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. 5.4 "Paraugprojekta programmas kodu ģenerācija no modeļiem" progresa pārskats

Pārskats Nr. 33 par periodu no 2015.gada 1.janvāra līdz 2015.gada 30.jūnijam.

SATURS

1. Kopsavilkums

Pārskata periodā (2015-01-01 – 2015-06-30) projekta „Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem” aktivitātes "Paraugprojekta programmas kodu ģenerācija no modeļiem" ietvaros veikti šādi darbi:

1. Paraugprojekts specifisko prasību realizācija;
2. Uzlaboti un noskaņoti kodu ģeneratori;
3. Uzlabota tehnoloģija, kā ģenerēto kodu papildināt ar programmētāju rakstīto kodu, kas nodrošina projekta specifiskās prasības;
4. Aktivitātes pētnieciskā darbība apspriesta ik nedēļas projekta semināros.

2. Paraugprojektā izmantotie risinājumi

Paraugprojekts izmanto WPF (Windows Presentation Foundation) [1] tehnoloģiju un tiek veidots kā WPF for Windows lietojumprogramma, kura izmanto MySQL datubāzi. [2]
Paraugprojekts tika realizēts izmantojot projekta gaitā izstrādātos metamodeļus: datubāzes metamodeli [3], join metamodeli [4], biznesa objektu metamodeli [5], WCF (Data View) metamodeli [6], skatu metamodeli [7], lietotāju tiesību metamodeli [8] un koda ģenerācijas šablonus (T4 templates).
Jāpiezīmē, ka ne visas paraugprojekta specifikācijas prasības varēja apmierināt ar jau izstrādātajiem koda ģeneratoriem. Protams, bija arī ļoti specifiskas prasības, kuru realizācijas iekļaušana vispārējā tehnoloģiskajā risinājumā nav lietderīga un kuru realizēšanai tika rakstīts kods.
Paraugprojekta izpildes laikā tika uzlaboti un noskaņoti kodu ģeneratori, kā arī uzlabota tehnoloģija, kā ģenerēto kodu papildināt ar programmētāju rakstīto kodu, kas nodrošina projekta specifiskās prasības.

3. Datu pieejas līmenis

3.1. Koda ģenerācija no datubāzes metamodeļa

No datubāzes metamodeļa tiek ģenerēti datu objekti, kas atbilst modelī definētajām datubāzes tabulām.
Atšķirībā no datubāzes metamodeļa izstrādes testa piemēriem, kuros tika izmantota MS SQL datubāze, paraugprojektā izmantotā datubāze ir MySQL. Datubāzes pieejas kodā izmantotā tehnoloģija Entity Framework Code First ļāva izmantot to pašu ģenerācijas kodu, kurš tika izmantots gadījumā, ja izmantotā datubāze ir MS SQL.
Izmantojot MySQL kā datubāzi (atšķirībā no MS SQL), lietojumprogrammas konfigurācijas failā bija jānorāda koda konfigurācijas tips (codeConfigurationType), lai atbalstītu datubāzes migrāciju modeļa izmaiņu gadījumā, jo MySQL datu ir ierobežojumi, kas neļauj izmantot noklusēto migrācijas tabulas struktūru. MySQL datu provaideris nodrošina šo tipu, kas ļauj izmantot Entity FrameFork Code First migrāciju arī MYSQL datubāzei.
Tas nozīmē, ka gan koda ģenerācija MS SQL un MySQL datubāzēm neatšķiras.
Atšķirības ir vienīgi lietojumprogrammas konfigurācijas failā.
Tabulā ir pārskaitītas projektā izmantotās T4 transformācijas datubāzes līmenim:

T4 transformāciju faili	**Ģenerētais kods**	**Apraksts**
DB_PocoGen.tt PocoGen.ttinclude	Entitiespile.cs	Datubāzei atbilstošo POCO objektu ģenerācija
DB_ContextGen.tt ContextGen.ttinclude	DB_TablesContextpile.cs	DBContex klase, kas precizē datubāzes modeļa aprakstu Entity Framework jēdzienos
DB_MEDUSContext.tt MEDUSContext.ttinclude	DB_MEDUSContext.cs	Universālā DBContext klase

4. Biznesa objektu līmenis

Biznesa objektu līmenis nodrošina darbības ar biznesa objektiem.
Tabulā ir pārskaitītas projektā izmantotās T4 transformācijas biznesa objektu līmenim:

T4 transformāciju faili	**Ģenerētais kods**	**Apraksts**
BL_JoinsGen.tt BL_JoinsGen.ttinclude	Joinspile.cs	Tabulu ierakstu kombinēšanas atbalsts
BL_Complexes.tt BL_Complexes.ttinclude	BL_Complexes.cs	Biznesa objektu definīcijas
BL_Complexes.enums.tt Complexes.enums.ttinclude	BL_Complexes.enums.cs	Pārskaitījumi biznesa objektu īpašībām
BL_DataViewsObjects.tt BL_DataViewsObjects.ttinclude	BL_DataViewsObjects.cs	DataView objektu definīcijas
BL_DataViews.tt BL_DataViews.ttinclude	BL_DataViews.cs	DataView objektu funkcionalitāte

4.1. Koda ģenerācija no join metamodeļa

No join metamodeļa tiek ģenerēts kods, kas nodrošina tabulu ierakstu kombinēšanas iespēju, kas balstīta uz objektiem, kuri definēti datubāzes modelī. Tiek ģenerētas arī join objektu CRUD operācijas.

4.2. Koda ģenerācija no biznesa objektu metamodeļa

No biznesa objekta metamodeļa tiek ģenerēts kods, kas definē biznesa objektu definīcijas, kas balstītas uz join objektiem. Tiek ģenerētas arī biznesa objektu CRUD operācijas, kā arī nodrošinājums lietot biznesa objektu atlases kritērijus.

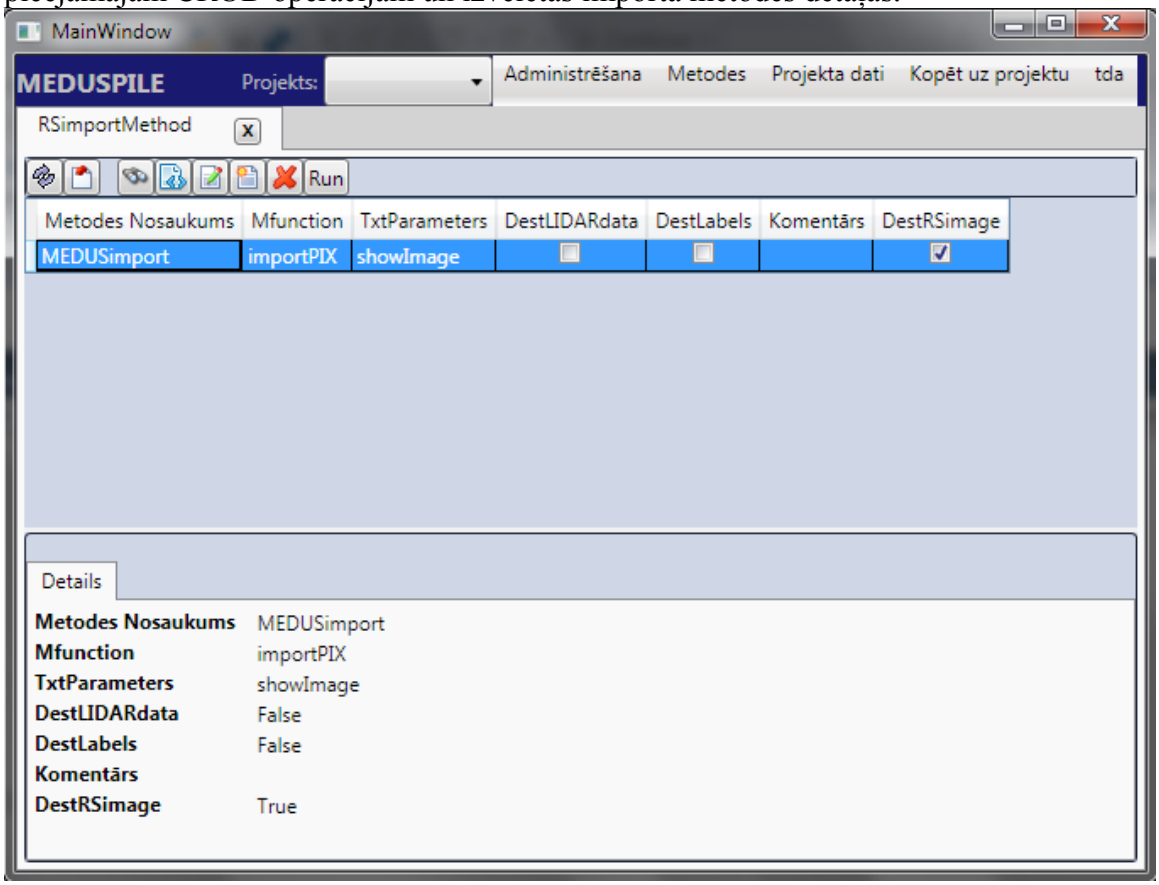
4.3. Koda ģenerācija no WCF (DataView) metamodeļa

No DataView metamodeļa tiek ģenerēts kods, kas definē objektus, kuri balstīti uz biznesa objektiem. Data View objektiem tiek nodrošināta CRUD funkcionalitāte.
Ar šiem datiem strādā lietotāja saskarnes. Ja projekts strādā klienta servera tehnoloģijā, tad nepieciešams nodrošināt DataView objektu transportu no servera uz klientu un no klienta uz serveri.

5. Lietotāja saskarnes līmenis

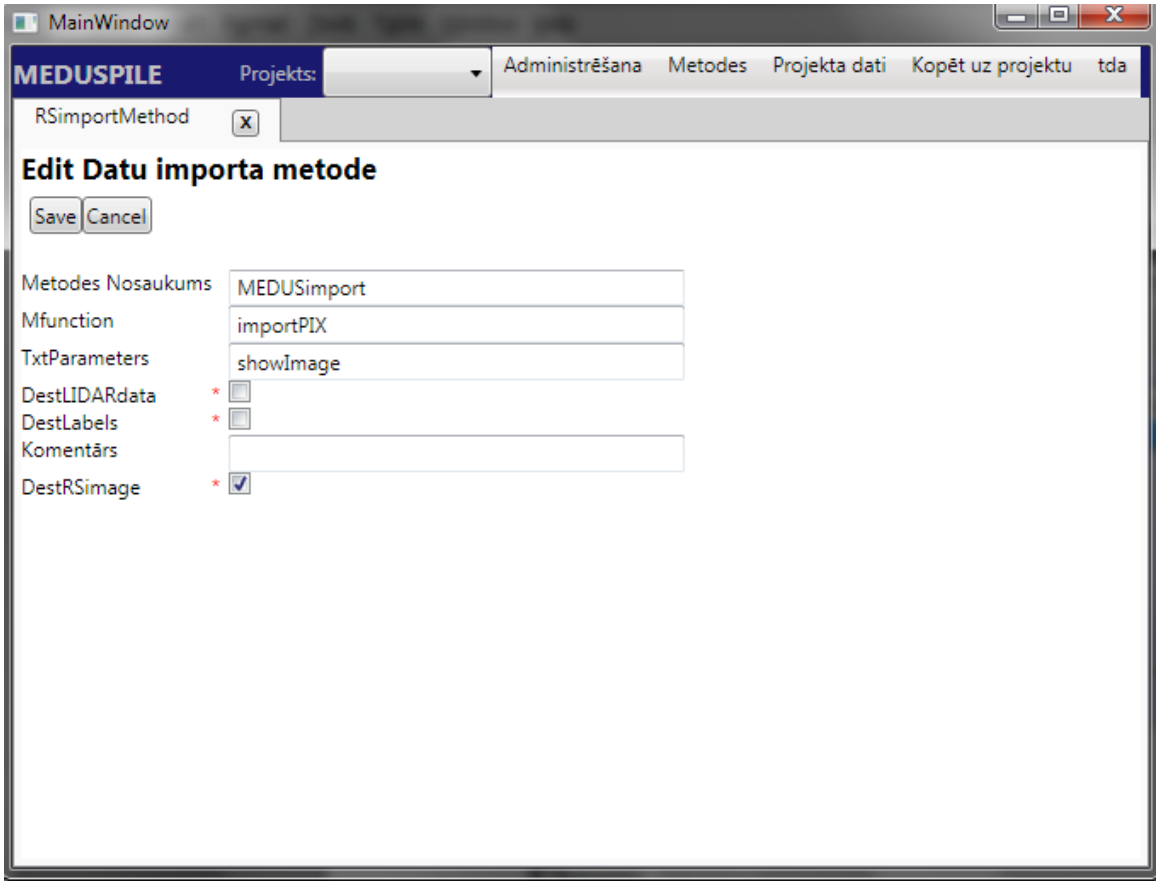
5.1. Koda ģenerācija no skatu metamodeļa

No skatu metamodeļa tiek ģenerēti XAML faili un atbilstošais C# kods.
Koda ģenerācija nodrošina pamatfunkcionalitāti: iespēju pēc dažādiem kritērijiem atlasīt izvēlētā tipa objektus, izveidot jaunus, labot esošos kā arī dzēst objektus.
Zīmējumā 1 ir redzams skats uz importa metodēm – importa funkciju saraksts ar pieejamajām CRUD operācijām un izvēlētās importa metodes detaļas.



Zīmējums 1

Zīmējumā 2 ir redzams izvēlētā importa funkcija labošanas režīmā.



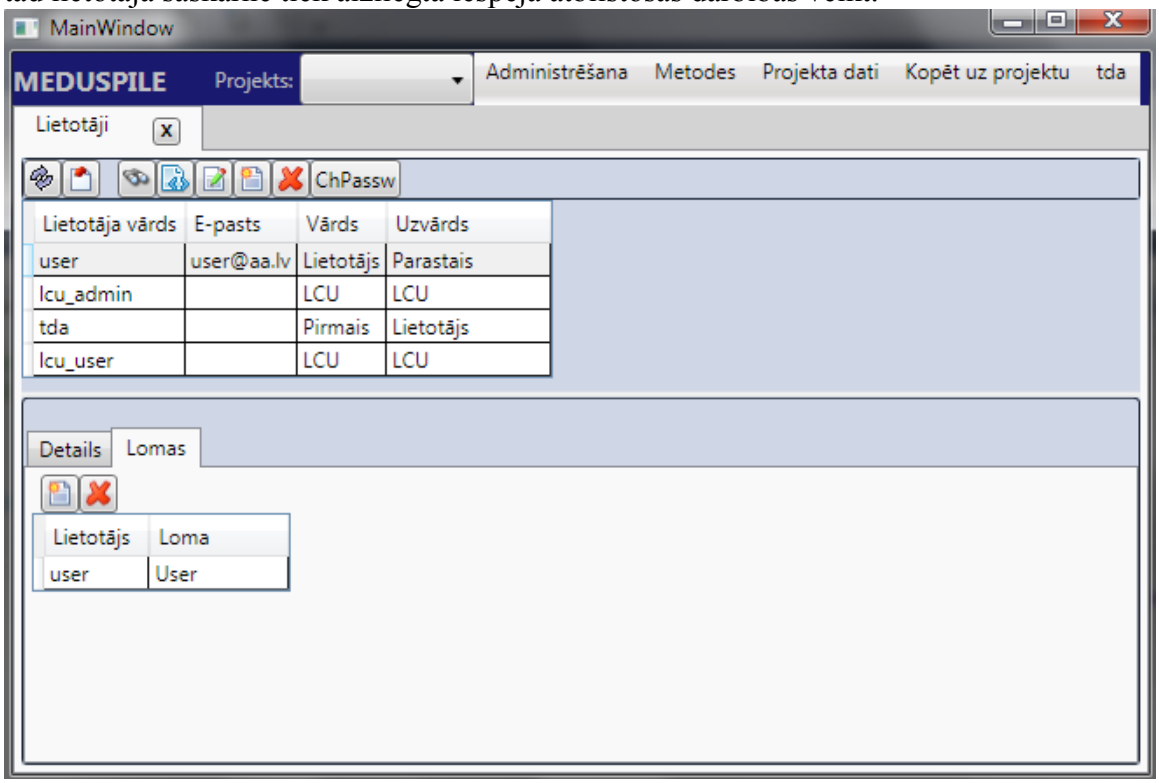
Zīmējums 2

Tabulā ir pārskaitītas projektā izmantotās T4 transformācijas lietotāju saskarnes līmenim:

T4 transformāciju faili	**Ģenerētais kods**	**Apraksts**
WPF_makeView.cs GuiBaseV__WPF.ttinclude	*.xaml	New, edit, index,details skati (XAML failu veidā) katram lietotāju saskarnes objektam no skatu modeļa
WPF_makeView.tt GuiBaseV__WPF.ttinclude	WPF_makeView.cs	Lietotāju saskarnes objektu .cs kods.
MEDUSPILE_subtables.tt	MEDUSPILE_subtables.cs	Apakšskatu definīcijas lietotāju saskarnes objektiem no skatu modeļa

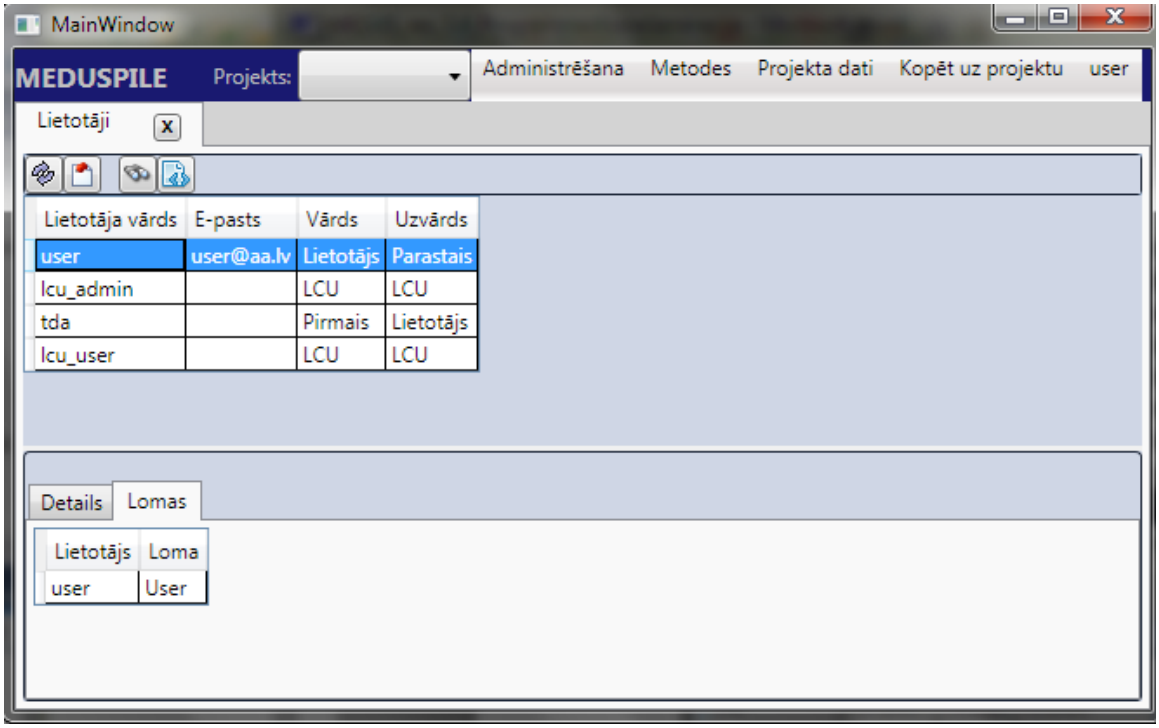
6. Koda ģenerācija no lietotāju tiesību metamodeļa

Paraugprojektā lietotāju tiesības tika nodrošinātas lomu līmenī. Tas nozīmē, ka datubāzē glabājas lietotāju informācija un lomas, bet lomu tiesību pārbaude izpildīt konkrētu operāciju tiek noģenerēta programmas kodā atbilstoši lietotāju tiesību modelim.
Lietotāju tiesības tiek pārbaudītas divos līmeņos. Pirmkārt, tiek pārbaudīts, vai lietotājam ir tiesības izpildīt operāciju. Ja lietotājam šādu tiesību nav, tad operācija netiek izpildīta un operācija atgriež paziņojumu, kurā teikts, ka lietotājam nav tiesības izpildīt operāciju.
Otrkārt, ar lietotāja tiesību palīdzību tiek regulēta lietotāja saskarnes elementu pieejamība. Tas ir – ja lietotājam nav tiesības doto objekta tipu dzēst, labot vai izveidot, tad lietotāja saskarnē tiek aizliegta iespēja atbilstošās darbības veikt.



Zīmējums 3

Zīmējumā 3 ir redzams skats uz lietotājiem gadījumā ja ir pieslēdzies administrators, bet zīmējumā 4 – ja pieslēdzies parasts lietotājs.
Kā redzams, administratoram ir iespējas veidot jaunus lietotājus, labot viņu datus, dzēst lietotājus, nomainīt lietotājiem paroli, kā arī pievienot lietotājam lomu un izmest to, bet parasts lietotājs var tikai apskatīties lietotājus.



Zīmējums 4

Tabulā ir redzama projektā izmantotās T4 transformācija lietotāju tiesību nodrošinājumam:

T4 transformāciju faili	**Ģenerētais kods**	**Apraksts**
UserRights.tt UserRights.ttinclude	UserRights.cs	Lietotāju tiesību pārbaudes funkcijas atbilstoši modelim.

7. Rezultāti

Aktivitātes ietvaros tika realizēta paraugprojekta programmas koda ģenerācija. Ne visas paraugprojekta specifikācijas prasības varēja apmierināt ar jau izstrādātajiem koda ģeneratoriem, jo bija arī specifiskas prasības, kuru realizācijas iekļaušana vispārējā tehnoloģiskajā risinājumā nav lietderīga un kuru realizēšanai tika rakstīts kods.
Paraugprojekta izpildes laikā tika uzlaboti un noskaņoti kodu ģeneratori, kā arī uzlabota tehnoloģija, kā ģenerēto kodu papildināt ar programmētāju rakstīto kodu, kas nodrošina projekta specifiskās prasības.

8. Literatūras saraksts

[1] Windows Presentation Foundation https://msdn.microsoft.com/en-us/library/ms754130%28v=vs.110%29.aspx
[2] https://www.mysql.com/
[3] Nr. 3.3 „Datu bāzes meta modeļa izstrāde un pētniecība, kas ietver dažādu datu bāzu vadības sistēmu atbalsta izpēti" progresa pārskats
[4] Nr. 3.6 „Savienošanas (join) meta modelis" progresa pārskats
[5] Nr. 3.7 „Biznesa objektu meta modeļa, kas kalpotu kā pamatmodelis biznesa objektiem, izstrāde" progresa pārskats
[6] Nr. 3.8.1 „Meta modelis WCF bāzētu aplikāciju serverim" progresa pārskats
[7] Nr. 3.9.1 „Skatu meta modelis" progresa pārskats
[8] Nr. 3.8.2 „Funkciju Objektorientētu Lietotāju Tiesību Modelis" progresa pārskats