



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.3.8.1 "WCF biznesa saskarnes meta modelis" progresa pārskats

Pārskats Nr.26. par periodu no 2014.gada 1.jūlija līdz 2014.gada 31.decembrim.

SATURS

1.	Kopsavilkums	3
2.	Ievads	4
3.	Biznesa saskarnes meta modelis	5
3.1.	Biznesa objekti un viņu transportēšana.....	5
3.2.	WCF.....	5
3.3.	Web API.....	5
3.4.	Prasības DataView meta modelim.....	5
4.	DataView tehnoloģiskais meta modelis.....	7
4.1.	DataView	7
4.2.	Complex	7
4.3.	CField.....	8
4.4.	DVColumn.....	8
4.5.	DVlink.....	8
4.6.	LinkPair.....	8
5.	DataView modeļu transformācijas.....	9
6.	Rezultāti	10
7.	Literatūras saraksts.....	11
8.	Pielikums.....	12

1. Kopsavilkums

Pārskata periodā (2014-07-01 – 2014-12-31.) projekta „Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem” aktivitātes Nr.3.8.1 "WCF biznesa saskarnes meta modelis" ietvaros veikti šādi darbi:

1. Prasību analīze biznesa saskarnes metamodelim.
2. Loģiskā DataView metamodela izstrāde.
3. Tehnoloģiskā DataView metamodela izstrāde.
4. DataView tehnoloģiskā modeļa automātiskās ražošanas transformācijas izstrāde.
5. WCF biznesa saskarnes koda ģeneratora izstrāde.
6. Aktivitātes pētnieciskā darbība apspriesta ik nedēļas projekta semināros.

2. Ievads

Šis pārskats ir veltīts projekta apakšaktivitātes Nr.3.8.1 "WCF biznesa saskarnes meta modelis" ietvaros veiktajai izstrādei.

Viens no mūsu pieejas pamatidejām ir multi-slāņu pieeja modeļu taisīšanā. [1] Biznesa saskarnes meta modelis ir domāts tam, lai atdalītu biznesa loģiku no datu glabāšanas, optimizācijas un pārējiem tehniskās realizācijas specifiskiem uzdevumiem, dotu iespēju manipulēt ar biznesa objektiem, maksimāli pietuvinot tos priekšmetu apgabala jēdzieniem un funkcijām.

3. Biznesa saskarnes meta modelis

3.1. Biznesa objekti un viņu transportēšana

Strādājot pie programmatūras izstrādes, ir jāveic biznesa analīze, kuras rezultātā tiks nodefinēti biznesa objekti [2], kuri ir nepieciešami biznesa funkcijas mērķu sasniegšanai. Loģiski, biznesa objekti, papildināti ar specifiskiem atribūtiem, kuri nav atkarīgi no biznesa loģikas, bet gan no tehnoloģiskajām un datu transportēšanas vajadzībām, arī ir tā datu kopa, kura tiks pārsūtīta starp datubāzi, servera un klienta lietojumprogrammām. Mūsdienās ir izstrādātas dažādas tehnoloģijas datu transportēšanai. Apakšā ir raksturojums divām vispopulārākajām Microsoft piedāvātajām tehnoloģijām.

3.2. WCF

Windows Communication Foundation (WCF) ir Microsoft piedāvātais vienots ietvars, lai izveidotu servīču orientētas (service-oriented) lietojumprogrammas, kas balstās uz SOA [3] principiem. Tas vislabāk ļauj realizēt drošu, stabilu, integrēto starp platformām biznesa aplikāciju risinājumu. [4]. WCF atbalsta dažādus transporta protokolus (HTTP, TCP, UDP un citi) un dod iespēju viegli pārslēgties starp viņiem. Izmantojot WCF tehnoloģiju, izstrādātājam jānodefinē metodes, kuras būs saskarnē, un jānokonfigurē vide, tālāk ietvars pats noģenerēs komponentes, nepieciešamas servīču lietošanai.

3.3. Web API

ASP.NET Web API ir ietvars, kas domāts tīmekļa servīču izstrādei. Web API balstās uz HTTP protokola un ir ļoti piemērots piekļuvei no plaša klientu klāsta, piemēram, dažādām pārlūkprogrammām un mobilajām ierīcēm [5]. Web API atvieglo un uzlabo REST servīču izstrādi.[6]

Kopumā, Web API izmantošana pieprasa no izstrādātāja vairāk manuālas koda rakstīšanās, bet ir tehnoloģiski skaidrāk un plānāk, strādā ātrāk. Savukārt, WCF ir lēnāks, viņa iekšēja uzbūve ir sarežģīta, tā prasa lielāku izpratni.

Izstrādes laikā tika piedāvāts mūsu biznesa objektus taisīt universāli, lai tos varētu lietot jebkurā no augšminētajām tehnoloģijām.

3.4. Prasības DataView meta modelim

Mūsu biznesa objekta realizāciju sauksim par DataView. Izstrādes procesā tika izvirzītas sekojošas prasības DataView meta modelim:

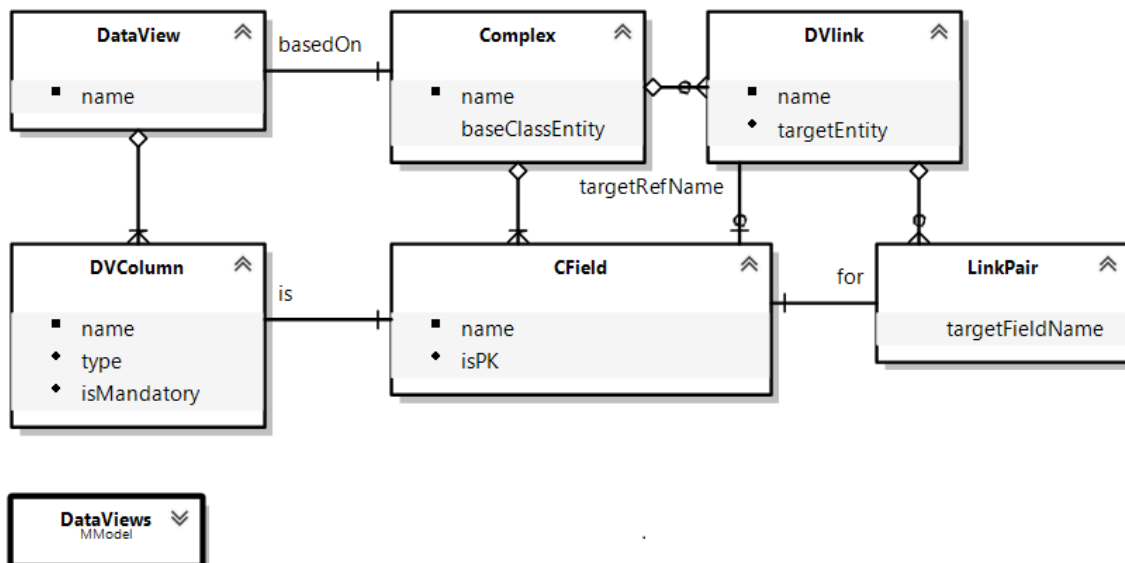
- iespēja izvēlēties datu konglomerātu un, tādējādi, izvēlēties atribūtu (kolonas) apakškopu;
- uzbūvēt DataView tā, lai tas derētu datu transportēšanas nolūkiem. DataView ir jūtīgs pret izmēru, kas ietekmē, savukārt, pārraides laiku.

Citiem vārdiem, DataView ir jāiekļauj tikai tas, kas ir tiešām nepieciešams.

- Jābūt plakanam, paslēpjot dažādu datu glabāšanas hierarhiju;
- Šis modelis būs pamats biznesa objektu klašu ģenerācijai.

4. DataView tehnoloģiskais meta modelis

DataView meta modelis ir virsslānis virs kompleksu (Complex) meta modeļa [2]. Zīmējumā 1 ir redzama meta modeļa shēma. DataView meta modelis dod iespēju aprakstīt biznesa objektus.



Zīm. 1.

Tālāk tiks aprakstīta katra metaentīcijas instance un tai piederošo metaatribūtu un metasaišu instances.

4.1. DataView

Reprezentē vienu *DataView*. Satur vairākas kolonas (*DVColum*)

Vārds	Datu tips	Apraksts
name	string	DataView vārds.
basedOn	link	Norāde uz kompleksu, uz kura pamata tiek uzbūvēts DataView

4.2. Complex

Reprezentē vienu kompleksu [2]. Satur vairākus laukus (*CField*) un references (*DVlink*)

Vārds	Datu tips	Apraksts
name	string	Kompleksa vārds.
baseClassEntity	string	Bāzes klases entīcijas nosaukums

4.3. CField

Reprezentē vienu skalāro lauku viena kompleksa ietvaros.

Vārds	Datu tips	Apraksts
name	string	Lauka vārds.
isPK	bool	Ja "true", tad lauks ir primārā atslēga (primary key)

4.4. DVColumn

Reprezentē vienu kolonu viena DataView ietvaros.

Vārds	Datu tips	Apraksts
name	string	Lauka vārds.
type	string	Lauka datu tips.
isMandatory	bool	Ja "true", tad lauks ir obligāts.
is	link	Norāde uz bāzes lauku (CField) bāzes kompleksā.

4.5. DVlink

Reprezentē norādi viena kompleksa ietvaros. Var saturēt salikto norādi uz *CField* lauku (*LinkPair*)

Vārds	Datu tips	Apraksts
name	string	Norādes vārds
targetEntity	string	Entītijas vārds, uz kuru rāda norāde (mērķa entītijā)
targetRefName	link	Norāde uz reprezentatīvo lauku, kuru rāda tad, kad vizuāli jāatspoguļo norāde. Saturīgi, tas ir viens no targetEntity laukiem.

4.6. LinkPair

Reprezentē salikto norādi. Domāta, lai nedefinēt saliktās norādes sastāvdaļas, bet vairumā gadījumu ir tikai viens *LinkPair* vienam *DVLink*.

Vārds	Datu tips	Apraksts
for	link	Norāde uz <i>CField</i> lauku kompleksā.
targetFieldName	string	Lauka vārds mērķa entītijā, uz kuru rāda šī norāde.

5. DataView modeļu transformācijas

Izstrādes procesā tika piedāvāts DataView loģiskais meta modelis, ar kuru ērtāk strādāt analītiķim. Realizācijā daudz ērtāk manipulēt ar tehnoloģiskā meta modeļa objektiem. Lai automātiski iepildītu datus tehnoloģiskajā modelī no loģiskā modeļa, tiek lietota modeļu transformāciju mašīna [2], [3], un tika izstrādāta speciāla transformācija DataView modeļiem (sk. pielikumu).

Transformācija taisa izgriezumu no loģiskā modeļa datiem un pārveido tos DataView tehniskā meta modeļa jēdzienos.

6. Rezultāti

Aktivitātes ietvaros veiktā darba rezultātā tika izstrādāts DataView meta modelis un WCF biznesa saskarnes koda ģenerators.

7. Literatūras saraksts

1. Nr.3.2 "Biznesa lēmumu atdalīšana no izstrādes lēmumiem" progresā pārskats
2. Nr.3.7 "Biznesa objektu meta modeļa, kas kalpotu kā pamatmodelis biznesa objektiem, izstrāde" progresā pārskats.
3. <http://msdn.microsoft.com/en-us/library/bb833022.aspx> - Service Oriented Architecture
4. <http://msdn.microsoft.com/en-us/library/dd456779%28v=vs.110%29.aspx> - WCF
5. <http://www.asp.net/web-api> - Web API
6. <http://www.restapitutorial.com/> - RESTful services

8. Pielikums

```

namespace LogicalDataView2 {
    block technoT(LogicalDataView LM, DataViews TV) : TtechnoT {
        cmplx:
            LM.Complex=$c:-Complex=$cT
            %($cT.name=$c.name;)%
                $c.Cpart=$cp
                $cp.Cfield=$cf
                $cf:-[$cT]CField=$cfT
            %($cfT.name=$cf.name;)%
                % (if($cf.isRefName)
                    return false;)%
                % (var x=$cf;
                    if(x.MDSisInMain==false)
                        return false;
                    if(x.basedOn!=null)
                        return x.basedOn.isPK;
                    if(x.basedOnSubField==null)
                        return false;
                    var l=x.basedOnSubField.MDScontainer as LogicalEntity.Link;
                    if(l.isPK==false)
                        return false;
                    )%
                % ($cfT.isPK=true;)%
            ;

        links_internal:
            LM.Complex=$c~=$cT(TV.Complex)
            $c.Cpart=$p
            $p-basedOn=$j
            $j-main=$main
            $j.Jpart=$jp
            $jp.Jlink=$jl
            $jl-basedOn=$l
            :~[$cT]DVlink=$jIT
            $jl-to=$tjp
            $l-target=$te
            % ( var x=$jIT;
                x.name=$jp.prefix+$l.name;
                x.targetEntity=$te.name;
                var k=$p.MDSfindRefNameCfield($tjp);
                if(k>0)
                    k=myBlockT.getMirror(new MEDUS.UniversalObject(k, false)).realObject;
                if(k>0){

```

```

        x.MDS_pp_targetRefName.setValue(k);
        if($!.isBase && $$jp==$main)
            $cT.baseClassEntity=x.targetEntity;
        }
    )%
    $!.SubField=$s
    :~[$!T]LinkPair=$!pT
    %(var x=$!pT;
    x.targetFieldName=$s.targetUnitName;
    var k=$p.MDSfindCfield($$jp, $$s);
    if(k>0)
        k=myBlockT.getMirror(new MEDUS.UniversalObject(k, false)).realObject;
    if(k>0)
        x.MDS_pp_for.setValue(k);
    )%
;

links_external:
LM.Complex=$c~=$cT(TV.Complex)
$c.Cpart=$p
%( myBlockT.dvs=new Dictionary<int, DataViews.DVlink>(); )%
$p.Cfield=$f~=$fT(TV.CField)
%(if($f.isBasedOnExternalLink==false)
return false;)%
$f-basedOn=$u
%( myBlockT.makeDVlinks($c, $cT, $p, $f, $fT, $u); )%
;

dv:
LM.Complex=$c~=$cT(TV.Complex)
$c.DataView=$v:-DataView=$vT
%( var x=$vT;
x.name=$v.name;
x.MDS_pp_basedOn.setValue($cT);
)%
$v.DVcolumn=$col
$col-is=$f~=$fT(TV.CField)
$col:-[$vT]DVColumn=$colT
%( var x=$colT;
x.name=$col.name;
x.MDS_pp_is.setValue($fT);
var f=$f;
if(f.isRefName){
x.type="String";
return false;
}
}

```

```
    )%
    $f-inJpart=$jp
    $f-basedOn=$u
    %( var bo=$f.basedOnSubField;
      string dbt=null;
      if(bo==null) // Scalar
        dbt=((LogicalEntity.Scalar)$u).type;
      else { // SubField
        dbt=bo.pkType.type;
      } // SubField
      bool isM=$jp.isMandatory && $u.isMandatory;
      var x=$colT;
      if(isM)
        x.isMandatory=true;
      x.type=dbt;
//    $colT.cType=MEDUS.TypeConvertor.ToNetTypeU(dbt, isM);
    )%
  ;
}
}
```