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. 3.9.1 "Skatu meta modelis" progresa pārskats

Pārskats Nr. 28 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 "Skatu meta modelis" ietvaros veikti šādi darbi:

1. Identificēti un analizēti skata elementi;
2. Veikta tipisko skatu analīze dažādiem biznesa objektiem;
3. Skata metamodeļa prasību analīze;
4. Skatu tehnoloģiskā metamodeļa izstrāde, kas ietver GuiControl, GuiControlType, GuiField, DataView, DVcolumn, DVLink, DVLinkPair, GuiSubclass, GuiParametricObject, GuiObject, GuiInstiable, Descriptor entītijas;
5. Skatu modeļu transformācijas izstrāde;
6. Skatu koda ģenerācijas izstrāde.

2. Ievads

Šī zinātniskā atskaite ir veltīta projekta apakšaktivitātes Nr.3.9.1. "Skatu meta modelis" ietvaros veiktajai izstrādei pārskata periodā (2015-01-01. – 2015-06-30.).

Būvējot lietotāju saskarni, liels manuālā darba apjoms tiek veltīts tam, lai sagatavotu biznesa objektiem dažāda veida prezentācijas vai rediģēšanas skatus (lietotāja saskarnes elementus, no kuriem tālāk tiks konstruētas sarežģītākas formas). Lielākoties šo skatu izstrāde ir tipveida un atkarīga no biznesa objekta, kuram skati ir veltīti.
Lai automatizētu šo tipveida skatu implementācijas procesu, ir nepieciešams modelis, kas saturētu parametrus skata ģenerācijai. Dotās apakšaktivitātes uzdevums ir izstrādāt metamodeli, kas ļautu aprakstīt skatu modeļus.

3. Skatu (*GuiBase*) metamodelis

3.1. Skatu implementācijas uzdevums

Tipveida skats kādam biznesa objektam ir lietotāja saskarnes elements, kas attēlo šī objekta atribūtus. Tas, kā šinī skatā izskatās katrs atribūts, ir atkarīgs no daudzām lietām: no atribūta tipa, no izstrādātā saskarnes dizaina, no apstākļiem, kādos tiks lietots šis saskarnes elements, no skata uzdevuma. Skats varētu reprezentēt tabulu ar kolonām, ievadformu utt. Kā arī tas var saturēt tipveida funkcionalitāti, kas ir nepieciešama gan elementa pareizai uzvedībai saskarnē, gan ērtai elementa izmantošanai izstrādē.

Iespējamo variācijas iespēju ir daudz, tomēr, kad tās visas ir nofiksētas, paliek tikai daudz manuālā darba, lai realizētu norunāto katram biznesa objektu tipam.
Faktiski, ja par katru skatu un katru tajā attēlojamo atribūtu aprakstīt norunātos parametrus, kas attiecas uz saskarni, tad skatu tipveida implementāciju varētu uzticēt automātiskai ģenerācijai.

Saskarnes elementu realizācijas parametru definēšanai varētu kalpot skatu modelis. To aizpildītu analītiķis, un tas dotu informāciju automātiskai ģenerācijai.
Attiecīgi, skatu modeļa metamodelis aprakstītu, kādus tieši parametrus ir nepieciešams norādīt, lai noģenerētu skatus.

3.2. Skatu metamodeļa uzdevuma ierobežojumi

Izstrādājot skatu modeļa metamodeli, ir stingri jāapzinās uzdevuma robežas.
Pirmkārt, saskarnes realizācijas parametriem nav jāaptver visas iespējamas saskarnes realizācijas nianses - skatu modelis nav domāts kā vispārināta saskarnes aprakstošā valoda. Pietiek ar to, ka skatu modelis specificēs tipveida skatus; individuālus gadījumus (atkarībā no biznesa prasībām) izstrādās cilvēki (iespējams, pielāgojot jau gatavu, automātiski uzģenerēto skatu).

Otrs aspekts, par kuru jāatceras, izstrādājot skatu metamodeli, ir skatu būtība. Skati ir kontroļi, saskarnes daļas, kuras pēc tam tiks lietotas īstās formās dažādos kontekstos. Ja pieļaus, ka skatu modelī ir entītijas, kas ietver sevī pārāk daudz saskarnes loģikas, tad šos elementus būs grūti pielietot citā ko tekstā (līdz ar to sanāks, ka citam kontekstam būs jādefinē citas entītijas utt. - kamēr ar ģenerāciju ieekonomētais manuālā darba apjoms netiks iztērēts, aizpildot skatu modeli).

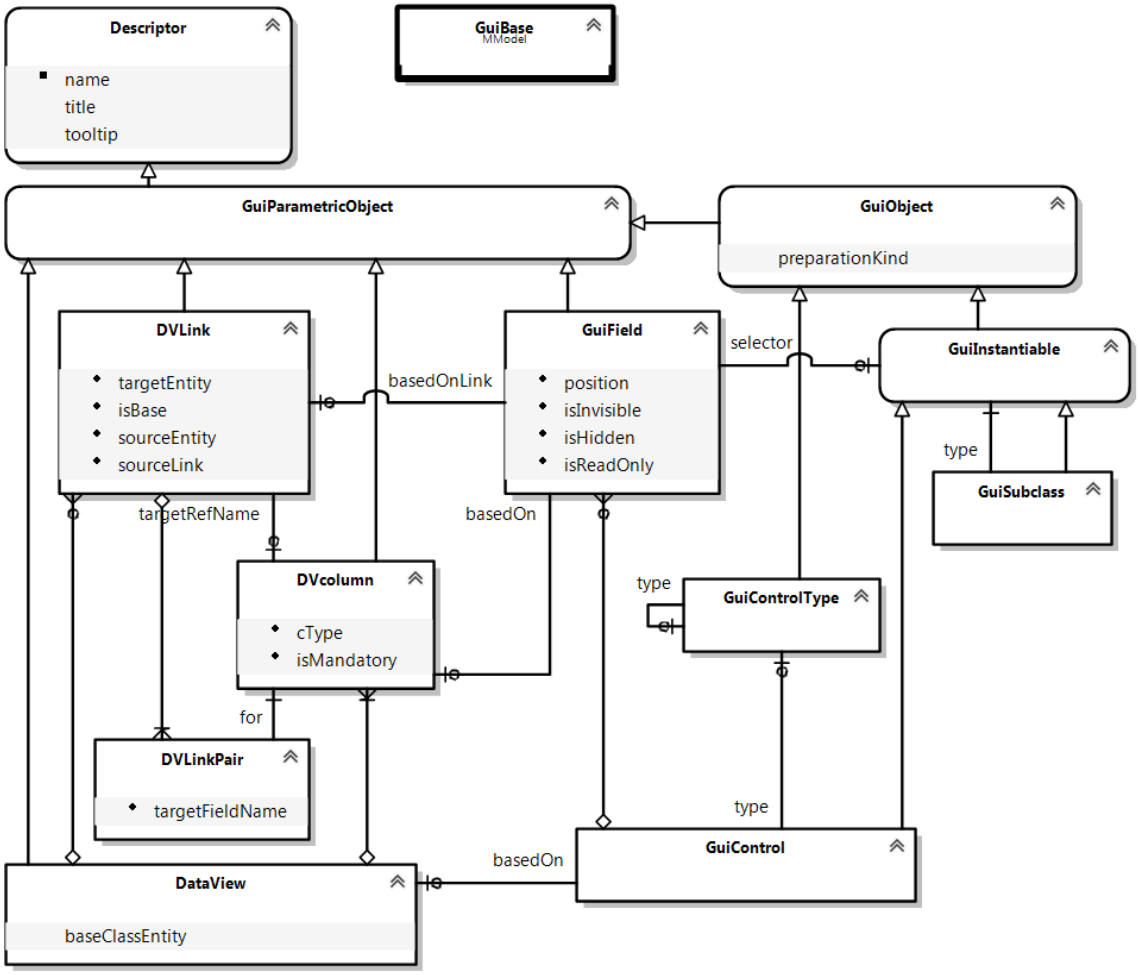
3.3. Prasības skatu metamodelim

Galvenā metamodeļa entītija ir skats. Tai ir saite ar attēlojamo biznesa objektu (*DataView*), tā satur aprakstus katram biznesa objekta laukiem ar lietotāja saskarnes tipveida parametriem. Katram laukam atbilstošajā aprakstā jābūt informācijai par:

1) kādam biznesa objekta atribūtam atbilst lauks,
2) vai dotajā skatā lauks ir redzams lietotājam,
3) rediģēšanas skatiem: vai dotajā skatā lauks ir pieejams rediģēšanai,
4) rediģēšanas skatiem: vai dotajā skatā lauks ir obligāts,
5) kāds ir lauka pozīcija dotajā skatā (tipiski - lauka kārtas numurs lauku secībā),
6) lauka datu tips,
7) norādēm uz citu biznesa objektu: kādi atribūti piedalās norādē, uz kuru biznesa objektu tipu rāda norāde, ko reprezentēt var norādes vērtību.

4. Skatu (*GuiBase*) tehnoloģiskais metamodelis

Skatu tehnoloģiskais modelis ir domāts, lai no tā ģenerētu lietotāja saskarnes skatus, tāpēc tam jāsatur visi nepieciešami skatu parametri. Skatu metamodelis apraksta šos parametrus un ir virsslānis virs *DataView* modeļa [1]. Zīmējumā 1. ir redzama metamodeļa shēma.



Zīmējums 1. Skatu tehnoloģiskais modelis.

Tālāk tiek aprakstītas visas metamodeļa entītijas un to atribūti.

4.1. GuiControl

GuiControl ir skats (lietotāja saskarnes elements, kas attēlo vienu biznesa objektu kaut kādā veidā (ko nosaka tips - *GuiControlType*)). Satur vairākus laukus (*GuiField*).

Vārds	Datu tips	Apraksts
basedOn	link	Norāde uz biznesa objektu (*DataView*), kuru šis skats attēlo.
type	link	Norāde uz skatu tipu (*GuiControlType*), kas nosaka

		attēlošanas veidu.

4.2. GuiControlType

GuiControlType nosaka attēlošanas veidu skatiem (*GuiControl*): sagrupē skatus, kas attēlo tabulas sarakstiem, ievadformas utt.

Vārds	**Datu tips**	**Apraksts**
Type	link	Norāde uz šī skatu tipa virstipu (*GuiControlType*), ja tāds ir (var nebūt).

4.3. GuiField

Lauks (*GuiField*) ir apakšelements, kas apraksta biznesa objekta viena atribūta (*DVcolumn*) attēlošanas parametrus sava skata (*GuiControl*) ietvaros.

Vārds	**Datu tips**	**Apraksts**
position	int	Lauka kārtas numurs viena skata (*GuiControl*) ietvaros.
isInvisible	bool	Ja ir uzstādīts "true", tad šis lauks ir pēc noklusēšanas neredzams lietotājam (ar potenciālu iespēju padarīt to redzamu).
isHidden	bool	Ja ir uzstādīts "true", tad šis lauks ir vienmēr neredzams lietotājam (tiek iekļauts skatā dienesta vajadzībām).
isReadOnly	bool	Ja ir uzstādīts "true", tad šis lauks ir nerediģējams, ja skatā ir pieejama rediģēšana.
basedOn	link	Norāde uz biznesa objekta parasto atribūtu (*DVcolumn*), kurai atbilst šis lauks.
basedOnLink	link	Norāde uz biznesa objekta relāciju (*DVLink*), kurai atbilst šis lauks.
selector	link	Norāde, kas dod iespēju izvēlēties citu skatu (*GuiControl*) vai apakšklasi (*GuiSubclass*), lai attēlotu šo lauku. Var būt nenorādīts.

Tikai viena no norādēm - *basedOn* vai *basedOnLink* - ir aizpildāma vienam laukam.

4.4. DataView

Reprezentē vienu biznesa objektu (*DataView*). Satur vairākus atribūtus *(DVColumn*) .

Vārds	**Datu tips**	**Apraksts**
Name	string	Biznesa objekta vārds.
baseClassEntity	string	Entītijas nosaukums, kas kalpo kā šī biznesa objekta galvenās entītijas bāzes entītija. (Var nebūt, ja galvenajai entītijai nav bāzes entītija.)

4.5. DVcolumn

Reprezentē vienu atribūtu viena biznesa objekta (*DataView*) ietvaros.

Vārds	Datu tips	Apraksts
cType	string	Atribūta datu tips
isMandatory	bool	Ja ir uzstādīts "true", tad šis atribūts ir obligāts.

4.6. DVLink

Reprezentē vienu relāciju viena biznesa objekta (*DataView*) ietvaros; satur salikto norādi uz *DVcolumn* lauku (*DVLinkPair*)

Vārds	Datu tips	Apraksts
targetEntity	string	Entītijas vārds, uz kuru rāda relācija (mērķa entītija)
isBase	bool	Ja ir uzstādīts "true", tad relācijas mērķa entītija ir bāzes entītija relācijas biznesa objektam.
sourceEntity	string	Entītijas vārds, kura satur doto relāciju (avota entītija)
sourceLink	string	Avota entītijas relācijas kolonas vārds
targetRefName	link	Norāde uz reprezentatīvo atribūtu mērķa biznesa objektā, kuru rāda tad, kad vizuāli jāatspoguļo norāde. Saturīgi, tas ir viens no *targetEntity* kolonām.

4.7. DVLinkPair

Reprezentē salikto norādi. Domāta, lai būtu iespēja definēt saliktās norādes sastāvdaļas, bet vairumā gadījumu ir tikai viens *DVLinkPair* vienam *DVLink*.

Vārds	Datu tips	Apraksts
For	link	Norāde uz atribūtu (*DVcolumn*) biznesa objektā.
targetFieldName	string	Kolonas vārds mērķa entītijā, uz kuru rāda šī norāde.

4.8. GuiSubclass

Reprezentē skata (*GuiControl*) apakšklasi.

Vārds	Datu tips	Apraksts
Type	link	Norāde uz virsklases skatu (*GuiControl*).

4.9. GuiParametricObject

Dienesta abstraktā klase, kas apvieno visas entītijas skatu modelī.

4.10. GuiObject

Dienesta abstraktā klase, kas apvieno skatu, to virsklašu un apakšklašu entītijas.

Vārds	Datu tips	Apraksts
preparationKind	string	Ģenerēšanas tipa nosaukums.

4.11. GuiInstantiable

Dienesta abstraktā klase, kas apvieno skatu un to apakšklašu entītijas (bez virsklasēm, kas ir vajadzīgi implementācijai).

4.12. Descriptor

Dienesta abstraktā klase, kas pieliek katrai metamodeļa entītijai dažāda rakstura aprakstošus atribūtus:

Vārds	Datu tips	Apraksts
Name	string	Iekšējais nosaukums.
Title	string	Lietotāja saskarnē rādāmais nosaukums.
Tooltip	string	Īss apraksts paskaidrei.

5. Skatu modeļu transformācija

Lai automātiski iepildītu datus tehnoloģiskajā modelī no loģiskā modeļa, kur tos liek analītiķis, tiek lietota modeļu transformāciju mašīna ([2] un [3]) un speciāli kompleksu modeļiem izstrādātā transformācija (sk. pielikumu).

Transformācija taisa izgriezumu no loģiskā modeļa datiem un pārveido tos kompleksu tehniskā metamodeļa jēdzienos.

Vēl viens uzdevums, ko izpilda transformācija, - tā saražo objektus, kuri nav aprakstīti atklāti loģiskajā modelī, bet kuru aprakstus var mehāniski noģenerēt no esošajiem objektiem. No vienas puses: loģiskais modelis nav piekrauts ar daudzajiem objektiem, kas atvieglo caurskatīšanos, un ir ieekonomēts roku darbs aizpildīšanā; no otrās puses - tehniskais modelis satur visus nepieciešamus objektus, un koda ģenerācijā nevajag iebūvēt sarežģītus mehānismus, kā rīkoties, ja nepieciešamais objekts trūkst, bet sistēmas infrastruktūrā tas tika ieplānots.

6. Rezultāti

Aktivitātes ietvaros veiktā darba rezultātā tika izstrādāts skatu metamodelis tehnoloģiskajam modelim ar vārdu *GuiBase*. Metamodelis ļauj aprakstīt biznesa objektiem atbilstošus lietotāja saskarnes elementus - skatus, kas tiks izmantoti lietotāja saskarnes formu izstrādē.

7. Literatūras saraksts

[1] Nr.3.8.1 "WCF biznesa saskarnes meta modelis" progresa pārskats.
[2] Nr.1.3.2 "Modeļa transformāciju atbalsta izstrāde (M2M transformāciju atbalsts)"
[3] Nr.1.3.3 "Modeļa apstrādes transformāciju mašīnas izstrāde, kas ietver daudz-modeļu manipulācijas"

8. Pielikumi

8.1. Skatu modeļu pārveidošanas transformācija

```
namespace LogicalGuiBase2 {
 block technoT(LogicalGuiBase FR, GuiBase TO) : TtechnoT {
  gcc($c(FR.GuiControlClass), $cT(TO.GuiObject))
        %(var x=$c; var y=$cT;
          y.name=x.name;
              var t=x.title; if(t!=null) y.title=t;
              t=x.tooltip; if(t!=null) y.tooltip=t;
              )%;

       controlType:
        FR.GuiControlType=$c:-GuiControlType=$cT
        [*@gcc];
       controlType_type:
        FR.GuiControlType=$c~=$cT(TO.GuiControlType)
        $c-type=$c2~=$c2T(TO.GuiObject)
        %($cT.MDS_pp_type.setValue($$c2T);)%
        ;

  dataviews:
        FR.DataView=$dv:-DataView=$dvT
        %($dvT.name=$dv.name;)%
        $dv.DVcolumn=$dvc:-[$dvT]DVcolumn=$dvcT
        %($dvcT.name=$dvc.name;)%
        $dvc-is=$f0
        $f0-inJpart=$jp
        %( var x=$f0;
         if(x.isRefName){
              var y=$dvcT;
              y.cType="string";
              if($jp.isMandatory) y.isMandatory=true;
              return false;
              })%
        $f0-basedOn=$u
        %(if($jp.isMandatory && $u.isMandatory) $dvcT.isMandatory=true;)%
        %(var u=$u;
          var f=u as LogicalEntity.Field;
          if(f!=null){
               var y=$dvcT;
               var t=f.cType;
               if(t==null){
```

```
                t=f.type;
                    var m=y.isMandatory;
                    t=MEDUS.TypeConvertor.ToNetTypeU(t, m);
                }
              y.cType=t;
              return false;
              }
        var s=u as LogicalEntity.Scalar;
        if(s!=null){
              var y=$dvcT;
              var t=s.cType;
              if(t==null){
                t=s.type;
                    var m=y.isMandatory;
                    t=MEDUS.TypeConvertor.ToNetTypeU(t, m);
                }
              y.cType=t;
              return false;
              })%
  $u=$l(FR.Link)
      %( var y=$dvcT; var l=$l;
              var t=l.myCtype;
              if(t==null){
                t=l.myType;
                    var m=y.isMandatory;
                    t=MEDUS.TypeConvertor.ToNetTypeU(t, m);
                }
              y.cType=t;
       )%;

      guicontrolsOnDV:
        FR.GuiControl=$c
        $c-basedOn=$dv~=$dvT(TO.DataView)
        $c:-GuiControl=$cT
        %($cT.MDS_pp_basedOn.setValue($$dvT);)%
        [*@gcc]
        $c-type=$ct2~=$c2T(TO.GuiObject)
        %($cT.MDS_pp_type.setValue($$c2T);)%
        ;

guicontrolsFree:
      FR.GuiControl=$c
       %(if($c.basedOn!=null) return false;)%
       $c:-GuiControl=$cT
       %( var x=$c; var y=$cT;
         y.name=x.name;
```

```
                var t=x.title;
                if(t!=null)
                  y.title=t;
                )%
          $c-type=$ct2~=$c2T(TO.GuiObject)
          %($cT.MDS_pp_type.setValue($$c2T);)%;

  guiFieldsInGuiControl_basedOn:
    FR.GuiControl=$c~=$cT(TO.GuiControl)
          $c-basedOn=$dv~=$dvT(TO.DataView)
          $c.GuiField=$gf:-[$cT]GuiField=$gfT
          %( var x=$gf; var y=$gfT;
           y.name=x.name;
           var t=x.title; if(t!=null) y.title=t;
           if(x.isInvisible) y.isInvisible=true;
           if(x.isHidden) y.isHidden=true;
           if(x.isReadOnly) y.isReadOnly=true;
           var p=x.position; if(p>0) y.position=p;
           )%
          $gf-basedOn=$bdv~=$dv2T(TO.DVcolumn)
          %($gfT.MDS_pp_basedOn.setValue($$dv2T);)%
           ;

  guiFieldsInGuiControl_basedOnLink:
    FR.GuiControl=$c~=$cT(TO.GuiControl)
          $c-basedOn=$dv~=$dvT(TO.DataView)
          $c.GuiField=$gf~=$gfT(TO.GuiField)
          $gf-basedOnLink=$dvl
          :~[$cT]DVLink=$glT
          $dvl-is=$cl
          $cl-basedOn=$l
          %(var x=$glT;
            x.sourceLink=$l.name;
                x.sourceEntity=((LogicalEntity.Entity)($l.MDScontainer)).name;)%
          $l-target=$e
          %($gfT.MDS_pp_basedOnLink.setValue($$glT);
            var nn=$dvl.name;
                if(nn==null)
                  nn=$dvl.@is.name;
            $glT.name=nn;
                $glT.targetEntity=$e.name;
                )%
          $dvl-is=$cl
          $cl-basedOn=$el
          %(if($el.isBase==true)
                  $glT.isBase=true;
```

```
            )%
        $cl.ClinkPair=$cp
        $cp-basedOn=$cf
        $dv.DVcolumn=$dvc
        $dvc-is=$dvcf
        %(if($$dvcf!=$$cf)
           return false;)%
        $dvc~=$dvcT(TO.DVcolumn)
        :~[$glT]DVLinkPair=$dvlpT
        %(var tc=$$dvcT;
          $dvlpT.MDS_pp_for.setValue(tc);
          if($cf.isRefName)
                $glT.MDS_pp_targetRefName.setValue(tc);)%
         ;

guiSubclass:
  FR.GuiSubclass=$c:-GuiSubclass=$cT
        %( var x=$c; var y=$cT;
           y.name=x.name;
                var t=x.title;
                if(t!=null)
                  y.title=t;
               )%
        $c-type=$ct2~=$c2T(TO.GuiObject)
        %($cT.MDS_pp_type.setValue($$c2T);)%
        ;



GuiObject_details:
 FR.GuiControlClass=$c~=$cT(TO.GuiObject)
       %( var x=$c; var y=$cT;
          var n=x.preparationKind;
          if(n!=null)
            y.preparationKind=n;
          n=x.name;
          if(n!=null)
            y.name=n;
          n=x.title;
          if(n!=null)
            y.title=n;
          n=x.tooltip;
          if(n!=null)
            y.tooltip=n;
          )%;
```

```
myBlockT.usedColsByGuiFields=$c.GuiFieldS.Where(o=>o.basedOn!=null).Select(o=>
o.basedOn.MDSisn).ToList();
myBlockT.maxPosGuiField=$c.GuiFieldS.Select(o=>o.position).DefaultIfEmpty().Max(
);
    }
  }
```