikatorus, kas attiecas uz darbinieku apmierinātību.

Teikums, kas var tikt izveidots pēc prasību šablona metamodeļa (32. attēls) ir „parādīt nodarbinātības statusus, saskaitīt personas, kam statuss ir „atlūgums” vai „līguma beigas” vai „pagaidu darbs” vai „līguma pārtraukšana” (“*show employment types, count (person occurrence) where employment type is 'resignation' or 'end of contract' or 'temporary staff'*”).

or 'termination' ”). 34. (b) attēls demonstrē tikai detalizētākā līmeņa elementus no piedāvātā prasību šablonu metamodeļa un atbilstošās teikuma daļas.

(a)		(b)	
	Indicator Component	Metamodel Element	
1	show	Refinement	1
2	delivery	Qualifying Data	2
3	where	Condition Type	3
4	delivery type	Qualifying Data	4
5	=	Comparison	5
6	'late'	Constant	6
			7
			8
			9
			10
			11
			12

rows 9-12 are repeated for 2 more comparisons with different constants

34. attēls. Prasību formalizācijas piemēri: (a) Tips – CF, IP; (b) Tips – ES.

## 5.6. Veiktspējas mērīšanas ietvars ar indikatoru dzīves cikla atbalstu

### 5.6.1. Indikatori un to dzīves cikls kā pamats veiktspējas mērīšanas ietvaram

Promocijas darbā tika aprakstīti veiktspējas mērīšanas jēdzieni un tika piedāvāts 5 soļu indikatoru dzīves cikls, kas var tikt izmantots par pamatu veiktspējas mērīšanas ietvara un atbilstošas veiktspējas mērīšanas sistēmas izveidei.

Efektīva biznesa procesu organizācija nodrošina organizācijas mērķu sasniegšanu. Veiktspējas mērīšana salīdzina mērījumu rezultātus ar mērķa vērtībām, lai sekotu līdz progresam. Būtiski ir izvēlēties atbilstošus mērījumus un mērīšanas ietvaru.

Veiktspējas mērījumi (Parmenter, 2010) ir indikatori, ko lieto organizācijas vadība, lai mērītu, ziņotu citiem un uzlabotu organizācijas veiktspēju.

Konkrētu veiktspējas indikatoru izvēli ietekmē pārvaldības modeļi organizācijās un šo pārvaldības modeļu mērīšanas perspektīvas, piemēram, *Business Scorecard* (Kaplan, Norton, 1996) definē četras mērījumu perspektīvas: finansu, klienta, iekšējo procesu, kā arī apmācības un izaugsmes perspektīvas. Citi autori pievieno vēl jaunas perspektīvas – piemēram, Vide/Sabiedrība un Darbinieku apmierinātība (Parmenter, 2010).

Lai veiktu efektīvu mērīšanu un adekvāti varētu reaģēt, atbilstoši atklātajam mērījumu rezultātos, būtu jāmodelē, jādokumentē un jāanalizē ne tikai dažādas perspektīvas, bet dažādi aspekti, piemēram, atbildība, reakcijas veidi, indikatoru saistība ar veiksmes faktoriem un citi.

Iepriekš promocijas darbā tika aprakstītas indikatoru īpašības un tās tika grupētas atbilstoši indikatoru dzīves ciklam. Indikatoru dzīves cikla jēdziens var tikt izmantots, lai atbalstītu piemērotu indikatoru lietošanu atbilstoši indikatoru īpašību vērtībām.

Lai implementētu veiktspējas mērīšanu atbilstoši izvēlētajai uzņēmuma (organizācijas) stratēģijai un atbilstoši kādam pārvaldības modelim, organizācijas izstrādā un lieto mērīšanas sistēmas. Viens no risinājumiem ir datu noliktavas, lai izstrādātu mērīšanas sistēmas, it īpaši, ja var izmantot jau eksistējošas datu noliktavas infrastruktūru. Eksistējošas datu noliktavas parasti glabā informāciju par klientiem un finansēm, bet ne par iekšējiem procesiem, vai kādu

citu veikspējas mērījumu perspektīvu. Tieši iekšējo procesu perspektīvai daži mēģinājumi ir jau bijuši datu noliktavu sistēmās, piemēram, (List, Machaczek, 2004), (Jarke et al., 2000), (Kueng et al., 2001).

Promocijas darbā netiek ieviesta jauna mērīšanas perspektīva datu noliktavās, bet datu noliktava tiek izmantota kā veikspējas mērījumu sistēmas sastāvdaļa, kas glabā indikatoru vērtības no dažādām perspektīvām un tiek izmantotas saskaņā ar tālāk piedāvāto veikspējas mērīšanas ietvaru.

Piedāvātais mērījumu ietvars apraksta dažādus mērīšanas aspektus ar mērķi sistematizēt aktivitātes tādā organizācijai nozīmīgā iniciatīvā, kā veikspējas mērīšana. Līdz ar šāda mērīšanas ietvara ieviešanu, mērīšanas procesa kvalitāte tiek uzlabota, nodrošinot to, ka tiek analizēti vajadzīgie indikatori pareizajā laikā un atbilstoši kontekstam, kā arī to, ka rezultātā tiek veiktas atbilstošas aktivitātes.

Esošas datu noliktavas infrastruktūras izmantošana dod papildus priekšrocības veikspējas mērīšanā. Indikatoru vērtību analīze var tikt veikta ar eksistējošu OLAP rīku, datu noliktavas atskaišu un dažādu grafisku rīku palīdzību.

Promocijas darbā iepriekš tika definēts indikatoru dzīves cikls ar pieciem soļiem – indikatoru definēšana, mērīšana, analīze, reakcija un uzlabošana. Katrā solī indikatori tiek aprakstīti ar dažādām īpašībām. Indikatoru definēšanas solis apraksta lietotāja informācijas vajadzības. Mērīšanas solī indikatoriem tiek piešķirtas vērtības. Analīzes solis reprezentē procesu, kad indikatori tiek izmantoti lēmumu pieņemšanā. Reakcijas solis reprezentē lēmumu implementāciju. Indikatoru dzīves cikls tiek pabeigts ar uzlabošanas soli, kurā notiek indikatoru definīciju un īpašību sākotnējo vērtību novērtēšana.

Iepriekš tika aprakstīts, kā indikatoru formulējumi var tikt formalizēti, lai precīzi aprakstītu mērīšanu – kas, kāpēc un kā tiek mērīts. No vienas puses – indikatori ir datu analīzes fokuss mērīšanas procesā, no otras puses datu noliktavas tiek būvētas un izstrādātas, lai reprezentētu informācijas vajadzības datu analīzei. Tādēļ var runāt par indikatoriem kā par informācijas prasībām datu noliktavu sistēmai. Promocijas darbā iepriekšējā nodaļā tika piedāvāts indikatoru definēšanas metamodelis

Indikatoru formālās definīcijas var tikt izmantotas, lai pusautomātiski ģenerētu datu noliktavas shēmu, tomēr promocijas darbā šāda metode netiek piedāvāta, tiek pieņemts, ka datu noliktavas modelis ir atbilstošs datu analīzes informācijas vajadzībām, tādēļ ir iespējams nedefinēt kartējumu starp indikatoriem un datu noliktavā ietverto informāciju.

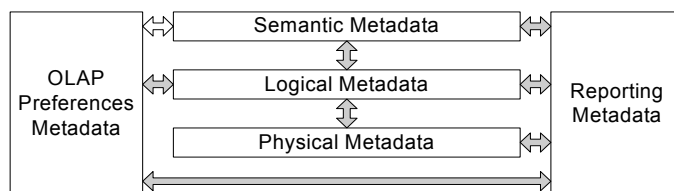
Promocijas darbā tālāk tiks piedāvāts veikspējas mērīšanas ietvars, kas izmanto datu noliktavu iespējas, kā, piemēram, atskaišu rīku un ETL procesu definīcijas.

### **5.6.2. Atskaišu rīka izmantošana mērījumu ietvarā**

Tā kā mērīšanas ietvars veidots, pēc iespējas izmantojot datu noliktavu tehnoloģijas, tad nozīmīga sastāvdaļa piedāvātajā mērījumu ietvarā ir atskaišu rīks, kas izstrādāts Latvijas Universitātē (Solodovnikova, 2007). Atskaišu rīks ir balstīts uz metadatiem un tā pēdējā versija izmanto piecus metadatu slāņus (skat. 35. attēlu), kas apraksta datu noliktavas shēmas glabāšanu, kā arī atskaišu definēšanu, kas balstās uz šiem metadatiem.

Semantiskie, Loģiskie un Fiziskie metadati apraksta datu noliktavas shēmu dažādos abstrakcijas līmeņos, sākot no shēmas elementu biznesa izpratnes semantiskajos metadatos, tālāk aprakstot shēmas elementus loģiskā līmenī ar OLAP jēdzienu palīdzību un beidzot ar datu noliktavas shēmas tabulu definīcijām fiziskajā līmenī. OLAP iestatījumu metadati rīkam ir ieviesti, lai aprakstītu lietotāju vēlnes, kas saistītas ar atskaišu struktūru un tiek izmantotas OLAP personalizācijas mērķiem.

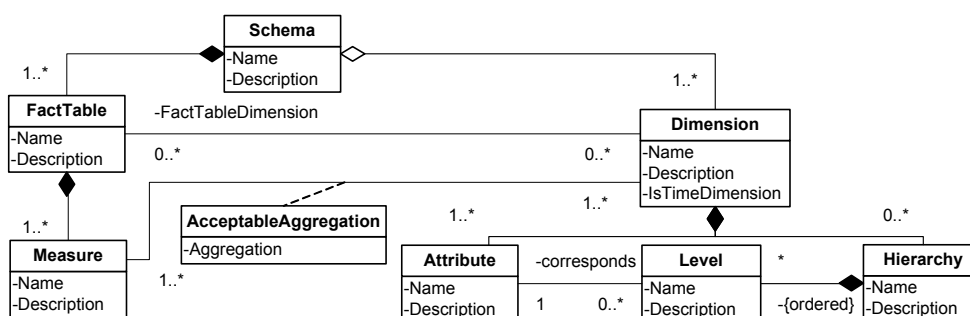




35. attēls. Atskaišu rīka metadatu līmeņi (Kozmina, Solodovnikova, 2011)

Metadatu līmeņi ir savstarpēji saistīti ar asociācijām starp atsevišķām klasēm. Piedāvātā mērīšanas ietvara kontekstā būtiski ir loģiskie un atskaišu metadata, kas tādēļ tiek aprakstīti sīkāk. Pilns metadatu detalizēts apraksts atrodams (Solodovnikova, 2010), (Solodovnikova, Niedrite, 2011).

Loģiskā līmeņa metadatai (skat. 36. attēlu) apraksta datu noliktavas shēmu no daudzdimensiju skatu punkta, un pamatā ir balstīts uz *Common Warehouse Metamodel* standarta (CWM) OLAP pakotni, tādēļ tajā ir OLAP jēdzieni –dimensijas un fakti (kubi CWM terminos). Faktu tabulas satur mērījumus, dimensijas sastāv no atribūtiem, bet hierarhijas sastāv no hierarhiju līmeņiem. Faktu tabulas un Dimensijas ir saistītas ar asociāciju *FactTableDimension*. Tikai dimensijas un faktu tabulas, kas saistītas ar šo asociāciju var vienlaicīgi tikt izmantotas vienā atskaitē, ko apraksta atskaišu metadatai. CWM OLAP pakotne ir papildināta ar *AcceptableAggregation* klasi, kas atļauj tikai tādu agregācijas funkciju (piemēram, SUM, AVG) definēšanu faktiem un dimensijām, kam konkrētajā situācijā ir jēga. Šie metadatai tiek izmantoti, lai definētu korektus vaicājumus atskaišu rīkā.

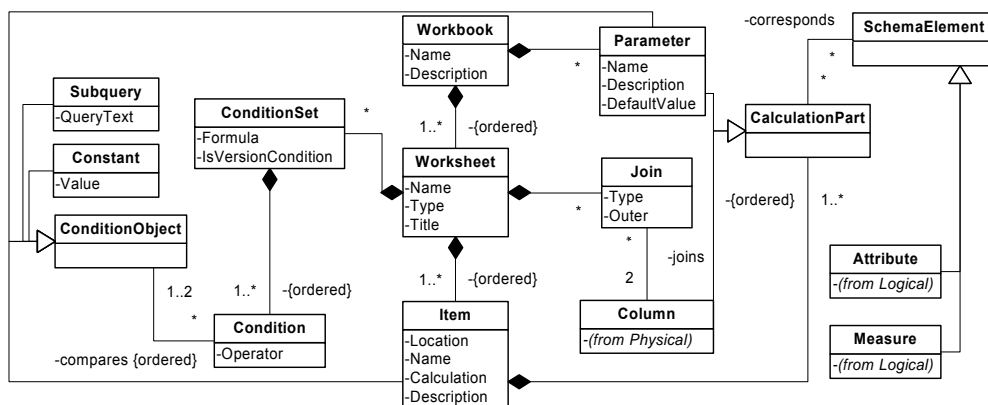


36. attēls. Loģiskā līmeņa metadatai (Solodovnikova, 2010)

Atskaišu metadatai (37. attēls) apraksta atskaišu struktūru. Šī modeļa izpratnē atskaites ir darba burtnīcas (*Worksheets*). Tās satur datu elementus, ko definē kalkūlācijas (*Calculations*). Kalkūlācijas specificē formulas, kas izmanto parametrus un tabulas kolonnas, kas atbilst shēmas elementiem no datu noliktavas modeļa. Atskaites balstās arī uz saitēm starp tabulām un var izmantot lietotāja definētus nosacījumus.

Loģiskā līmeņa metadatai un atskaišu metadatai ir saistīti. Atskaišu elementus definē ar formulas sastāvdaļām (atskaišu metadatai), kas katra atsaucas uz kādu no dimensiju atribūtiem vai mērījumiem (loģiskā līmeņa metadatai). Šo saistību apraksta ar asociāciju ‘*corresponds*’

Atskaišu rīkā atskaites tiek definētas, izvēloties paredzētos elementus no datu noliktavas shēmas, kā arī definējot nosacījumus, parametrus. Tikai vienai shēmai piederīgi mērījumi un dimensijas var tikt iekļauti vienā atskaitē.



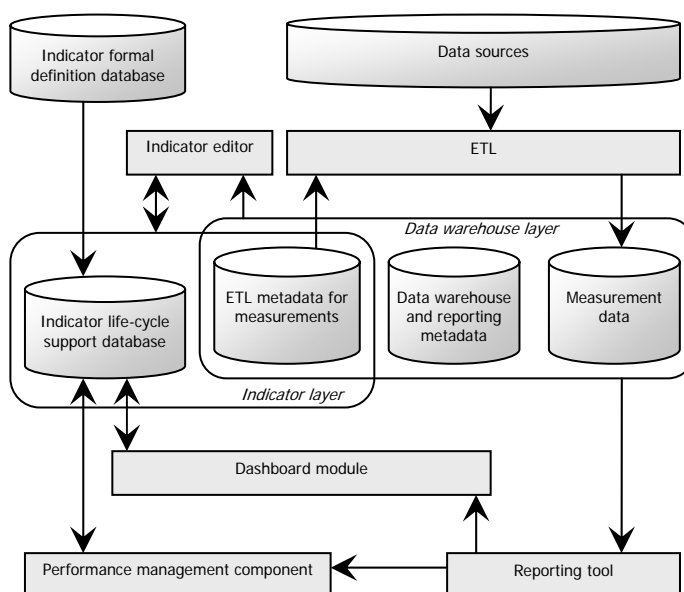
37. attēls. Atskaišu metadati (Solodovnikova, Niedrite, 2011)

### 5.6.3. Veiktspējas mērīšanas sistēmas komponenti

Promocijas darbā tiek piedāvāts ietvars veiktspējas mērīšanas sistēmu izveidei, kas būtiski balstās uz datu noliktavu tehnoloģiju izmantošanu. Tas balstās uz sekojošiem izstrādes un izmantošanas principiem:

- indikatoru informācijas apstrāde tiek veikta saskaņā ar indikatoru dzīves ciklu un formālo indikatoru modeli, kas tika aprakstīts iepriekš promocijas darbā;
- mērījumu dati tiek iegūti ar datu noliktavas ETL procesu palīdzību un tiek glabāti datu noliktavā;
- indikatoru analīzes aspekti tiek nodrošināti ar eksistējoša atskaišu rīka palīdzību gan indikatora vērtības aprēķināšanai, gan detalizētākas informācijas piegādei atskaišu formā.

Veiktspējas mērījumu sistēmas kodols (skat. 38. attēlu) sastāv no pārvaldības komponenta (*performance management component*) un no indikatoru dzīves cikla atbalsta datubāzes (*indicator life-cycle support database*), kā arī no informācijas paneļa (*dashboard module*).



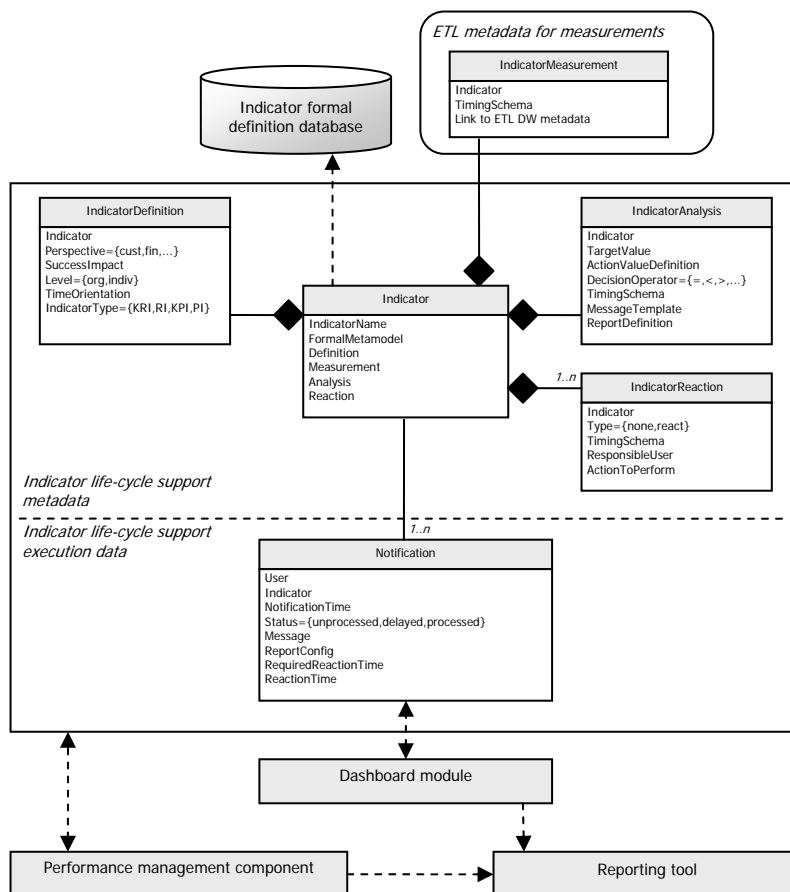
38. attēls. Veiktspējas mērīšanas sistēmas komponenti

Indikatoru dzīves cikla atbalsta datubāze (*Indicator life-cycle support database*) glabā saites uz indikatoru formālām definīcijām Indikatoru formālo definīciju datubāzē (*Indicator formal definition database*), kas izstrādāts atbilstoši iepriekš aprakstītajam indikatoru modelim.

Indikatoru redaktors (*Indicator editor*) ir administrēšanas rīks un ir paredzēts diviem mērķiem: 1) lai nodefinētu saites starp indikatoru formālo definīciju un veikspējas mērīšanas sistēmas indikatoru slāni un 2) lai konfigurētu indikatoru mērīšanas metadatus (ar datu noliktavas ETL procesiem saistītus metadatus).

Mērīšanas process, kura laikā indikatori iegūst savas vērtības, tiek realizēts ar datu noliktavas ETL procesu palīdzību, kas izmanto ETL metadatus mērīšanai (*ETL metadata for measurements*). ETL komponenti ir sistēmas ārēji komponenti. Promocijas darbā tiek pieņemts, ka ir realizēta virkne procedūru, kas veic datu atjaunošanu datu noliktavā atbilstoši ETL mērīšanas metadatiem (piemēram, laika plānojumu).

ETL procesi apstrādā datus no ārējiem datu avotiem un ielādē tos datu noliktavā, kas veikspējas mērījumu sistēmā atbilst mērījumu datiem (*Measurement data*). Pārējā sistēmas daļa ir metadati – Datu noliktavas un atskaišu rīka metadati, atbilstoši tam, kā tika aprakstīts iepriekš promocijas darbā.



39. attēls. Indikatoru dzīves cikla atbalsta metadati

Veikspējas pārvaldības komponents (*Performance management component*) ir galvenā sistēmas daļa, kas paredzēta biznesa procesu monitorēšanai, analizējot indikatoru mērījumu rezultātus. Komponents ir balstīts uz dažādu indikatoru īpašību aprakstiem indikatoru dzīves cikla atbalsta datubāzē, kas ļauj lietotājam analizēt indikatoru vērtības vispiemērotākajā veidā, izmantojot divus citus komponentus- indikatoru paneli (*Dashboard module*) un atskaišu rīku (*Reporting tool*). Indikatoru panelis vizualizē galvenos indikatorus un to salīdzinājumu ar

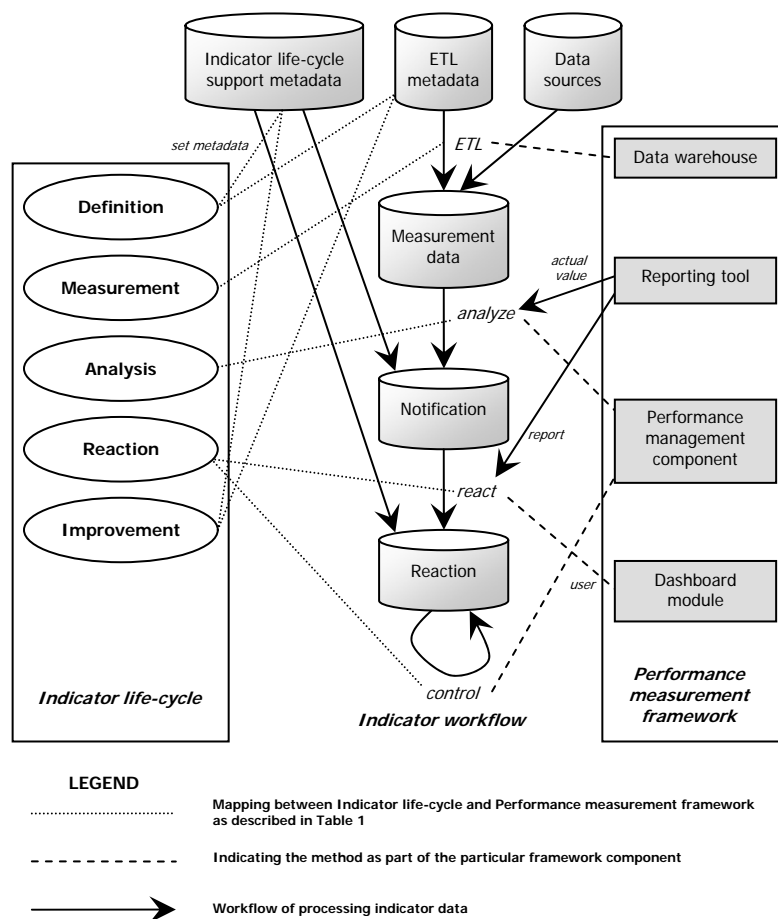
mērķa vērtībām. Atskaišu rīks piegādā lietotājam detalizētu informāciju, ar rīkā jau iepriekš definētu atskaišu palīdzību, kas piekārtotas indikatoriem. Šim nolūkam izmanto eksistējošu atskaišu rīku, kas ir izstrādāts, saskaņā ar iepriekš aprakstītajiem atskaišu metadatiem. Vairāk par atskaišu rīku atrodams (Solodovnikova, Niedrite, 2011).

Indikatoru dzīves cikla atbalsta datubāzes pamatā ir Indikatoru dzīves cikla atbalsta metadati *'Indicator life-cycle support metadata'* (skat. 39. attēlu), kas nosaka veikspējas mērījumu sistēmas darbību. Šie metadati tiek izmantoti Veikspējas pārvaldības komponentā, kura izmantošanas mērķis ir koordinēt indikatoru dzīves cikla darbu plūsmu.

Tā kā mērīšanas uzdevumi ir pilnībā uzticēti datu noliktavai, ETL metadati tiek sagatavoti un glabāti ārpus Indikatoru dzīves cikla atbalsta datubāzes. ETL procesu darba plūsmas izpildes rezultāti tiek glabāti Indikatoru dzīves cikla izpildes datos (*'Indicator life-cycle support execution data'*) un tos tiešā veidā izmanto lietotājs ar informācijas paneļa palīdzību. Darba plūsmas izpildes rezultātus kontrolē arī veikspējas pārvaldības komponents kopā ar informācijas paneli, tie ietver arī ziņojumu sūtīšanu lietotājiem un lietotāju reakciju uz tiem (piemēram, detalizētu atskaišu izsaukumu atskaišu rīkā).

#### 5.6.4. Indikatoru dzīves cikla soļi veikspējas mērīšanas ietvarā

Piedāvātais veikspējas mērīšanas ietvars atbalsta indikatora dzīves cikla piecus soļus. Tālāk tiks apskatīts, kā šis atbalsts tiek nodrošināts atbilstoši katra soļa specifikai.



40. attēls. Indikatoru dzīves cikla atbalsts veikspējas mērīšanas ietvarā

Attēlā (skat.40. attēls) redzamas indikatoru dzīves cikla soļu secība, tajos izmantotie veikspējas pārvaldības ietvara komponenti un iesaistītie datu avoti. Veikspējas mērīšana notiek sekojoši:

- mērīšanas solis tiek izpildīts ar ETL soļu palīdzību,
- analīzes laikā tiek apstrādāti mērījumu dati saskaņā ar indikatoru dzīves cikla atbalsta metadatiem, un to veic veikspējas pārvaldības komponents (skat. 41. attēls). Tiek izmantots atskaišu rīks, lai iegūtu indikatoram atbilstošā mērījuma aktuālo vērtību. Šajā solī tiek pievienots ieraksts indikatoru dzīves cikla izpildes datos, rezultātā informācija par veiktajiem indikatoru mērījumiem informācija ziņojuma veidā kļūst redzama lietotājam īpašā informācijas panelī,
- lietotāja reakcija (skat. 42. attēls) tiek reģistrēta informācijas panelī, un tā var būt divu veidu:
  - 1.1 detalizētas informācijas pieprasījums, tad tiek izsaukts atskaišu rīks ar indikatoram piekārtotu detalizētu atskaiti;
  - 1.2 reakcija, kas gadījumā, ja to pieprasa indikatoru īpašību apraksts, var tikt veikta saskaņā ar šo īpašību vērtībām ārpus ietvara.
- kontroles laikā tiek pārbaudīts, vai lietotājs ir reaģējis laicīgi un paredzētajā veidā uz indikatora izmaiņām, ja to pieprasa indikatora apraksts. (skat. 43. attēls).

Tabula (skat.4. tabulu) attēlo sakarības starp indikatoru dzīves cikla soļiem un tā implementāciju veikspējas mērīšanas ietvarā.

#### 4. Tabula. Attēlojums “Indikatoru dzīves cikls ↔ veikspējas mērīšanas ietvars”

<i>Dzīves cikla soļi</i>	<i>Implementācijas apraksts veikspējas mērīšanas ietvarā</i>
<b>1. Definēšana</b>	Indikatoru definīcijas ir aprakstītas indikatoru formālās definīcijas datubāzē, kā arī indikatoru dzīves cikla atbalsta datubāzē. Indikatoru definēšana ietver metadatu sagatavošanu pārējiem dzīves cikla soļiem.
<b>2. Mērīšana</b>	Mērīšanas process tiek uzticēts datu noliktavas ETL procesiem.
<b>3. Analīzes</b>	Analīze ir uzticēta veikspējas pārvaldības komponentam, mērījumu dati tiek apstrādāti un attēloti lietotājam.
<b>4. Reakcija</b>	Lietotājs izlasa, un ja nepieciešamas reaģē uz ziņojumu. Veikspējas pārvaldības komponents kontrolē reaģēšanu.
<b>5.Uzlabošana</b>	Indikatoru uzlabošana tehniski atbilst indikatoru definēšanai, bet tiek ņemta iepriekšējā cikla izpildes pieredze.

<pre> Procedure analyze Begin   Repeat Forever     Foreach indicator From Indicator Do       analysis := Indicator.Analysis       Wait for the next report according to analysis.TimingSchema       actual_value := run report according to analysis.ActualValueDefinition       If analysis.DecisionOperator(actual_value, analysis.TargetValue) = True Then         Forall reaction In Indicator.Reaction Do           add record to Notification With             User := reaction.ResponsibleUser             Indicator := indicator             NotificationTime := current time             Status := 'unprocessed'             Message := compute according to analysis.MessageTemplate               and indicator.Definition and reaction.ActionToPerform             ReportConfig := set according to analysis.ReportDefinition             RequiredReactionTime := compute according to reaction.TimingSchema             ReadTime := Null             ReactionTime := Null </pre>
---

#### 41. attēls. Analīzes soļa algoritms veikspējas pārvaldības komponentam

```

Procedure react
Begin
  Foreach user
    Display all from Notification in the dashboard Where User = user
    Foreach notification From Notification Where User = user Do
      Wait for user action Do
        Case user asks to show detalized information Do
          run report according to notification.ReportConfig and display it
        Case performs an action according to report.Indicator.Reaction.ActionToPerform Do
          notification.ReactionTime = current time

```

#### 42. attēls. Reakcijas soļa algoritms informācijas panelim

```

Procedure control
Begin
  Repeat Forever
    Foreach notification From Notification Where ReportStatus <> 'processed' Do
      If notification.Indicator.Reaction.Type = 'none' Then
        notification.Status = 'processed'
      Else
        If notification.ReactionTime Is Not Null Then
          notification.Status = 'processed'
        Else If current time > notification.RequiredReactionTime Then
          notification.Status = 'delayed'

```

#### 43. attēls. Kontroles soļa algoritms veikspējas pārvaldības komponentam

### 5.6.5. Integrācija ar datu noliktavas komponentiem

ETL metadati mērīšanai (*IndicatorMeasurement* klase) ir daļa no indikatoru dzīves cikla atbalsta datubāzes (skat.39. attēls). Atribūts *Indicator* identificē konkrēto atribūtu, kas tiek mērīts, *TimingSchema* apraksta laika atribūtus mērīšanai, piemēram, biežums, precīzs mērīšanas laiks). Atribūts *ETLprocess* norāda uz datu noliktavas repozitoriju, kurā glabājas ETL metadati, kas apraksta attēlojumu (kartējumu) starp datu avotu shēmām un datu noliktavas shēmu, šie metadati arī satur ETL procedūru izsaukumus, kas šos kartējumus un nepieciešamās datu transformācijas implementē. Piedāvātajam mērīšanas ietvaram tiek pieņemts, ka *IndicatorMeasurement* klase satur ETL procedūru izsaukumus, kas atjauno datu noliktavas shēmas datus, kas nepieciešami izvēlētajā indikatora vērtības aprēķināšanai.

*IndicatorAnalysis* klase no indikatoru dzīves cikla atbalsta datubāzes (skat.39. attēls) un tās atribūts *ReportDefinition* ir paredzēts norādes glabāšanai uz atskaites definīciju atskaišu metadatos atskaišu rīkam.

Atskaišu metadati (skat. 37.attēls) satur klasi *Worksheet*, kas identificē konkrētu atskaiti, kas var tikt izsaukta, kad tiek veikta mērīšanas rezultātu analīze. Atskaite var būt vienkārša, kad tiek aprēķināta tikai viena vērtība, vai kompleksa, kad rezultāts ir atskaite detalizētas analīzes vajadzībām. Atskaites sarežģītība ir atkarīga no konkrētās atskaites definīcijas metadatos.

### 5.6.6. Veikspējas mērījumu sistēmas novērtējums

Veikspējas mērījumu sistēmas, kas implementētas ar datu noliktavu palīdzību ir aprakstītas vairākos darbos. Pētījumos par eksistējošām sistēmām centrālais jautājums tipiski ir saistīts ar piemērota datu noliktavas konceptuālā daudzdimensiju modeļa izstrādi, kas būtu piemērots procesu datu glabāšanai. Parasti procesu izpildes žurnālfaili tiek izmantoti kā viens no papildus datu avotiem, un tajos esošie dati tiek integrēti eksistējošā uzņēmuma datu noliktavā kā papildus informācija.

Procesu datu noliktavu piemēri jau tika apskatīti promocijas darbā, lai argumentētu, ka veikspējas mērīšanas sistēmas var tikt implementētas, izmantojot datu noliktavu tehnoloģijas (skat. 5.4. apakšnodaļu), savukārt, šajā apakšnodaļā esošie risinājumi tiks skatīti, lai izvērtētu

promocijas darbā piedāvāto veiktspējas mērīšanas ietvaru un tai atbilstošu sistēmu. Tālāk dots īss kopsavilkums par eksistējošām pieejām, kā arī tabulas formā šo pieeju vērtējums pēc dažādiem kritērijiem, kas apskatīti tālāk darbā.

Procesu datu noliktavu Jarke et al. (2000) definēja „kā datu noliktavu, kas glabā datus par ražošanas procesu izpildi un rezultātiem pieredzes iegūšanai un atkārtotai izmantošanai”.

Veiktspējas pārvaldības sistēma (Kueng et al., 2001) glabā procesu izpildes datus, gan finansiālos, gan nefinansiālos centralizētā datu noliktavā. Tiek piedāvāta metode, lai varētu uzbūvēt veiktspējas pārvaldības sistēmai atbilstošu datu noliktavas modeli. Tiek noskaidrotas ieinteresēto pušu informācijas prasības, veicot organizācijas un tās mērķu analīzi. Veiktspējas mērījumu sistēmas satur mērījumu vērtības un papildus informāciju par kompānijas struktūru, biznesa procesiem, mērķiem. Papildus tradicionālajām datu noliktavu perspektīvām veiktspēju mērījumu gadījumā, šeit tiek analizēta arī procesu perspektīva.

Korporatīvajā izpildes mērīšanas sistēmā (*Corporate Performance Measurement System - CPMS*) (List, Machaczek, 2004) procesu izpildes dati no darba plūsmu sistēmas žurnālfailiem tiek integrēti uzņēmuma datu noliktavā. CPMS datu noliktavas projektējuma izveidei tiek noskaidroti biznesa procesu mērķi no uzņēmuma mērķiem, tiek definēti jautājumi par mērķu sasniegšanu, tiek dokumentēti atbilstoši indikatori un datu avoti šo indikatoru aprēķināšanai.

Darba plūsmas datu noliktava (Bonifati et al., 2001) paredzēta tieši darba plūsmu sistēmu datu glabāšanai. Šīs sistēmas vispārīgais datu noliktavas modelis būvēts tieši no procesu izpildes skatu punkta, izejot no žurnālfailu formāta un datu piedāvājuma, saistība ar „ne-procesu” datiem ir paredzēta, ir papildus faktu tabula – „Procesu dati”, bet šeit dati tiek vākti saistībā ar procesu izpildi, nevis informācijas vajadzībām. Modeļa oriģinalitāte slēpjas dimensijā „Behaviour”, kas ļauj analizēt tipiskus procesu izpildes ceļus, un var tikt izmantots kā atbalsts lēmumu pieņemšanā, klasificējot jaunu procesu izpildes instanci pēc līdzības ar iepriekšējiem.

Tiek pētītas arī metodoloģijas, kas apraksta, kā veiktspēja ir vērtējama. Piemēram, metodoloģija, kas balstās uz dinamisku procesu veiktspējas novērtējumu (Tan et al., 2008), piedāvā mērījumu modeli dažādu procesu plūsmu analīzei. Analīzes mērķis ir kontrolēt procesu izpildes kvalitāti. Tiek mērītas aktivitāšu plūsma, informācijas plūsma un resursu plūsma. Kā vērtēšanas kritēriji tiek izmantoti laiks, kvalitāte, serviss, izmaksas, efektivitāte un nozīmīgums.

Tālāk dots tabulas formā (skat. 5. Tabula) šo pieeju vērtējums pēc dažādiem kritērijiem: a) tiek skatīta pieejas saistība ar procesu perspektīvu, vai tā ir integrēta ar jau eksistējošu datu noliktavu, b) kāds ir metodoloģisks atbalsts indikatoru definēšanai – kā noskaidrot ko mērīt, vai vismaz ir nosaukti indikatoru piemēri c) kāds ir metodoloģisks atbalsts indikatoru mērīšanai un izmantošanai (kāpēc tiek veikta mērīšana, kā indikatoru vērtības jāizmanto) un d) vai pieeja balstās uz kādu no pārvaldības modeļiem.

Tabulā dotais izvērtējums parāda, ka nevienu no minētajām pieejām nav pārlicinoši labāka visu kritēriju izvērtējumā. Veiktspējas pārvaldības sistēma (Kueng et al., 2001) un Korporatīvā veiktspējas mērīšanas sistēma (List, Machaczek, 2004) atrisina procesu datu glabāšanu un integrēšanu ar eksistējošu datu noliktavu, bet tikai netieši risina uzkrāto indikatoru efektīvas izmantošanas atbalstu, savukārt, dinamiska procesu pārvaldības vērtēšana (Tan et al., 2008), kas vienīgā no minētām pretendē uz metodoloģijas statusu mērīšanas pielietošanā, par indikatoru noteikšanu, glabāšanu nesniedz detalizētu informāciju. Neviena no minētām pieejām nepārklāj kāda eksistējoša pārvaldības modeļa visas perspektīvas, vai nesniedz par to informāciju.

5. Tabula. Eksistējošas datu noliktavu sistēmas un pieejas procesu pārvaldībai

Sistēma	Procesu perspektīva, saistība ar citiem datu noliktavas datiem	Metodoloģija – indikatoru definēšanai	Metodoloģija – Indikatoru mērīšanai un izmantošanai	Balstās uz pārvaldības modeli
Procesu datu noliktava (Jarke et al., 2000)	glabā datus par ražošanas procesu izpildi	Nav minēts	Glabā datus par procesu izpildi pieredzes iegūšanai un atkārtotai izmantošanai	Nav minēts
Veiktspējas pārvaldības sistēma (Kueng et al., 2001)	Procesu izpildes dati – finansu un ne-finansu	Organizācijas un tās mērķu modelēšana, iesaistīto pušu informācijas vajadzības, noskaidro intervējot	Datu noliktavā glabā indikatorus, papildus atsevišķi – informāciju par organizācijas struktūru un biznesa procesiem, bet metodoloģija kā visu izmantot - nav tieši aprakstīta	Tieši netiek pieminēta, procesu izpildes datus izmanto kā papildus datu avotu
Korporatīvā veiktspējas mērīšanas sistēma (List, Machaczek, 2004)	Procesu izpildes dati tiek integrēti uzņēmuma datu noliktavā	Tiek definēti jautājumi par procesu mērķu sasniegšanu, tiek dokumentēti atbilstoši indikatori un datu avoti šo indikatoru aprēķināšanai, noskaidro intervējot	Nav tieši minēts, izmanto atbilstoši indikatoriem	Tieši netiek pieminēta, procesu izpildes datus izmanto kā papildus datu avotu
Darba plūsmas datu noliktava (Bonifati et al., 2001)	Tikai procesu izpildes žurnālfailu datu glabāšana	Modeļi un iespējamās indikatorus nosaka žurnālfaila formāts	Ir ieviesta dimensija „Behaviour”, kas ļauj klasificēt procesu izpildes ceļus pēc līdzības, tā palīdzot pieņemt līdzīgu lēmumu	Nebalstās, jo datu noliktavas mērķis ir tieši procesu izpildes datu analīze
Dinamiska procesu pārvaldības vērtēšana (Tan et al., 2008)	Nav minēts	Tiek izmantoti mērījumi: laiks, kvalitāte, serviss, izmaksas, efektivitāte un nozīmīgums	Metodoloģija mēra aktivitāšu plūsmu, informācijas plūsmu un resursu plūsmu	Nav minēts

Promocijas darbā veiktspējas pārvaldības ietvars ļauj definēt indikatorus dažādām mērīšanas perspektīvām, ļauj glabāt papildus informāciju par katru indikatoru, kā arī nodrošina metodoloģisku atbalstu mērīšanas procesam, balstot šo atbalstu uz indikatora dzīves cikla modeli. Indikatoru noskaidrošanai metodoloģija netiek piedāvāta, jo iepriekš pieminētām metodēm izmantotā organizācijas mērķu modelēšana var tikt izmantota arī citos risinājumos, tai skaitā promocijas darbā piedāvātajā sistēmā. Indikatoru definēšana tiek, savukārt papildināta ar iespēju formāli pierakstīt indikatorus atbilstoši darbā dotajam indikatoru modelim, lai iegūtu maksimāli precīzu priekšstatu par informācijas vajadzībām.

Promocijas darbā piedāvātā veiktspējas pārvaldības ietvara priekšrocība ir tā, ka tiek izmantoti, kur ir iespējams, datu noliktavu komponenti – ETL procesi, atskaišu rīks, datu noliktavu metadati, lai varētu sagatavot un glabāt indikatorus atbilstoši dažādām mērīšanas perspektīvām. Datu noliktavu komponenti tiek integrēti ar jauniem komponentiem, kas izmantojot metadatus indikatoru dzīves cikla atbalstam, nodrošina mērījumu sistemātisku un kontrolētu izmantošanu veiktspējas mērīšanā.



## 5.7. Biznesa procesu divu līmeņu mērīšanas pieeja

Tālāk apskatītā divu līmeņu mērīšanas pieeja, kuras izstrādē piedalījās promocijas darba autors (Medvedis et al., 2008), (Medvedis et al., 2008b), tika aprobēta e-universitātes iniciatīvas laikā, tāpēc tālāk izmantoti sekojoši galvenie jēdzieni:

- E-universitātes jēdziens: E-universitāte ir galveno servisu un attiecību, kā arī akadēmisko un biznesa procesu transformēšana, izmantojot Interneta tehnoloģiju lietošanu (Edgar, McQuesten, 2000),
- E- Universitātes ietvars: E-universitātes ietvars sastāv no cilvēkiem, e-biznesa, biznesa procesiem un IKT infrastruktūras (Dodds, 2002).

E-universitātes iniciatīva Latvijas Universitātē kā viena no stratēģiskajām prioritātēm tika definēta ar 2001. gada novembra Senāta lēmumu un uzsākta 2002. gada aprīlī. E-universitātes iniciatīvas izveides mērķis bija izmantot IKT, lai uzlabotu, dažādotu un paplašinātu universitātes servisu, lai būtu efektīvāka, pievilcīgāka un konkurētspējīgāka lokālajā un globālajā tirgū (Treimanis, Stonis, 2005).

E-universitātes iniciatīvas dažādos izpausmes veidos, lielākā vai mazākā apjomā ir tikušas veiktas daudzās universitātēs (Edgar, McQuesten, 2000), (Dodds, 2002). Tipiski šajos projektos ir izstrādātas pārvaldības informācijas sistēmas (Mincer-Daszkiwicz, 2003), (Šorm, Netrefová, 2006), (Desnos, 2002), portāli (uPortal), (HC), IKT infrastruktūra, e-studiju sistēmas (Robal, Kalja, 2004), (Jasinska et al., 2006), (Granow, 2006), bet šie risinājumi parasti neietver biznesa procesu un biznesa mērķu definēšanu, mērīšanu un pastāvīgu uzlabošanu, izņemot dažu universitāšu gadījumos (Adenso-Díaz, Cantelli, 2001), kas ietver dažus biznesa procesu pārbūves aspektus. Tajā pat laikā vajadzība pēc biznesa procesu pārbūves un pieejas procesu mērīšanai komerciālās sfērās ir labi zināmas (Harrington et al., 1997), (El Sawy, 2000).

Tālāk piedāvāta metode apskata, kā novērtēt informācijas sistēmas (IS) lietošanu, kas tiek izmantota biznesa procesu atbalstam.

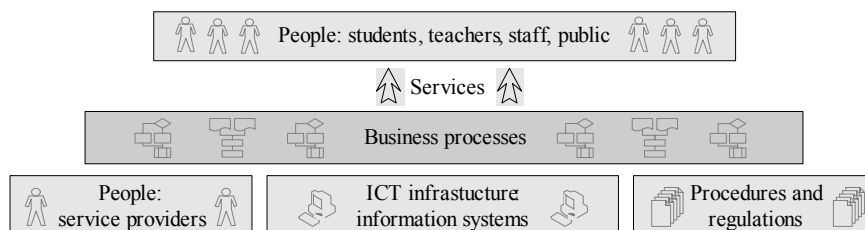
Eksistē daudzi pētījumi, kas mēģina noskaidrot, kāds ir piemērotākais veids, kā mērīt informācijas sistēmu efektivitāti. IS efektivitāte ir jāmēra attiecībā pret produktu, procesu un servisa aspektu šai informācijas sistēmai (Whyte, Bytheway, 1996). Katrs no minētajiem aspektiem ir jāvērtē, izmantojot atšķirīgus mērījumus. Promocijas darbā tiek pētīts informācijas sistēmas produkta aspekts, ko var raksturot ar dažādiem mērījumiem, tai skaitā, piemēram, ar sistēmas reakcijas laiku, ar sistēmas lietošanas raksturlielumiem. Dažādi pētījumi aplūko dažādus individuālus mērījumus, piemēram, sistēmas lietošanu (Trice, Treacy, 1988), (Cuellar et al., 2006). Savukārt nav daudzu modeļu, kas reprezentētu vairākus IS efektivitātes rādītājus un to savstarpējās atkarības.

DeLone un McLean piedāvāja „IS veiksmes modeli” (DeLone, McLean, 1992), kur „IS veiksmes” tika izmantota efektivitātes kontekstā. Atsevišķi pētnieki (Seddon, 1997) nepiekrīt DeLone un McLean ieviestajam „sistēmas lietošanas” mērījumam, kas tika ietverts kā viena no sešām dimensijām IS efektivitātes mērīšanas modelī. Neskatoties uz iebildumiem, sistēmas lietošana tika atkārtoti iekļauta arī 10 gadus vēlāk pārskatītajā „IS veiksmes” (efektivitātes) modelī (DeLone, McLean, 2002). Autori apgalvo, ka, lai nebūtu pārpratumu ar sistēmas lietošanas mērījumu, ir papildus jāraksturo arī sistēmas lietošanas pakāpe, lietošanas veids jeb lietošanas daba, kā arī lietošanas piemērotība.

### 5.7.1. Divu līmeņu mērīšanas pieeja

Organizācijas klientiem un darbiniekiem pakalpojumus piegādā ne tikai citi cilvēki vai informācijas sistēmas, bet tos piegādā biznesa procesi (El Sawy, 2000), kas būtība ir tas faktors, kas saista kopā darbiniekus un citus lietotājus, informācijas sistēmas, procedūras un

likumus (skat. 44. attēlu). Šāda pieeja organizējot organizācijā IS pakalpojumu piegādi klientiem, atļauj sasniegt labākus rezultātus, savienojot kopā visus nepieciešamos komponentus, tā arī dod pamatu, lai izmantotu ierobežotus informāciju sistēmu izstrādes resursus kritiskiem uzdevumiem, kas citādi būtu apgrūtināši. Precīzu informāciju par resursu izmantošanu, par organizācijas veiktspējas dažādām perspektīvām, var iegūt veicot sistemātiskus mērījumus.



44. attēls. Biznesa procesu bāzēta pieeja

Pirms uzsākt mērīšanas aktivitātes, ir nepieciešams pieņemt lēmumu, kas un kādā veidā tiks mērīts. Tiek piedāvāta divu līmeņu mērīšana (Medvedis et al., 2008), (Medvedis et al., 2008b).

**Pirmā**, augsta līmeņa mērīšana aplūko informācijas sistēmas kā atsevišķus mērījumu objektus, piemēram, tās sistēmas, kas ietvertas e-universitātes ietvarā, tiek detalizētāk apskatītas tālāk promocijas darbā.

Informācijas sistēmu efektivitāte bieži tiek mērīta, lai novērtētu ieguldījumu un pārvaldības aktivitāšu nozīmīgumu un lietderību. Tādēļ promocijas darbā izvēlēta pieeja novērtēt IS efektivitāti, izmantojot augsta līmeņa mērījumus. „IS veiksmes modelis” (DeLone, McLean, 1992), (DeLone, McLean, 2002), kas apraksta dažādus efektivitātes aspektus, apraksta 6 dimensijas efektivitātes mērījumiem – sistēmas kvalitāti, informācijas kvalitāti, servisu kvalitāti, sistēmas lietošanu, lietotāju apmierinātību un ieguvumus. Tai pat laikā DeLone un McLean (DeLone, McLean, 1992), (DeLone, McLean, 2002) iesaka ievērojami samazināt dažādu mērījumu skaitu, lai mērītu IS efektivitāti. Tātad pirms uzsākt IS efektivitātes mērījumu aktivitātes, ir jāizvēlas ierobežots skaits piemērotāko mērījumu.

Augsta līmeņa mērīšanai ir būtiski sekojoši uzdevumi:

- organizācijas IS sistēmu apzināšana, integrācijas interfeisu noskaidrošana, dokumentēšana, lai varētu noskaidrot iespējamās alternatīvos vienas funkcijas izpildes ceļus, kas var palīdzēt analizēt un izskaidrot mērīšanas rezultātus,
- ir jānosaka un jādokumentē īpašības, kas ietekmē sistēmas lietošanu: vai sistēmas lietošana ir obligāta vai neobligāta (piemēram, vai funkcijas jāpilda darba pienākumu izpildei, vai nē), piekļuves kontroles veids, kā arī vai eksistē lietotāju piekļuves mehānisms, kas ierobežo sistēmas plašu lietošanu,
- procesu mērīšanas sistēmas izveide, kas jāveido saskaņā ar izvēlētiem mērīšanas indikatoriem, un paredzot, ka tā tiks izmantota arī detalizētā līmeņa mērīšanai,
- mērīšanas mērķu definēšana katrā no līmeņiem. Pirmā līmeņa mērīšanā šādi mērķi var būt, piemēram, lai noskaidrotu, kuras sistēmas lietošanas indikatori mainās, lai nepieciešamības gadījumā varētu veikt padziļinātu analīzi, lai noskaidrotu izmaiņu cēloņus, vai piemēram, pēc organizatorisku pasākumu veikšanas saistībā ar kādas IS atbalstītiem biznesa procesiem, vai mainās IS lietošanas paradumi. Cits mērīšanas mērķis varētu būt saistībā ar finansējumu, piemēram, vai ieguldījumi atmaksājas, vai, piemēram, SaaS arhitektūru gadījumā – kā viens no informācijas avotiem maksas noteikšanai par IS lietošanu no organizācijas,

- indikatoru izvēle, definēšana, aprakstīšana, atbilstoši katram mērīšanas mērķim, pirmā līmeņa mērīšanā – saistībā ar IS lietošanu.

**Otrais** mērījumu līmenis ietver detalizētu mērīšanu atsevišķiem biznesa procesiem. Šajā līmenī tiek pētītas un mērītas konkrētas informācijas sistēmas funkcijas, kas atbalsta izvēlēto biznesa procesu. Mērīšanas mērķi šajā gadījumā parasti ir saistīti ar biznesa procesu pārbūvi, mēra ar procesu saistītus indikatorus.

Biznesa procesu pārbūves nepieciešamību var noteikt vajadzība attīstīt informācijas sistēmas. Izstrāde un ieviešana informācijas sistēmām ir sarežģīta, izmaiņu pieprasījumi – bieži, resursi, kas nepieciešami esošo sistēmu uzturēšanai, bieži ievērojami lielāki nekā jaunas funkcionalitātes izstrādei. No IS viedokļa mērīšanas mērķi var būt IS tālāka attīstīšana, optimizēšana u.c., šai gadījumā IS funkciju lietošanas rādītāji var palīdzēt noteikt prioritātes prasībām.

Otrā līmeņa mērīšanā var lietot pilnu biznesa procesu pārbūves ciklu, pielietojot vienā konkrētā biznesa apgabalā: 1) esošo procesu dokumentēšana, analīze un uzlabošana; 2) uzlaboto procesu implementēšana; 3) regulāra uzlaboto procesu novērošana, balstīta uz savāktiem efektivitāti raksturojošiem indikatoriem un atgriezenisko saiti no lietotājiem.

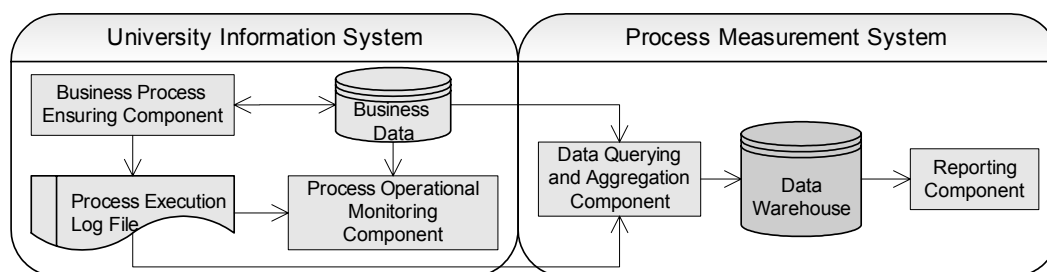
Detalizēta biznesa procesu mērīšana var tikt veikta ar šim nolūkam izveidotu procesu mērīšanas un monitoringa sistēmu.

Tālākajos apakšnodaļas punktos aprakstīts, kā divu līmeņu mērīšanas pieeja tika aprobēta LU e-universitātes iniciatīvas ietvaros. Rezultāti atsevišķu biznesa procesu mērīšanas aktivitātēm atbilstoši piedāvātajai pieejai e-universitātes iniciatīvas ietvaros, ir atspoguļoti (Medvedis et al., 2008) un pielikumā 1.3.

### 5.7.2. *Procesu mērīšanas sistēmas implementācija*

Procesu mērīšanas sistēma, kas aprakstīta šajā punktā, kā sastāvdaļu izmanto datu noliktavu, kas DNATIS arhitektūras kontekstā iekļaujas veikspējas mērījumu sistēmas sastāvā (skat. 5.6. apakšnodaļu).

Procesu mērīšanas un monitorēšanas sistēma (PMMS) tika izveidota kā pārvaldības informācijas sistēmas LUIS integrēta sastāvdaļa (Niedrite et al., 2007a). Kā viena no sastāvdaļām PMMS tika izmantota arī datu noliktava, citas būtiskas komponentes ir: 1) procesu operatīvās monitorēšanas komponente, 2) procesu mērīšanas sistēma, un 3) procesu izpildes žurnālfails. Kopējā PMMS shēma redzama 45. attēlā.



45. attēls. Procesu monitorēšana un mērīšana

Procesu operatīvās monitorēšanas komponents ļauj veidot atskaites no žurnālfailu datiem un biznesa datiem, kas glabājas informācijas sistēmas tabulās. Procesu Operatīvās

monitorēšanas komponents ir izstrādāts kā pārvaldības informācijas sistēmas LUIS sastāvdaļa, izmantojot tās pašas tehnoloģijas kā LUIS.

Procesu operatīvās monitorēšanas komponents ļauj analizēt indikatorus par procesu darba plūsmu tieši no žurnālfaila un procesu darba plūsmas izpildes laikā. Pēc tam indikatori tiek ielādēti datu noliktavā un izmantoti procesu mērīšanas un analīzes uzdevumiem.

Procesu mērīšanas sistēma ir bāzēta uz datu noliktavu modeļiem, kas tika izstrādāti atbilstoši trim jaunām metodēm – lietotāju vadītai (Benefelds, Niedrite, 2004), datu vadītai (Solodovnikova, Niedrite, 2005) un mērķu vadītai metodei (Niedrite et al., 2007a).

Mērīšana, kura var tikt veikta, izmantojot datu noliktavas datus, var būt divu veidu – var mērīt procesu izpildi raksturojošus mērījumus (kad, cik ilgi, kas izpilda procesu utt.) un var mērīt procesu izpildes rezultātus (biznesa datus).

Procesu Mērīšanas Sistēmai ir arī ETL daļa (*Extract, Transform, Load* – izguve, transformācija, ielāde). Šī komponente ir tā, kas veic datu izguvi no procesu izpildes žurnālfailiem un IS, veic nepieciešamās transformācijas, procesu izpildes datu integrēšanu ar biznesa datiem no IS, tiek veikta arī šo integrēto datu transformācijas, lai tie atbilstu datu noliktavas daudzdimensiju modeļiem un pēc tam dati tiek ielādēti datu noliktavas datu struktūrās. Datu glabāšanai tiek izmantota Oracle datu bāze.

Procesu izpildes žurnālfails ir vēl viena būtiska PMMS komponente, kas nodrošina vajadzīgo datu par procesu izpildi savākšanu. Žurnālfails un tā formāts ir veidots tieši PMMS vajadzībām, tā formātu noteica datu noliktavai vajadzīgie dati un prasības, kas bija izvirzītas procesu mērīšanas sistēmai. Izstrūkstošo datu savākšanai tika izstrādāts atbilstošs žurnālfaila formāts, katrai procesa izpildei tiek veidots jauns ieraksts, piemēram, biznesa procesam „Studenta reģistrācija uz kursu” tiek veidots formāts: <procedūra>,<kurss>,<lietotājs>,<datums>

Formāts var būt atšķirīgs, ja biznesa procesu realizējošai sistēmas funkcijai (procedūrai ir jā saglabā papildus parametri, kā iepriekš dotajā piemērā – ir viens parametrs – kursa kods.

Atskaišu veidošanas komponents tiek izmantots, lai attēlotu rezultātus no atšķirīgām datu noliktavas daļām (datuvēm).

### **5.7.3. Divu līmeņu mērīšanas pieejas aprobācija**

Uzsākot jaunas iniciatīvas, piemēram, e-universitātes ietvaros (piem., vai uzlabojot un paplašinot eksistējošas sistēmas, ir būtiski saprast situāciju ar eksistējošām informācijas sistēmām. Šajā gadījumā ir piemērota sistēmas līmeņa mērīšana. Piemēram, IS efektivitātes noteikšanai izvēlēta sistēmas lietošanas dimensija. Saskaņā ar shēmu (50. attēls), universitātē tiek izmantotas dažādas sistēmas dažādiem mērķiem, bet tās izmanto vienu un to pašu infrastruktūru, tās savstarpēji ir saistītas ar dažādiem interfeisiem un tādēļ ietekmē viena otru. Tām ir arī viena un tā pati potenciālo lietotāju kopa. Šādā situācijā var mērīt sistēmas lietošanu, jo rezultāti būs salīdzināmi savā starpā.

Tālāk detalizētāk tiks aplūkotas sistēmas, kas tiks novērtētas. Īpašības, kas ietekmē sistēmas lietošanu un kas katrai no sistēmām ir jānosaka un jādokumentē ir sekojošas:

- vai sistēmas lietošana ir obligāta vai neobligāta,
- piekļuves kontroles veids,
- vai eksistē lietotāju piekļuves mehānisms, kas ierobežo sistēmas lietošanu.

## LUIS

Pārvaldības IS sistēma (LUIS) tika jau iepriekš aprakstīta, tāpēc kopsavilkuma veidā šeit galvenais: sistēmas piekļuvei tiek izmantotas trīs dažādu veidu pieejas – pašapkalpošanās funkcijām, pamatfunkcijām un publiskajām funkcijām.

Sistēmas (LUIS) lietošana pamatfunkciju gadījumā ir obligāta, pašapkalpošanās funkciju gadījumā dažkārt eksistē alternatīvs veids kā sasniegt vajadzīgo mērķi, piemēram, students tā vietā, lai pats veiktu vajadzīgo darbību sistēmā ar pašapkalpošanās funkcijām, iet pie studiju padomnieka un lūdz palīdzēt un šai gadījumā tās pašas darbības studiju padomnieks var izdarīt sistēmā ar pamatfunkciju palīdzību studenta vietā.

Daudzas funkcijas pašapkalpošanās funkciju daļā ir informatīvas, piemēram, personisko datu apskatīšana dažādos griezumos, tā kā šāda veida funkciju izmantošana ir neobligāta. Tātad var teikt, ka sistēmas lietošana pašapkalpošanās funkciju sadaļā nav stingri obligāta, bet arī viennozīmīgi neobligāta tā nav. LUIS publiskās daļas lietošana ir pilnībā neobligāta.

Pamatfunkciju gadījumā eksistē pieejas kontroles mehānisms, kas nosaka, kādām un cik funkcijām katrs konkrētais sistēmas lietotājs var piekļūt. Pašapkalpošanās funkciju gadījumā ir jāpaskata divas atsevišķas lietotāju grupas, kur katrai no tām ir sava funkciju kopa – studenti (studentiem paredzētās pašapkalpošanās funkcijas) un darbinieki (darbiniekiem paredzētās pašapkalpošanās funkcijas). Katrā no šīm funkciju kopām ierobežojumi attiecas arī uz pieejamo informāciju katram konkrētajam lietotājam – pieejami ir tikai ar konkrēto lietotāju saistīti dati. Publiskās funkcijas ir pieejamas lietotājiem bez jebkādas pieejas kontroles un bez papildus ierobežojumiem uz datiem.

## Portāls

Portālam ir divas daļas- viena ar piekļuves kontroli, otra ir publiska. Visas portālā veicamās funkcijas ir neobligātas.

## E-studiju vide Moodle

E-studiju vide Moodle arī izmanto pieejas kontroli. Ja e-kurss tiek izmantots studiju procesā tikai kā papildus materiālu publicēšanas vieta, tad var šādos atsevišķos gadījumos runāt par neobligātu izmantošanu, bet pamatā gadījumu, ja pasniedzējs šādu e-kursu ir izveidojis, tad ar mērķi, lai students to izmantotu. Tomēr ja vērtē sistēmu kopumā sistēmas līmeņa mērīšanai – tad eksistē neobligātas izmantošanas iespēja, jo eksistē alternatīvi veidi, kā izdarīt to pašu darbību (klātienē studijas, materiālu ieguve ar citu studentu starpniecību u.c.).

Tātad sistēmas lietošanas kontekstā būtiski ir izvērtēt, vai sistēmu mērījumu rezultāti būs salīdzināmi. Iepriekš minētajām sistēmām (LUIS, portāls, Moodle) eksistē sistēmu daļas (LUIS pašapkalpošanās funkcijas, portāla ierobežotās piekļuves daļa) vai arī pati sistēma (Moodle), kas pieļauj zināmu sistēmas lietošanas izvēles brīvības vai neobligātuma pakāpi.

Protams, mērīt var arī LUIS pamatfunkciju izmantošanu, lai pieņemtu lēmumus par sistēmas tālāku attīstīšanu, bet šie mērījumi nav viennozīmīgi salīdzināmi ar pārējo sistēmu lietošanas rādītājiem, jo to nosaka šo funkciju lietošanas obligātums. Tāpat arī LUIS publiskās daļas un portāla publiskās daļas izmantošanu var mērīt un salīdzināt (būs salīdzināmi lielumi).

Tālāk aprakstītie mērījumi ir par sistēmu un sistēmu daļu ar zināmu brīvības (neobligātuma) pakāpi salīdzināšanu (nav ietvertas publiskās daļas un LUIS pamatfunkcijas). Mērījumi, ko var noteikt ir sekojoši: 1) lietotāju skaits, 2) potenciālo lietotāju skaits, 3) sesiju skaits.

Datu savākšanai sistēmas mērīšanas līmenī tika izmantoti sistēmu žurnālfaili vai specializētas žurnāltabulas, kurās tika savākta informācija par katras sistēmas lietošanu. Katras sistēmas gadījumā bija tikai jāanalizē, kurš ieraksts un kādā veidā interpretējams mērījumu kontekstā, kā arī jāprecizē failu un specializēto žurnāltabulu ierakstu formāti.

Pārvaldības IS sistēma LUIS glabā lietošanas informāciju datu bāzes tabulā, kur izvēlēta detalizācijas pakāpe katram ierakstam lielo datu apjomu dēļ reprezentē vienas dienas vienas sistēmas vienas konkrētas funkcijas lietošanas kopsavilkumu (skaitu) atsevišķam lietotājam. Gadā šādā veidā tiek savākti 50 miljoni ierakstu atbilstoši šai detalizācijas pakāpei. E-pastu lietošana tiek iegūta no Unix žurnālfailā, Moodle lietošana tiek fiksēta Apache tīmekļa servera žurnālfailā.

Visi šie žurnālfaili vai tabulas tika analizēti un tika secināts, ka iespējams iegūt datus par sistēmas lietošanu dienas detalizācijas līmenī par katru no sistēmām sekojošā formātā:

<user\_name><activity><IP address><date><the number of performed activities>, kur <user\_name> apzīmē atsevišķu sistēmas lietotāju, <activity> - apzīmē atsevišķu sistēmas funkciju, bet sistēmas līmeņa mērījumu rezultātā šī aktivitāte ir pieslēgšanās sistēmai (login), bet <the number of performed activities> vērtība apzīmēs interesējošo mērījuma vērtību.

Tika veikta sistēmas līmeņa mērīšana iepriekš minētajām sistēmām 2008. gada pavasara semestrī, un rezultāti 3 galvenajām e-universitātes sistēmām doti 6. tabulā. Šajā tabulā nav ietverti rādītāji uz to brīdi, kas raksturo katras sistēmas potenciālo lietotāju skaitu, jo var uzskatīt, ka tas nemainās mērījumu veikšanas perioda laikā. Tā LUIS sistēmai un portālam bija 20478 potenciālie lietotāji, kas reprezentē visu uz atbilstošo semestri reģistrēto studentu skaitu. Savukārt, Moodle gadījumā var runāt par 6861 potenciāliem sistēmas lietotājiem mērījumu periodā, kas reprezentē uz tādiem studiju kursiem reģistrēto studentu skaitu, kam eksistē atbilstošs e-kurss.

Tabulā ietvertais pirmais mērījums katrai sistēmai atspoguļo sistēmas lietotāju skaitu (unikālo lietotāju skaits mēnesī), otrais mērījums atspoguļo, cik procentu no potenciāliem lietotājiem ir aktīvie lietotāji, trešais mērījums ir sesiju skaits mēnesī, un pēdējais mērījums ir vidējais sesiju skaits uz vienu aktīvo lietotāju.

**6. Tabula. Sistēmas līmeņa mērīšanas rezultāti**

	Moodle				Portal				LUIS			
Mar	4158	61%	37225	9,0	6227	30%	52329	8,4	12909	63%	77403	6,0
Apr	4090	60%	35102	8,6	5895	29%	56271	9,5	12083	59%	84300	7,0
May	5256	77%	39488	7,5	6195	30%	57845	9,3	14628	71%	121495	8,3
Jun	3150	46%	26384	8,4	5702	28%	58896	10,3	14976	73%	202448	13,5

Mērījumu rezultāti var tikt izmantoti, lai noskaidrotu, vai vadības stratēģiskie lēmumi tiek realizēti, piemēram, ja portāls tika plānots kā ieejas punkts visām sistēmām, tad mērījumiem būtu jāatspoguļo šis aspekts. Kā redzams tabulā, LUIS lietotāju skaits ir apmēram divas reizes lielāks nekā portālam, kas rāda, ka lietotāji izmanto alternatīvus eksistējošus piekļuves veidus LUIS sistēmai tā vietā, lai pieslēgtos portālam.

Otrs analīzes mērķis var būt sistēmu mērķauditoriju analīze, var novērtēt, cik no potenciāliem lietotājiem ir iesaistīti sistēmu lietošanā un noskaidrot, kurām sistēmām eksistē problēmas to lietošanā. Pētījuma rezultāti uzrāda līdzīgus mērījumus divām sistēmām – vidēji robežās no 60% līdz 70%, bet uz pusi mazāk trešajai sistēmai. Šo mērījumu rezultātu cēloņi tad ir noskaidrojami tālākos pētījumos, kur nepieciešama detalizēta analīze, kas raksturo lietotājus, kas lieto un kādas īpašības raksturo lietotāju, kas nelieto sistēmu, piemēram, gadījumā, ja lietošanas rādītāju grib paaugstināt.

No mērījumu rezultātiem- kaut arī lietotāji portālam ir salīdzinoši neliels % no iespējamajiem lietotājiem, tad sesiju skaits uz 1 lietotāju ir līdzvērtīgs pārējām sistēmām. Atkal iespējami izskaidrojumi, ko pārbaudīt var jau ar detalizēta līmeņa mērīšanu – piemēram, portāla funkcijas tiešām noder noteiktai lietotāju grupai, ko tā lieto intensīvi, vai arī ir tehniski

ienesli – piemēram, caur portālu lietojot kādu no sistēmām, piemēram, e-pastu, lai ekrānā atjaunotu redzamos datus, tiek veikta automātiska atkārtota pieslēgšanās sistēmai.

Līdz ar to rodas nepieciešamība tālākai lietotāju paradumu pētīšanai. Kā dažas no hipotēzēm var minēt:

1) mazs pieejamo funkciju skaits, salīdzinoši ar citām sistēmām,

2) kādu no portālā piedāvātajām funkcijām var realizēt pa tiešo citā sistēmā, apejot portāla ieeju, līdz ar to statistika pieaug attiecīgi citai sistēmai.

Tālāki pētījumi, izmantojot detalizēto mērīšanu, apliecina, ka spēkā bija otrā hipotēze. Cits iespējams tālāku pētījumu virziens – hipotēze “aktīvie studenti”- kas lieto visas sistēmas. Tas izskaidrotu līdzīgos lietošanas procentus, būtu jāveic to lietotāju analīzi, kas sistēmas nelieto, lai varētu noskaidrot cēloņus un uzlabotu mērījumus.

Detalizētā analīzē tālāk būtu jāpēta arī funkciju klases, ko lieto, atkal lielāka uzmanība būtu jāpievērš mazāk lietotām – vai iemesls objektīvs, piemēram, laika periodā funkcijas retu lietošanu izraisa biznesa procesu specifika (uz semestri students reģistrējas tikai 1 reizi periodā), vai iemesli ir citi.

Detalizētai procesu mērīšanai un mērījumu rezultātu tālākas izpētes nodrošināšanai tiek izmantota Procesu mērīšanas sistēma, kas izveidota kā LUIS integrēta sastāvdaļa. Latvijas Universitātē ir ap 250 biznesa procesiem (skaitlis mainīgs laika gaitā), bet tikai daļa no tiem tiek analizēti un uzlaboti. Dažos biznesa procesa pārbūves gadījumos, kuros piedalījies autors (uzņēmšana, reģistrācija uz semestri, reģistrācija uz kursiem), procesu analīzei, aprakstīšanai un uzturēšanai tika izmantots GRADE rīks (Kalnins et al., 1996).

Detalizētā līmeņa mērīšana tika pielietota diviem procesiem – Reģistrācijai uz semestri un E-kursu lietošanai, aprakstu par detalizētās mērīšanas rezultātiem var skatīt pielikumā.

## 5.8. Secinājumi

Lai organizācijā veiktu veikspējas mērīšanu, tad izstrādājot vai izvēloties piemērotu sistēmu, tai jābūt balstītai uz kādu pārvaldības modeli, kas palīdz izvēlēties un precīzi definēt piemērotus mērījumus.

Atbilstoši pārvaldības modelim, uzņēmumi var tikt novērtēti attiecībā pret dažādām perspektīvām, piemēram, finansu, klientu, iekšējo procesu un citām. Lai to varētu izdarīt, jāmodelē un jādokumentē veikspējas indikatoru dažādi aspekti, piemēram, saistība ar veiksmes faktoriem, kādas atskaites nepieciešamas analīzei, kas ir atbildīgie par procesiem un citi.

Procesu mērīšanas sistēma var tikt realizēta kā datu noliktava, galvenokārt, ja ir paredzēta ilgtermiņa analīze, ja ir nepieciešama visu organizācijas datu avotu integrācija, un ja daļa no nepieciešamajiem datiem jau tiek uzkrāta uzņēmumā eksistējošā datu noliktavā. Datu noliktava ir piemērots risinājums arī gadījumā, kad analīze ir nepieciešama uzņēmuma, ne atsevišķa departamenta līmenī.

Veikspējas mērījumu sistēmu implementācijā, izmantojot kā līdzekli datu noliktavu, ir jāizveido, piemērots daudzdimensiju modelis, lai atspoguļotu veikspējas mērīšanas datus – indikatorus, indikatoru īpašības. Lai šādu modeli izstrādātu, par pamatu var kalpot indikatoru formālais pieraksts, kur indikatoru definīcijās ir izdalāmi tādi datu noliktavu daudzdimensiju modelim būtiski elementi kā kvantitatīvi dati (potenciāli fakti) un kvalitatīvi dati (potenciāli dimensiju atribūti).

Lai nodrošinātu lietotājus ar informāciju par indikatoriem, jāizvērtē, kā informāciju par indikatoriem attēlot – ko var nodrošināt tipiski datu noliktavas infrastruktūras komponenti, piemēram, atskaišu rīki (attēlo indikatoru vērtības), un kādi ir iespējamie jaunie komponenti,

kas jāizstrādā, lai lietotāju nodrošinātu ar papildus informāciju par indikatoriem, to īpašību aprakstiem.

Veiktspējas mērīšanas sistēmas izstrādes balstīšana uz indikatoru dzīves ciklu, ļauj lietotājam veikt veiktspējas mērīšanu pēc noteiktas sistēmas, izmantojot pilnīgu informāciju par indikatoriem, ļauj pamatoti veikt pārveidojumus biznesā, un ļauj uzlabot pašu mērīšanas sistēmu, mainot indikatoru īpašību vērtības.

Datu noliktavas izmantošana veiktspējas mērīšanas sistēmās ir jau diezgan bieži izmantotas. Promocijas darbā piedāvātais veiktspējas mērīšanas ietvars ir izstrādāts, lai iegūtu maksimālu labumu no pārbaudītu datu noliktavu tehnoloģiju izmantošanas, realizējot sistēmu indikatoru dzīves cikla atbalstam.

Piedāvātais un veiktspējas pārvaldības ietvaram izmantotais indikatoru dzīves cikla modelis kalpo kā kvalitātes nodrošinājums mērīšanā. Veiktspējas mērīšanas ietvarā izmantotās datu noliktavu tehnoloģijas pārklāj divus būtiskus aspektus indikatoru dzīves ciklā: a) indikatoru mērīšanu un b) daļu no indikatoru analīzes, ko veic atskaišu modulis.

Piedāvātā metode veiktspējas mērīšanai nodrošina laicīgu un dotajam kontekstam atbilstošu lēmumu pieņemšanas procesu. Indikatoru dzīves cikla atbalsta datubāze glabā metadatus, kas definē un plāno indikatoru mērīšanas un kontroles darbības, izmantojot laika shēmas, atbildīgos par procesiem, veicamās aktivitātes atkarībā no rezultātiem u.c. metadatus. Piedāvātais ietvars piedāvā iespēju izveidot veiktspējas kontroles mehānismu, kas tiek iniciēta no mērījumu sistēmas puses, tiklīdz sistēma konstatē problēmu un nepieciešamību pēc detalizētākas analīzes.

Piedāvātais veiktspējas mērījumu ietvars un tam pamatā esošais indikatoru dzīves cikls aprakstīts divos rakstos (Niedritis et al., 2012) un (Niedritis et al., 2011a).

Nepieciešamas arī metodes un pieejas, kā veikt mērīšanu, t.i. kā izmantot veiktspējas mērījumu sistēmu, kas balstīta uz indikatoru dzīves ciklu.

Runājot par piedāvāto divu līmeņu mērīšanas pieeju, universitāte nav vieglākā vieta kur veikt biznesa procesu mērīšanu, analīzi un pārbūvi. Ar tās tradīcijām un akadēmisko brīvību, kas bieži tiek attiecināta arī uz pārvaldības funkcijām universitātē, biznesa procesi Latvijas Universitātē bija (un joprojām ir) neskaidrāki nekā biznesa struktūrās. Tomēr divu līmeņu mērīšana ļauj gūt priekšstatu gan par sistēmu izmantošanu, gan par nepieciešamajiem uzlabojumu virzieniem un problēmu cēloņiem. Rezultāti aprakstīti vairākos darbos (Medvedis et al., 2008), (Medvedis et al., 2008b), (Niedrite et al., 2007a).



## 6. Nobeigums

Promocijas darbā izvirzītā hipotēze “Ir iespējama efektīva, uz inovatīvu arhitektūru balstīta tīmekļa informācijas sistēmu izstrāde, lai realizētu programmatūras kā pakalpojuma pieeju” ir apstiprinājusies. To pierāda promocijas darba rezultāti un to aprobācija divos projektos.

Tīmekļa informācijas sistēmas, kas balstīta uz piedāvāto arhitektūru, izstrādes efektivitāti var izvērtēt, izmantojot sekojošus kritērijus:

- vienas organizācijas ietvaros izstrādātā informācijas sistēma nodrošina organizācijas biznesa procesiem atbilstošu adaptētu funkcionalitāti, kas tiek regulāri lietota un uzlabota. Šī kritērija izpildi apliecina divu līmeņu mērīšanas pieejas pielietojums, kur tiek veikta sistēmas lietošanas un procesu izpildes analīze,
- tīmekļa informācijas sistēma tiek izstrādāta, paredzot iespēju adaptētas šīs sistēmas instances uzturēt un lietot vienlaicīgi vairākās organizācijās. Šī kritērija izpildi apliecina piedāvātās arhitektūras izmantošana divos aprobācijas projektos, kur tika izstrādātas divas tīmekļa informācijas sistēmas, kas pēc tam izmantotas daudzās organizācijās. Viena no tām izmantota projektā, kurā 13 universitātes izmanto vienas TIS 13 adaptētas instances, un otrs projekts, kur citu TIS – Transportlīdzekļu reģistrācijas IS izmanto Ceļu Satiksmes Drošības Direkcijā un citās ārējās organizācijās.

Prakse rāda, ka šo arhitektūru var izmantot dažādās biznesa jomās. Šādas arhitektūras pielietošana var tikt apsvērta sekojošās situācijās: 1) pastāv daļa lietotāju, kas izmanto vairāk nekā vienu TIS instanci vienā laika periodā vai arī secīgi, tā piekļūstot visiem saviem datiem; 2) daudziem lietotājiem jāveic viena un tā pati funkcija, bet katram atšķirīgā veidā.

Promocijas darbā piedāvāto DNATIS arhitektūru raksturo vairākas inovatīvas īpašības.

- TIS adaptācijai piedāvāts izmantot divus adaptācijas līmeņus – organizācijas līmeni un detalizētu lietotāju līmeni. Organizācijas līmeņa adaptācijas rezultātā tiek izveidota TIS adaptēta instance, balstoties uz organizācijas profilu, kas apraksta, kādas funkcijas un kādā veidā tiek izmantotas organizācijā. Lietotāja līmeņa adaptācija balstās uz lietotāja profilu un adaptē TIS individuāla lietotāja specifikai darbā ar sistēmu,
- Arhitektūras ietvaros nedefinēti gan komponenti, kas nepieciešami adaptācijas nodrošināšanai, gan mijiedarbības veidi starp komponentiem. Arhitektūras komponentu mijiedarbības vajadzībām ir nedefinētas adaptācijas operācijas, kas definētas abos adaptācijas līmeņos.
- Arhitektūra atbalsta situāciju, kad lietotāji var strādāt ar vienas TIS sistēmas vairākām instancēm, ja lietotājs ir saistīts ar vairākām organizācijām, kas ir TIS nomnieces. Adaptētās TIS instances tiek integrētas vienā konkrētam lietotājam adaptētā lietotnē, ko lietotājam piegādā, izmantojot integrētu lietotāja saskarni,
- Atbalstīta arī TIS saskarnes dinamiska ģenerēšana, adaptēšana atbilstoši lietotāja darbībām ar sistēmas objektiem,
- Tika piedāvātas metodes, kā risināt datu konfliktus, ko rada daudzo nomnieku vienas instances pieeja SaaS realizācijai, kas tika izmantota šajā arhitektūrā. Piedāvāts aprakstīt katra ieraksta piederību, spēkā būšanas laiku, un biznesa loģikas līmenī definēt kontroles procedūras, kas nosaka, kas un kādas darbības var veikt ar atbilstoša statusa ierakstiem.
- DNATIS arhitektūrā būtiska loma ir arī lietotāja modelim, kura izveidē tiek izmantotas lietotāju profilu integrācijas idejas, lai varētu attīstīt lietotāja modeli un varētu savākt datus par lietotāju. Lietotāja modelis tiek izmantots gan TIS adaptācijā, gan

organizācijas veikspējas mērīšanas sistēmā, attīstot datu noliktavas modeli, lai tas atbilstu lietotāju vajadzībām pēc datiem.

- DNATIS sastāvā tiek piedāvāts arī mērīšanas ietvars, formāls indikatoru pieraksts, kā arī metode biznesa procesu mērīšanai un uzlabošanai. Piedāvātais organizācijas veikspējas mērīšanas ietvars balstās uz datu noliktavu tehnoloģiju izmantošanu. Datu noliktavu risinājumi papildināti ar informāciju par mērīšanas procesa organizāciju. Promocijas darbā piedāvātais mērīšanas ietvars dod papildus metodi, kas nodrošina gan informatīvu atbalstu mērīšanai un analīzei, gan papildus kontroles iespējas šo uzdevumu veikšanai. Jāņem vērā, ka piedāvātais ietvars ir paredzēts situācijai, kad organizācijā datu noliktava, tās izveides un uzturēšanas procesi un datu noliktavas atskaišu rīki jau eksistē.

## Literatūras saraksts

- [Abbar et al., 2008] Abbar, S., Bouzeghoub, M., Kostadinov, D., Lopes, S., Aghasaryan, A., Betge-Brezetz, S.: A Personalized Access Model: Concepts and Services for Content Delivery Platforms. In: Kotsis, G., Taniar, D., Pardede, E., Khalil, I. (eds.) Proc. of the 10th Int. Conf. on Information Integration and Web-Based Applications and Services, iiWAS '08, pp. 41--47. ACM, New York (2008)
- [Abel et al., 2010] Abel, F., Henze, N., Herder, E., Houben, G.J., Krause, D., Leonardi, E., Building Blocks for User Modeling with Data from the Social Web. Proc. of International Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems at UMAP, (2010)
- [Adenso-Díaz, Cantelli, 2001] Adenso-Díaz, B., Cantelli, A.F. Business process reengineering and university organisation: a normative approach from the Spanish case, Journal of Higher Education Policy and Management, 23(1), pp. 63-73, (2001)
- [ANSI/IEEE Std. 1471-2000], Recommended Practice for Architectural Description of Software-Intensive Systems.
- [Bass et al., 2003] Bass, L., Clements, P., Kazman, R. Software Architecture in Practice, Second Edition, Addison-Wesley Professional, p. 560, (2003)
- [Benefelds, Niedrite, 2004] Benefelds, J., Niedrite, L. Comparison of Approaches in Data Warehouse Development in Financial Services and Higher Education. In: Seruca, I. et al. (eds), Proceedings of International Conference on Enterprise Information Systems, Vol.1, Porto, April 14 - 17, pp. 552-557., (2004)
- [Berkovsky et al., 2008] Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of usermodels for enhanced personalization in recommender systems. Journal of User Modeling and User Adapted Interaction, 18(3), 245–286 (2008)
- [Berkovsky et al., 2009] Berkovsky, S., Kuflik, T., Ricci, F. Cross-representation mediation of user models, Journal of User Modeling and User Adapted Interaction, 19. 35–63, (2009)
- [Berners-Lee et al. 1992] Berners-Lee, T., Cailliau, R., Groff, J., Pollermann, B. World-Wide Web: The Information Universe. Electronic Networking: Research, Applications and Policy, 1(2):74-82, (1992).
- [Body et al., 2002] Body, M., Miquel, M., Bedard, Y., Tchounikine, A. A Multidimensional and Multiversion Structure for OLAP Applications. In: ACM 5th International Workshop on Data Warehousing and OLAP, pp 1–6. ACM, McLean, VA (2002)
- [Bonifati et al., 2001] Bonifati, A., Casati, F., Dayal, U., Shan, M.C.: Warehousing Workflow Data: Challenges and Opportunities. In: Proceedings of the 27th International Conference VLDB2001, pp. 649--652 (2001)
- [Brusilowsky, 1996] Brusilowsky P. Adaptive Hypermedia: An Attempt to Analyze and Generalize. Proceedings of First International Conference on Multimedia, Hypermedia and Virtual Reality 1994. Brusilowsky, P., Streit, N. (Eds.) LNCS 1077, Springer Verlag, 288-304. . (1996)
- [Brusilowsky, 2001] Brusilowsky, P.: Adaptive hypermedia. Journal of User Modeling and User Adapted Interaction, 11 (1/2), pp. 87-110 (2001)
- [Brusilowsky et al., 1998] Brusilowsky, P., Kobsa, A., Vassileva, J., (Eds). Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, (1998).

- [Brusilowsky et al., 2000] Brusilovsky, P., Stock, O., and Strapparava, C., Eds. Adaptive hypermedia and adaptive Web-based systems, LNCS, Vol. 1892, Springer-Verlag, Berlin, (2000).
- [Brusilowsky, Maybury, 2002] Brusilovsky, P., Maybury, M.T. From adaptive hypermedia to adaptive Web. *Communications of the ACM*, 45, 5 (2002), 31-33.
- [Ceri et al., 2002] Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M. *Designing Data-Intensive Web Applications*. Morgan Kaufmann Publishers Inc., 2002.
- [Clements, Northrop, 2002] Clements, P., Northrop, L. *Software Product Lines: Practices and Patterns*. Addison-Wesley, (2002).
- [Cuellar et al., 2006] Cuellar, M. J., McLean, E. R., Johnson, R. D. The measurement of information system use: preliminary considerations. In *Proceedings of the 2006 ACM SIGMIS CPR Conference on Computer Personnel Research*, Claremont, California, USA, April 13 - 15, pp. 164-168., (2006)
- [CWM] Object Management Group: Common Warehouse Metamodel Specification, v1.1,  
<http://www.omg.org/cgi-bin/doc?formal/03-03-02>
- [Desnos, 2002] Desnos, J. A National Data Warehouse Project for French Universities. In Knop, J., Schirmbacher, P., & Mahnic, V. (eds) *Proceedings of the 7th international Conference of European University Information Systems on the Changing Universities - the Role of Technology*, Berlin, March 28 - 30, 2001, pp. 146-152, (2002)
- [De Bra et al., 1999] De Bra, P., Houben, G.J., Wu, H.: AHAM: a Dexter-based Reference Model for Adaptive Hypermedia. In: *Proc. of the 10th ACM Conf. on Hypertext and Hypermedia Hypertext'99*, pp. 147-156, ACM Press (1999)
- [De Bra et al., 2004] De Bra, P., Aroyo, L., Chepegin, V.: The Next Big Thing: Adaptive Web-Based Systems. *Journal of Digital Information*, 5 (1), (2004).
- [DeLone, McLean, 1992] DeLone, W.H., McLean, E.R. Information Systems success: The Quest for the dependent variable, *Information Systems Research*, 3(1), pp. 60-95., (1992)
- [DeLone, McLean, 2002] DeLone, W.H., McLean, E.R. Information Systems Success Revisited. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS-35'02)*, Hawaii, USA, pp. 238-249., (2002)
- [De Virgilio, Torlone, 2005] De Virgilio, R., Torlone, R.: A General Methodology for Context-Aware Data Access. In: *Proceedings of the 4th ACM international Workshop on Data Engineering for Wireless and Mobile Access, MobiDE '05*, pp. 9-15. ACM, New York (2005)
- [Dodds, 2002] Dodds T.: *The UBC e-Strategy Framework*. – University of British Columbia, October 2, 2002
- [Edgar, McQuesten, 2000] Edgar, T., McQuesten, P. (2000) Report of the e-University Task Force,  
<http://www.utexas.edu/faculty/council/1999-2000/minutes/min032000.html>
- [El Sawy, 2000] El Sawy, O. A. *Redesigning Enterprise Processes for e-Business*. McGraw-Hill, 224 p., (2000)
- [Facca et al., 2005] Facca, F., Ceri, S., Armani, J., Demalde, V., *Building reactive Web Applications*, *Proceedings of WWW'05.*, (2005)

- [Fenton, Whitty, 1995] Fenton, N.E., Whitty, R.: Software Quality Assurance and Measurement: A Worldwide Perspective. International Thomson Computer Press, pp. 1--19 (1995)
- [Frank et al., 2008] Frank, U., Heise, D., Kattenstroth, H., Schauer, H.: Designing and Utilizing Business Indicator Systems within Enterprise Models - Outline of a Method. In: Loos, P., Nüttgens, M., Turowski, K., Werth, D. (eds.) Modellierung Betrieblicher Informationssysteme (MobIS 2008) - Modellierung zwischen SOA und Compliance Management, LNI, vol. 141, pp. 89-105. Saarbrücken, Germany (2008)
- [Frasincar et al., 2002] Frasincar, F., Houben, G. J., and Vdovjak, R. (2002). Specification framework for engineering adaptive web applications. In The Eleventh International World Wide Web Conference, Web Engineering Track. <http://www2002.org/CDROM/alternate/682/>.
- [Frasincar et al., 2006] Frasincar, F., Houben, G. J., Barna, P. Hpg: The Hera presentation generator. *Journal of Web Engineering*, 5(2), pp. 175–200, (2006).
- [Fraternali, 1999] Fraternali, P.: Tools and Approaches for Developing Data Intensive Web Applications: a Survey. *J ACM Comput. Surv.* 31(3), 227--263 (1999)
- [Garcia et al., 2006] Garcia, F., Bertoa, M., Calero, C., et al.: Towards a Consistent Terminology for Software Measurement. In: *Information and Software Technology*, Vol. 48, Issue 8, pp. 631-644 (2006)
- [Gauch et al., 2007] Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User Profiles for Personalized Information Access. In: Brusilovsky, P., Kobsa, A., Nejdil, W. (eds.): *The Adaptive Web*. LNCS 4321, pp. 54--89. Springer, Heidelberg (2007)
- [Garlan . et al., 1995] Garlan, D., Allen, R., Ockerbloom, J. Architectural Mismatch or Why It's Hard to Build Systems Out of Existing Parts. In: *Proceedings of 17th International Conference on Software Engineering*, pp.170-185, (1995).
- [Giorgini et al., 2005] Giorgini, P., Rizzi, S., Garzetti, M., Goal-Oriented Requirement Analysis for Data Warehouse Design. In: *8th ACM Int. Workshop DOLAP*, pp 47-56 (2005)
- [Goepp et al., 2006] Goepp, V., Kiefer, F., Geiskopf, F. Design of information system architectures using a key-problem framework, *Computers in Industry*, 57, pp. 189–200, (2006)
- [Golfarelli et al., 1998] Golfarelli, M., Maio, D., Rizzi, S. The Dimensional Fact Model: a Conceptual Model for Data Warehouses. *International Journal of Cooperative Information Systems*, Vol. 7, No 2&3, pp 215-247 (1998)
- [Golfarelli et al., 2006] Golfarelli, M., Lechtenböcker, J., Rizzi, S., Vossen, G. Schema Versioning in Data Ware-Houses: Enabling Cross-Version Querying via Schema Augmentation. *Data and Knowl. Eng.* 59(2), pp. 435–459 (2006)
- [Golfarelli, Rizzi, 2009] Golfarelli, M., Rizzi, S.: Expressing OLAP Preferences. *Scientific and Statistical Database Management*, LNCS, Vol. 566/2009, pp. 83-91, Springer, Heidelberg (2009).
- [Gorton, 2011] Gorton, I., *Essential Software Architecture*, 2nd Edition Springer, 2011
- [Granow, 2006] Granow, R. Strategic E-Learning Implementation at Lübeck University of Applied Sciences: Experiences and Roadmap. In *Book of Abstracts, Online Educa 2006*, Berlin, pp. 265-267, (2006)

- [Halasz, Schwartz, 1990] Halasz, F., Schwartz, M. The Dexter Hypertext reference model. In Moline J.; Benigni, J., Baronas, J. (eds), Proceedings of the Hypertext Standardization Workshop, pp. 95-133 (1990).
- [Harrington, 1991] Harrington, J. H.: Business Process Improvement – The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness. McGraw-Hill (1991)
- [Harrington et al., 1997] Harrington, J. H., Esseling, K. C., Van Nimwegen Business Process Improvement Workbook. McGraw-Hill, 314 p., (1997)
- [HC] HyperContent: Content Management System, <http://hypercontent.sourceforge.net>
- [Houben, 2004] Houben, G.-J. Challenges in Adaptive Web Information Systems: Do Not Forget The Link!, In: Proceedings of ICWE Workshops, pages 3-11, 2004.
- [Imhoff et al., 2003] Imhoff, C., Galemme, N., Geiger, J. G.: Mastering Data Warehouse Design: Relational and Dimensional Techniques, Wiley Publishing, USA, (2003)
- [IMS] IMS Learner Information Package Information Model Specification, <http://www.msglobal.org/profiles/lipinfo01.html>
- [IP] IP Address Geolocation to Identify Website Visitor's Geographical Location. <http://www.ip2location.com/>
- [Isakowitz et al., 1998] Isakowitz, T., Bieber, M., Vitali, F. Web Information Systems. Communications of ACM, 41 (7), pp. 78-80 (1998)
- [Jablonski et al., 2004] Jablonski, S., Petrov, I., Meiler, C., Mayer, U.: Guide to Web Application and Platform Architectures (Springer Professional Computing), SpringerVerlag (2004)
- [Jarke et al., 2000] Jarke, M., List, T., Koller, J.: The Challenge of Process Data Warehousing. In: Proceedings of the 26th International Conference VLDB 2000, pp. 473-483 (2000)
- [Jasinska et al., 2006] Jasinska, M., Grad-Grudzinska, M., Wodecki, A. Training Methodology and Organization of Virtual Campus within Traditional University: What Works, What doesn't Work? In Book of Abstracts, Online EDUCA 2006, Berlin, pp. 272-275, (2006)
- [Jazayeri, 2007] Jazayeri, M.: Some Trends in Web Application Development, In: Future of Software Engineering (FOSE '07). IEEE Computer Society, Washington, DC, USA, 199-213. (2007).
- [Jerbi et al., 2009] Jerbi, H., Ravat, F., Teste, O., Zurfluh, G.: Preference-Based Recommendations for OLAP Analysis. In Proceedings of DaWaK'09, pp. 467-478, (2009)
- [Jones, Song, 2005] Jones, M.E., Song, I. Dimensional Modeling: Identifying, Classifying and Applying Patterns. In: Proceedings of DOLAP '05, pp 29–38. ACM, New York (2005)
- [Kaldeich, Oliveira, 2004] Kaldeich, C., Oliveira, J. Data Warehouse Methodology: A Process Driven Approach. In: LNCS, Vol. 3084, pp 536 – 549 (2004)
- [Kalnins et al., 1996] Kalnins, A., Barzdins, J., Podnieks, K., Zarins, A. et al. Business Modelling Language GRAPES-BM and Related CASE Tools. In Proceedings of International Conference Baltic DB&IS'96, Institute of Cybernetics, Tallinn, pp. 3-16., (1996)

- [Kaplan, Norton, 1996] Kaplan, R.S., Norton, D.P.: The Balanced Scorecard. Harvard Business School Press (1996)
- [Kay et al., 2003] Kay, J., Kummerfeld, B., Lauder, P.: Managing private user models and shared personas. In: Proceedings of the UbiUM Workshop (2003)
- [Kiesling, 2002] Kiesling, W., Foundations of Preferences in Database Systems, VLDB Journal, 2002.
- [Koutrika, Ioannidis, 2004] Koutrika, G., Ioannidis, Y. E.: Personalization of Queries in Database Systems. In Proceedings of 20th International Conference on Data Engineering, pp. 597--608, (2004)
- [Kobsa, 2001] Kobsa, A. Generic user modeling systems. User Modeling and User-Adapted Interaction, 11 (1-2), pp. 49-63, (2001)
- [Koch, 2000] Koch, N. Software Engineering for Adaptive Hypermedia Systems. PhD thesis, Ludwig-Maximilians-University Munich/Germany, (2000).  
<http://www.pst.informatik.uni-muenchen.de/personen/kochn/PhDThesisNoraKoch.pdf>
- [Koch, Wirsing, 2002] Koch, N., Wirsing, M.: The Munich reference model for adaptive hypermedia applications. In De Bra, P., Brusilovsky, P., Conejo, R., eds.: Proc. of Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'02, Springer, LNCS 2347, pp. 213--222, (2002)
- [Kozmina, Niedrite, 2010] Kozmina, N., Niedrite, L.: OLAP Personalization with User-Describing Profiles. In: Forbrig, P., Günther, H. (eds.) BIR 2010. LNBIP, vol. 64, pp. 188--202. Springer, Heidelberg (2010)
- [Kozmina, Solodovnikova, 2011a] Kozmina, N., Solodovnikova, D. On Implicitly Discovered OLAP Schema-Specific Preferences in Reporting Tool. In: J.Grunspenķis (Ed.), Scientific Journal of Riga Technical University, Computer Science, Ser. 5, Vol. 46, Special issue for 10th International Conference on Perspectives in Business Informatics Research, Riga, RTU, pp. 35.-42. (2011)
- [Kozmina, Solodovnikova, 2011] Kozmina, N., Solodovņikova, D. Determining Preferences from Semantic Metadata in OLAP Reporting Tool. In: Niedrite, L., Strazdina, R., Wangler, B. (eds.), Perspectives in Business Informatics Research, Local Proceedings, 10th International Conference, BIR 2011 Associated Workshops and Doctoral Consortium, Riga Technical University, pp. 363-370. (2011)
- [Kozmina et al., 2008] Kozmina, N., Niedrite, L., Solodovnikova, D.: A Knowledge-based Method for Transforming Requirements to Conceptual Model of a Data Warehouse. In: Proceedings of the 8th International Baltic Conference on Databases and Informational Systems (DBIS '08), Estonia, pp. 63-74 (2008)
- [Kuechler, Vaishnavi, 2008] Kuechler, B., Vaishnavi, V.K. On Theory Development in Design Science Research: Anatomy of a Research Project. European Journal of Information Systems 17(5), pp 489--504 (2008)
- [Kueng et al., 2001] Kueng, P., Wettstein, T., List, B.: A Holistic Process Performance Analysis through a Process Data Warehouse. In: Proceedings of the 7th American Conference on Information Systems (AMCIS'01), pp. 349--356 (2001)
- [Kwok et al., 2008] Kwok, T., Thao, N., Linh, L.: A Software as a Service with Multi-tenancy Support for an Electronic Contract Management Application. In: IEEE International Conference on Services Computing, SCC 2008, pp. 179--186 (2008)

- [Laplante et al., 2008] Laplante, P.A., Jia, Z., Voas, J.: What's in a Name? Distinguishing between SaaS and SOA. *IT Professional* 10, 46–50 (2008)
- [Letonika] <http://termini.letonika.lv/>
- [List, Machaczek, 2004] List, B., Machaczek, K.: Towards a Corporate Performance Measurement System. In: *Proceedings of the ACM Symposium SAC'04*, ACM Press, pp. 1344-1350 (2004)
- [LUMoodle] E-learning System Moodle at University of Latvia, [www.estudijas.lu.lv](http://www.estudijas.lu.lv)
- [LUIIS] Management Information System of University of Latvia, [www.luis.lv](http://www.luis.lv)
- [McGarry et al., 2001] McGarry, J., Card, D., Jones, C., et al.: *Practical Software Measurement: Objective Information for Decision Makers*, Addison-Wesley (2001)
- [Medvedis et al., 2008] Medvedis, I., Niedrite, L., Niedritis, A., Treimanis M, Voitkans A. E-University Initiative: Approach, Solutions and Case Studies. In: *Proceedings of the 8th International Baltic Conference on Databases and Information Systems*, Estonia, pp. 141-152 (2008)
- [Medvedis et al., 2008b] Medvedis I., Niedrite L., Niedritis A., Treimanis M, Voitkans A. Approach and Technical Solutions for e-University Initiative', *Postconference proceedings of the 8th International Baltic Conference on Databases and Information Systems*, Tallinn, Estonia, *Frontiers in Artificial Intelligence and Applications* by IOS Press (2008).
- [Mincer-Daszkiewicz, 2003] Mincer-Daszkiewicz, J. Deploying University Study-Oriented System at Polish Universities. In: *Proceedings of 9th International Conference of European University Information Systems*, Amsterdam, 2-4 July, pp. 513–517, (2003)
- [Moodle] Moodle, a Free, Open Source Course Management System for Online Learning, [www.moodle.com](http://www.moodle.com)
- [MS06] Microsoft, “Architecture Strategies for Catching the Long Tail”, April, (2006)
- [Muehlen, 2001] Muehlen, M.: Process-driven Management Informations Systems – Combining Data Warehouses and Workflow Technology. In: *ICECR-4* (2001)
- [Munson, 2003] Munson, J.: *Software Engineering Measurement*. Auerbach Publications (2003)
- [Nair et al., 2007] Nair, R., Wilson, C., Srinivasan, B. A Conceptual Query- Driven Design Framework for Data warehouse. In: *Transactions on Engineering, Computing and Technology*, Vol.19, pp 141–146 (2007)
- [Niederee et al., 2004] Niederee, C., Stewart, A., Mehta, B.,Hemmje, M.:A Multi-Dimensional, Unified User Model for Cross-System Personalization. *The Proceeding of the AVI 2004 Workshop on Environments for Personalized Information Access*.pp. 34-54. (2004)
- [Niedrite et al., 2007a] Niedrite, L., Solodovnikova, D., Treimanis, M., Niedritis, A.: The Development Method for Process-Oriented Data Warehouse. In: Mastorakis, N.E. (ed.) *WSEAS Transactions on Computer Research*, 2(2), pp. 183-190 (2007)
- [Niedrite, Niedritis, 2003] Niedrite, L., Niedritis, A. The Development and Implementation of Unified Information System for Universities in Latvia – a Retrospective View and Conclusions, *Proceedings of the congress EUNIS2003*, Amsterdam, (2003)



- [Niedritis, 1999] Niedritis, A. Latvian University Information System security aspects in different application environments, Proceedings of the congress EUNIS99, Espoo, (1999)
- [Niedritis, 2011] Niedritis, A. Delivery of Consistent and Integrated User's Data within a Multi-Tenant Adaptive SaaS Application. In: Niedrite, L., Strazdina, R., Wangler B. (Eds.), Perspectives in Business Informatics Research. Local Proceedings of 10th International Conference, BIR2011 Associated Workshops and Doctoral Consortium, Riga Technical University, pp. 307-314, (2011)
- [Niedritis, Niedrite, 1998] Niedritis, A., Niedrite, L. Latvian University Information System, Proceedings of the congress EUNIS98, Prague, (1998).
- [Niedritis, Niedrite, 2000] Niedritis, A., Niedrite, L. LAIS – a New Stage in the Informatization of Latvian Universities, Proceedings of the Baltic IT&T'2000, Riga, (2000)
- [Niedritis, Niedrite, 2011] Niedritis, A., Niedrite L. The Adaptation of a Web Information System: a Perspective of Organizations, In: Pokorny, J. Repa, V., Richta K. et al. (Eds) Proceedings of the 19th International Conference on Information Systems Development (ISD2010), Springer, pp. 539- 550, (2011).
- [Niedritis, Niedrite, 2012] Niedritis, A., Niedrite, L. „Adaptation of the Presentation in a Multi-tenant Web Information System”, In: Skersys,T., Butleris R., Butkiene R. (Eds.), Information and Software Technologies, CCIS , Vol. 319, Springer, pp. 176 – 186 (2012).
- [Niedritis et al., 2007] Niedritis, A., Bumanis, A., Niedrite, L. (2007) Student and Employee Self Information/Registration Services. In Proceedings of the International Conference of European University Information Systems EUNIS 2007, Grenoble, (2007)
- [Niedritis et al., 2011a] Niedritis, A., Niedrite, L., Kozmina, N. Performance Measurement Framework with Formal Indicator Definitions. In: Grabis, J., Kirikova M. (Eds), Perspectives in Business Informatics Research, Lecture Notes in Business Information Processing, Volume 90, Springer, p. 44-58. (2011)
- [Niedritis et al., 2011b] Niedritis, A., Niedrite, L., Kozmina, N. Integration Architecture of User Models. In: Niedrite, L., Strazdina, R., Wangler B. (Eds.), Perspectives in Business Informatics Research, Local Proceedings of 10th International Conference BIR2011 Associated Workshops and Doctoral Consortium, Riga Technical University, pp. 323 – 330. (2011)
- [Niedritis et al., 2012] Niedritis, A., Niedrite, L., Zuters, J. Performance Measurement Framework with Indicator Life-cycle Support, In: A.Caplinskas, G.Dzemyda, A.Lupeikiene, O.Vasilecas (Eds.) Databases and Information Systems. Tenth International Baltic Conference on Databases and Information Systems. Local Proceedings, Materials of Doctoral Consortium. Vilnius, Žara, pp. 115-127, (2012).
- [Niemi et al., 2001] Niemi, T., Nummenmaa, J., Thanisch, P. Constructing OLAP cubes based on queries. In: Proc. of the 4th ACM International Workshop on Data Warehousing and OLAP, pp 9–15. ACM Press, (2001)
- [OCL] <http://www.omg.org/spec/OCL/>
- [Oreizy, et al., 1998] Oreizy, P., Rosenblum, D.S., Taylor, R.N.: On the role of connectors in modeling and implementing software architectures. Technical Report 98-04, University of California, Irvine (1998)

- [Pang et al., 2004] Pang, C., Taylor, K., Zhang, X., Cameron, M. Generating Multidimensional Schemata from Relational Aggregation Queries. In: Zhou, X. et al. (eds.) WISE'2004, LNCS, vol. 3306, pp 584–589. Springer (2004)
- [Panhilla, 2006] Panhilla, S. Accessing the usage of personalized web information systems, OULU university press (Finland), 2006.
- [Parmenter, 2010] Parmenter, D.: Key Performance Indicators: Developing, Implementing, and Using Winning KPIs. Jon Wiley & Sons, Inc., Second Edition (2010)
- [Poole et al., 2003] Poole, J., Chang, D., Tolbert, D., Mellor, D.: Common Warehouse Metamodel Developers Guide, Wiley Publishing, USA (2003)
- [Popova, Treur, 2007] Popova, V., Treur, J.: A Specification Language for Organizational Performance Indicators. In: Applied Intelligence Journal, 27 (3), pp. 291-301 (2007)
- [Razmerita et al., 2003] Razmerita, L., Angehrn, A., Maedche, A.: Ontology Based User Modeling for Knowledge Management Systems. Proceedings of the 9th International Conference on User Modeling, pp. 213-217, Springer-Verlag (2003)
- [Robal, Kalja, 2004] Robal, T., Kalja, A. e-EDU – An Information System for e-Learning Services. In Barzdins, J., & Caplinskas, A. (eds) Databases and Information Systems: Selected Papers from Sixth International Baltic Conference DB&IS'2004, IOS Press, pp. 288 – 298, (2005)
- [Schwinger, Koch, 2006] Schwinger, W., Koch, N. Modeling Web Applications. In: Kappel, G., Pröll, B., Reich, S., Retschitzegger, W. (Eds), Web Engineering: Systematic Development of Web Applications, Chapter 3, pp. 39–64. John Wiley, 2006.
- [Seddon, 1997] Seddon, P. B. A Respecification and Extension of the DeLone and McLean Model of IS Success, Information Systems Research, 8(3), pp. 240-253., (1997)
- [Stonis et al., 2001] Stonis, J., Niedrite, L., Niedritis, A. The influence of the Information System on administrative processes at the University of Latvia. In Knop, J., Schirmbacher, P., Mahnic, V. (eds) Proceedings of the 7th international Conference of European University Information Systems on the Changing Universities - the Role of Technology, Berlin, March 28 - 30, pp. 119-126., (2001)
- [Solodovnikova, 2007] Solodovnikova, D.: Data Warehouse Evolution Framework. In: Proc. of the Spring Young Researcher's Colloquium On Database and Information Systems (SYRCoDIS'07), Moscow, Russia (2007), [http://ceur-ws.org/Vol-256/submission\\_4.pdf](http://ceur-ws.org/Vol-256/submission_4.pdf)
- [Solodovnikova, 2010] Solodovnikova, D.: Metadata to Support Data Warehouse Evolution. In: Information Systems Development, pp. 627-635, Springer, Heidelberg (2010)
- [Solodovnikova, Niedrite, 2005] Solodovnikova, D., Niedrite, L. Using Data Warehouse Resources for Assessment of E-Learning Influence on University Processes. In: Eder, J. et al. (eds.), Communications of the Ninth East-European Conference on Advances in Databases and Information Systems, Tallinn, Estonia, September 12-15, 2005, pp. 233-248., (2005)
- [Solodovnikova, Niedrite, 2006] Solodovnikova, D., Niedrite, L.: Data Warehouse Adaptation after the Changes in Source Schemata. In: 7th International Baltic Conference on Databases and Information Systems, pp. 52-63, (2006)

- [Solodovnikova, Niedrite, 2011] Solodovnikova, D., Niedrite, L. Evolution-Oriented User-Centric Data Warehouse. In: Pokorny, J., Repa, V., Richta, K. et al. (eds) Proceedings of the 19th International Conference on Information Systems Development, Springer, pp. 721 - 734. (2011)
- [Solodovnikova et al., 2012] Solodovnikova, D, Niedrite, L., and Niedritis, A. Query-Driven Method for Improvement of Data Warehouse Conceptual Model, Proceedings of the 21st International Conference on Information Systems Development (ISD2012), Prato, Italy, Springer (in print).
- [Suh, 2005] Suh., W, Web Engineering principles and techniques, Idea Group publishing, 2005.
- [Šorm, Netrefová, 2006] Šorm, M., Netrefová, H. Complex University Information System at Mendel University Brno. In Lillemaa, T. (ed), Proceedings of 12th International Conference of European University Information Systems, Estonia, Tartu, pp. 94-97, (2006)
- [Tan et al., 2008] Tan, W., Shen, W., Xu, L., Zhou, B., Li, L. A Business Process Intelligence System for Enterprise Process Performance Management. IEEE Transactions on Systems, Man, and Cybernetics, Part C 38(6), pp. 745–756. (2008)
- [Taylor, Van der Hoek, 2007] Taylor, R.N., Van der Hoek, A.: Software design and architecture the once and future focus of software engineering. In: International Conference on Software Engineering, ICSE 2007 (FoSE 2007) (2007).
- [Treimanis, Stonis, 2005] Treimanis, M., Stonis, J. Unified Approach to Implementation of E-university Initiative. In: Proceedings of 11th International Conference on Technology Supported Learning and Training, Berlin, Germany, pp. 186 – 188, (2005)
- [Trice, Treacy, 1988] Trice, A.W., Treacy, M.E. Utilization as a dependent variable in MIS research, SIGMIS Database, 19(3-4), ACM, pp. 33-44., (1988)
- [Tvarožek et al., 2007] Tvarožek, M., Barla, M., Bieliková, M.: Personalized Presentation in Web-Based Information Systems. In: Leeuwen, J., Italiano, G.F., Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) Proc. of the 33rd Conference on Current Trends in Theory and Practice of Computer Science. LNCS, vol. 4362, pp. 796--807. Springer-Verlag (2007)
- [uPortal] uPortal: portal under development by institutions of higher education, <http://www.uportal.org>
- [UWE] UML-based Web Engineering: <http://uwe.pst.ifi.lmu.de/>
- [Valeriano et al., 2006] Valeriano, D., De Virgilio, R., Torlone, R., Di Federico, D.: An Efficient Implementation of a Rule-based Adaptive Web Information System. In: Frasinca, F., Houben, G.-J., Thiran, P. (eds.) Proc. of Int. CAISE Workshop on Web Information Systems Modeling (WISM'06), CEUR Workshop Proceedings, vol. 239 (2006)
- [Velegrakis et al., 2003] Velegrakis, Y., Miller, R.J., Popa, L. Mapping Adaptation under Evolving Schemas. In: 29th Int. Conf. VLDB, pp 584–595. Morgan Kaufmann, Berlin, Germany (2003)
- [Videira, Silva, 2005] Videira, C., Silva, A.R.: Patterns and Metamodel for a Natural-language-based Requirements Specification Language. In: Belo, O., et al. (eds) The 17th Conference on Advanced Information Systems Engineering (CAiSE '05), Porto, Portugal, pp. 189--194 (2005)

- [Wang, Zheng, 2010] Wang, H., Zheng, Z. Software Architecture Driven Configurability of Multi-tenant SaaS Application. Wang, F.L. et al. (Eds.): WISM 2010, LNCS 6318, pp. 418–424, (2010).
- [Wei et al., 2008] Wei, S., Xin, Z., Chang Jie, G., Pei, S., Hui, S.: Software as a Service: Configuration and Customization Perspectives. In: Congress on Services Part II, 2008. SERVICES-2, pp. 18– 25. IEEE, (2008)
- [Westerman, 2001] Westerman P. Data Warehousing Using the Wal-Mart Model. Morgan Kaufmann, (2001)
- [Whyte, Bytheway, 1996] Whyte, G., Bytheway, A. Factors affecting information systems' success, International Journal of Service Industry Management, 7(1), pp. 74-93., (1996)
- [Wrembel, Bebel, 2005] Wrembel, R., Bebel, B. Metadata Management in a Multiversion Data Warehouse. In: OTM Conferences (2), LNCS, vol. 3761, pp 1347–1364. Springer, Heidelberg (2005)
- [W3C] W3C Working Group, Delivery Context: Client Interfaces (DCCI) 1.0, Accessing Static and Dynamic Delivery Context Properties, 2010, <http://www.w3.org/TR/DPF/>
- [Yankelovich et al. 1988] Yankelovich, N., Haan, B., Meyrowitz, N., Drucker, S. Intermedia: The Concept and the Construction of a Seamless Information Environment. IEEE Computer, 21(1):81-96, 1988.
- [Zachman, 2003] Zachman, J. A.: The Zachman Framework: A Primer for Enterprise Engineering and Manufacturing. Zachman International, (2003).
- [Zhang et al., 2007] Zhang, H., Song, Y., Song, H.: Construction of Ontology-Based User Model for Web Personalization. Proceedings of the 11th International Conference on User Modeling. pp. 67-76, (2007)
- [Zhang et al., 2005] Zhang, J., Wang, W., Liu, H., Zhang, S. X- warehouse: Building Query Pattern-Driven Data Warehouse for XML Data. In: Proc. of the International World Wide Web Conference, pp 896–897 (2005)

## 1. Pielikums

### 1.1. DNATIS arhitektūras pielietojums Augstskolu informatīvajā sistēmā

Pielikumā dota informācija par iesaistīto augstskolu izmantotajām funkcijām un lietotāju skaitu (skat. 7. un 8. tabulu), kā arī doti adaptēto lietotāju saskarņu paraugi divās Augstskolu informatīvās sistēmas instancēs (skat. 46. un 47. attēlus).

Augstskolu informatīvā sistēma realizēta, izmantojot DNATIS arhitektūru. 2012. gada novembrī šo sistēmu izmanto 13 augstskolas.

Funkciju skaitu pašapkalpošanās funkciju grupā pa gadiem un pa augstskolām, kuras dažādos gados ir ieviesušas šo funkciju grupu, var redzēt tabulā (skat. 7. tabulu). Pašapkalpošanās funkcijas izmanto 10 augstskolas, visas 13 augstskolas izmanto pilnvaroto funkciju grupu.

Kaut arī Augstskolu informatīvā sistēma tika uzsākta pirms 2000. gada, par nozīmīgu ieviešanu pašapkalpošanās funkciju grupai var runāt no 2006. gada. Funkciju skaita samazinājums pārsvarā gadījumu nozīmē jaunas versijas realizāciju TIS, kur apvienotas vairākas funkcijas. Var pieņemt, ka visas realizētās funkcijas tiek izmantotas LU, un uz doto brīdi ir 101 funkcija pašapkalpošanās funkciju grupā.

#### 7. Tabula. Pašapkalpošanās funkciju grupa

PUB izmantošana (piekļuve saviem datiem): funkcijas	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
1. Latvijas Universitāte	39	77	50	54	78	63	62	69	74	112	103	103	101
2. Liepājas Universitāte		30	18	12	1		30	38	35	40	72	64	45
3. Latvijas Lauksaimniecības universitāte								20	26	27	30	35	27
4. Vidzemes Augstskola							24	33	36	36	34	56	44
5. Ekonomikas un kultūras augstskola									22	26	23	21	20
6. Rēzeknes Augstskola							23	32	33	37	30	30	32
7. Daugavpils Universitāte							21	23	27	32	25	23	20
8. Latvijas Kultūras koledža									21	27	22	19	16
9. Latvijas Policijas akadēmija								29	30	32	23		
10. Sarkanā Krusta medicīnas koledža								32	32	34	30	26	

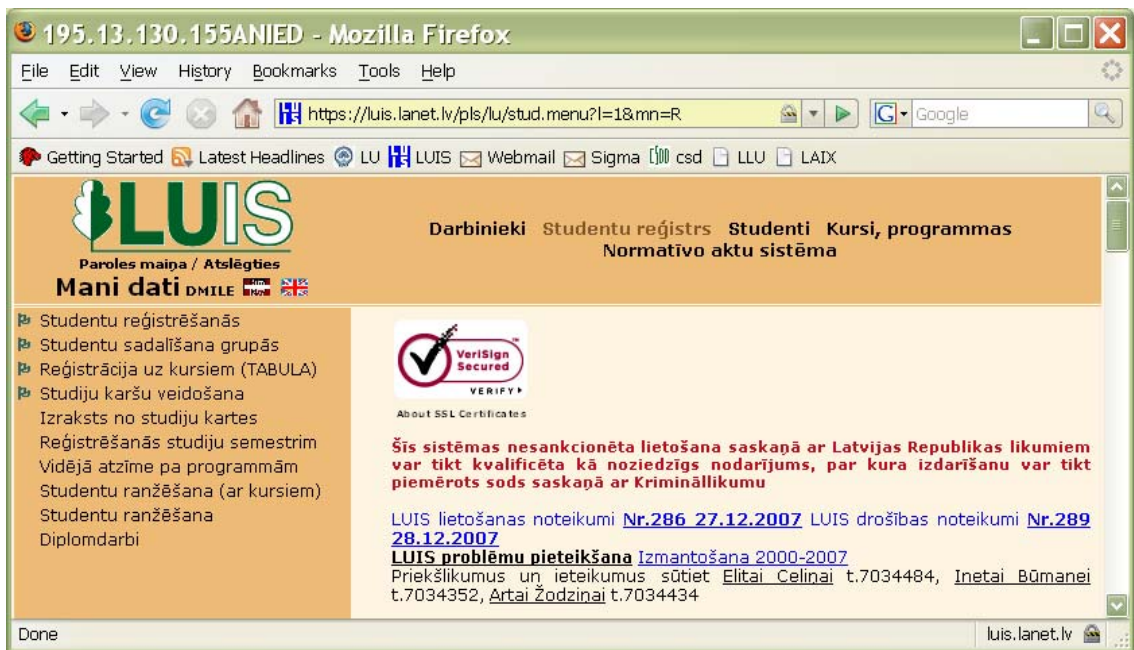
Pamatfunkciju (pilnvaroto funkciju) grupas funkciju un lietotāju skaits pa gadiem un augstskolām dots pielikumā (skat. 8. tabulu).

Daļa augstskolu sākumā (līdz 2005. gadam) izmantoja Augstskolu informatīvo sistēmu LIIS projekta ietvaros, un kad beidzās finansējums no IZM par datu sistēmas ieviešanu un uzturēšanu augstskolām, daļa atsacījās no sistēmas. Daļai augstskolu paralēli iepriekš uzturēja vēl citas sistēmas, ko izvēlējās par piemērotākām katras augstskolas specifikai. No 2006. gada augstskolas pašas no sava budžeta apmaksā Augstskolu Informācijas sistēmas uzturēšanu un jaunu funkciju ieviešanu. Latvijas Policijas akadēmijas gadījumā izmantošana pārtraukta sakarā ar šīs augstskolas pievienošanu LU 2010. gadā.

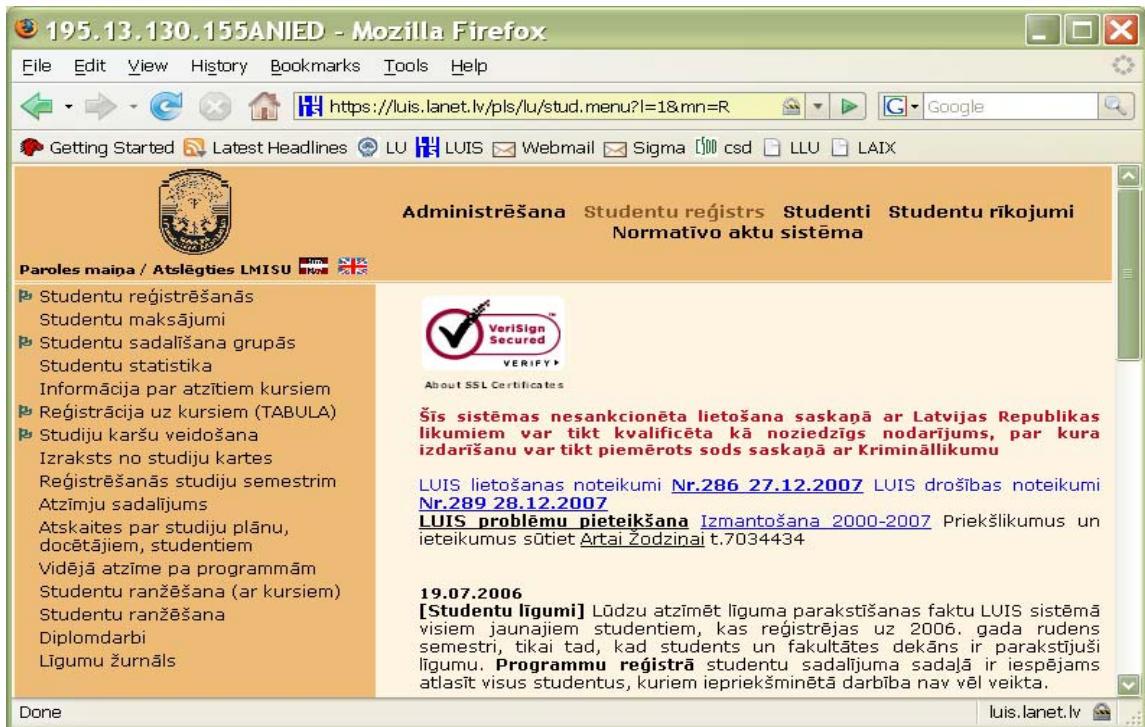
**8. Tabula. Augstskolu informatīvās sistēmas pamatfunkciju grupas funkcijas un lietotāji**

AUTH izmantošana (sankcionētas pieejas funkcijas): lietotāji funkcijas	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
1. Latvijas Universitāte	113 378	252 1107	292 1411	320 1667	371 1922	404 2042	427 2088	447 2251	480 2348	504 2456	512 2635	533 2705	535 2686
2. Latvijas Lauksaimniecības universitāte			3 100	44 471	51 766	74 1031	103 1001	107 998	134 1083	136 1159	135 1186	229 1192	213 1233
3. Liepājas Universitāte	8 74	28 570	33 568	38 679	40 753	40 854	47 819	49 741	48 835	74 964	60 983	49 1073	54 1061
4. Daugavpils Universitāte					1 68	1 137	60 762	56 726	58 795	109 866	113 1027	120 1057	120 1029
5. Rēzeknes Augstskola		6 267	12 206	13 451	30 603	55 780	48 717	40 649	42 683	41 629	45 703	44 753	40 812
6. Ekonomikas un kultūras augstskola	1 13	1 1	1 186	2 254	4 302	20 573	18 488	24 566	21 556	17 576	16 544	19 592	14 559
7. Vidzemes Augstskola		4 138	4 201	3 250	10 347	17 558	31 535	38 540	53 617	39 692	43 679	47 710	40 803
8. Latvijas Policijas akadēmija			1 49				4 186	24 544	22 559	24 492	9 162		
9. Studiju Fonds		2 38	2 77	3 80	6 102	3 87	3 92	3 108	5 104	2 102	2 102	3 116	3 112
10. Latvijas Mākslas akadēmija			3 215	2 242	3 233	3 263	2 180	3 260	5 370	4 389	6 417	6 466	6 476
11. Latvijas Kultūras koledža									7 339	9 423	8 359	6 371	6 334
12. J.Vītola Latvijas Mūzikas akadēmija			5 231	5 241	7 226	6 281	5 223	11 403	14 504	13 458	17 466	16 438	13 419
13. Sarkanā Krusta medicīnas koledža					4 205	4 471	5 352	4 378	4 462	5 418	4 432	4 388	
14. Rīgas Pedagoģijas un izglītības vadības augstskola	4 50	22 151	4 171	3 176	2 142	1 154	1 106						
15. Rīgas Juridiskā augstskola		8 210	3 188	4 256	5 312	4 291	3 290	2 249	1 116	3 213	2 208	5 415	6 403
16. Ventpils Augstskola			6 244	9 299	9 206	7 268	5 142						15 527
17. Latvijas Sporta pedagoģijas akadēmija		7 142	6 210	6 263	7 284	6 195							
18. Latvijas Kultūras akadēmija	1 4	1 6	4 188	3 336	4 315	4 154							
19. Latvijas Nacionālā Aizsardzības akadēmija			2 264	2 254	2 218	3 237							
20. Latvijas Jūras akadēmija			2 98	1 190	2 158	2 156	1 191						
21. Banku augstskola			1 129	1 175				1 13					
22. Rīgas Starptautiskā ekonomikas un biznesa				1 163	1 78	1 13							
23. Rīgas Tehniskā universitāte			1 30	1 21				3 1					
24. Rīgas Ekonomikas augstskola													1 41

Augstskolu IS, kas veidota atbilstoši TIS adaptācijas arhitektūrai, aprobācijas projektā realizētās adaptētās instances LU un LiepU vajadzībām – lietotāju saskarņu paraugi- redzamas 46. un 47. attēlos.



46. attēls. LUIS sistēmas lietotāja ekrāns atbilstoši lietotājam definētajām tiesībām



47. attēls. LiepU sistēmas lietotāja ekrāns atbilstoši lietotājam definētajām tiesībām

## 1.2. DNATIS arhitektūras pielietojums Transportlīdzekļu reģistrā

Pielikumā dota informācija par iesaistīto organizāciju izmantotajām funkcijām un lietotāju skaitu (skat. 9. tabulu), kā arī doti adaptēto lietotāju saskarņu paraugi, divām dažādām organizācijām (skat. 48. attēlu un 49. attēlu). Transportlīdzekļu reģistrācijas sistēmas funkcijas pieejamas vairāk kā 9000 lietotājiem no 36 dažādām ārējām organizācijām, kā arī no 24 rajonu nodaļām. Informāciju par nomnieku skaitu, lietotāju skaitu, izmantoto funkciju skaitu var redzēt 9. tabulā. Tabulas rindu virsraksti atbilst dažādiem ārējiem lietotājiem, kuru nosaukumi aizstāti ar organizāciju kodiem, bet rajonu nodaļas – ar atbilstošo rajonu nosaukumiem (skat. 9. tabulu).

**9. Tabula. Funkcijas un lietotāju skaits dažādās transportlīdzekļu reģistra TIS instancēs**

Lietotāji Funkcijas	2007		2008		2009		2010		2011		2012	
*0			7	27	21	36	25	30	46	44	54	37
*1	45	42	41	35	43	50	41	43	45	49	40	53
*2	24	46	33	40	31	37	25	39	27	52	27	56
*3	107	49	147	44	143	47	126	49	139	56	144	60
*4	33	45	59	43	52	46	47	54	46	67	50	61
*5	75	49	80	45	83	66	85	57	82	66	75	70
*6	283	51	326	50	333	52	327	53	356	72	380	73
*7	21	36	36	36	35	36	33	41	35	52	36	51
*8	36	41	50	137	45	107	40	41	36	52	35	53
*9	98	51	134	48	129	56	114	65	169	77	154	75
*A	75	49	101	45	104	51	90	52	91	56	91	53
*B	23	48	29	45	53	53	52	47	65	58	68	62
*C	45	50	67	44	54	42	53	47	59	54	58	53
*D	217	54	242	58	248	71	234	60	252	73	227	71
*E	290	83	268	81	247	93	241	99	244	117	240	117
*F	34	48	49	39	58	44	51	57	54	78	68	70
*G	166	54	216	51	215	53	194	55	191	58	193	60
*H	28	49	43	49	46	44	40	41	38	52	42	60
*I	28	50	38	52	36	51	37	51	53	69	52	81
*J	28	46	74	46	62	43	55	43	59	54	54	56
*K	81	50	102	55	106	81	107	80	128	94	142	105
*L	1097	163	1263	176	1230	180	1140	157	1183	191	1168	216
*M	130	50	139	71	147	67	150	60	153	75	152	89
*N	52	215	68	191	58	242	57	302	68	363	61	328
*O	56	148	74	146	59	158	62	159	66	188	61	192
*P	28	43	38	37	36	41	40	37	38	53	48	55
*Q							61	39	265	43	321	52
*R	114	52	109	61	89	53	89	62	105	71	103	60
*S	158	46	179	45	156	48	173	50	188	71	185	70
*T	78	56	102	53	111	54	105	53	107	90	106	94
*U	4	18	6	32	8	31	11	26	8	27	8	30
*V	7	37	17	36	13	23	8	26	14	33	18	22
*W	444	62	693	76	774	61	774	63	749	65	702	64
*X	2	9	52	18	44	14	63	14	86	13	155	15
*Y	2	18	1	20	19	18	28	34	25	26	19	33
*Z					23	4	21	5	22	5	23	6



10. Tabula. Funkcijas un lietotāju skaits dažādās transportlīdzekļu reģistra TIS instancēs (Turpinājums)

Lietotāji Funkcijas	2007		2008		2009		2010		2011		2012	
1-NodaļaE	10	141	13	141	7	138	9	151	8	161	12	169
2-Nodaļa	11	132	13	135	17	148	15	149	16	165	12	161
3-Nodaļa	7	147	9	145	9	159	10	151	10	168	11	180
4-Nodaļa	14	144	11	148	10	159	10	159	16	167	11	185
5-Nodaļa	24	138	16	151	21	161	11	169	14	171	17	181
6-Nodaļa	23	149	27	156	28	178	23	176	20	193	22	192
7-Nodaļa	9	133	12	141	11	159	12	157	14	160	9	170
8-Nodaļa	14	141	11	118	16	158	17	169	18	165	23	166
9-Nodaļa	33	153	31	153	32	169	32	173	30	185	30	190
A-Nodaļa	37	158	33	154	21	161	21	168	22	177	24	183
B-Nodaļa	13	130	17	142	13	146	16	159	16	170	12	183
C-Nodaļa	7	130	9	136	7	149	6	148	7	154	7	165
D-Nodaļa	11	126	8	131	15	147	12	148	13	152	8	164
E-Nodaļa	16	146	20	151	24	164	23	168	27	182	20	182
F-Nodaļa	8	140	8	145	10	156	10	163	8	168	8	179
H-Nodaļa	17	132	18	133	15	145	8	156	11	161	16	167
I-Nodaļa	17	146	18	158	15	159	14	166	18	181	16	190
J-Nodaļa	10	125	13	130	16	151	13	155	13	168	8	165
K-Nodaļa	20	150	21	157	16	174	15	184	15	202	17	200
L-Nodaļa	188	211	182	245	143	226	108	231	122	240	121	243
M-Nodaļa	191	248	135	261	160	295	171	322	125	397	113	401
N-Nodaļa	20	150	14	155	15	162	14	169	11	187	11	183
O-Nodaļa	11	62	9	69	11	64	13	69	11	74	14	147
P-Nodaļa	11	147	12	149	11	165	9	168	14	185	12	190
R-Nodaļa	8	121	9	127	9	135	7	142	8	145	9	168
S-Nodaļa	24	150	24	155	24	169	19	172	20	182	23	179
T-Nodaļa	20	144	12	141	13	147	18	144	24	172	25	176

Tālāk doti piemēri, kas raksturo adaptētās lietotāju saskarnes divām dažādām organizācijām - Transportlīdzekļu reģistrācijas TIS sistēmas nomniecēm (skat. 48. attēlu un 49. attēlu).

<a href="#">Dienas atskaite TL Ipašnieki Informācijas meklēšana</a> <a href="#">Vienas dienas atļauja Velo</a> <a href="#">Apliecinājumu drukas Statistika, atskaites</a> <a href="#">Stāvēšanas liegumu noņemšana</a> <a href="#">RN izgatavošana RN atdošana</a> <a href="#">KL reģistrācija TL reģistrācija</a> <a href="#">Invalida karte Reģistrācijas labošana</a> <a href="#">Dokumentu izsniegšana Numuru utilizācija</a> <a href="#">Deklarācija par tehniskajiem datiem</a> <a href="#">Asis, atsperojums labošana</a> <a href="#">Numuru salīdzināšanas meklēšana</a> <a href="#">Numuru salīdzināšana Kuģa salīdzināšana</a> <a href="#">Kuģa motora salīdzināšana</a> <a href="#">Pieteik. tehn. eksp. akta meklēšana</a> <a href="#">Tehniskās ekspertīzes akts</a> <a href="#">Pieteikums transportlīdzekļa pārbūvei</a> <a href="#">Piet. pašmasas noteikš. Pašmasas protokols</a> <a href="#">Numuru salīdzināšanas darbības</a> <a href="#">Vadītāja apliecinājuma dati Nodokļu kalkulators</a> <a href="#">CSDD IZZINAS Brīvie rezervētie numuri</a> <a href="#">Izmantojamie numuri</a>	<a href="#">Daudz bērnu, zemnieku nodeva</a> <a href="#">Komerču apliecinājuma</a> <a href="#">Nomnieku datu labošana</a> <a href="#">RN sodu noņemšana</a> <a href="#">Maksājumu izmantošana</a> <a href="#">NEW RA saraksts RA saraksts</a> <a href="#">Materiālās vērtības RN rezervēšana</a> <a href="#">Adreses Problēmu pieteikšana</a> <a href="#">CoC6 7 8 9 10 11V 11K Likv.sert</a> <a href="#">KAT Maiņa</a> <a href="#">Labojumi (plānotie un realizētie)</a> <a href="#">Administrēšana Vecās Kūlas</a> <a href="#">Neskanētās apliecināšanas</a> <a href="#">Kodifikatori Tipu kodifikators</a> <a href="#">Meklēšana, labošana Fox dati</a> <a href="#">Juridiskie liegumi Žurnāli</a> <a href="#">Veiktās darbības</a>	<a href="#">Testa Lietotājs (Rīga) Iziet 0 1 2 3 4 5 6</a> <a href="#">VK maksājumi Org.mašinas Polises no LTAB</a> <a href="#">Klasifikatori Policijas dati TL reģistrators</a> <a href="#">rokasgrāmata Izdotie sertifikāti TL nodevas pārskaitījumi Apliecinājums par īpašnieka maiņas reģistrāciju CSDD</a> <a href="#">ES RN piesaiste / KL RN izsniegšana</a> <a href="#">Apdr.uzlīmes sagatavošana</a>
---	--	--

48. attēls. Transportlīdzekļu reģistrācijas IS funkciju izvēle (organizācija NR 1)

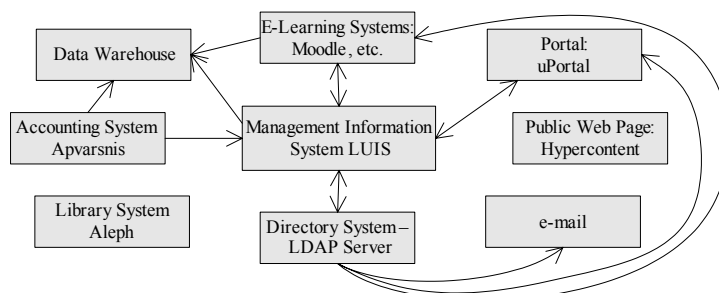
<a href="#">Informācijas meklēšana</a>	<a href="#">Maksājumu izmantošana</a>	<b>Testa Lietotājs2 (Rīga) Iziet 0 1 2 3 4 5 6</b>
<a href="#">Vienas dienas atļauja Velo</a>	<a href="#">RN rezervēšana</a>	<a href="#">TL reģistrators rokasgrāmata</a> <a href="#">Sertifikāts lielgarāta TL</a>
<a href="#">Numuru salīdzināšanas meklēšana</a>	<a href="#">Adrešu <b>Problēmu pieteikšana</b></a>	<a href="#">Izdote sertifikāti</a> <a href="#">Apliecinājums par īpašnieka maiņas reģistrāciju CSDD</a>
<a href="#">Numuru salīdzināšana Kuģa salīdzināšana</a>	<a href="#">CoC6 7 8 9 10 11V 11K Likv.sert</a>	
<a href="#">Kuģa motora salīdzināšana</a>	<a href="#">KAT Maiņa</a>	
<a href="#">Pieteik. tehn. eksp. akta meklēšana</a>	<a href="#">Labojumi (plānotie un realizētie)</a>	
<a href="#">Tehniskās ekspertīzes akts</a>		
<a href="#">Pieteikums transportlīdzekļa pārbūvei</a>		
<a href="#">Piet.pašmasas noteikš. Pašmasas protokols</a>		
<a href="#">Numuru salīdzināšanas darbības</a>		
<a href="#">Vadītāja apliecības dati</a>		

49. attēls. Transportlīdzekļu reģistrācijas IS funkciju izvēlne (organizācija NR 2)

### 1.3. Divu līmeņu mērīšanas pieejas pielietojums Latvijas Universitātē

Pielikumā aprakstītas divas galvenās IS Latvijas Universitātē, kas izmantotas divu līmeņu mērīšanas aprobācijā – LUIS un e-studiju sistēmas, kā arī aprakstīta detalizētā līmeņa mērīšana un tās rezultāti diviem procesiem „Reģistrācija uz semestri” un „E-kursu lietošana”.

Latvijas Universitātes gadījumā ir 9 galvenās informācijas sistēmas, kas tiek izmantotas dažādām vajadzībām, ar dažādiem interfeisiem, un atkarīgi viens no otra datiem (50. attēls). Dotā shēma atspoguļo LU sistēmas uz divu līmeņu mērīšanas pieejas aprobācijas brīdi (Medvedis et al., 2008), kur laika gaitā nomainītas vides atsevišķām sistēmām, piemēram, portāls ir izstrādāts, izmantojot Typo3, un notiek darbs pie daļas no LUIS funkciju lietotāju saskarnes nomaiņu.

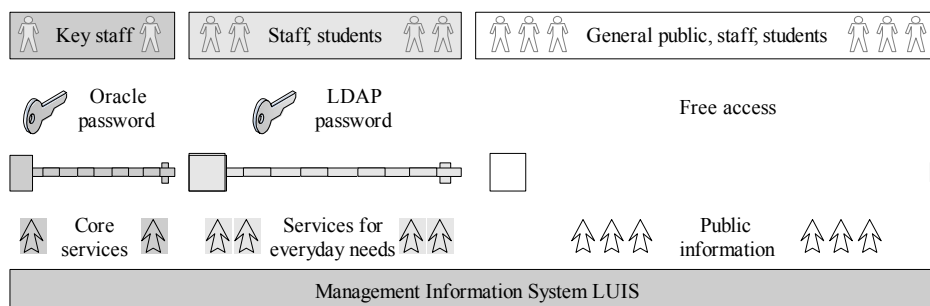


50. attēls. Galvenās informācijas sistēmas un datu plūsmas

#### 1.3.1. Pārvaldības Informācijas Sistēma LUIS

Galvenais Pārvaldības Informācijas sistēmas (LUIS) izveidošanas mērķis Latvijas Universitātē [LUIS] ir atbalstīt studiju procesu administrēšanu [Stonis et al., 2001]. Tā sastāv no sekojošām apakšsistēmām – darbinieku reģistrs un rīkojumu reģistrs, algu sistēma, studentu uzņemšana, studentu reģistrācija uz semestri, studiju maksu uzskaitē, kursu katalogs un studiju plāns, studentu reģistrācija uz studiju kursiem, diplomu pielikumu sagatavošana un citi.

Visi servisi, ko lietotājiem nodrošina biznesa procesi, ir iedalāmi 3 grupās (51. attēls): 1) pamatservisi, kas atbalsta galvenos biznesa procesus, un kas implementēti iepriekš pieminētajās galvenajās LUIS apakšsistēmās, 2) pašapkalpošanās funkcijas un 3) publiskie servisi.



51. attēls. LUIS trīs servisu grupas

Svarīgākie sistēmas lietotāji, kas uztur galveno sistēmas informāciju ir darbinieki. Viņi strādā ar pamatservisiem. Katram lietotājam ir ierobežota pieeja sistēmai, saskaņā ar viņu darba pienākumiem un viņiem sistēmā piešķirtām atbilstošām ORACLE datubāzes privilēģijām.

Cita nozīmīga LUIS funkciju daļa ir tādas, kas nodrošina pieeju personīgai informācijai studentiem un darbiniekiem. Tā ir LUIS daļa, kas paredzēta pašapkalpošanās servisa vajadzībām (Niedritis et al., 2007). Šajā gadījumā studenti un darbinieki izmanto LDAP (*Lightweight Directory Access Protocol*) kontus, lai pieslēgtos sistēmai. LDAP kontu administrēšana ir vēl viena no LUIS funkcijām. Šis LDAP konts vienlaicīgi tiek izmantots arī, lai lietotājs varētu pieslēgties vairākām citām sistēmām: SSH, Moodle un Portāls, kur implementēta piekļuve visām šeit nosauktajām sistēmām no viena piekļuves punkta.

LDAP kontu izveide visiem studentiem notiek, kad potenciālie studenti piesakās studijām. Viņi saņem informāciju par savu kontu, pēc tam, kad viņi ir pierēģistrēti studentu uzņemšanas sistēmā. Konti tiek veidoti vienreiz dienā, izpildot LUIS ģenerētus skriptus kontu izveidei. Uz doto brīdi sistēmā ir vairāk nekā 50000 aktīvi LDAP konti, kā arī vairāk nekā 1000 definētu datubāzes lietotāju (LUIS pamatfunkciju servisu lietotāji). Vidējais dažādu unikālu LUIS lietotāju skaits dienā ir tuvu 500. Atsevišķos laika periodos mērījums var būt ievērojami lielāks (studentu reģistrācijas laikā uz kursiem) vai arī mazāks (brīvlaikos).

Trešā LUIS servisu grupa ir publiskās funkcijas, kas ir pieejamas visiem, bez pieejas kontroles. Šīs galvenokārt ir informatīvas funkcijas, tās publicē informāciju par studiju programmām, par pieejamiem studiju kursiem, dažādu statistiku, piemēram, statistiku par studentu uzņemšanu.

LUIS ir integrēta ar citām informācijas sistēmām, kopējā shēma, kas atspoguļo gan sistēmas, gan interfeisus starp tām, ir dota 50. attēlā. Piemērs tam, kā notiek sistēmu mijiedarbība, ko nodrošina interfeiss, tālāk ir aprakstīts gadījumam, kad sistēmas ir LUIS un e-studiju sistēma Moodle.

Sistēma ir izstrādāta Oracle vidē, lietotāju saskarnes izveide ir tīmekļa bāzēta.

### **1.3.2. E-Studiju sistēmas**

Universitātes e-studiju atbalsta sistēmas var sastāvēt no daudzām atšķirīgām komponentēm, Latvijas Universitātes gadījumā var runāt par Moodle – Studiju pārvaldības sistēmu (*Learning Management System* - LMS), emuāru sistēmu WordPress; Universitātes Portālu –lietotāji var izmantot portālu kā vienotu piekļuves punktu citām universitātes sistēmām. Laika gaitā Latvijas Universitātē ir izmantotas dažādas e-studiju vides - WebCT CE 4 (2002 – 2008) un Moodle no 2007. gada (Moodle), (LUMoodle).

Moodle ir integrēts ar citām galvenajām Informācijas sistēmām: LDAP, Portālu, LUIS un datu noliktavu.

Lietotājs var piekļūt saviem kursiem divos veidos – vai nu caur portālu vai tieši, pieslēdzoties Moodle. Abos gadījumos autentifikācija tiek veikta attiecībā pret LDAP, bet portāla gadījumā nav vajadzīga papildus autentifikācija, jo tiek izmantots Single Sign-On mehānisms. Brīdī, kad lietotājs pieslēdzas sistēmai notiek sinhronizācija – Moodle, izmantojot interfeisu ar LUIS sinhronizē datus par lietotāju (vārdu, uzvārdu, e-pastu) un pārbauda, vai nav mainījušies dati par studenta reģistrāciju uz kursiem. Ja mainījusies studenta reģistrācija uz kursiem LUIS, piemēram, students reģistrējies uz jaunu studiju kursu, tad Moodle automātiski pierēģistrē studentu uz atbilstošu e-kursu. Līdzīgi notiek arī pasniedzēju informācijas sinhronizācija. Ja pasniedzējs ir pievienots kādam studiju kursam LUIS, tad viņš automātiski tiek pievienots arī atbilstošam e-kursam Moodle sistēmā.

### **1.3.3. Reģistrācija uz semestri**

Reģistrācija uz semestri ir biznesa process, kas jāveic katram studentam semestra sākumā. Procesa laikā students apstiprina, ka piedalīsies nākošā semestra studijās un maksas

studenta gadījumā – samaksās studiju maksu. Students veic arī reģistrāciju uz izvēles kursiem, kas tiek piedāvāti studiju programmas plānā.

Reģistrācija uz semestri bija pirmais biznesa process, kas tika uzlabots, izmantojot integrēto pieeju. Galvenās problēmas ar šo biznesa procesu, pirms uzsākt pārbūvi bija sekojošas:

1) liela studentu skaita dēļ (24 000), studiju padomnieki nevarēja pietiekami īsā laikā veikt reģistrāciju uz semestri,

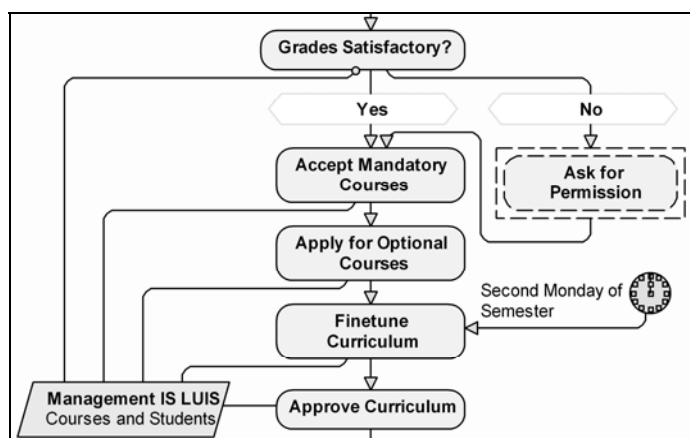
2) ierobežots vietu skaits populāros studijuursos, kopā ar nepārdomātu lēmumu piešķirt vietas pēc principa „kas pirmais brauc, tas pirmais maļ”, bija novedis pie ažiotažas pirmajā dienā, kad notika reģistrācija uz kursiem, tā kā visi studenti gribēja būt pirmie, kas reģistrējas, lai nodrošinātu sev vietu.

Tika veikts biznesa procesu pārbūves pilns cikls (skat. 52. attēlu), kas attēlo vienkāršotu augsta līmeņa skatījumu uz biznesa procesu. Daudzi aspekti tika precizēti, mainīti, un galvenie secinājumi bija sekojoši:

1) studentiem jāveic viss reģistrācijas process Internetā. Klātienē reģistrācija pie studiju padomnieka tiek paredzēta tikai tiem studentiem, kam ir noteikts skaits akadēmisko parādu, finansiālo parādu, vai individuāls studiju plāns,

2) vietas populāros studijuursos jāpiešķir uz iepriekšējā semestrī iegūto atzīmju pamata.

Uzlabotais reģistrācijas process tika implementēts 2005. gadā, turpmāk sekojot līdzīgai procesa norisei, veicot šī procesa monitorēšanu un mērīšanu, saskaņā ar mērījumiem, ko nodrošināja gan operatīvās monitorēšanas dati, gan indikatoru, kas tika uzkrāti datu noliktavā, gan atsauksmēm no studentiem un darbiniekiem.



52. attēls. Reģistrācija uz semestri – vienkāršots modelis

Datu noliktavā tika ielādēti gan operatīvie rādītāji, gan mērījumi procesa beigās. Abu tipu rādītāji tika analizēti procesa beigās, un galvenie secinājumi pirmajā mērīšanas reizē pēc procesa pārbūves bija sekojoši: servera noslodze bija vienmērīga, izņemot pirmo reģistrācijas dienu (ko var izskaidrot ar iepriekšējiem paradumiem par reģistrāciju, kas bija atkarīga no laika), pietrūka vietu populārajās kursos, bet operatīvi reaģējot bija iespējams laicīgi reaģēt, lai plānotu (ja pieejami papildus resursi – telpas, pasniedzēji) papildus nodarbības.

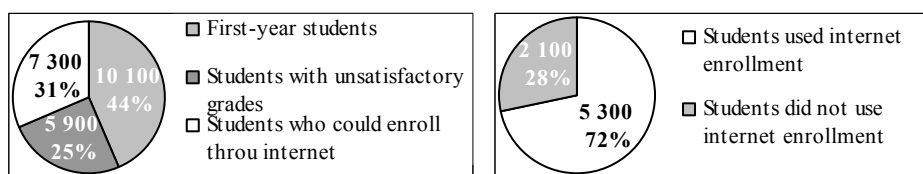
Izmaiņas kursu piedāvājumā turpinājās visu reģistrācijas procesa laiku (pamatā tas tika balstīts uz operatīvās monitorēšanas datiem), bet būtiska bija arī procesa mērīšana pēc procesa beigām, lai varētu pieņemt lēmumus par procesa organizāciju.

Tika analizēta mērķauditorija – studenti (53. attēls). 1. kursa studentiem ir īpaša situācija – rudens semestrī nav iespējami akadēmiskie parādi, tādēļ tiek izdalīti īpaši (10 100

jeb 44%). Interesēja pārējo studentu sadalījums: 7300 jeb 31% teorētiski varēja reģistrēties internetā, 5900 jeb 25% nevarēja izmantot šo pakalpojumu parādu dēļ

Tā kā mērķis ir palielināt studentu skaitu, kam šis pakalpojums ir pieejams, tad interesē divi aspekti: 1) cik no minētajiem 31% reāli izmantoja servisu un kā uzlabot situāciju ar parādniekiem (kāds ir precīzāks parādnieka raksturojums).

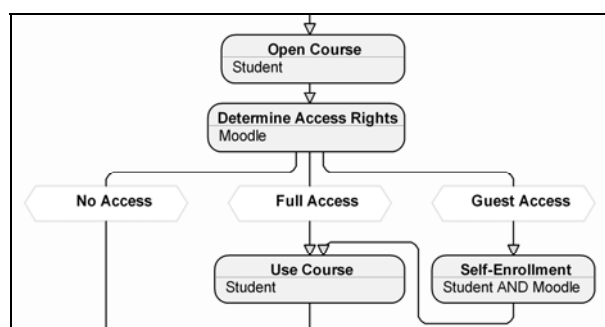
Sadalījums šiem 31% potenciāli iespējamiem procesa dalībniekiem tālāk bija 70% pret 30% (reāli izmantoja/neizmantoja reģistrāciju internetā). Attiecība labāka bija pilna laika studentiem (79% pret 21%) un sliktāka nepilna laika studentu vidū (53% pret 47%). Tā kā mērķis bija palielināt studentu skaitu, kas reģistrējas uz semestri un kursiem Internetā, tad tika secināts, ka jāveic pasākumi, lai iesaistītu nepilna laika studentus pašapkalpošanās servisu izmantošanā. Analizējot parādu veidus, kļuva skaidrs, ka lielāko daļu veido akadēmiskie parādnieki (lielu daļu veido studenti ar 1 parādu), kas ļāva ierosināt izskatīt priekšlikumu, ka studenti ar vienu akadēmisko parādu varētu reģistrēties internetā.



53. attēls. Mērķauditorijas analīze

#### 1.3.4. E-kursu lietošana

E-studijas Latvijas Universitātē ir organizēta, izmantojot e-kursus e-studiju sistēmās (agrāk WebCT, tagad Moodle). E-kursi vienmēr ir piesaistīti vienam no pārvaldības IS LUIS studiju plānā pievienotam kursam. Tas nozīmē, ka e-kursu vienmēr tiek piedāvāts kā daļa no tradicionāli pasniegta kursa. E-kursa izmantošana (apjoms, metodoloģija) notiek pēc pasniedzēju ieskatiem. Daži e-kursi tiek izmantoti tikai kā satura publicēšanas vieta, bet citi tiek izmantoti arī studiju procesa atbalstam (testi, patstāvīgo darbu un pārbaudes darbu iesniegšana un pārbaude, komunikācija u.c.). E-studiju process var tikt aprakstīts kā jebkurš cits process ar procesu modeli, vienkāršots modelis redzams 54. attēlā.



54. attēls. Pieteikšanās uz e-kursiem –frags

Uz pētījuma veikšanas brīdi ir vairāk nekā 560 e-kursu, kas tiek piedāvāti e-studijās, vēl 70 ir izstrādes statusā Moodle sistēmā, e-studijas izmanto vairāk nekā 5000 studenti un 200 pasniedzēji katru semestri.

Lai stimulētu e-studiju attīstību, e-universitātes iniciatīvas ietvaros, vairākus gadus tika finansēta vairāku e-kursu izstrāde. Šiem kursiem tika uzstādītas noteiktas prasības e-kursu uzbūvei un satura apjomam (t.i. kādi ir obligātie komponenti, piemēram, satura lappuses, vārdnīca, testi utt.)