



## IEGULDĪJUMS TAVĀ NĀKOTNĒ!

Eiropas Reģionālās attīstības fonds

Prioritāte: 2.1. Zinātne un inovācijas

Pasākums: 2.1.1. Zinātne, pētniecība un attīstība

Aktivitāte: 2.1.1.1. Atbalsts zinātnei un pētniecībai

### **Projekts: "Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem"**

Projekta sākuma datums: 2014.gada 1.janvāris.

Projekta beigu datums: 2015.gada 30.jūnijs.

Līguma Nr. 2013/0031/2DP/2.1.1.1.0/13/APIA/VIAA/010

ESF finansējuma saņēmējs: SIA, SWH SETS

Sadarbības partneris: Elektronikas un datorzinātņu institūts (EDI)

### **Projekta aktivitātes Nr. 5.2 "Datu modeļa izstrāde" progresu pārskats**

Pārskats Nr. 37 par periodu no 2015.gada 1.janvāra līdz 2015.gada 30.jūnijam.

## SATURS

1.	Kopsavilkums .....	3
2.	Ievads .....	4
3.	Datu modeļa izstrāde.....	5
3.1.	MEDUS pielietošana MEDUSPILE datu modeļa definēšanai .....	5
3.1.1.	Loģiska entīšu metamodeļa pielietošana .....	5
3.1.2.	Modeļu redaktora izmantošana datu modeļa definēšanas procesā .....	6
	Datu modeļa definēšanas procesā tiek izmantots MEDUS ietvaros izstrādātais Modeļu redaktors [3]. Zīm. 2.....	6
3.2.	MEDUSPILE datu modelis.....	7
3.2.1.	Datu modeļa shēmas apraksts .....	7
3.2.2.	Datu modeļa konstrukcijas īpatnības .....	11
4.	Rezultāti .....	13
5.	Literatūras saraksts.....	14

## 1. Kopsavilkums

Pārskata periodā (2015-01-01 – 2015-06-30) projekta „Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem” aktivitātes "Datu modeļa izstrāde" ietvaros veikti šādi darbi:

1. Datu modeļa MEDUSPILE izstrādi, kas ietver šādu loģisko daļu izstrādi:
  - Lietotāju datu pārvalde;
  - Auditēšanas dati;
  - Dati sadarbībai ar MATLAB;
  - Paraugprojekta pamatdati.
2. Aktivitātes pētnieciskā darbība apspriesta ik nedēļas projekta semināros.

## 2. Ievads

Šis pārskats ir veltīts projekta apakšaktivitātes Nr. 5.2 "Datu modeļa izstrāde" progresā ietvaros veiktajiem darbiem.

Datu modelis apraksta datu elementus, struktūras, kā arī relācijas starp tiem. Tiks uzbūvēts loģiskais datu modelis paraugprojektam, izmantojot MEDUS daudzmodeļu izstrādes tehnoloģiju. Turpmāk, šis modelis tiks izmantots gan datubāzes objektu izveidošanai, gan parēju infrastruktūras objektu automātiskai ģenerācijai transformāciju procesā.

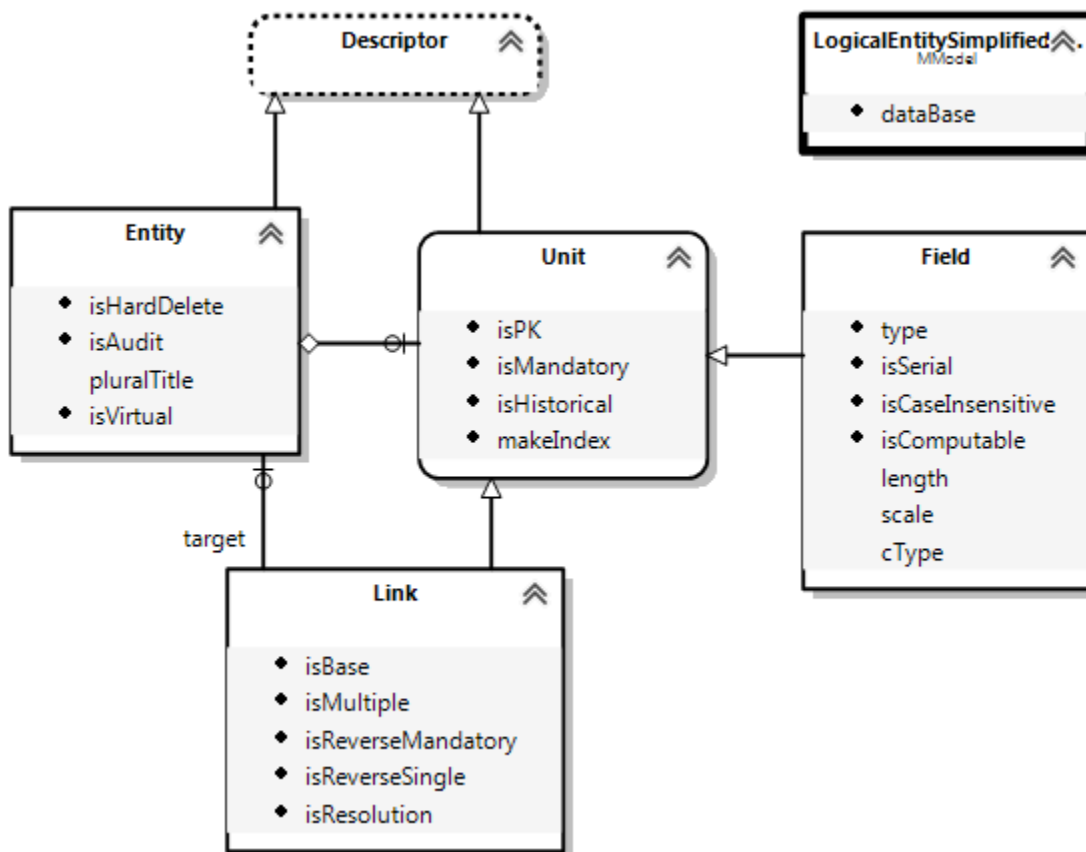
## 3. Datu modeļa izstrāde

### 3.1. MEDUS pielietošana MEDUSPILE datu modeļa definēšanai

MEDUS daudzmodeļu izstrādes tehnoloģija dod plašas iespējas datu modeļa definēšanai. Tehnoloģija ļauj nodefinēt un izmantot savu specifisko metamodeli, ja to pieprasa biznesa īpatnības realizējamajā projektā. Bet mēs savā izstrādē izmantosim jau gatavu entīšu metamodeli [1], uz kura tiks bāzēts mūsu datu modelis. Entīšu metamodelis pilnība atbilst prasību specifikācijā [2] aprakstītām vajadzībām un konceptuālajam datu modelim.

#### 3.1.1. Loģiska entīšu metamodeļa pielietošana

Zīmējumā 1 ir redzama vienkāršota loģiska entīšu metamodeļa shēma.



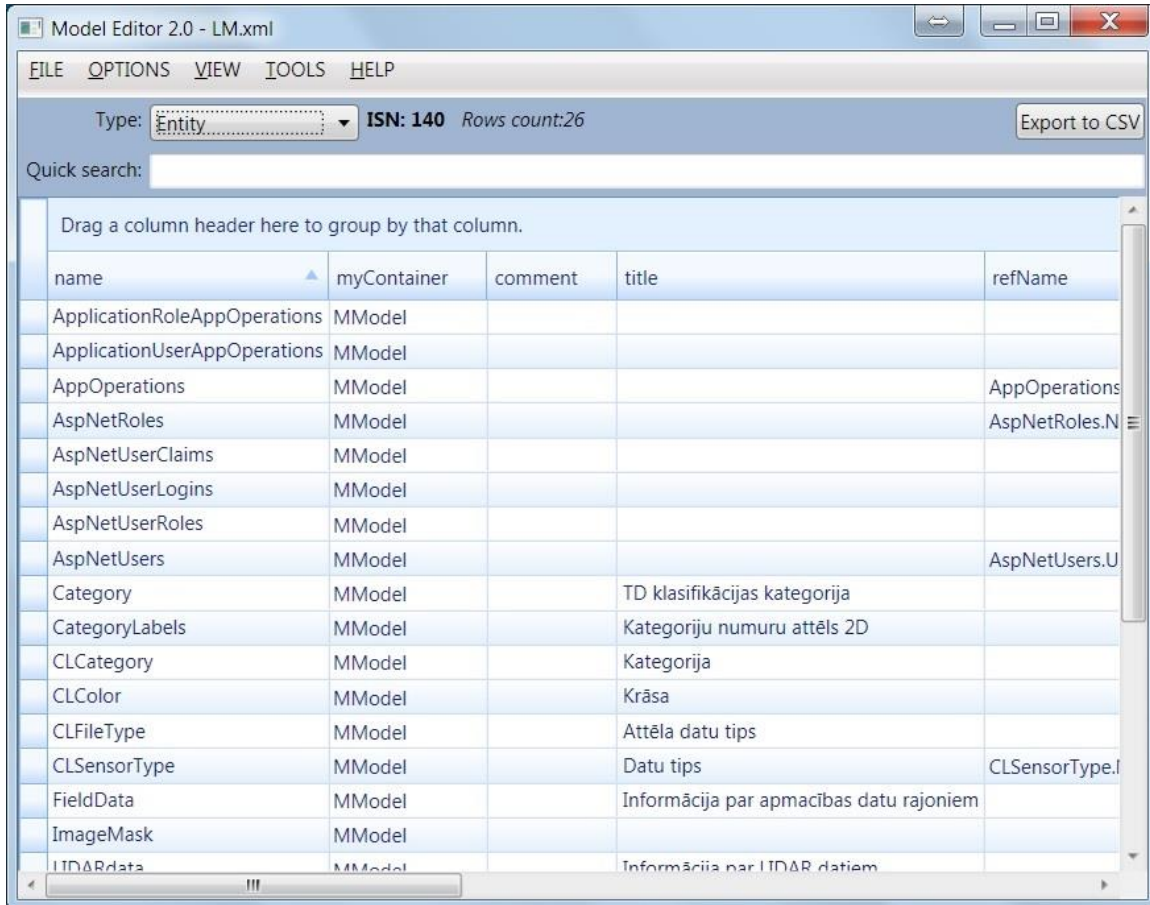
Zīm. 1

Lai attēlotu paraugprojekta datu modeli, sekojošie objektu tipi (metamodeļa objekti) tiks izveidoti:

- Entity
- Field
- Link

### 3.1.2. Modeļu redaktora izmantošana datu modeļa definēšanas procesā

Datu modeļa definēšanas procesā tiek izmantots MEDUS ietvaros izstrādātais Modeļu redaktors [3]. Zīm. 2.



The screenshot shows the Model Editor 2.0 - LM.xml interface. The window title is "Model Editor 2.0 - LM.xml". The menu bar includes FILE, OPTIONS, VIEW, TOOLS, and HELP. The main area displays a table of model entities. The table has columns: name, myContainer, comment, title, and refName. The table contains 26 rows, with the first row being the header. The table is scrollable, and the bottom of the table is partially obscured by a scrollbar. The interface also includes a "Type" dropdown menu set to "Entity", "ISN: 140", "Rows count:26", and an "Export to CSV" button. A "Quick search:" field is located above the table. A message above the table reads "Drag a column header here to group by that column." The table data is as follows:

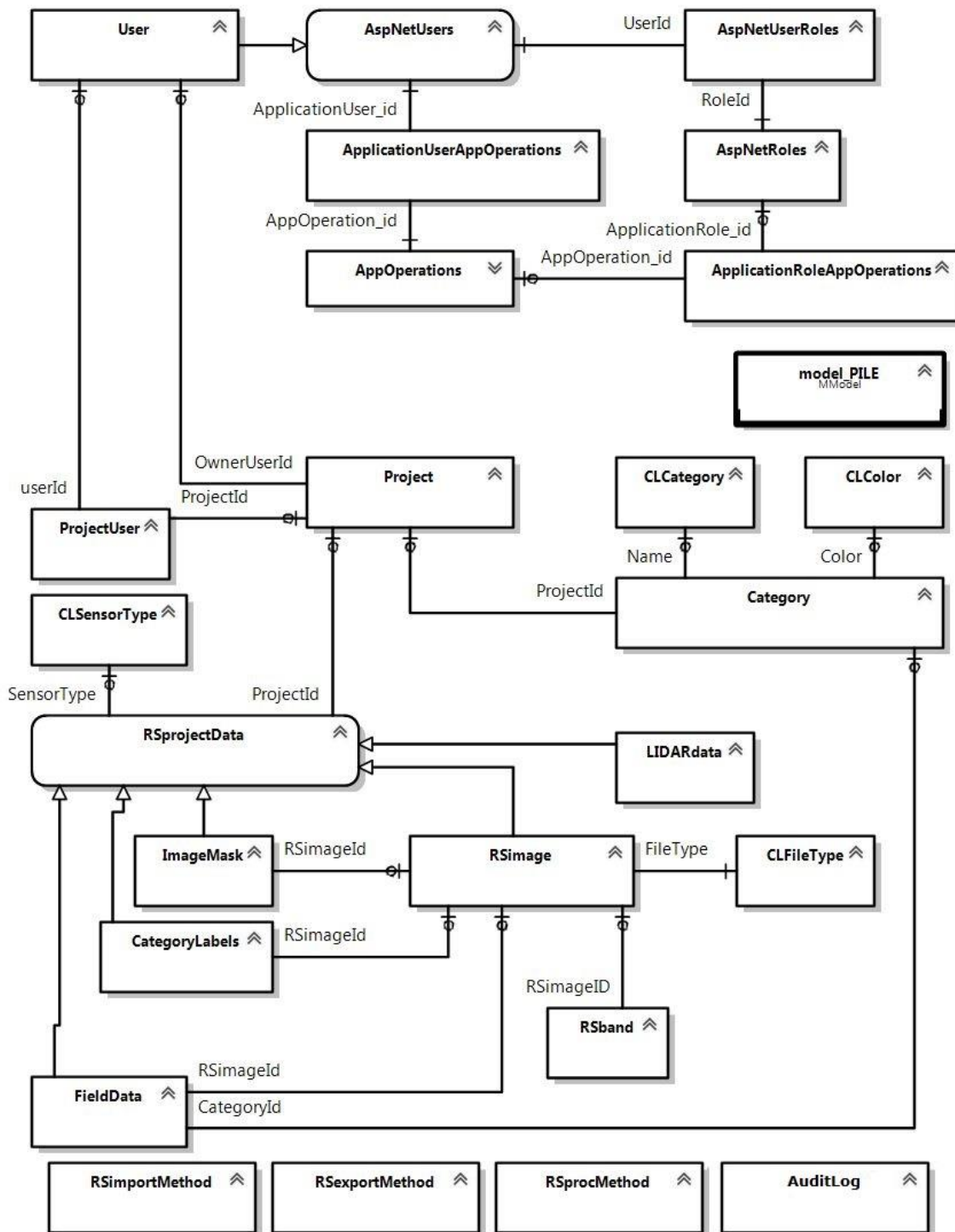
name	myContainer	comment	title	refName
ApplicationRoleAppOperations	MModel			
ApplicationUserAppOperations	MModel			
AppOperations	MModel			AppOperations
AspNetRoles	MModel			AspNetRoles.N
AspNetUserClaims	MModel			
AspNetUserLogins	MModel			
AspNetUserRoles	MModel			
AspNetUsers	MModel			AspNetUsers.U
Category	MModel		TD klasifikācijas kategorija	
CategoryLabels	MModel		Kategoriju numuru attēls 2D	
CLCategory	MModel		Kategorija	
CLColor	MModel		Krāsa	
CLFileType	MModel		Attēla datu tips	
CLSensorType	MModel		Datu tips	CLSensorType.f
FieldData	MModel		Informācija par apmacības datu rajoniem	
ImageMask	MModel			
LIDARdata	MModel		Informācija par LIDAR datiem	

Zīm. 2

## **3.2. MEDUSPILE datu modelis**

### **3.2.1. Datu modeļa shēmas apraksts**

Vadoties no paraugprojekta prasību specifikācijas [2], tika uzbūvēts sekojošs datu modelis (Zīm. 3):



Zīm. 3.

Tabulā 1 tiek dots datu modeļa shēmas apraksts pa entītēm (Entity) un relācijām (Link) starp tām. Entīšu atribūti (Field) tika izveidoti atbilstoši paraugprojekta specifikācijā [2] izvirzītām prasībām un konceptuālam modelim un atstāti ārpus shēmas lai neapgrūtinātu to lasāmību un vizuālo uztveršanu.



<b>Entītija</b>	<b>Apraksts</b>
<b>AspNetUsers</b>	Apraksta lietotāju pamatinformāciju. [4]
<b>AspNetRoles</b>	Apraksta pieejamās lietotāja lomas. [4]
<b>AspNetUserRoles</b>	Piesaista lomas lietotājiem. [4]
<b>ApplicationRoleAppOperations</b>	Piesaista operācijas lomām. [4]
<b>ApplicationUserAppOperations</b>	Piesaista operācijas lietotājiem. [4]
<b>Project</b>	Datu, metožu un rezultātu apkopojums, kas raksturo vienu ar MEDUSPILE palīdzību risināmo uzdevumu. Projektam ir viens īpašnieks-MEDUSPILE lietotājs, kas organizē darbus. Projekta aktivitātes tiek piefiksētas žurnāla ierakstos šim projektam. Projektam tiek izveidota datu mape failu sistēmā, kurā tiek glabāti ar projektu saistītie TD faili, lauka dati, starprezultātu faili (tai skaitā masku un iezīmju faili) un galējie rezultāti.
<b>ProjectUser</b>	Ieraksti, kas sasaista projektus ar lietotājiem. Katram lietotājam, kurš nav administrators, būs redzami tikai projekti, kuru īpašnieki viņu ir iesaistījuši.
<b>User</b>	Programmas lietotājs. Parole tiek glabāta šifrētā veidā. Administrators lietotāju var deaktivēt.
<b>RSImage</b>	Informācija par TD attēlu, kas var saturēt MS, HS un SAR datus, uz ko norāda lauks FileType ('MS'   'HS'   'SAR' ). Paši dati binārā formā glabājas failu sistēmas failā ar nosaukumu Filename. TD attēla faili tiek izveidoti ar MATLAB importa metožu palīdzību, kuru rezultātā dati tiek pārveidoti iekšējā formātā (3D vai 2D matrica). MS un HS attēli tiek glabāti kā 3D matricas ar izmēru Lines x Samples x Bands. SAR attēls ir 2D attēls (Bands=1).
<b>RSband</b>	TD attēla spektrālā josla. SAR attēla gadījumā, kā arī MS un HS attēlu palīgjoslām tai nav definēts viļņa garums (WavelengthNM) un joslas platumš (WidthFWHM). Joslai ir: numurs, tas vienāds ar attēla matricas trešās dimensijas indeksu; vārds, ko attēlo GUI; viļņa garums nm.
<b>LIDARdata</b>	Informācija par LIDAR datiem. Paši dati iekšējā formātā glabājas failā ar nosaukumu Filename un tiek izveidoti ar importa metožu palīdzību.
<b>RSprojectData</b>	Ieraksti, kas saista TD datus (RSImage   LIDARdata) vai citus attēlus (FieldData   ImageMask   CategoryLabels) ar projektu. RecordType norāda

	datu tipu: 'RSImage'   'LIDAR'   'Field'   'Mask'   'Labels'.
<b>Category</b>	TD klasifikācijas (piem. zemes izmantošanas) kategorija. Projektos, kas saistīti ar TD klasifikāciju, jādefinē kategorijas, uz kurām klasificēt datus.
<b>FieldData</b>	Katrai klasificējamajai kategorijai var būt definēti apmācības datu rajoni, t.i. rajoni, kuri ir zināmi kā šai kategorijai piederoši. Katrs FieldData ieraksts satur informāciju par šiem rajoniem, kas definēta SHP faila veidā, kurš saistīts ar kādu no TD attēliem.
<b>ImageMask</b>	Binārs 2D attēls pikseļu kopas izdalīšanai no pamatattēla, uz kuru norāda RSImageID. Attēla izmēri vienādi ar pamatattēla Lines x Samples.
<b>CategoryLabels</b>	Kategoriju numuru attēls 2D, saistīts ar pamatattēlu, uz kuru norāda RSImageID. Attēla izmēri vienādi ar pamatattēla Lines x Samples. Pikseļu vērtības $k$ , kuras ir $>0$ , norāda uz pikseļa piederību kategorijai ar numuru $k$ . Šāds attēls var kalpot gan kā lauka dati, gan klasifikācijas rezultāta fiksēšanai.
<b>RSimportMethod</b>	Informācija par datu importa metodi. Lai metodi izmantotu, jābūt sagatavotam .NET modulim, no kura šo metodi var izsaukt. Ar DestRSImage, DestLIDARdata un DestLabels lauku palīdzību tiek norādīts importa mērķa tips (tieši vienam no šiem laukiem jāpiešķir vērtība "true", pārējiem- "false"). Definētie parametri TxtParameters obligāti jāievada, lai metodi izsauktu. Importa avots jāizvēlas pirms metodes izsaukšanas un jānodod kā viens no teksta parametru virknes <i>TxtParametri</i> locekļiem. <i>TxtParametri</i> ir formātā "param1=<value>; param2=<value>;...", tos izsauktā metode pati izanalizē. Ja metodes izsaukums ir veiksmīgs, no tās atgrieztās struktūras jāizveido attiecīgā tipa (RSImage, LIDARdata vai CategoryLabels) mainīgo, kurš jāieraksta datubāzē ar izvēlēto vārdu un par kuru jāizveido arī <i>RSprojectData</i> ieraksts. Paši importētie dati paliks failu sistēmā, uz tiem norādīs attiecīgais lauks Filename, kurš norādīts izejas parametrā.
<b>RSprocMethod</b>	Informācija par datu apstrādes metodi. Lai metodi izmantotu, jābūt sagatavotam .NET modulim, no kura šo metodi var izsaukt. Ar SourceRSdata1, SourceRSdata2, SourceLIDARdata, MaskImage, LabelImage lauku palīdzību tiek norādīts, kuri no

	<p>ieejas un izejas parametriem tiek izmantoti metodes izsaukumā. Vismaz vienam no izejas parametriem DestRSimage1, DestRSimage2, DestMask vai DestLabels jāpiešķir vērtība "true". Definētie parametri TxtParameters obligāti jāievada, lai metodi izsauktu. TxtParametri ir formātā "param1=&lt;value&gt;;param2=&lt;value&gt;;...", tos izsauktā metode pati izanalizē. Ja metodes izsaukums ir veiksmīgs, no tās atgrieztajām struktūrām jāizveido mainīgo(s), kurš(-i) jāieraksta datubāzē ar izvēlēto(-ajiem) vārdu(-iem) un par kuru(-iem) jāizveido arī <i>RSprojectData</i> ieraksts(-i). Izejas parametriem atbilstošie dati paliks failu sistēmā, uz tiem norādīs attiecīgais lauks Filename, kurš norādīts izejas parametrā.</p>
<b>RSexportMethod</b>	<p>Informācija par datu eksporta metodi. Lai metodi izmantotu, jābūt sagatavotam .NET modulim, no kura šo metodi var izsaukt. Ar SourceRSimage, SourceLIDARdata un SourceLabels lauku palīdzību tiek norādīts eksporta avota tips (tieši vienam no šiem laukiem jāpiešķir vērtība "true", pārējiem - "false"). Definētie parametri TxtParameters obligāti jāievada, lai metodi izsauktu. TxtParametri ir formātā "param1=&lt;value&gt;;param2=&lt;value&gt;;...", tos izsauktā metode pati izanalizē. Eksporta mērķis (faila vārds) jāizvēlas pirms metodes izsaukšanas un jānodod kā viens no teksta parametru virknes <i>TxtParametri</i> locekļiem.</p>
<b>AuditLog</b>	<p>Auditēšanas informācija - informāciju par to, kāds lietotājs un kad ir veicis kādu noteiktu darbību datubāzē. [5]</p>

Tabula 1

### 3.2.2. Datu modeļa konstrukcijas īpatnības

Kopumā, paraugprojekta datu modelis sastāv no četrām loģiskām daļām:

1. Dati lietotāju pārvaldībai – datu pamatstruktūra lietotāju pārvaldībai jau ir nodrošinātā MEDUS ietvari, tam kalpos **AspNetUsers**, **AspNetRoles**, **AspNetUserRoles**, **ApplicationRoleAppOperations**, **ApplicationUserAppOperations** entītes. Paraugprojekta ietvaros tika nodefinēta relācija ar šo struktūru un biznesa atkarīgie papildus dati lietotājiem (User) (sīkāk sk. Nr.3.9.4 "Lietotāju tiesību modelēšanas risinājumu izpēte" [4])
2. Auditēšanas dati – MEDUS ietvars realizē informācijas par lietotāju aktivitātēm datubāzē ierakstīšanu un novērošanu, izmantojot AuditLog entīti. Paraugprojekta tika izmantota šī pieeja.

3. Dati sadarbībai ar MATLAB bibliotēkām – entītes **RSimportMethod**, **RSprocMethod**, **RSexportMethod** ir neatkarīgas ( nav saites ar citām entītēm) un domātas lai nodrošinātu ērtāku izsaušanas veidu MATLAB bibliotēkas metodēm.
4. Paraugprojekta pamatdati – visas pārējas entītes apraksta biznesa specifiskus datus. Pēc prasību specifiskācijas analīzes tika izlemts , ka projekta datu (**LIDARdata**, **Rimage ImageMask**, **CategoryLabel**, **FieldData**) piederība projektam (**Project**) būs ērtāk un efektīvāk aprakstīt ar mantošanas palīdzību, ko piedāvā MEDUS ietvarā jau realizētais loģiskais metamodelis (sk. Zīm.3). **RSprojectData** kļūst par bāzes entīti jebkuram projekta datu objektam. Tā satur norādi uz projektu un kopīgus atribūtus. Līdz ar to, katra tipa projektu objektus, biznesa līmenī, var uzskatīt gan par sava tipa objektiem, gan par **RSprojectData** tipa objektiem (to nodrošinās izstrādātā daudzmodeļu vides arhitektūra). Tas turpmāk dos iespēju saformēt vienotu biznesa objektu un atvieglos tā vizuālo parādīšanu.

## 4. Rezultāti

Aktivitātes ietvaros tika nokonstruēts MEDUSPILE paraugprojekta datu modelis, izmantojot MEDUS ietvara iespējas. Par pamatu tika ņemts ietvara jau realizētais loģiskais metamodelis. Datu modeļa aprakstīšanai tika izmantots MEDUS ietvara modeļu redaktors.

## 5. Literatūras saraksts

1. Nr.3.2 "Biznesa lēmumu atdalīšana no izstrādes lēmumiem";
2. Nr.5.1 "Prasību specifikācijas izstrāde";
3. Nr.2.2 "Universālā modeļa redaktora izstrāde"
4. Nr.3.9.4 "Lietotāju tiesību modelēšanas risinājumu izpēte"
5. Nr.3.5.1 "Auditēšanas izstrāde" progresa pārskats