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.1.3.3 "Modeļa apstrādes transformāciju mašīnas izstrāde, kas ietver daudz-modeļu manipulācijas" progresa pārskats

Pārskats Nr.20. par periodu no 2014.gada 1.jūlija līdz 2014.gada 31.decembrim.

SATURS

1. Kopsavilkums

Pārskata periodā (2014-06-01 – 2014-12-31.) projekta „Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem" aktivitātes Nr.1.3.3 "Modeļa apstrādes transformāciju mašīnas izstrāde, kas ietver daudz-modeļu manipulācijas" ietvaros turpināti iesāktie darbi un veiktas šādas darbības:

1. Modeļu apstrādes transformācijas mašīnas izstrādes projektējuma pārveide.
2. Transformāciju definīcijas metamodeļa izstrāde.
3. Transformācijas likumu pārveide.
4. Transformācijas bloka pārveide.
5. Transformācijas valodas analizatora savietošana ar modeļa iepildīšanas funkcijām.
6. T4 transformācijas izstrāde, kas, izmantojot modeļa apstaigāšanas funkcijas, ģenerē C# kodu.
7. Aktivitātes pētnieciskā darbība apspriesta ik nedēļas projekta semināros.

2. Ievads

Šis pārskats ir veltīts projekta apakšaktivitātes Nr.1.3.3 "Modeļa apstrādes transformāciju mašīnas izstrāde, kas ietver daudz-modeļu manipulācijas" ietvaros veiktajiem pētījumiem.

Aktivitātes ietvaros tika izstrādāta transformāciju struktūra, kas aprakstīta sadaļā 3. Transformāciju struktūra, kā arī tika izstrādāta transformācijas valodas izpildes mašīna (skatīt 4. Transformāciju valodas izpildes mašīna).

3. Transformāciju struktūra

M2M transformācijas tiek veidotas, izmantojot [1] aprakstītos transformācijas elementus. No visām potenciāli iespējamajām transformācijām tiks atbalstīta saprātīga apakškopa (tiek izslēgtas transformācijas, kurās elementa apstrādē nepieciešami vairāk kā 2 modeļi).

Transformācijas sastāv no transformācijas blokiem, kas savukārt sastāv no transformācijas likumiem.

3.1. Transformācijas likums

Jebkuras transformācijas pamats ir *transformācijas likums* - aprakstīto elementu virkne. Likumi ir divu veidu:

1. izpildāms likums tiek izpildīts aktuālajā transformācijas kontekstā neatkarīgi no pārējiem likumiem (tas drīkst atsaukties uz citiem likumiem, izmantojot *Clause_callNonLoop*).
2. referencējams likums var tikt tikai izsaukts no cita likuma izmantojot *Clause_callNonLoop* elementu. Referencējamie likumi ir analogi apakšprogrammām programmēšanas valodās (izpildāmie likumi - analogi galvenajām programmām). Referencējams likums izmanto izsaucošā likuma kontekstu, no kura tipiski informācija tiek nodota caur *transformācijas mainīgajiem*.

3.2. Transformācijas bloks

Transformācijas likumi tiek apvienoti ***transformācijas blokos*** - visi bloka likumi strādā uz viena modeļu pāra. Vienu no modeļiem ērtības labad sauc par izejas modeli un otru - par mērķa modeli. Ja bloka likumi paredzēti viena modeļa modificējošai apstrādei, tad tas blokam ir reizē gan izejas gan mērķa modelis (jāņem vērā, ka šādā situācijā "spoguļošana" netiek atļauta).

Transformācijas bloka izpildāmie likumi tiek izpildīti to uzrakstīšanas secībā.

3.3. Transformācija

Transformācija apvieno transformācijas blokus. Transformācija strādā uz patvaļīga skaita modeļiem, no kuriem katrā transformācijas blokā piedalās 2 (vai 1). Jebkurš transformācijas modelis var būt jebkurā lomā (izejas/mērķa), t.i. modelis dažādos blokos drīkst būt dažādās lomās. Šāda pieeja ļauj noformēt sarežģītu manipulāciju sēriju kā vienu transformāciju.

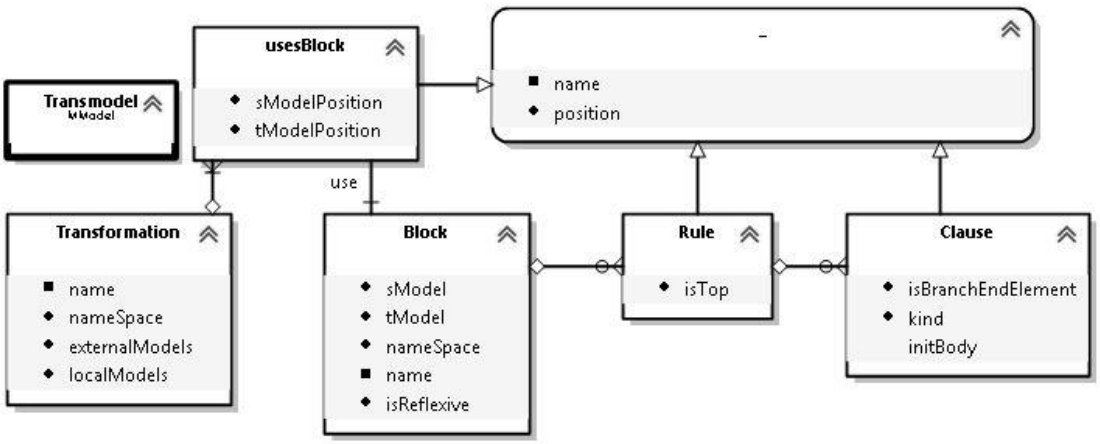
Transformācijas bloki izpildās to uzrakstīšanas secībā.

4. Transformāciju valodas izpildes mašīna

Transformāciju bāzes realizācija C# tieši atbilst iepriekšējā nodaļa un [1] aprakstītajai hierarhijai: transformācija - transformācijas bloks - transformācijas likums - transformācijas elements. Loģiskajā līmenī izklāstītie algoritmi realizēti atbilstošo klašu samērā sarežģītās metodēs, kas lielā mērā ir tehnisks jautājums.

Interesantāka ir transformāciju valodas realizācija:

- tika izveidots transformāciju definīcijas metamodelis (skat. zīm. 1)
- [1] ietvaros tika izstrādāts šeit definētās valodas analizators, izmantojot LEX un YACC
- [1] izstrādātais analizators tika savienots ar modeļa iepildīšanas funkcijām, tādējādi iepildot analīzes rezultātu modelī (būtībā specifiska T2M transformācija)
- tika izveidota T4 transformācija, kas, izmantojot modeļa apstaigāšanas funkcijas, ģenerēja C# kodu (Būtībā specifiska M2T transformācija)



Zīmējums 1. TDL meta modelis.

Tātad šādā veidā tika iegūts transformators no transformāciju definīcijas valodas uz atbilstošo C# kodu, kas būtiski atvieglo transformāciju izstrādi un uzturēšanu.

5. Secinājumi un rezultāti

Aktivitātes ietvaros ir izstrādāta modeļa apstrādes transformāciju mašīna.

6. Literatūras saraksts

[1] 1.3.2. "Modeļa transformāciju atbalsta izstrāde (M2M transformāciju atbalsts)" progresa pārskats

7. Pielikumi.

7.1. Transformāciju valodas izpildes mašīnas resultāta piemērs.

Piemērā ir noģenerētais C# kods transformācijai, kas pārveido Join modeļu datus - no loģiskās uz tehnisko.

```
using System.Collections.Generic;
using System.Linq;

// =========== BLOCK ======= bridge ================================
namespace LogicalJoin2 {
  public partial class @bridge :
MEDUS.TransformationBlockTyped<LogicalJoin.MDSmo_LogicalJoin,
LogicalJoin.MDSmo_LogicalJoin> {

    partial class E_baseModel_1 : MEDUS.Clause_external {
      public LogicalJoin2.bridge myBlockT { get { return
(LogicalJoin2.bridge)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         var tr=new LogicalEntity2.Tbridge();
            tr.logger=myBlock.logger;
               tr.transform(myBlockT.sModel, myBlockT.sModel);
        initManualAfter();
        return true;
        }
      }

    partial class E_preprocess_1 : MEDUS.Clause_external {
      public LogicalJoin2.bridge myBlockT { get { return
(LogicalJoin2.bridge)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         myBlockT.sModelTyped.MDSpreprocess_LogicalJoin(myBlock.logger, true);
        initManualAfter();
        return true;
        }
      }

    partial class E_orderJoins_1 : MEDUS.Clause_list {
      public LogicalJoin2.bridge myBlockT { get { return
(LogicalJoin2.bridge)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Join>(0).Select(o=>o.MDSobject.ISN)
.ToList();
        initManualAfter();
        return true;
        }
```

```
      }

    partial class E_orderJoins_2 : MEDUS.Clause_varSet {
      public LogicalJoin2.bridge myBlockT { get { return
(LogicalJoin2.bridge)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_orderJoins_3 : MEDUS.Clause_external {
      public LogicalJoin2.bridge myBlockT { get { return
(LogicalJoin2.bridge)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         ((LogicalJoin.Join)myBlock.getDobject("j")).makePartNumbering();
        initManualAfter();
        return true;
        }
      }

    public @bridge(MEDUS.Transformation transformation, int sourceModelId, int
targetModelId, bool initModelMirrors=false)
                : base(transformation, sourceModelId, targetModelId,
initModelMirrors) {
      isReflexive=true;
      allRules=new Dictionary<string, MEDUS.RuleDescriptor>(){
            { "baseModel", new MEDUS.RuleDescriptor(){
               new MEDUS.ClauseDescriptor(typeof(E_baseModel_1))
             } }
          , { "preprocess", new MEDUS.RuleDescriptor(){
               new MEDUS.ClauseDescriptor(typeof(E_preprocess_1))
             } }
          , { "orderJoins", new MEDUS.RuleDescriptor(){
               new MEDUS.ClauseDescriptor(typeof(E_orderJoins_1))
             , new MEDUS.ClauseDescriptor(typeof(E_orderJoins_2))
             , new MEDUS.ClauseDescriptor(typeof(E_orderJoins_3))
             } }
        };
      topRuleNames=new List<string>() {
            "baseModel"
          , "preprocess"
          , "orderJoins"
        };
      }
    }
  }

// ========== BLOCK ======= techno ================================
namespace LogicalJoin2 {
  public partial class @techno :
MEDUS.TransformationBlockTyped<LogicalJoin.MDSmo_LogicalJoin, Joins.MDSmo_Joins> {

    partial class E_model_dataBase_1 : MEDUS.Clause_external {
```

```
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         var x=myBlockT.onDataBase;
            myBlockT.tModelTyped.dataBase=
                  (string.IsNullOrWhiteSpace(x))?myBlockT.sModelTyped.dataBase:x;
        initManualAfter();
        return true;
        }
      }

    partial class E_base_1 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Join>(0).Select(o=>o.MDSobject.ISN)
.ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_base_2 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_base_3 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         var x=((LogicalJoin.Join)myBlock.getDobject("j"));
if(x.MDS__isSelected(myBlockT.onDataBase)==false) return false;
        initManualAfter();
        return true;
        }
      }

    partial class E_base_4 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
```

```
      }

    partial class E_base_5 : MEDUS.Clause_create {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        me=myBlockT.tModelTyped.MDScreateE("Join", 0).MDSbox.ISN;
        isTarget=true;
        initManualAfter();
        return true;
        }
      }

    partial class E_base_6 : MEDUS.Clause_mirrorSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_base_7 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="jT";
        initManualAfter();
        return true;
        }
      }

    partial class E_base_8 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){

((Joins.Join)myBlock.getDobject("jT")).name=((LogicalJoin.Join)myBlock.getDobject(
"j")).name;
        initManualAfter();
        return true;
        }
      }

    partial class E_base_9 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_base_10 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
```

```
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Jpart>(currentISN()).Select(o=>o.MD
Sobject.ISN).ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_base_11 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p";
        initManualAfter();
        return true;
        }
      }

    partial class E_base_12 : MEDUS.Clause_create {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        me=myBlockT.tModelTyped.MDScreateE("JPart",
(myBlock.getVisn("jT"))).MDSbox.ISN;
        isTarget=true;
        initManualAfter();
        return true;
        }
      }

    partial class E_base_13 : MEDUS.Clause_mirrorSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_base_14 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="pT";
        initManualAfter();
        return true;
        }
      }

    partial class E_base_15 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){

if(((LogicalJoin.Jpart)myBlock.getDobject("p")).basedOn.isHardDelete==true)
          ((Joins.JPart)myBlock.getDobject("pT")).isHardDelete=true;
```

```
        initManualAfter();
        return true;
        }
      }

    partial class E_base_16 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p";
        initManualAfter();
        return true;
        }
      }

    partial class E_base_17 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;
        chain=(((LogicalJoin.Jpart)currentD).MDS_pp_basedOn.MDSfullValueI);
        initManualAfter();
        return true;
        }
      }

    partial class E_base_18 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="e";
        initManualAfter();
        return true;
        }
      }

    partial class E_base_19 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         var x=((Joins.JPart)myBlock.getDobject("pT"));
x.name=((LogicalJoin.Jpart)myBlock.getDobject("p")).name;
x.tableName=((LogicalJoin.Entity)myBlock.getDobject("e")).name;
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_1 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;
```

```
chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Join>(0).Select(o=>o.MDSobject.ISN)
.ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_2 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_3 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         var x=((LogicalJoin.Join)myBlock.getDobject("j"));
if(x.MDS__isSelected(myBlockT.onDataBase)==false) return false;
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_4 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_5 : MEDUS.Clause_mirror {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_main_next_6 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="jT";
        initManualAfter();
        return true;
        }
      }
```

```
    partial class E_main_next_7 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_8 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;
        chain=(((LogicalJoin.Join)currentD).MDS_pp_main.MDSfullValueI);
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_9 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_10 : MEDUS.Clause_mirror {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_main_next_11 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="pT";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_12 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){

((Joins.Join)myBlock.getDobject("jT")).MDS_pp_main.addValue((myBlock.getVisn("pT")
));
```

```
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_13 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class orderedP : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Jpart>(currentISN()).Select(o=>o.MD
Sobject.ISN).ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_15 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p1";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_16 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        if((myBlock.getVisn("p"))==(myBlock.getVisn("p1"))) return false;
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_17 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p";
```

```
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_18 : MEDUS.Clause_mirror {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_main_next_19 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="pT";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_20 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p1";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_21 : MEDUS.Clause_mirror {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_main_next_22 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p1T";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_23 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){

((Joins.JPart)myBlock.getDobject("pT")).MDS_pp_next.addValue((myBlock.getVisn("p1T
")));
        initManualAfter();
        return true;
```

```
        }
      }

    partial class E_main_next_24 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p1";
        initManualAfter();
        return true;
        }
      }

    partial class E_main_next_25 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="p";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_1 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Join>(0).Select(o=>o.MDSobject.ISN)
.ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_link_2 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_3 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         var x=((LogicalJoin.Join)myBlock.getDobject("j"));
if(x.MDS__isSelected(myBlockT.onDataBase)==false) return false;
        initManualAfter();
```

```
        return true;
        }
      }

    partial class E_link_4 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="j";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_5 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Jpart>(currentISN()).Select(o=>o.MD
Sobject.ISN).ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_link_6 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalJoin.Jlink>(currentISN()).Select(o=>o.MD
Sobject.ISN).ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_link_7 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="l";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_8 : MEDUS.Clause_create {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
```

```
      protected sealed override bool init(){
        me=myBlockT.tModelTyped.MDScreateE("JCondition", 0).MDSbox.ISN;
        isTarget=true;
        initManualAfter();
        return true;
        }
      }

    partial class E_link_9 : MEDUS.Clause_mirrorSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_link_10 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="lT";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_11 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="l";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_12 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;
        chain=(((LogicalJoin.Jlink)currentD).MDS_pp_to.MDSfullValueI);
        initManualAfter();
        return true;
        }
      }

    partial class E_link_13 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="pt";
        initManualAfter();
        return true;
        }
      }
```

```
    partial class E_link_14 : MEDUS.Clause_mirror {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_link_15 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="ptT";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_16 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="l";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_17 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;
        chain=(((LogicalJoin.Jlink)currentD).MDS_pp_basedOn.MDSfullValueI);
        initManualAfter();
        return true;
        }
      }

    partial class E_link_18 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="lb";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_19 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        if(((LogicalJoin.Link)myBlock.getDobject("lb")).isBase)
           ((Joins.JCondition)myBlock.getDobject("lT")).isBase=true;
```

```
        initManualAfter();
        return true;
        }
      }

    partial class E_link_20 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="pt";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_21 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;
        chain=(((LogicalJoin.Jpart)currentD).MDS_pp_basedOn.MDSfullValueI);
        initManualAfter();
        return true;
        }
      }

    partial class E_link_22 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="e";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_23 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         myBlock.variables["tPK"]=new MEDUS.UniversalObject();
        initManualAfter();
        return true;
        }
      }

    partial class E_link_24 : MEDUS.Clause_callNonLoop {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        ruleName="findPK"; kind=callKind.exists;
        initManualAfter();
        return true;
```

```
        }
      }

    partial class E_link_25 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="tPK";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_26 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         myBlock.variables["pf"]=new
MEDUS.UniversalObject(((LogicalJoin.Jlink)myBlock.getDobject("1")).MDSbox.myContai
ner, false);
        initManualAfter();
        return true;
        }
      }

    partial class E_link_27 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="pf";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_28 : MEDUS.Clause_mirror {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      }

    partial class E_link_29 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="pfT";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_30 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
```

```
      protected sealed override bool init(){
         var l=((LogicalJoin.Jlink)myBlock.getDobject("l"));
          var ex=l.extend;
          var pf=((LogicalJoin.Jpart)myBlock.getDobject("pf"));
          var pt=((LogicalJoin.Jpart)myBlock.getDobject("pt"));
          var tPK=((string)(myBlock.getCobject("tPK")));
          var lT=((Joins.JCondition)myBlock.getDobject("lT"));
          if(pf.position==pt.position){
            var m=string.Format("{0}[{1}].{2}->{3}[{4}].{5}{6}",
              pf.name, pf.position, l.basedOn.name, pt.name, pt.position, tPK,
ex==null?"":"("+ex+")");
       //     lT.sourceField=m;
              return true;
              }
          if(pf.position<pt.position){

((Joins.JPart)myBlock.getDobject("ptT")).MDS_pp_condition.addValue((myBlock.getVis
n("lT")));
              lT.MDS_pp_sourceJPart.addValue((myBlock.getVisn("pfT")));
       //     lT.sourceField=l.basedOn.name;
       //     lT.targetField=tPK;
              switch(ex){
                case null: break;
                case "target": lT.isLeftJoin=true; break;
                case "source": lT.isRightJoin=true; break;
                case "both":   lT.isLeftJoin=true; lT.isRightJoin=true; break;
                }
             } else {

((Joins.JPart)myBlock.getDobject("pfT")).MDS_pp_condition.addValue((myBlock.getVis
n("lT")));
              lT.MDS_pp_sourceJPart.addValue((myBlock.getVisn("ptT")));
       //     lT.sourceField=tPK;
       //     lT.targetField=l.basedOn.name;
              switch(ex){
                case null: break;
                case "target": lT.isRightJoin=true; break;
                case "source": lT.isLeftJoin=true; break;
                case "both":   lT.isLeftJoin=true; lT.isRightJoin=true; break;
                }
               }

        initManualAfter();
        return true;
        }
      }

    partial class E_link_31 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="lb";
        initManualAfter();
        return true;
        }
      }
```

```
    partial class E_link_32 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalEntity.SubField>(currentISN()).Select(o=
>o.MDSobject.ISN).ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_link_33 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="sf";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_34 : MEDUS.Clause_create {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        me=myBlockT.tModelTyped.MDScreateE("JoiningPair",
(myBlock.getVisn("lT"))).MDSbox.ISN;
        isTarget=true;
        initManualAfter();
        return true;
        }
      }

    partial class E_link_35 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="jpT";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_36 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="sf";
        initManualAfter();
        return true;
        }
```

```
      }

    partial class E_link_37 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;
        chain=(((LogicalJoin.SubField)currentD).MDS_pp_pkType.MDSfullValueI);
        initManualAfter();
        return true;
        }
      }

    partial class E_link_38 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="fts";
        initManualAfter();
        return true;
        }
      }

    partial class E_link_39 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
         var x=((LogicalEntity.SubField)myBlock.getDobject("sf")); var
y=((Joins.JoiningPair)myBlock.getDobject("jpT"));
            y.type=((LogicalEntity.Scalar)myBlock.getDobject("fts")).type;

if(((LogicalJoin.Jpart)myBlock.getDobject("pf")).position<((LogicalJoin.Jpart)myBl
ock.getDobject("pt")).position){
                 y.sourceField=x.name;
                 y.targetField=x.targetUnitName;
                    if(((LogicalJoin.Link)myBlock.getDobject("lb")).isMandatory)
                      y.sourceFieldIsMandatory=true;
                    y.targetFieldIsMandatory=true;
                } else {
                 y.sourceField=x.targetUnitName;
                 y.targetField=x.name;
                    if(((LogicalJoin.Link)myBlock.getDobject("lb")).isMandatory)
                      y.targetFieldIsMandatory=true;
                    y.sourceFieldIsMandatory=true;
                 }

        initManualAfter();
        return true;
        }
      }

    partial class E_findPK_1 : MEDUS.Clause_varGet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
```

```
      protected sealed override bool init(){
        varName="e";
        initManualAfter();
        return true;
        }
      }

    partial class E_findPK_2 : MEDUS.Clause_list {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        isTarget=false;

chain=myBlockT.sModelTyped.MDSgetD<LogicalEntity.Unit>(currentISN()).Select(o=>o.M
DSobject.ISN).ToList();
        initManualAfter();
        return true;
        }
      }

    partial class E_findPK_3 : MEDUS.Clause_varSet {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        varName="u";
        initManualAfter();
        return true;
        }
      }

    partial class E_findPK_4 : MEDUS.Clause_external {
      public LogicalJoin2.techno myBlockT { get { return
(LogicalJoin2.techno)myBlock; }}
      partial void initManualAfter();
      protected sealed override bool init(){
        if(((LogicalEntity.Unit)myBlock.getDobject("u")).isPK!=true) return false;
         myBlock.variables["tPK"]=new
MEDUS.UniversalObject(((LogicalEntity.Unit)myBlock.getDobject("u")).name);
        initManualAfter();
        return true;
        }
      }

    public @techno(MEDUS.Transformation transformation, int sourceModelId, int
targetModelId, bool initModelMirrors=false)
                : base(transformation, sourceModelId, targetModelId,
initModelMirrors) {
      allRules=new Dictionary<string, MEDUS.RuleDescriptor>(){
            { "model_dataBase", new MEDUS.RuleDescriptor(){
               new MEDUS.ClauseDescriptor(typeof(E_model_dataBase_1))
             } }
          , { "base", new MEDUS.RuleDescriptor(){
               new MEDUS.ClauseDescriptor(typeof(E_base_1))
             , new MEDUS.ClauseDescriptor(typeof(E_base_2))
             , new MEDUS.ClauseDescriptor(typeof(E_base_3))
             , new MEDUS.ClauseDescriptor(typeof(E_base_4))
```

```
        , new MEDUS.ClauseDescriptor(typeof(E_base_5))
        , new MEDUS.ClauseDescriptor(typeof(E_base_6))
        , new MEDUS.ClauseDescriptor(typeof(E_base_7))
        , new MEDUS.ClauseDescriptor(typeof(E_base_8))
        , new MEDUS.ClauseDescriptor(typeof(E_base_9))
        , new MEDUS.ClauseDescriptor(typeof(E_base_10))
        , new MEDUS.ClauseDescriptor(typeof(E_base_11))
        , new MEDUS.ClauseDescriptor(typeof(E_base_12))
        , new MEDUS.ClauseDescriptor(typeof(E_base_13))
        , new MEDUS.ClauseDescriptor(typeof(E_base_14))
        , new MEDUS.ClauseDescriptor(typeof(E_base_15))
        , new MEDUS.ClauseDescriptor(typeof(E_base_16))
        , new MEDUS.ClauseDescriptor(typeof(E_base_17))
        , new MEDUS.ClauseDescriptor(typeof(E_base_18))
        , new MEDUS.ClauseDescriptor(typeof(E_base_19))
        } }
    , { "main_next", new MEDUS.RuleDescriptor(){
          new MEDUS.ClauseDescriptor(typeof(E_main_next_1))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_2))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_3))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_4))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_5))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_6))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_7))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_8))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_9))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_10))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_11))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_12))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_13))
        , new MEDUS.ClauseDescriptor(typeof(orderedP))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_15))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_16))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_17))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_18))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_19))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_20))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_21))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_22))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_23))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_24))
        , new MEDUS.ClauseDescriptor(typeof(E_main_next_25))
        } }
    , { "link", new MEDUS.RuleDescriptor(){
          new MEDUS.ClauseDescriptor(typeof(E_link_1))
        , new MEDUS.ClauseDescriptor(typeof(E_link_2))
        , new MEDUS.ClauseDescriptor(typeof(E_link_3))
        , new MEDUS.ClauseDescriptor(typeof(E_link_4))
        , new MEDUS.ClauseDescriptor(typeof(E_link_5))
        , new MEDUS.ClauseDescriptor(typeof(E_link_6))
        , new MEDUS.ClauseDescriptor(typeof(E_link_7))
        , new MEDUS.ClauseDescriptor(typeof(E_link_8))
        , new MEDUS.ClauseDescriptor(typeof(E_link_9))
        , new MEDUS.ClauseDescriptor(typeof(E_link_10))
        , new MEDUS.ClauseDescriptor(typeof(E_link_11))
        , new MEDUS.ClauseDescriptor(typeof(E_link_12))
        , new MEDUS.ClauseDescriptor(typeof(E_link_13))
        , new MEDUS.ClauseDescriptor(typeof(E_link_14))
```

```
                , new MEDUS.ClauseDescriptor(typeof(E_link_15))
                , new MEDUS.ClauseDescriptor(typeof(E_link_16))
                , new MEDUS.ClauseDescriptor(typeof(E_link_17))
                , new MEDUS.ClauseDescriptor(typeof(E_link_18))
                , new MEDUS.ClauseDescriptor(typeof(E_link_19))
                , new MEDUS.ClauseDescriptor(typeof(E_link_20))
                , new MEDUS.ClauseDescriptor(typeof(E_link_21))
                , new MEDUS.ClauseDescriptor(typeof(E_link_22))
                , new MEDUS.ClauseDescriptor(typeof(E_link_23))
                , new MEDUS.ClauseDescriptor(typeof(E_link_24))
                , new MEDUS.ClauseDescriptor(typeof(E_link_25))
                , new MEDUS.ClauseDescriptor(typeof(E_link_26))
                , new MEDUS.ClauseDescriptor(typeof(E_link_27))
                , new MEDUS.ClauseDescriptor(typeof(E_link_28))
                , new MEDUS.ClauseDescriptor(typeof(E_link_29))
                , new MEDUS.ClauseDescriptor(typeof(E_link_30))
                , new MEDUS.ClauseDescriptor(typeof(E_link_31))
                , new MEDUS.ClauseDescriptor(typeof(E_link_32))
                , new MEDUS.ClauseDescriptor(typeof(E_link_33))
                , new MEDUS.ClauseDescriptor(typeof(E_link_34))
                , new MEDUS.ClauseDescriptor(typeof(E_link_35))
                , new MEDUS.ClauseDescriptor(typeof(E_link_36))
                , new MEDUS.ClauseDescriptor(typeof(E_link_37))
                , new MEDUS.ClauseDescriptor(typeof(E_link_38))
                , new MEDUS.ClauseDescriptor(typeof(E_link_39))
                } }
            , { "MDSgnr1", new MEDUS.RuleDescriptor(){
                } }
            , { "findPK", new MEDUS.RuleDescriptor(){
                  new MEDUS.ClauseDescriptor(typeof(E_findPK_1))
                , new MEDUS.ClauseDescriptor(typeof(E_findPK_2))
                , new MEDUS.ClauseDescriptor(typeof(E_findPK_3))
                , new MEDUS.ClauseDescriptor(typeof(E_findPK_4))
                } }
          };
        topRuleNames=new List<string>() {
              "model_dataBase"
            , "base"
            , "main_next"
            , "link"
          };
        }
      }
    }

// ========== TRANSFORMATION ======= Tbridge ================================
namespace LogicalJoin2 {
  public partial class @Tbridge : MEDUS.TransformationSimple<LogicalJoin2.bridge>
{
//   public static void Transform(LogicalJoin.MDSmo_LogicalJoin modelS,
LogicalJoin.MDSmo_LogicalJoin modelT) {
//      new @Tbridge().transform(modelS, modelT);
//      }
//    public static LogicalJoin.MDSmo_LogicalJoin
Transform(LogicalJoin.MDSmo_LogicalJoin modelS) {
//      var modelT=new @LogicalJoin.MDSmo_LogicalJoin();
//      Transform(modelS, modelT);
//      return modelT;
```

```
//      }
    }
  }

// ========== TRANSFORMATION ======= Ttechno ==============================
namespace LogicalJoin2 {
  public partial class @Ttechno : MEDUS.TransformationSimple<LogicalJoin2.techno>
{
//   public static void Transform(LogicalJoin.MDSmo_LogicalJoin modelS,
Joins.MDSmo_Joins modelT) {
//      new @Ttechno().transform(modelS, modelT);
//      }
//    public static Joins.MDSmo_Joins Transform(LogicalJoin.MDSmo_LogicalJoin
modelS) {
//      var modelT=new @Joins.MDSmo_Joins();
//      Transform(modelS, modelT);
//      return modelT;
//      }
    }
  }
//============================END OF FILE==================================
```