



# RĒZEKNES AUGSTSKOLAS INFORMĀCIJAS SISTĒMAS SERVISORIENTĒTĀS ARHITEKTŪRAS RISINĀJUMS REZEKNES AUGSTSKOLA INFORMATION SYSTEM SERVICE ORIENTED ARCHITECTURE SOLUTION

**Mihails Kijaško**

Rēzeknes Augstskola

Atbrīvošanas alejā 115, Rēzeknē, tālr.+37126625775; e-mail: Mihails.Kijasko@ru.lv

---

**Abstract.** *The goal of this work is to analyze information about main principles of creating information systems using service oriented architecture and create RA integrated information system conceptual model. The proposed system model was developed based on service oriented architecture, with additional description of main principles. Final part contains RA information system service oriented architecture conceptual model with development principles. Using knowledge gathered from analyzes and examinations performed in this work, author gives proposals for integrated information system architecture development principles.*

**Keywords:** *Service oriented architecture, information system, Software requirements specification*

---

## Ievads

Darba ietvaros tika izanalizēti informācijas sistēmu servisorientētās arhitektūras (SOA) izveidošanas galvenie principi un izstrādāts RA vienotās informācijas sistēmas konceptuālais modelis.

Darbā lielākais ieguldījums ir veiktie secinājumi un izstrādātās rekomendācijas. Balstoties uz SOA arhitektūras pamatprincipiem un pašreizējo informācijas sistēmu veicamajām funkcijām, tika noteikti svarīgākie arhitektūras servisi, kā arī atklāta to nozīme funkciju veikšanā.

Darba izstrādāšanas gaitā tika iegūtas ne vien teorētiskas zināšanas par SOA arhitektūras veidošanas principiem un metodoloģiju, bet arī papildinātas autora informācijas analīzes un modelēšanas prasmes.

## 1. Pētījuma objekti un metodes

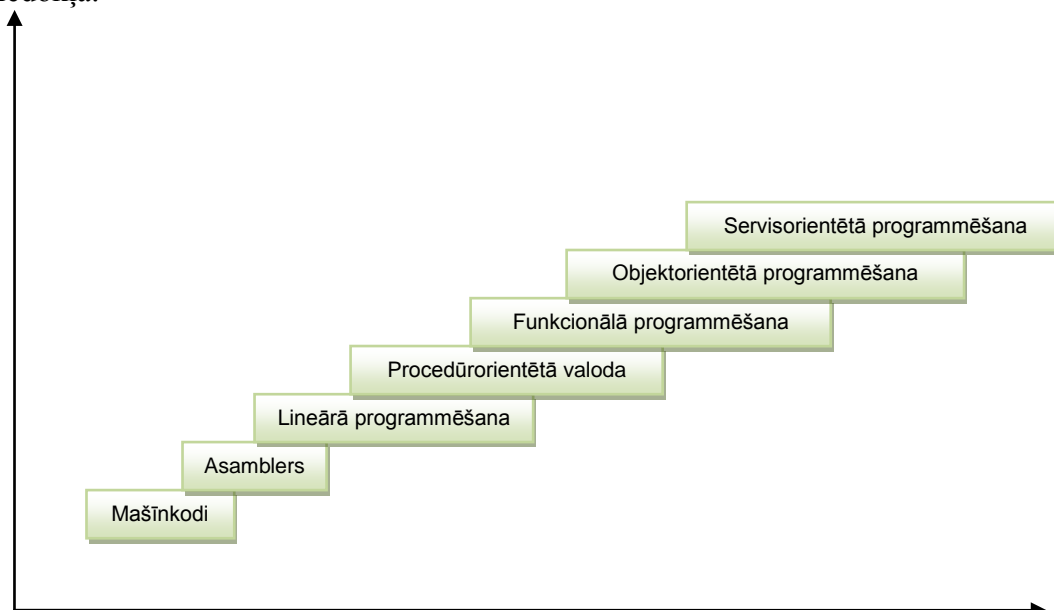
Galvenais iemesls, kāpēc autors izvēlējās tieši SOA arhitektūru, ir tas, ka atšķirībā no klientservisa arhitektūras šeit tiek integrēti biznesa procesi, nevis dati, kas ievērojami paaugstina sistēmas efektivitāti.

SOA – tā ir koncepcija, zināmā mērā ideāls modelis atšķirībā no klientservisa modeļa, kurš vai nu ir, vai nav. Tāpēc, veidojot sistēmu, var ievērot SOA principus, bet pilnībā sasniegt mērķi būtībā nav iespējams. SOA galvenā ideja ir sadalīto biznesa sistēmu komponentu modelis, kas izmanto gatavus, iespējams, dažādu ražotāju produktus, kas automātiski paredz platformneatkarīgu pieeju un atvērto standartu izmantošanu. Šobrīd SOA pamata tehnoloģija ir *Web Services*, kas, protams, nav vienīgais SOA arhitektūras realizācijas variants. Kā arī nepareizi tiks uzskatīts, ka SOA un *Web Services* ir viens un tas pats.

SOA paradīšanai bija vairāki priekšnosacījumi: SQL datubāzu, tīkla protokolu, web servisu, biznesa procesu aprakstīšanas standartu izstrāde.

SOA – ir ne tikai modelis, bet arī ir IT sistēmas realizācijas princips ar orientēšanos uz biznesa procesiem. Ideālā gadījumā SOA ir biznesa procesu kopums, kas nodrošina uzņēmuma darbību, balstoties uz IT komponentiem. Šajā gadījumā otrais tehnoloģiskais komponents ir biznesa procesu vadības (BPM - *Business process management*) līdzekļi.

Apkopojot iepriekšteikto, ir skaidrs, ka SOA projekta realizēšanai ir nepieciešami divi nosacījumi – atbilstoša IT infrastruktūra un pasūtītāja gatavība no darba organizācijas viedokļa.



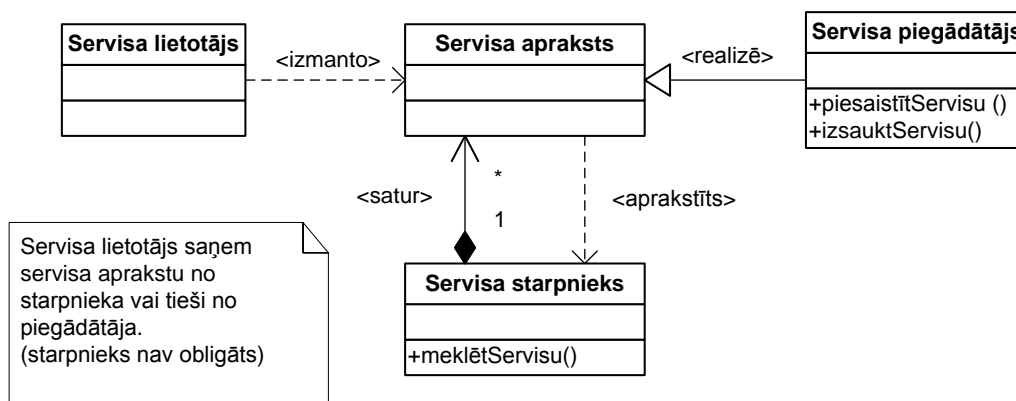
1.att. **Programmēšanas evolūcija**

SOA arhitektūra nav nekāds jauns risinājums, bet tomēr šo tehnoloģiju ļoti vāji piedāvā sistēmu integratori – jo tiem nav izdevīgi piegādāt sistēmu, kura ir platformneatkarīga un kuras atsevišķus komponentus var vienkārši aizvietot. Savukārt tieši programmproduktu izstrādātāji intensīvi cenšas reklamēt un popularizēt SOA. Turklāt integratori, saprotot, ka visa atbildība par sistēmas, kas sastāv no dažādu ražotāju komponentiem, korektu darbību gulstas uz viņu pleciem, nevēlas piedāvāt SOA klientiem.

### 1.1. SOA konceptuālais modelis

Koncepcijas pamatā ir arhitektūras stils, kas definē mijiedarbības modeli starp trim pusēm: servisa lietotāju, piegādātāju un starpnieku. Servisa piegādātājs publicē servisu un nodrošina servisa darbību. Lietotājs, lai atrastu un izmantotu nepieciešamo servisu, var griezties pie servisa repozitorija vai arī caur universālo resursa identifikatoru (URI - *uniform resource identifier*) uzreiz izmantot nepieciešamo servisu.

Metamodelis, kas atspoguļo šīs saites, ir dots 2.attēlā.



2.att. **SOA arhitektūras stila meta modelis**

### **1.2. Arhitektūras stils un principi**

SOA arhitektūras stilu veido parametru un rekomendāciju kopums, kas definē vāji atkarīgu biznesorientēto servisu veidošanas principus. Šie servisi, pateicoties sadalījumam, nodrošina sistēmas elastīgumu un neierobežotu modificēšanas brīvību, kas ļauj reaģēt uz manīgajām biznesa prasībām.

SOA arhitektūra ir uzņēmuma līmeņa arhitektūra, kas nodrošina resursu apvienošanu pēc pieprasījuma. Arhitektūra nodrošina resursu pieejamību visiem uzņēmuma dalībniekiem, izmantojot servisu grupēšanas pieeju. Servisu grupas var veidot kā saliktas aplikācijas, kuras savukārt var inicializēt, izmantojot standarta protokolus (HTTP, SOAP).

Serviss ir programmatūras resurss, ko iespējams atrast un identificēt un kuram ir ārējais apraksts. Ārējais apraksts ir pieejams lietotājiem servisu meklēšanai, sasaistei un izsaukšanai. Servisa piegādātājs nodrošina gan servisa aprakstu, gan servisa apkalpošanas kvalitātes prasības. Servisu vadībai jānotiek, izmantojot politikas, un tādējādi nodrošinātu dinamisku servisu rekonfigurāciju.

Biznesa procesu izmaiņu adaptācijas ātrumu nodrošina IT sistēma uz SOA pamata. Šīs sistēmas ir ļoti elastīgas, pateicoties saskarnes, realizācijas un protokolu piesaistes sadalījumam, kas savukārt ļauj samazināt sistēmas izmaiņu ieviešanas izmaksas.

Servisus ir iespējams izmantot atkārtoti starp iekšējām biznesa vienībām vai arī starp sadalītiem biznesa partneriem.

### **1.3. SOA modelēšanas principi**

Lai būtu iespējams pāriet uz SOA arhitektūru, ir jāņem vērā vairāki jautumi, kuri nav saistīti ar servisu modelēšanu. Tie ir:

- Vai uzņēmums ir gatavs pāriet uz SOA arhitektūru?
- Vai esošā arhitektūra apmierina jūsu prasības?
- Vai SOA ir piemērota arhitektūra jūsu uzņēmumam?
- Kā tiek plānota pārēja uz SOA arhitektūru?
- Vai eksistējošās sistēmas komponentiem arī jāpāriet uz SOA?
- Kā ir plānots vadīt servisu?
- Kāds drošības līmenis ir nepieciešams?

### **1.4. SOA komponenti**

Galvenos SOA arhitektūras komponentus vislabāk aprakstīt ar korporācijas IBM produktu piemēru. IBM ir viens no aktīvākajiem SOA atbalstītājiem un līderis šajā jomā. Kompānija uzskata visu savu programmatūru kā SOA platformu ar pamatu *WebSphere*, papildinātu ar citu zīmolu produktiem (*Tivoli*, *Lotus*, *Rational*, *DB2*). Kopumā SOA infrastruktūra no IBM ietver programmatūru, kas nodrošina pilnu integrāciju visā programmatūras izstrādes ciklā: modelēšana, kodēšana, ieviešana, vadība.

Modelēšanai IBM piedāvā *WebSphere Business Modeler* — līdzekli, ar kura palīdzību speciālisti var modelēt un projektēt procesu struktūru. Papildus ir iespējams izmantot *Rational Software Architect*.

Kodēšanai IBM piedāvā *WebSphere Integration Developer* - aplikāciju izstrādes līdzeklis uz *Eclipse* bāzes, kas ļauj būvēt un ieviest biznesa procesus.

Ar *Integration Developer* palīdzību var izstrādāt kompozitaplikācijas, uzskatot esošos IT resursus, kā servisu, kurus var saistīt savā starpā biznesa procesos. Kā papildus līdzekli var arī izmantot arī *Rational Application Developer*.

Sistēmas ieviešanai tiek izmantots *WebSphere Enterprise Service Bus (ESB)*, kas veido elastīgu komunikāciju infrastruktūru aplikāciju un servisu mijiedarbībai un integrācijai.

ESB uzdevumi:

- ziņojumu pārsūtīšana starp servisiem;

- transporta protokolu konvertēšana starp ziņojuma avotu un servisiem;
- ziņojumu formāta konvertēšana starp ziņojuma avotu un servisiem;
- dažādu biznesa notikumu vadība.

Vadībai ir domāts produkts *WebSphere Business Monitor*, kas ļauj kontrolēt procesu gaitu un galvenos radītājus.

IBM piedāvā arī speciālus produktus testēšanai: *IBM Rational Tester for SOA Quality* (funkcionālo iespēju testēšana) un *IBM Rational Performance Tester Extension for SOA Quality* (veiktspējas testēšana). Šie produkti automatizē darbības un savietojamības testēšanu, pārbaudot sistēmas izturību pie liela lietotāju daudzuma un nosakot maksimālo veiktspēju, kā arī ļauj noteikt funkcionālās problēmas daudzservisu vidē.

Vēl viens IBM produkts *Tivoli Composite Application Manager* ļauj nepārtraukti kontrolēt servisu gatavību, veiktspēju un saturu, veicot regulāras pārbaudes un gatavojot atskaites.

## 2. Rezultāti un to izvērtējums

### 2.1. Konceptuālais modelis

SOA ieviešanas mērķis ir izveidot servisu, kuri realizētu biznesa procesiem nepieciešamo funkcionalitāti. Tas tiek darīts, izmantojot biznesa procesu arhitektūru un biznesa procesu analīzes rezultātus. Konceptuālais modelis attēlo sistēmas arhitektūras uzbūves principus.

SOA izveidē ir nepieciešama šādu augstskolas darbības arhitektūras sastāvdaļu izmantošana:

- 1) biznesa apgabali, t.i., augstskolas darbības sfēras, piemēram, studentu imatrikulācija/eksmatrikulācija, studentu maksājumu uzskaitē;
- 2) biznesa procesi – darbības soļu secība, lai sasniegtu kādu biznesa apgabalam nepieciešamo rezultātu, piemēram, studenta kartītes izveide;
- 3) biznesa aktivitātes – atsevišķi darbības soļi, piemēram, studenta datu ievade vai eksports no reflektantu reģistra;
- 4) biznesa objekti – materiāli vai nemateriāli biznesa pasaules objekti, piemēram, studenta dati, rīkojuma dokumenti.

SOA servisu klasifikācija un steka šablons ir dots 3.attēlā.

Veidojot arhitektūras modeli, ir svarīgi atcerēties, ka:

- 1) katrs augšējā slāņa serviss var izmantot apakšējā slāņa servisu, bet ne otrādi;
- 2) bāzes slāņa datu centriskie servisi viens otru nevar izmantot.

Pareizai SOA slāņu skaita izvēlei ir liela nozīme. Ja to ir maz, tad var rasties problēmas ar atkārtotu servisu izmantošanu, ja to ir daudz – grūtības var sagādāt sistēmas ātrdarbība.

### 2.2. Studenta eksmatrikulācijas uzdevuma servisu arhitektūra

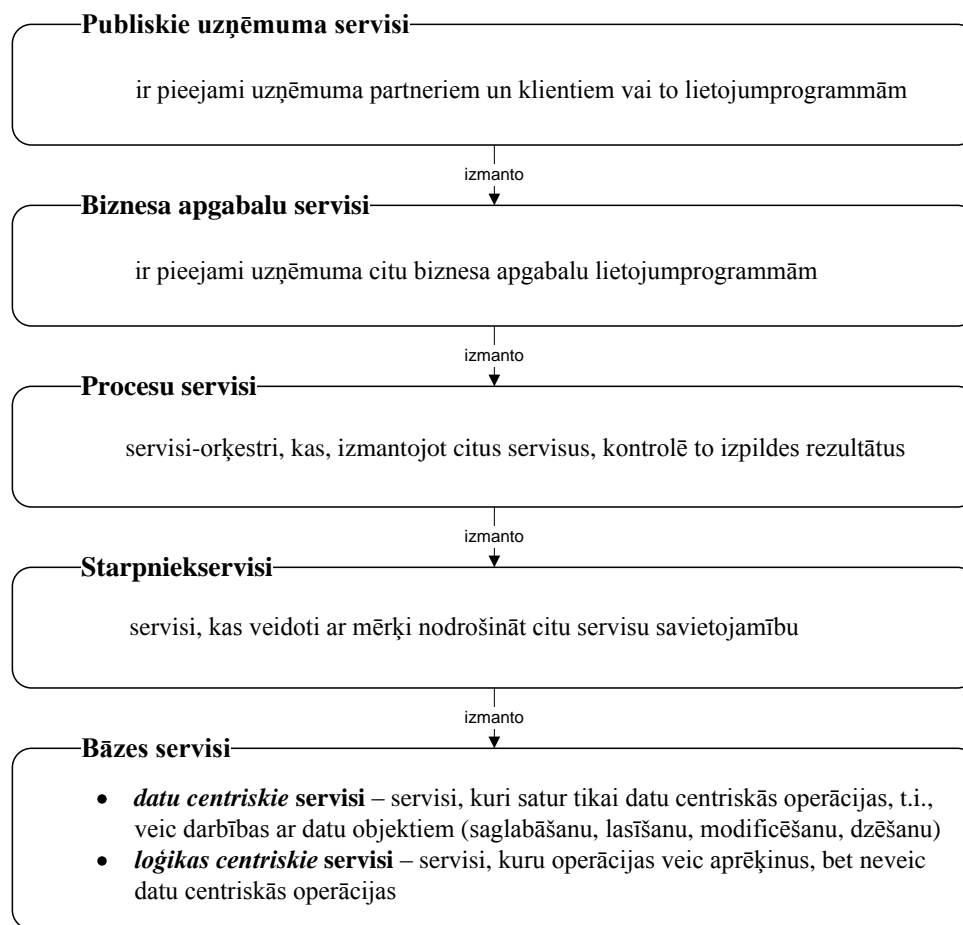
Šajā apakšnodaļā ir aprakstīts lietojuma gadījuma „Studenta eksmatrikulācija” servisorientētās analīzes process, un analīzes rezultātā izveidots modelis, kas var kalpot kā paraugs citu lietojuma gadījumu projektēšanai.

Saskaņā ar SOA izveides tehnoloģiju servisu analīzi var veidot, izmantojot vienu no šādiem paņēmieniem:

- 1) ‘no apakšas’ – analizējot biznesa procesu analīzes rezultātā iegūtās aktivitātes, definējot tām atbilstošās servisu operācijas un integrējot tās servisu komponentos;
- 2) ‘no augšas’ – analizējot uzņēmuma vispārīgākās funkcijas, sadalot tās zemāka līmeņa funkcijās un biznesa procesos.

Pamatojoties uz RA studiju procesa nodrošinājuma darbības veiktās analīzes rezultātiem, tika izveidots lietojuma piemērs, balstoties uz paņēmieni „no augšas” un veicot šādus soļus:

- 1) biznesa servisa definēšana (skat. 1.tabulu);
- 2) procesu servisu noteikšana (skat. 2.tabulu);
- 3) nepieciešamo aplikāciju un atbilstošu saskarņu izvēle (skat. 3.tabulu);
- 4) servisu savstarpējās izmantošanas saikņu konstatēšana (skat. 4.attēlu).



3.att. SOA arhitektūras servisu steka šablons

1.tabula

**Biznesa apgabala servisi**

Nosaukums	Apraksts
<b>Studenta eksmatrikulācijas procedūras serviss</b>	Šajā servisā ir integrētas visas darbības, kuras tiek inicializētas, veicot RA studenta eksmatrikulācijas procedūru: <ul style="list-style-type: none"> <li>• studenta iespējamo saistību pārbaude: ar bibliotēkām, kopmītnēm, personāldaļu, fakultāti;</li> <li>• eksmatrikulācijas rīkojuma izveide;</li> <li>• diploma pielikumu vai atzīmju izraksta sagatavošana;</li> <li>• studenta datu arhivēšana visās sistēmās.</li> </ul>

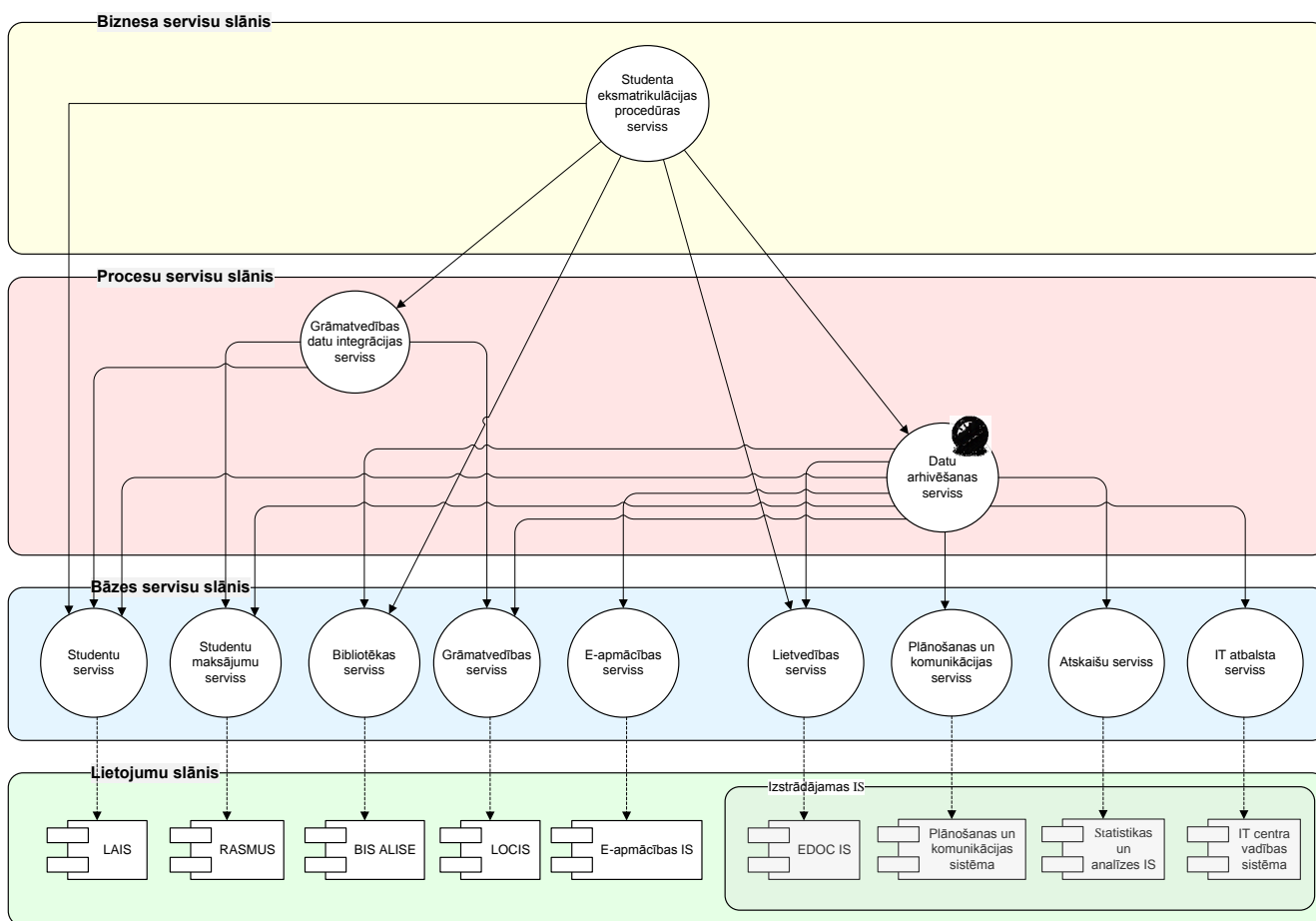
2.tabula

**Procesu servisi**

Nosaukums	Apraksts
<b>Grāmatvedības datu integrācijas serviss</b>	Servisa uzdevums ir iegūt aktuālāko informāciju par studenta finansiālajām saistībām, izmantojot bāzes servissus „Studentu serviss”, „Studentu maksājumu serviss” un „Grāmatvedības serviss”.
<b>Datu arhivēšanas serviss</b>	Datu arhivēšanas serviss veic eksmatrikulēta studenta datu pārvietošanu no visām RA sistēmām uz datu noliktavu.

**Bāzes servisi**

Nosaukums	Apraksts
<b>Studentu serviss</b>	Veic datu ievades un modifikācijas operācijas ar bāzes aplikācijām.
<b>Studentu maksājumu serviss</b>	
<b>Bibliotēkas serviss</b>	
<b>Grāmatvedības serviss</b>	
<b>E-apmācības serviss</b>	
<b>Lietvedības serviss</b>	
<b>Plānošanas un komunikācijas serviss</b>	
<b>Atskaišu serviss</b>	
<b>IT atbalsta serviss</b>	



4.att. Lietojuma gadījuma „Studenta eksmatkulācija” arhitektūra un servisi

**Secinājumi**

Darba izpildes gaitā sasniegti šādi rezultāti:

- izanalizētas RA darbības plūsmas;
- izpētīti informācijas sistēmu lietošanas gadījumi;
- izanalizētas SOA arhitektūras iespējas un izmantošanas iespējas RA IS darbā;
- izstrādāts RA vienotās informācijas sistēmas konceptuālais modelis.

### **Summary**

Working in the course was achieved the following result:

- analyzed RA activity stream;
- researched the information system use cases;
- analysis of SOA architectural options and the possible use in RA IS;
- developed the single RA information system conceptual model.