Mdd4cca

Model-driven Development of Composite Content Applications
www.mdd4cca.com

International Workshop on Advanced Topics in Software Engineering (ATSEN'14)
07.11.2014, Istanbul Kultur University, Istanbul, Turkey.

Ferhat Erata

About me…
Responsibilities and Research Interests

- **Responsibilities**
  - PhD student in Ege University, International Computer Institute
  - Co-founder and President, UNIT Information Technologies R&D Ltd.
  - TUBITAK TEYDEB 1501 – MDD4CCA – Software Architect (Completed)
  - EUREKA - ITEA2 - ModelWriter – Project Leader (Labelled)
  - EUREKA - ITEA3 - ModelingEdge – National Consortium Leader (Submitted)
  - EUREKA - ITEA3 - Assume – National Consortium Leader- (Submitted)

- **Research Interests:**
  - Model-driven Engineering, Domain Specific Languages
  - Software Product Line Engineering, Software Variability Management
  - Formal Specification Languages, Verification of Model Transformation
  - Multi-Paradigm Modeling, Formalism Transformation
  - Language Engineering, Language Semantics

Presentation Outline

UNIT

Mdd4cca

- Model-driven Development of Composite Content Applications
  - Domain Specific Modeling Languages
  - Model-driven Software Product Line

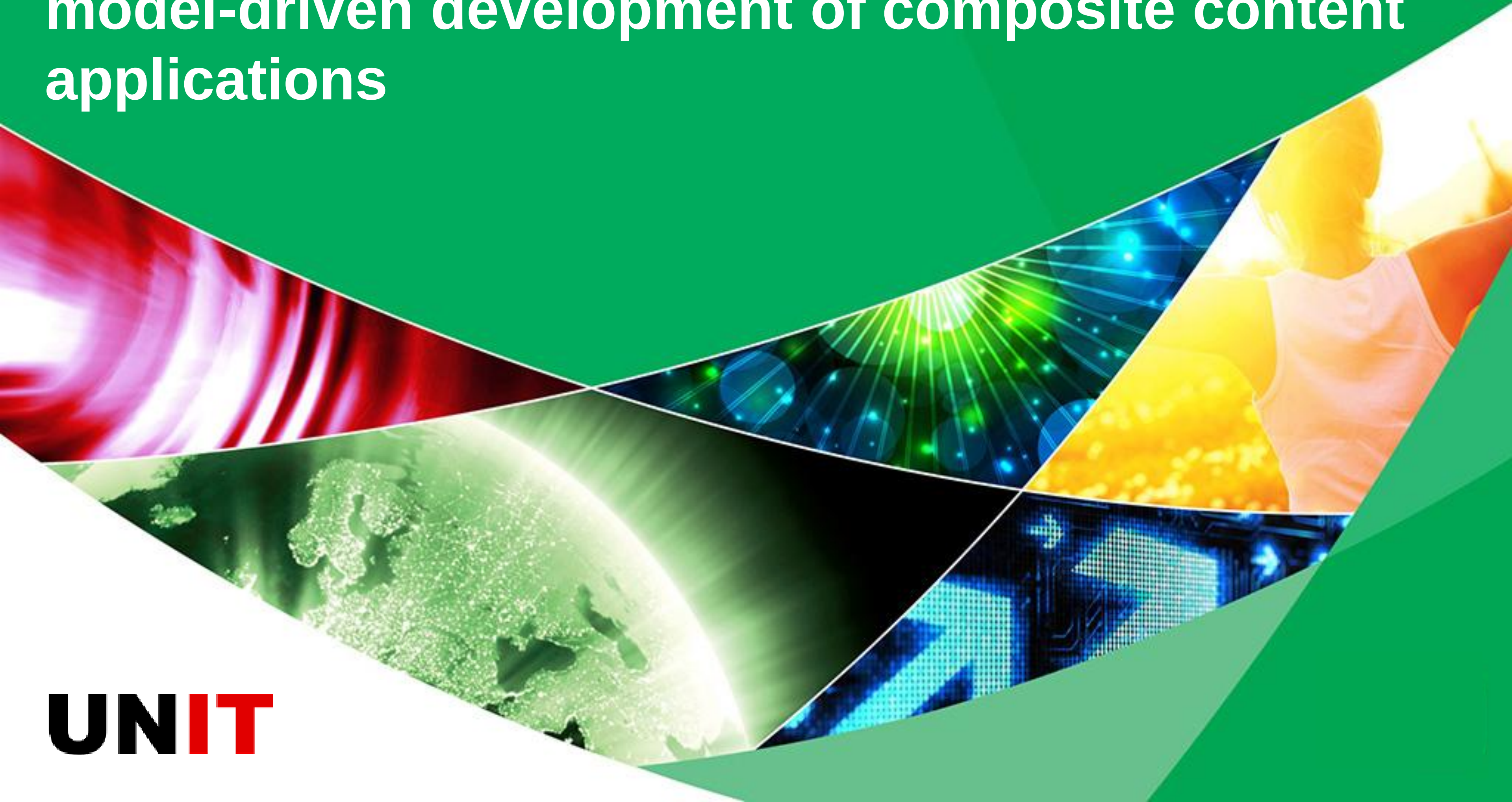
Clafer4EMF

- Clafer for Eclipse Modeling Project
  - Eclipse-based Model Verification Tool
  - Use Case Scenario

Conclusion

- ?

Mdd4cca
model-driven development of composite content applications



UNIT

mdd4cca
http://www.mdd4cca.com

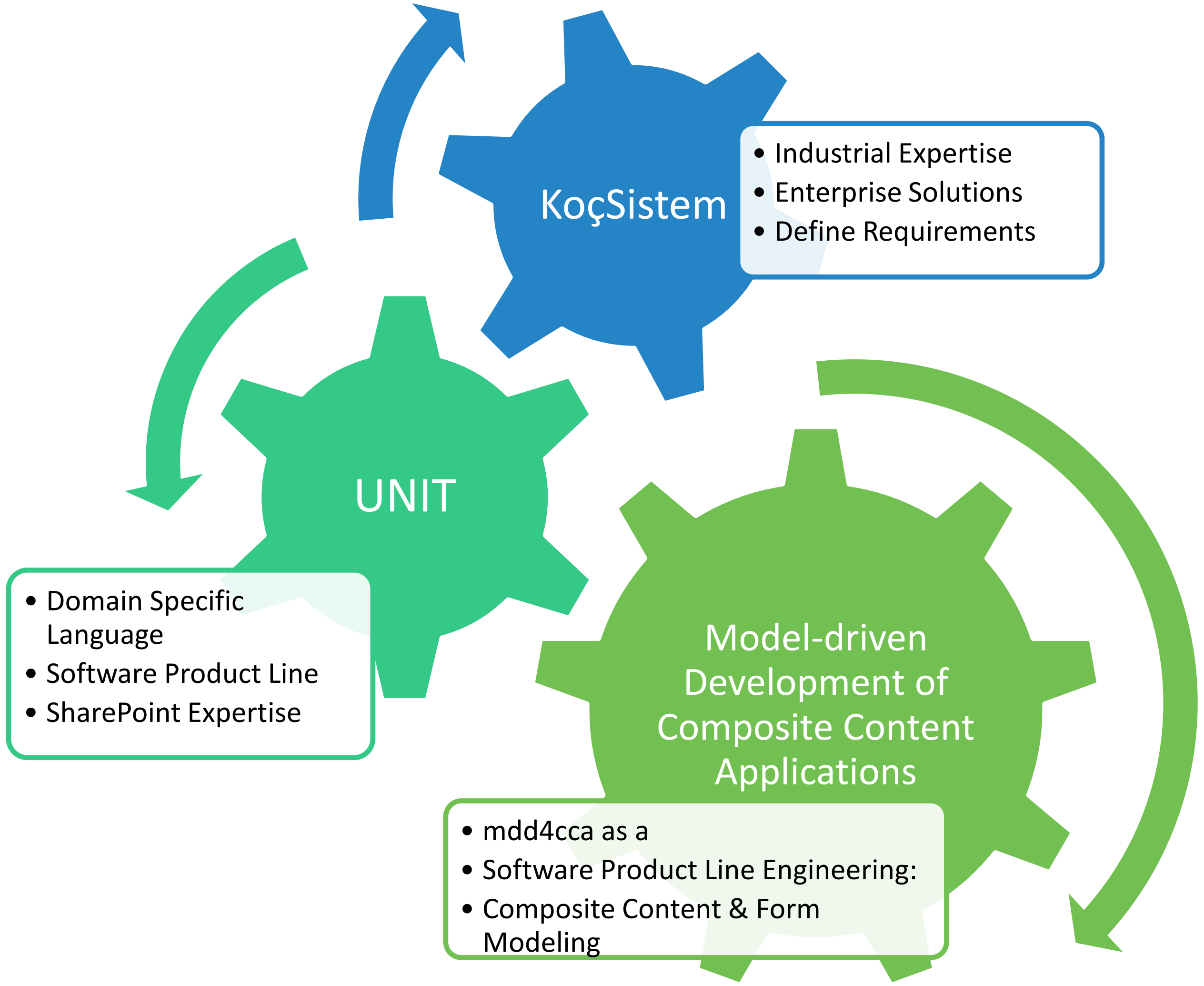
- Model Driven Development (MDD) of Composite Content Applications (CCA)
  - built on top of one or more Enterprise Content Management (ECM) platforms
  - supports singular or composite architectures
- A joint R&D project:

Composite Content Modeling in SharePoint

mdd4cca extends Entity Modeling Diagram (EDM) of Entity Framework (EF)

Content Modeling in both SharePoint Lists and EF + Form, Navigation and Workflow Modeling Ability

mdd4cca
Research & Development


KoçSistem
• Industrial Expertise
• Enterprise Solutions
• Define Requirements
UNIT
• Domain Specific Language
• Software Product Line
• SharePoint Expertise
Model-driven Development of Composite Content Applications
• mdd4cca as a
• Software Product Line Engineering:
• Composite Content & Form Modeling

mdd4cca
1501 - Industrial R&D Funding Programme of TUBITAK

UNIT

- Resource Allocation: 100 man-month
- Duration: 16 month
- Budget: 428.682,61 USD
- Start and Finish Date: 01.09.2012 – 01.01.2014
- Application Code: 3110712

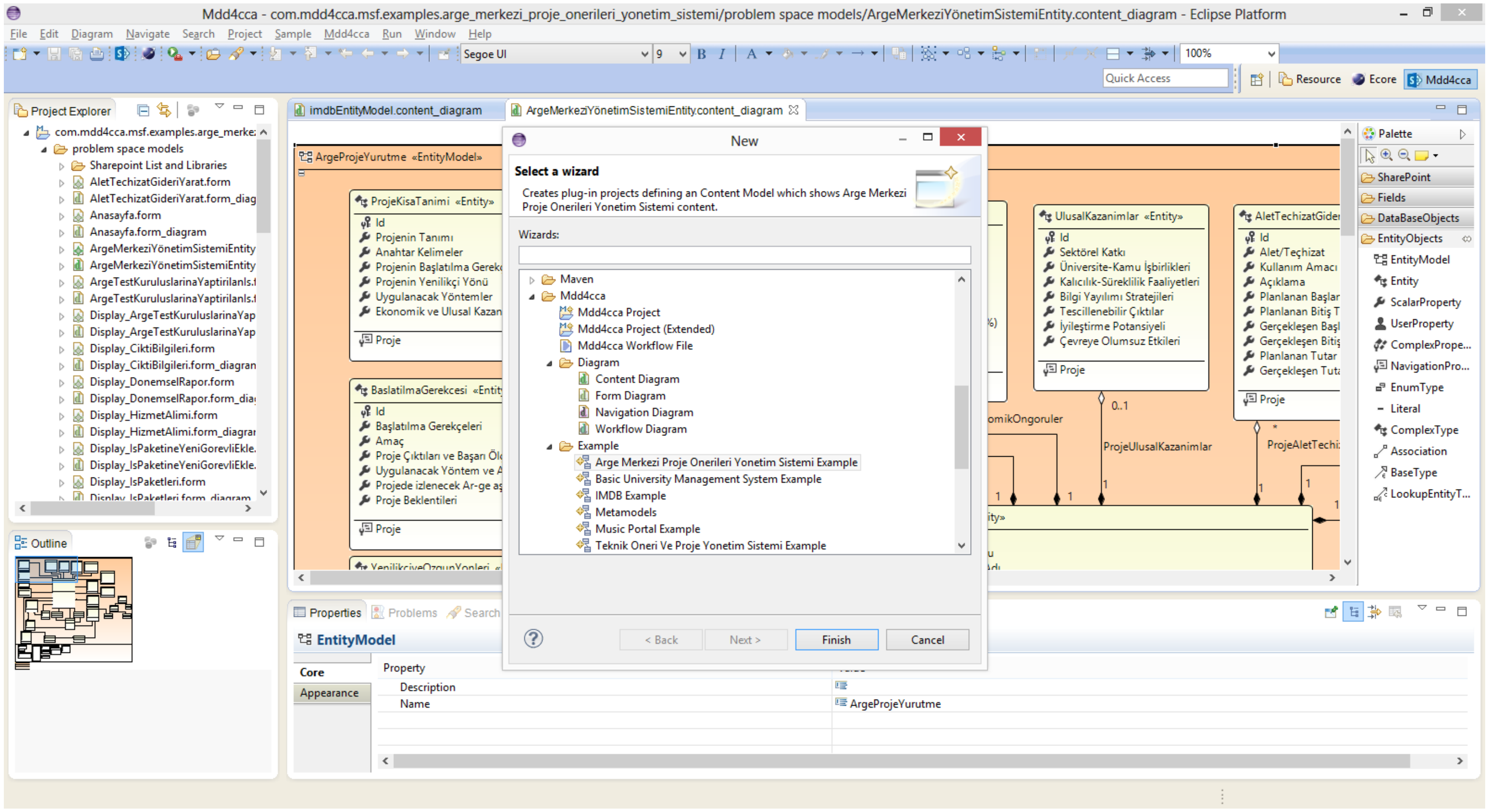
Mdd4cca
Project Team Members

- **Main Contributors**
  - Ferhat Erata[1,2], *Software Architect*
  - Moharram Challenger[2] , *Researcher*
  - Serhat Gezgin[1] , *Committer*
  - Akgün Demirbaş[1,2] , *Committer*
  - Mehmet Önat [3] , *Project Manager*
  - Prof. Geylani Kardaş[1,2] , *Tech. Consultant*

- **Acknowledgements**
  - Prof. Çağatay Çatal[4], *Tubitak Reviewer*
  - Dr. Michał Antkiewicz[5] , *Support*

[1] UNIT Bilişim Teknolojileri Ar-Ge Ltd., Ar-Ge Bölümü, Urla, İzmir
[2] Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, Izmir, Turkey
[3] KoçSistem Bilgi ve İletişim Hizmetleri A.Ş., Ar-Ge Merkezi, Istanbul, Turkey
[4] Istanbul Kultur University, Dept. of Computer Engineering, Istanbul, Turkey
[5] University of Waterloo, Generative Software Development Lab, Canada

mdd4cca
Editors, Languages, Wizards, Templates …



Mdd4cca - com.mdd4cca.msf.examples.arge_merkezi_proje_onerileri_yonetim_sistemi/problem space models/ArgeMerkeziYönetimSistemiEntity.content_diagram - Eclipse Platform
File Edit Diagram Navigate Search Project Sample Mdd4cca Run Window Help
Segoe UI
9
100%
Quick Access
Resource
Ecore
Mdd4cca
Project Explorer
imdbEntityModel.content_diagram
ArgeMerkeziYönetimSistemiEntity.content_diagram
Palette
Outline
Properties
Problems
Search

Project Explorer:

- com.mdd4cca.msf.examples.arge_merkez
  - problem space models
    - Sharepoint List and Libraries
    - AletTechizatGideriYarat.form
    - AletTechizatGideriYarat.form_diag
    - Anasayfa.form
    - Anasayfa.form_diagram
    - ArgeMerkeziYönetimSistemiEntity
    - ArgeMerkeziYönetimSistemiEntity
    - ArgeTestKuruluslarinaYaptirilanls.
    - ArgeTestKuruluslarinaYaptirilanls.
    - Display_ArgeTestKuruluslarinaYap
    - Display_ArgeTestKuruluslarinaYap
    - Display_CiktiBilgileri.form
    - Display_CiktiBilgileri.form_diagran
    - Display_DonemselRapor.form
    - Display_DonemselRapor.form_dia
    - Display_HizmetAlimi.form
    - Display_HizmetAlimi.form_diagrar
    - Display_IsPaketineYeniGorevliEkle.
    - Display_IsPaketineYeniGorevliEkle.
    - Display_IsPaketleri.form
    - Display_IsPaketleri.form_diagram

Diagram:

ArgeProjeYurutme «EntityModel»

ProjeKisaTanimi «Entity»
- Id
- Projenin Tanımı
- Anahtar Kelimeler
- Projenin Başlatılma Gereki
- Projenin Yenilikçi Yönü
- Uygulanacak Yöntemler
- Ekonomik ve Ulusal Kazan
- Proje

BaslatilmaGerekcesi «Entity»
- Id
- Başlatılma Gerekçeleri
- Amaç
- Proje Çıktıları ve Başarı Öl
- Uygulanacak Yöntem ve A
- Projede izlenecek Ar-ge aş
- Proje Beklentileri
- Proje

YenilikciveOzgunYonleri

UlusalKazanimlar «Entity»
- Id
- Sektörel Katkı
- Üniversite-Kamu İşbirlikleri
- Kalıcılık-Süreklilik Faaliyetleri
- Bilgi Yayılımı Stratejileri
- Tescillenebilir Çıktılar
- İyileştirme Potansiyeli
- Çevreye Olumsuz Etkileri
- Proje

AletTechizatGider
- Id
- Alet/Teçhizat
- Kullanım Amacı
- Açıklama
- Planlanan Başlar
- Planlanan Bitiş T
- Gerçekleşen Başl
- Gerçekleşen Bitiş
- Planlanan Tutar
- Gerçekleşen Tuta
- Proje

omikOngoruler — 0..1 — ProjeUlusalKazanimlar — ProjeAletTechi

New

Select a wizard

Creates plug-in projects defining an Content Model which shows Arge Merkezi Proje Onerileri Yonetim Sistemi content.

Wizards:

- Maven
- Mdd4cca
  - Mdd4cca Project
  - Mdd4cca Project (Extended)
  - Mdd4cca Workflow File
  - Diagram
    - Content Diagram
    - Form Diagram
    - Navigation Diagram
    - Workflow Diagram
  - Example
    - Arge Merkezi Proje Onerileri Yonetim Sistemi Example
    - Basic University Management System Example
    - IMDB Example
    - Metamodels
    - Music Portal Example
    - Teknik Oneri Ve Proje Yonetim Sistemi Example

< Back | Next > | Finish | Cancel

Palette:

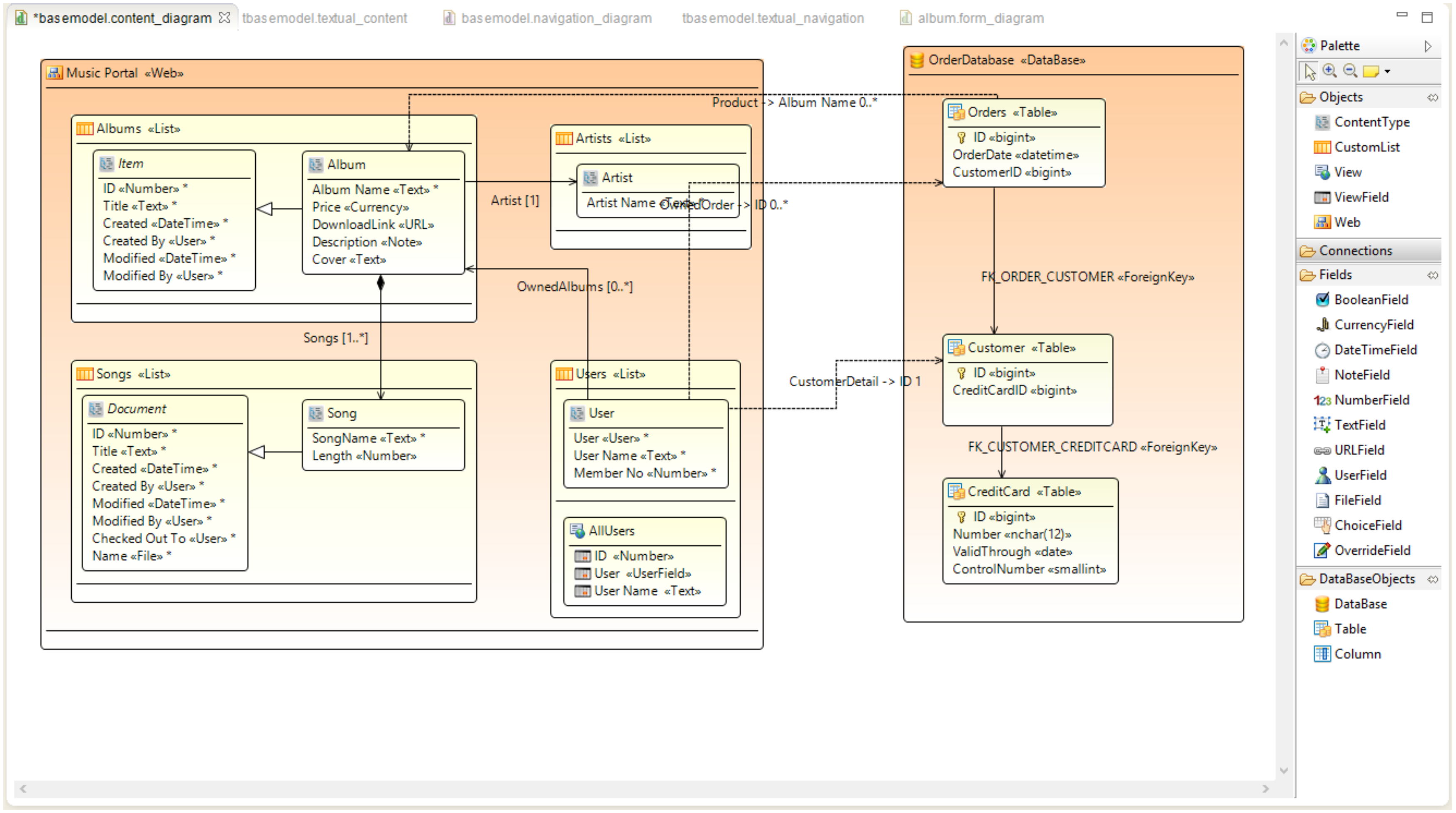
- SharePoint
- Fields
- DataBaseObjects
- EntityObjects
  - EntityModel
  - Entity
  - ScalarProperty
  - UserProperty
  - ComplexPrope...
  - NavigationPro...
  - EnumType
  - Literal
  - ComplexType
  - Association
  - BaseType
  - LookupEntityT...

Properties: **EntityModel**

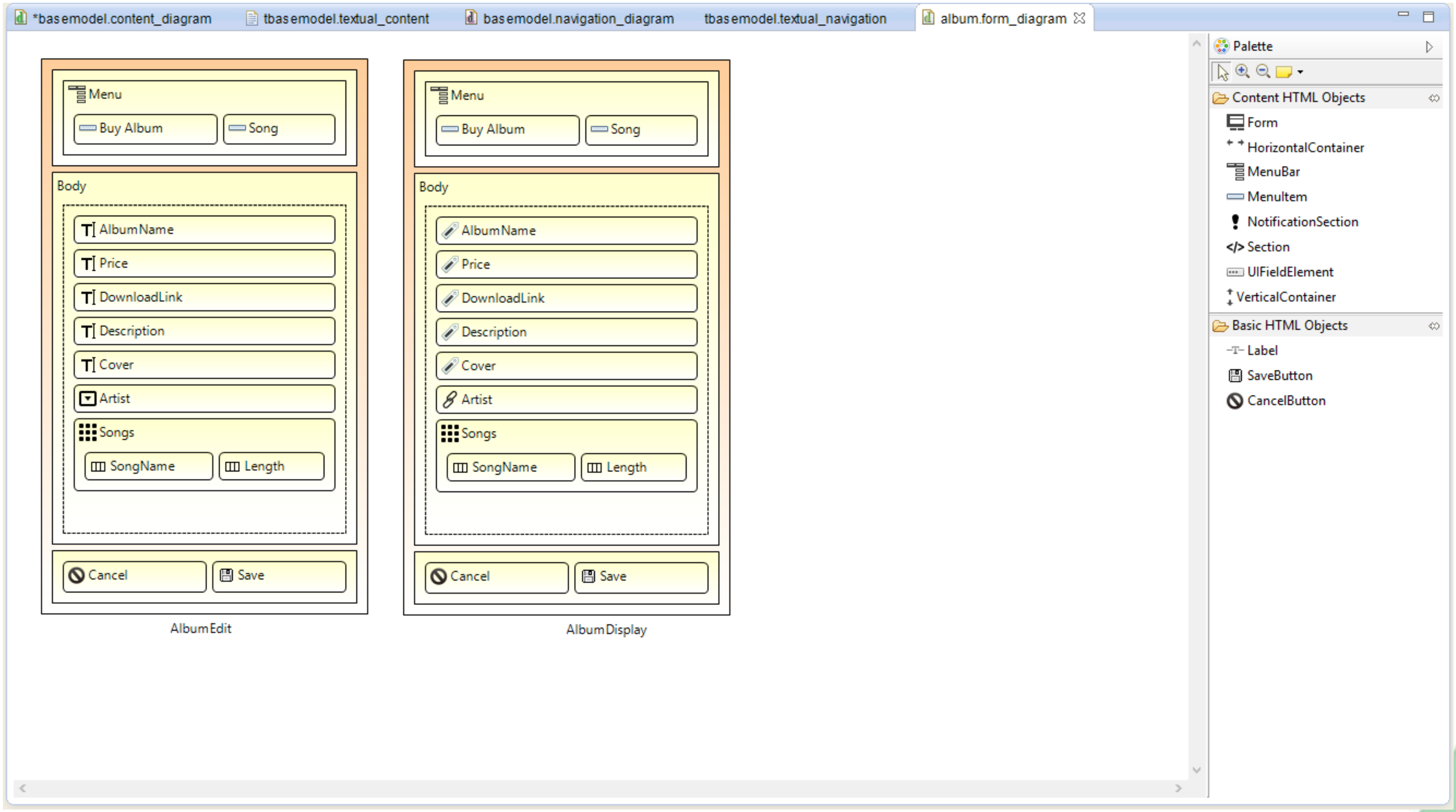
	Property	Value
Core	Description	
Appearance	Name	ArgeProjeYurutme

mdd4cca

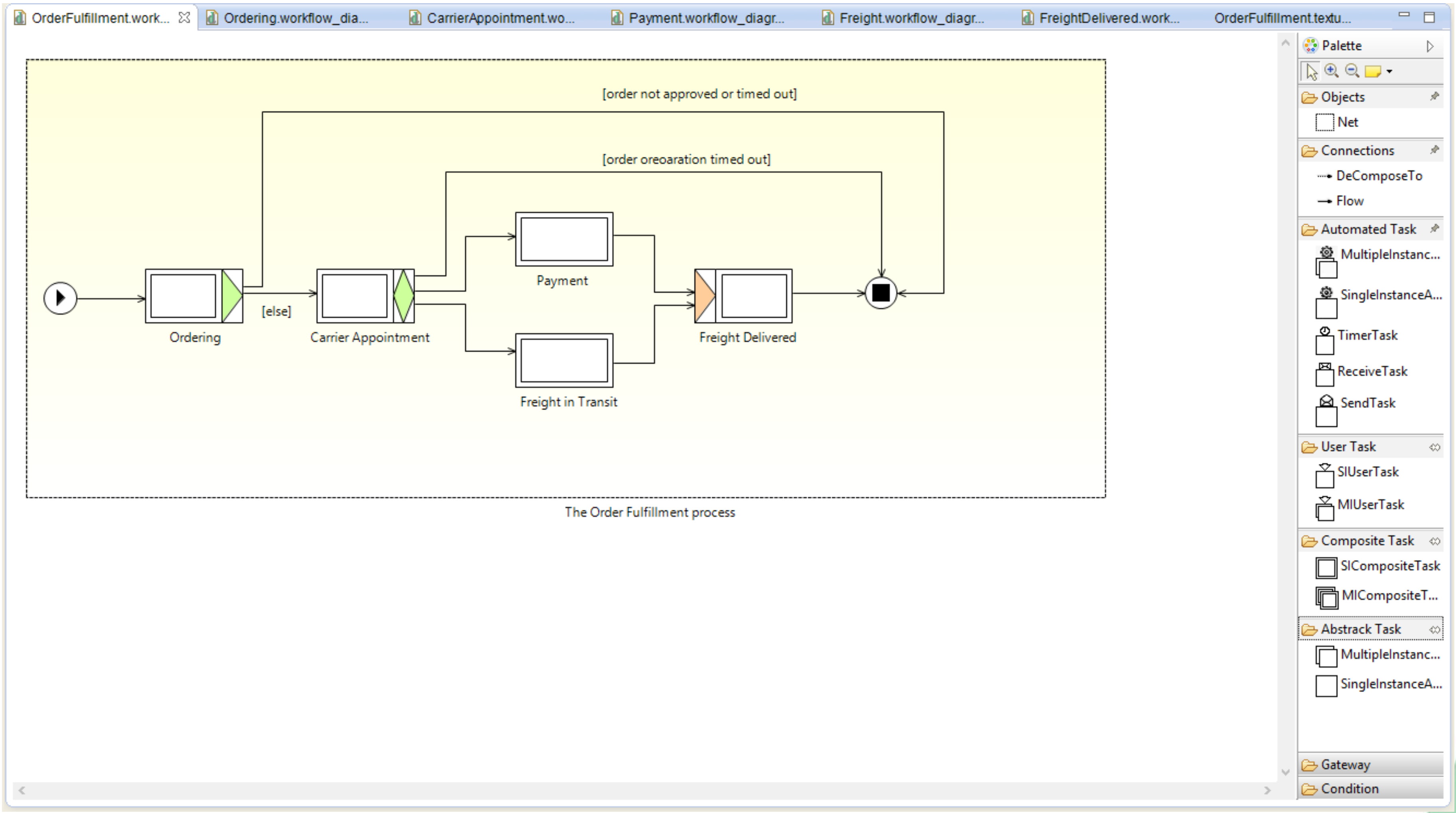
Graphical Domain Specific Languages



mdd4cca
Form Modeling

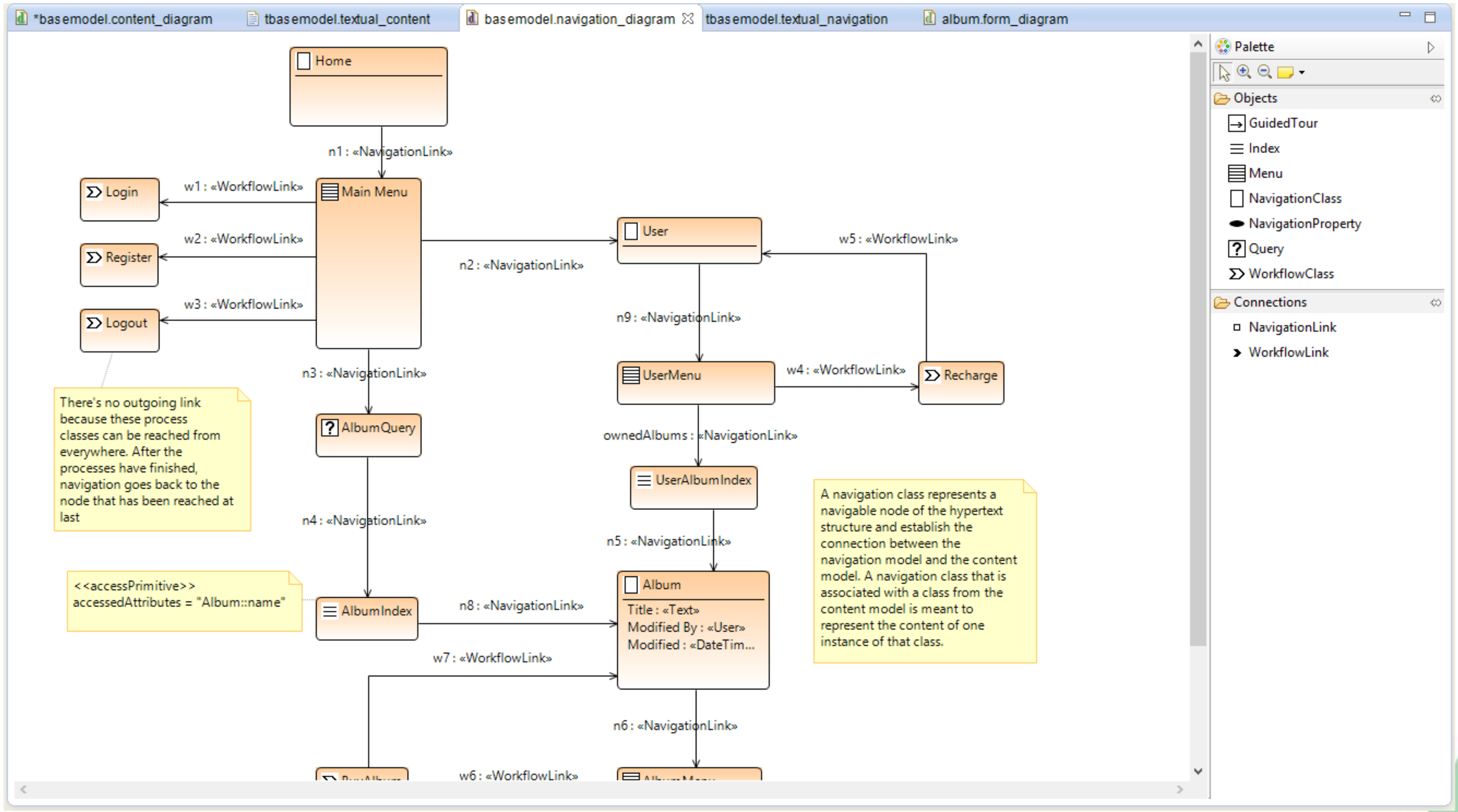


mdd4cca
Workflow Modeling



mdd4cca

Navigation Modeling

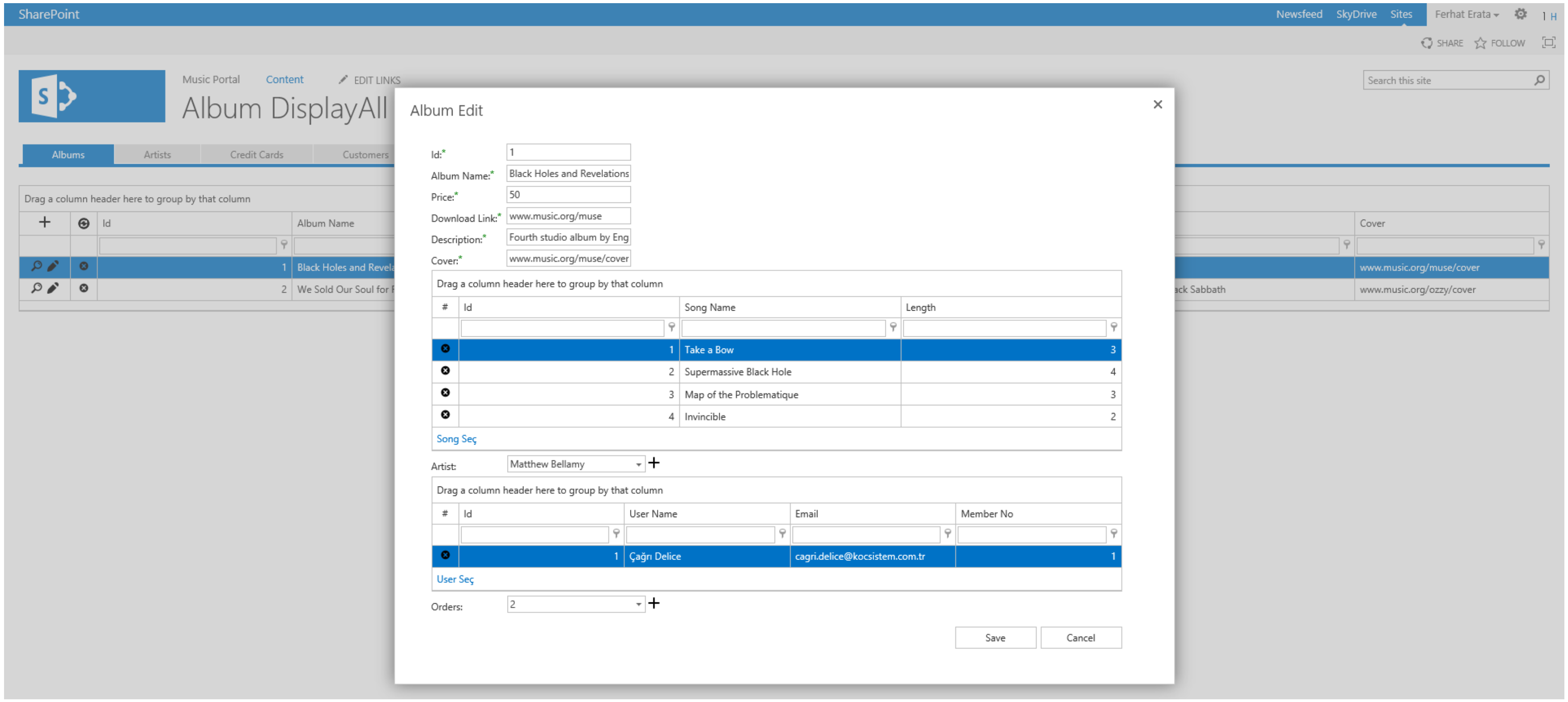


There's no outgoing link because these process classes can be reached from everywhere. After the processes have finished, navigation goes back to the node that has been reached at last

<<accessPrimitive>>
accessedAttributes = "Album::name"

A navigation class represents a navigable node of the hypertext structure and establish the connection between the navigation model and the content model. A navigation class that is associated with a class from the content model is meant to represent the content of one instance of that class.

mdd4cca
auto-generated sample application



Album DisplayAll

Drag a column header here to group by that column

		Id	Album Name	…	Cover
		1	Black Holes and Revela		www.music.org/muse/cover
		2	We Sold Our Soul for F	ack Sabbath	www.music.org/ozzy/cover

Album Edit

Id:*	1
Album Name:*	Black Holes and Revelations
Price:*	50
Download Link:*	www.music.org/muse
Description:*	Fourth studio album by Eng
Cover:*	www.music.org/muse/cover

Drag a column header here to group by that column

#	Id	Song Name	Length
	1	Take a Bow	3
	2	Supermassive Black Hole	4
	3	Map of the Problematique	3
	4	Invincible	2

Song Seç

Artist: Matthew Bellamy

Drag a column header here to group by that column

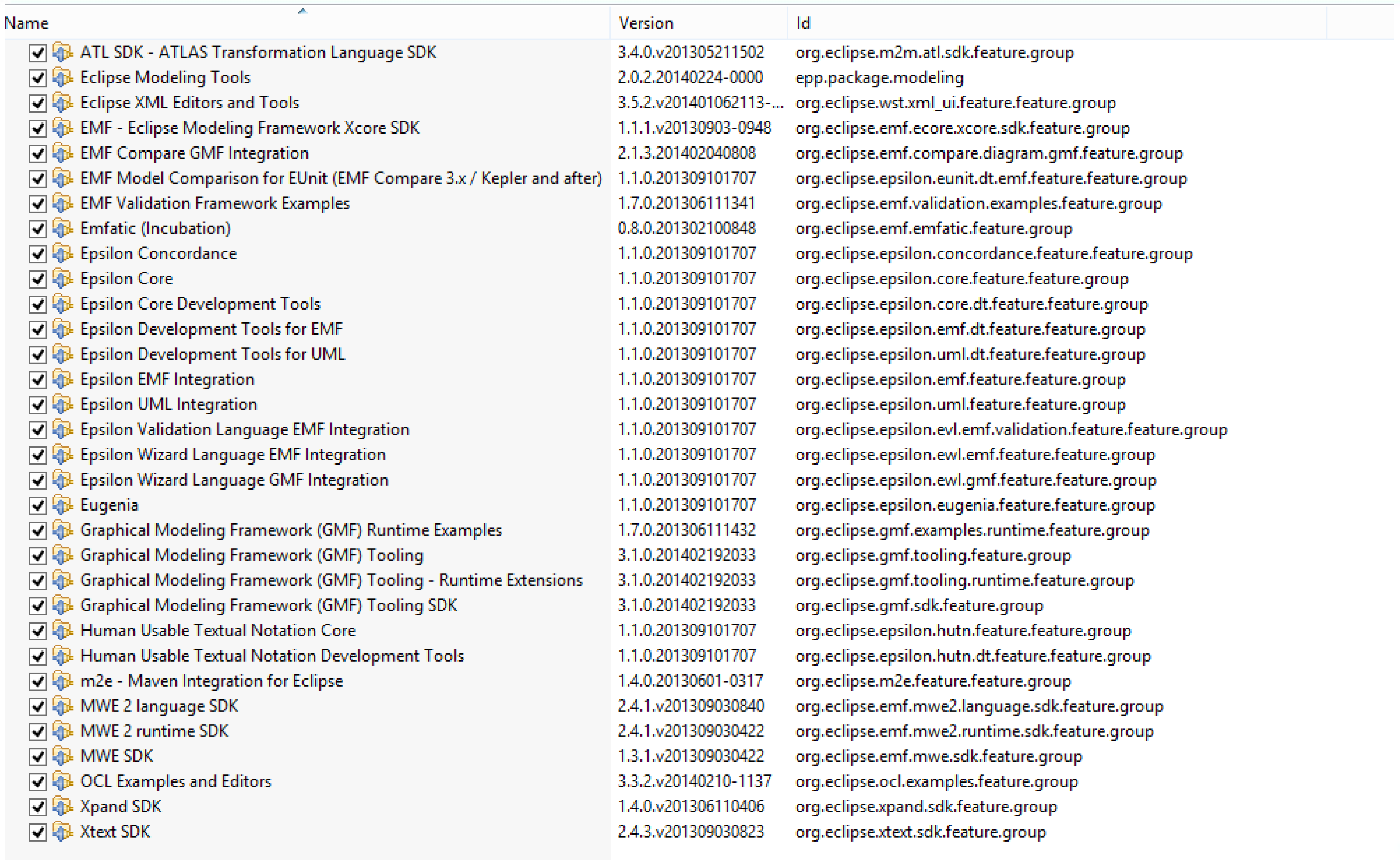
#	Id	User Name	Email	Member No
	1	Çağrı Delice	cagri.delice@kocsistem.com.tr	1

User Seç

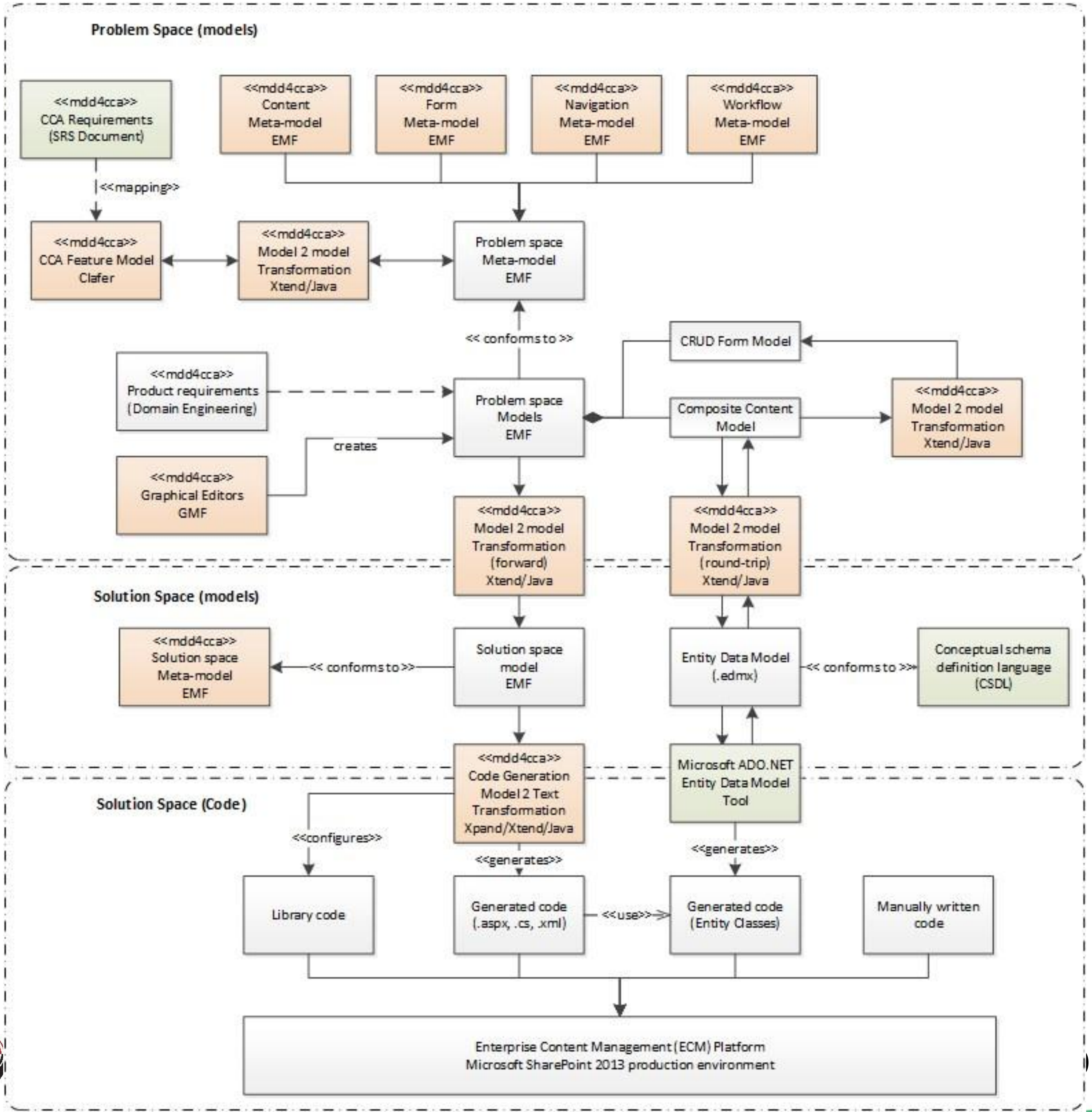
Orders: 2

mdd4cca

State-of-the-Art (SoA) on Eclipse Platform



Name	Version	Id
☑ ATL SDK - ATLAS Transformation Language SDK	3.4.0.v201305211502	org.eclipse.m2m.atl.sdk.feature.group
☑ Eclipse Modeling Tools	2.0.2.20140224-0000	epp.package.modeling
☑ Eclipse XML Editors and Tools	3.5.2.v201401062113-...	org.eclipse.wst.xml_ui.feature.feature.group
☑ EMF - Eclipse Modeling Framework Xcore SDK	1.1.1.v20130903-0948	org.eclipse.emf.ecore.xcore.sdk.feature.group
☑ EMF Compare GMF Integration	2.1.3.201402040808	org.eclipse.emf.compare.diagram.gmf.feature.group
☑ EMF Model Comparison for EUnit (EMF Compare 3.x / Kepler and after)	1.1.0.201309101707	org.eclipse.epsilon.eunit.dt.emf.feature.feature.group
☑ EMF Validation Framework Examples	1.7.0.201306111341	org.eclipse.emf.validation.examples.feature.group
☑ Emfatic (Incubation)	0.8.0.201302100848	org.eclipse.emf.emfatic.feature.group
☑ Epsilon Concordance	1.1.0.201309101707	org.eclipse.epsilon.concordance.feature.feature.group
☑ Epsilon Core	1.1.0.201309101707	org.eclipse.epsilon.core.feature.feature.group
☑ Epsilon Core Development Tools	1.1.0.201309101707	org.eclipse.epsilon.core.dt.feature.feature.group
☑ Epsilon Development Tools for EMF	1.1.0.201309101707	org.eclipse.epsilon.emf.dt.feature.feature.group
☑ Epsilon Development Tools for UML	1.1.0.201309101707	org.eclipse.epsilon.uml.dt.feature.feature.group
☑ Epsilon EMF Integration	1.1.0.201309101707	org.eclipse.epsilon.emf.feature.feature.group
☑ Epsilon UML Integration	1.1.0.201309101707	org.eclipse.epsilon.uml.feature.feature.group
☑ Epsilon Validation Language EMF Integration	1.1.0.201309101707	org.eclipse.epsilon.evl.emf.validation.feature.feature.group
☑ Epsilon Wizard Language EMF Integration	1.1.0.201309101707	org.eclipse.epsilon.ewl.emf.feature.feature.group
☑ Epsilon Wizard Language GMF Integration	1.1.0.201309101707	org.eclipse.epsilon.ewl.gmf.feature.feature.group
☑ Eugenia	1.1.0.201309101707	org.eclipse.epsilon.eugenia.feature.feature.group
☑ Graphical Modeling Framework (GMF) Runtime Examples	1.7.0.201306111432	org.eclipse.gmf.examples.runtime.feature.group
☑ Graphical Modeling Framework (GMF) Tooling	3.1.0.201402192033	org.eclipse.gmf.tooling.feature.group
☑ Graphical Modeling Framework (GMF) Tooling - Runtime Extensions	3.1.0.201402192033	org.eclipse.gmf.tooling.runtime.feature.group
☑ Graphical Modeling Framework (GMF) Tooling SDK	3.1.0.201402192033	org.eclipse.gmf.sdk.feature.group
☑ Human Usable Textual Notation Core	1.1.0.201309101707	org.eclipse.epsilon.hutn.feature.feature.group
☑ Human Usable Textual Notation Development Tools	1.1.0.201309101707	org.eclipse.epsilon.hutn.dt.feature.feature.group
☑ m2e - Maven Integration for Eclipse	1.4.0.20130601-0317	org.eclipse.m2e.feature.feature.group
☑ MWE 2 language SDK	2.4.1.v201309030840	org.eclipse.emf.mwe2.language.sdk.feature.group
☑ MWE 2 runtime SDK	2.4.1.v201309030422	org.eclipse.emf.mwe2.runtime.sdk.feature.group
☑ MWE SDK	1.3.1.v201309030422	org.eclipse.emf.mwe.sdk.feature.group
☑ OCL Examples and Editors	3.3.2.v20140210-1137	org.eclipse.ocl.examples.feature.group
☑ Xpand SDK	1.4.0.v201306110406	org.eclipse.xpand.sdk.feature.group
☑ Xtext SDK	2.4.3.v201309030823	org.eclipse.xtext.sdk.feature.group



Problem Space (models)
<<mdd4cca>> CCA Requirements (SRS Document)
<<mdd4cca>> Content Meta-model EMF
<<mdd4cca>> Form Meta-model EMF
<<mdd4cca>> Navigation Meta-model EMF
<<mdd4cca>> Workflow Meta-model EMF
<<mapping>>
<<mdd4cca>> CCA Feature Model Clafer
<<mdd4cca>> Model 2 model Transformation Xtend/Java
Problem space Meta-model EMF
<< conforms to >>
CRUD Form Model
<<mdd4cca>> Product requirements (Domain Engineering)
Problem space Models EMF
Composite Content Model
<<mdd4cca>> Model 2 model Transformation Xtend/Java
creates
<<mdd4cca>> Graphical Editors GMF
<<mdd4cca>> Model 2 model Transformation (forward) Xtend/Java
<<mdd4cca>> Model 2 model Transformation (round-trip) Xtend/Java
Solution Space (models)
<<mdd4cca>> Solution space Meta-model EMF
<< conforms to >>
Solution space model EMF
Entity Data Model (.edmx)
<< conforms to >>
Conceptual schema definition language (CSDL)
<<mdd4cca>> Code Generation Model 2 Text Transformation Xpand/Xtend/Java
Microsoft ADO.NET Entity Data Model Tool
Solution Space (Code)
<<configures>>
<<generates>>
<<generates>>
Library code
Generated code (.aspx, .cs, .xml)
<<use>>
Generated code (Entity Classes)
Manually written code
Enterprise Content Management (ECM) Platform
Microsoft SharePoint 2013 production environment

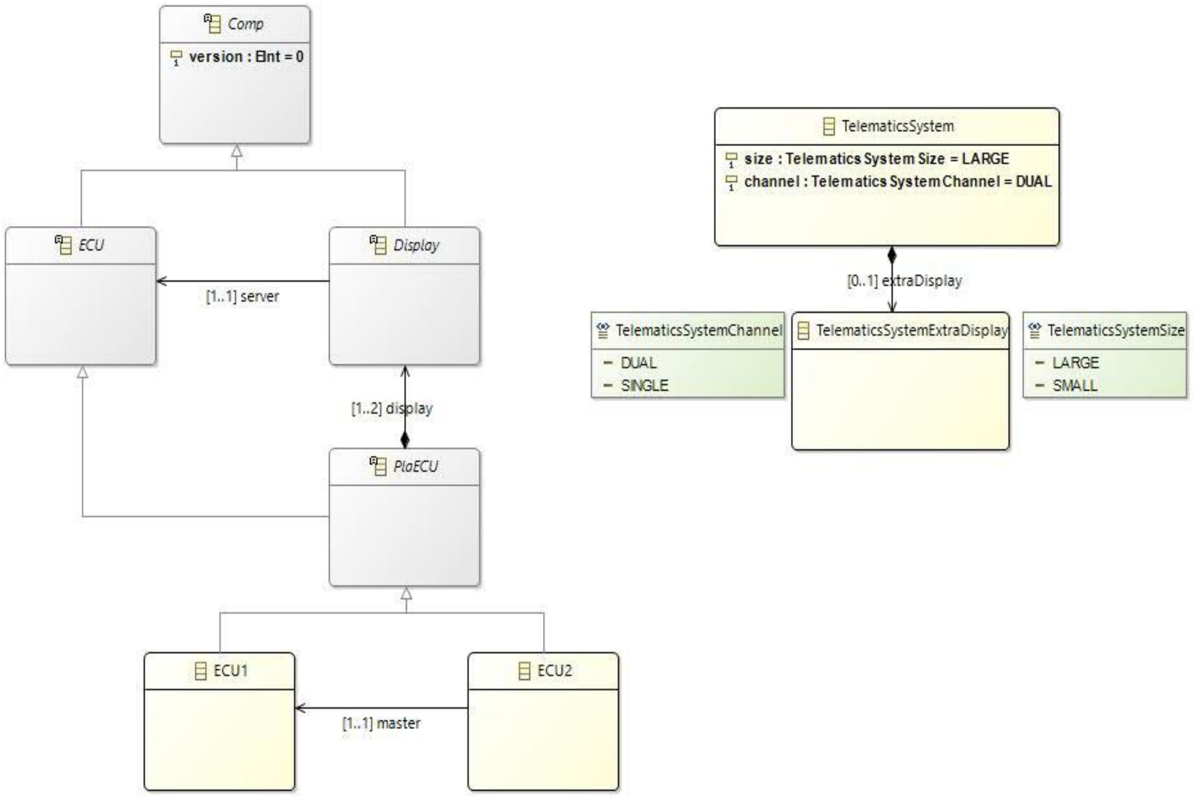
EMF Model Verification Tool
Clafer4EMF – Clafer for Eclipse Modeling Project

UNIT

Sample Clafer Model
Eclipse4EMF tool

UNIT

```
telematicsSystem
    xor channel
        single
        dual

    extraDisplay ?

    xor size
        small
        large

abstract comp
    version : integer = 1 + 2

abstract ECU : comp

abstract display : comp
    server -> ECU
    [this.version >= server.version]


abstract plaECU : ECU
    display : display 1..2
        [server = parent]

ECU1 : plaECU

ECU2 : plaECU ?
    master -> ECU1

 [dual
 extraDisplay
 telematicsSystem.size.large]
```



Clafer: Lightweight Modeling Language
Class, Feature, Reference

- **Domain and structural modeling.**

- A single Clafer model can encode feature, class, and meta-models augmented with complex constraints.

- **Model verification and validation.**

Clafer instance generators (IGs) use the Alloy Analyzer or Choco3 to:

- Check consistency of models.
- Check if given examples are correct instances of models.
- Derive examples from models.

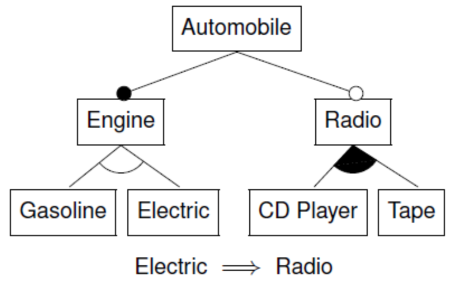
- **Model completion.**

- An IG helps to automatically configure models and specify attribute values to derive fully-specified model instances.
- When configuring a model, the engineer can specify only some properties; the rest will be automatically completed by the reasoner.

Clafer: Lightweight Modeling Language
Class, Feature, Reference

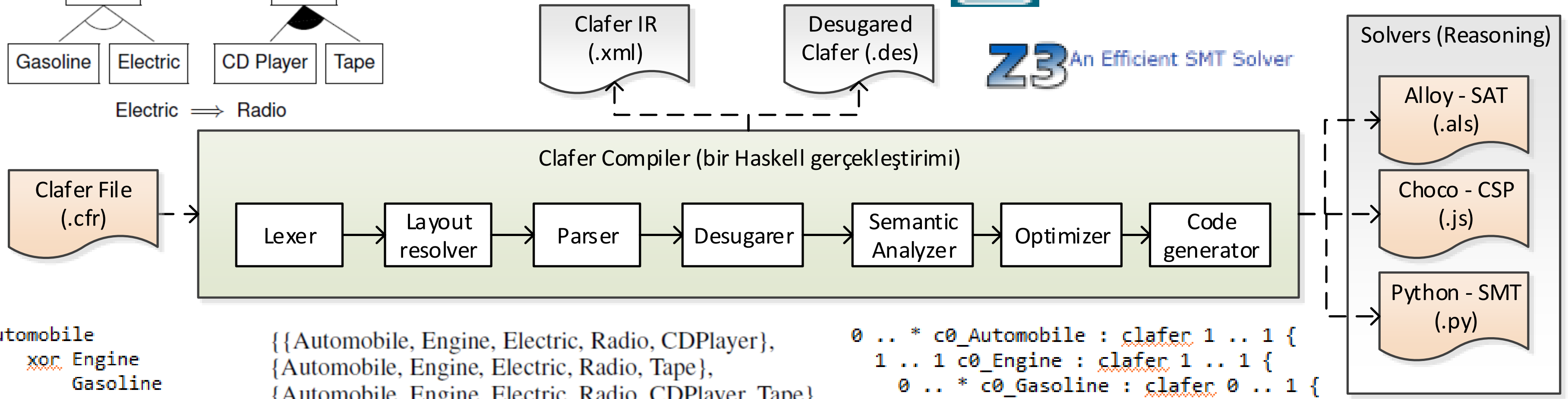
- **Features**
  - Unification of classes, associations, and properties, and arbitrary property nesting.
  - Hierarchical modeling with subclassing.
  - Support for partial instances and (partial) completions.
  - Concise concrete syntax.
  - Model verification and validation

- **Advantages:**
  - Minimalistic modeling language
  - Mixing feature and meta-models
  - Uniform semantics
  - Constraints (First Order Logic - FOL)
  - Reasoning (SAT, CSP, SMT Solver Backends)



Choco3 is an open-source Java library for Constraint Programming.

Z3 An Efficient SMT Solver



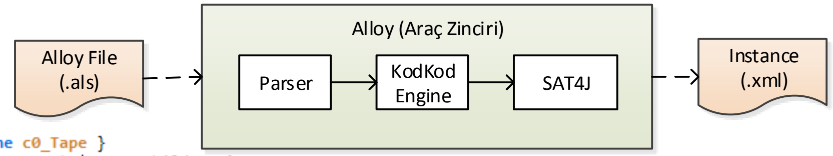
```
Automobile
    xor Engine
        Gasoline
        Electric
    or Radio ?
        CDPlayer
        Tape

[Electric => Radio]
```

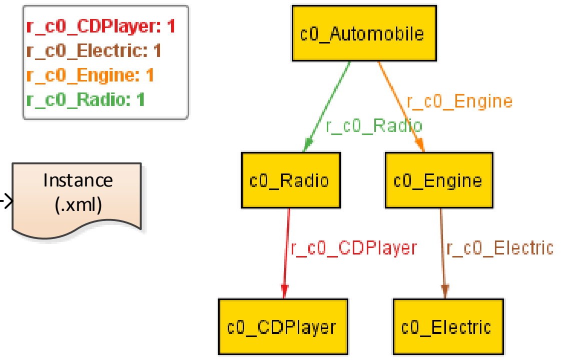
{{Automobile, Engine, Electric, Radio, CDPlayer},
{Automobile, Engine, Electric, Radio, Tape},
{Automobile, Engine, Electric, Radio, CDPlayer, Tape},
{Automobile, Engine, Gasoline},
{Automobile, Engine, Gasoline, Radio, CDPlayer},
{Automobile, Engine, Gasoline, Radio, Tape},
{Automobile, Engine, Gasoline, Radio, CDPlayer, Tape}}

```
0 .. * c0_Automobile : clafer 1 .. 1 {
  1 .. 1 c0_Engine : clafer 1 .. 1 {
    0 .. * c0_Gasoline : clafer 0 .. 1 {
     }
    0 .. * c0_Electric : clafer 0 .. 1 {
     }
   }
  1 .. * c0_Radio : clafer 0 .. 1 {
    0 .. * c0_CDPlayer : clafer 0 .. 1 {
     }
    0 .. * c0_Tape : clafer 0 .. 1 {
     }
   }
```

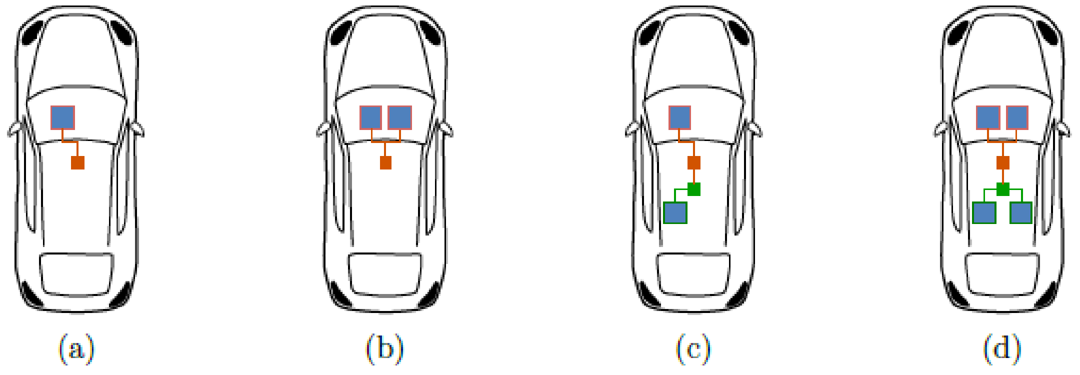
```
pred show {}
run show for 1

one sig c0_Automobile
{ r_c0_Engine : one c0_Engine, r_c0_Radio : lone c0_Radio }

one sig c0_Engine
{ r_c0_Gasoline : lone c0_Gasoline, r_c0_Electric : lone c0_Electric }
{ let children = (r_c0_Gasoline + r_c0_Electric) | one children }

lone sig c0_Gasoline
{} { one r_c0_Gasoline }

lone sig c0_Electric
{} { one r_c0_Electric }

lone sig c0_Radio
{ r_c0_CDPlayer : lone c0_CDPlayer , r_c0_Tape : lone c0_Tape }
{ one r_c0_Radio let children = (r_c0_CDPlayer + r_c0_Tape) | some children }

lone sig c0_CDPlayer
{} { one r_c0_CDPlayer }

lone sig c0_Tape
{} { one r_c0_Tape }

fact { (some (c0_Automobile.@r_c0_Engine).@r_c0_Electric) => (some c0_Automobile.@r_c0_Radio) }
```

KODKOD a constraint solver for relational logic





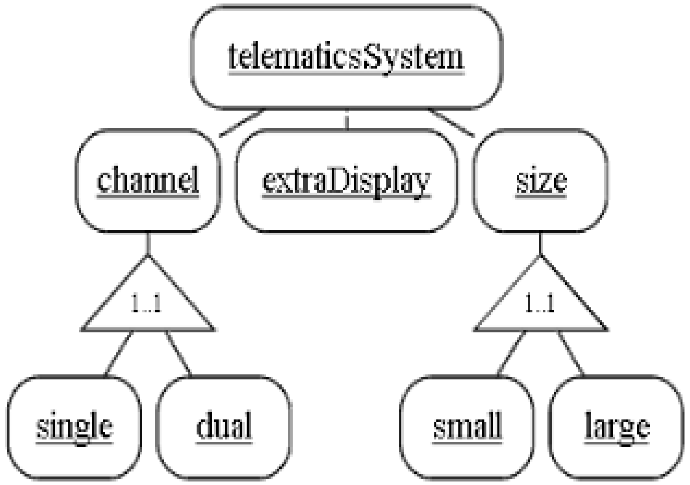
Sample Clafer Model
Simple Telematics System (Feature Model)



Clafer model

```
telematicsSystem
    xor channel
        single
        dual

    extraDisplay ?

    xor size
        small
        large
```

Visualization (CVL notation)



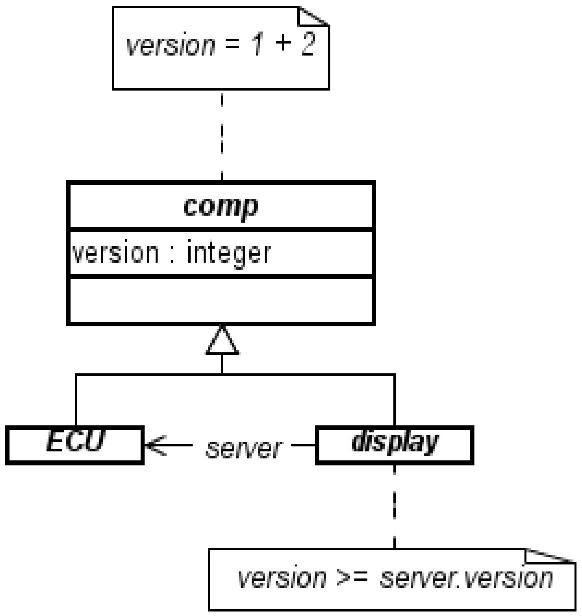
Credit: Clafer.org, used with permission

Sample Clafer Model
Simple Telematics System (Metamodel)

Clafer model

```
abstract comp
    version : integer = 1 + 2

abstract ECU : comp

abstract display : comp
    server -> ECU
    [this.version >= server.version]
```

Visualization (UML notation)



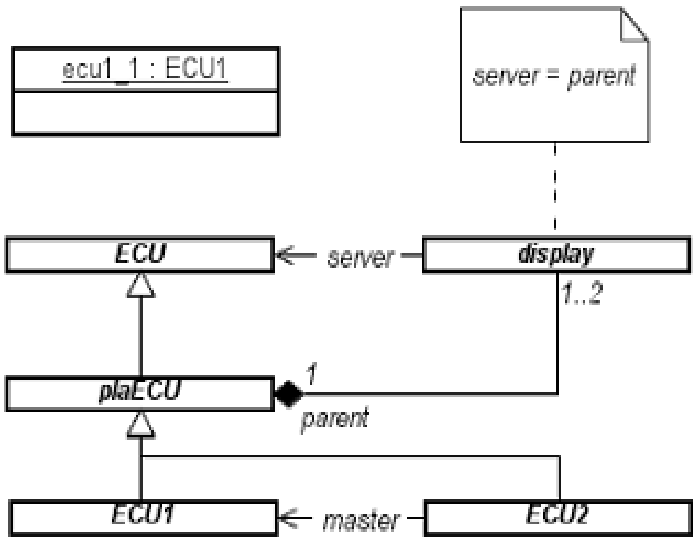
Credit: Clafer.org, used with permission

Sample Clafer Model
Basit bir Telematik Sistemi (Sınıf ve Nesneler)

Clafer model

```
abstract plaECU : ECU
    `display 1..2
        [server = parent]

ECU1 : plaECU

ECU2 : plaECU ?
    master -> ECU1
```

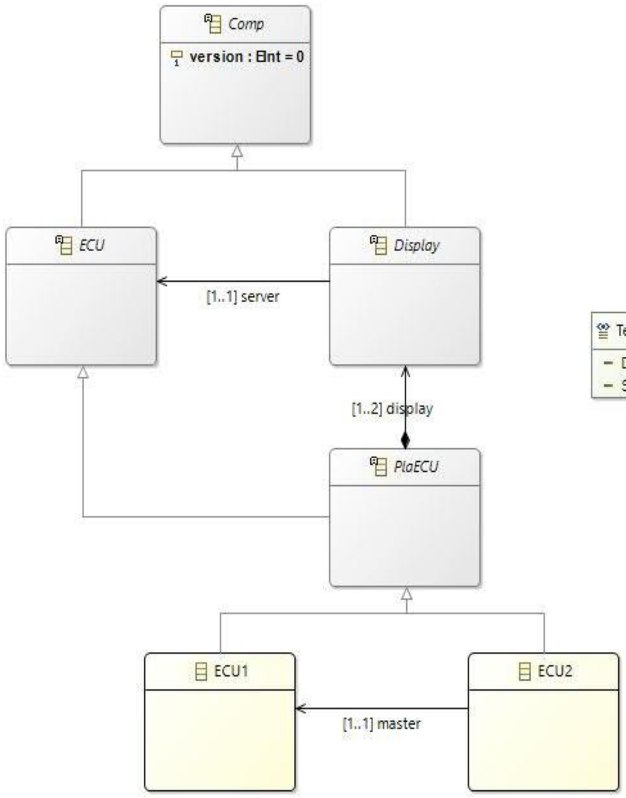
Visualization (UML notation)

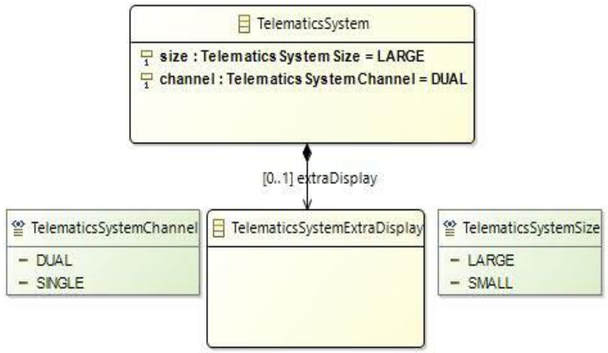


Credit: Clafer.org, used with permission

Sample Clafer Model
Eclipse4EMF tool

```
telematicsSystem
    xor channel
        single
        dual

    extraDisplay ?

    xor size
        small
        large

abstract comp
    version : integer = 1 + 2

abstract ECU : comp

abstract display : comp
    server -> ECU
    [this.version >= server.version]


abstract plaECU : ECU
    display : display 1..2
        [server = parent]

ECU1 : plaECU

ECU2 : plaECU ?
    master -> ECU1

 [dual
 extraDisplay
 telematicsSystem.size.large]
```





TelematicsSystem
size : Telematics System Size = LARGE
channel : Telematics System Channel = DUAL
[0..1] extraDisplay
TelematicsSystemChannel
DUAL
SINGLE
TelematicsSystemExtraDisplay
TelematicsSystemSize
LARGE
SMALL

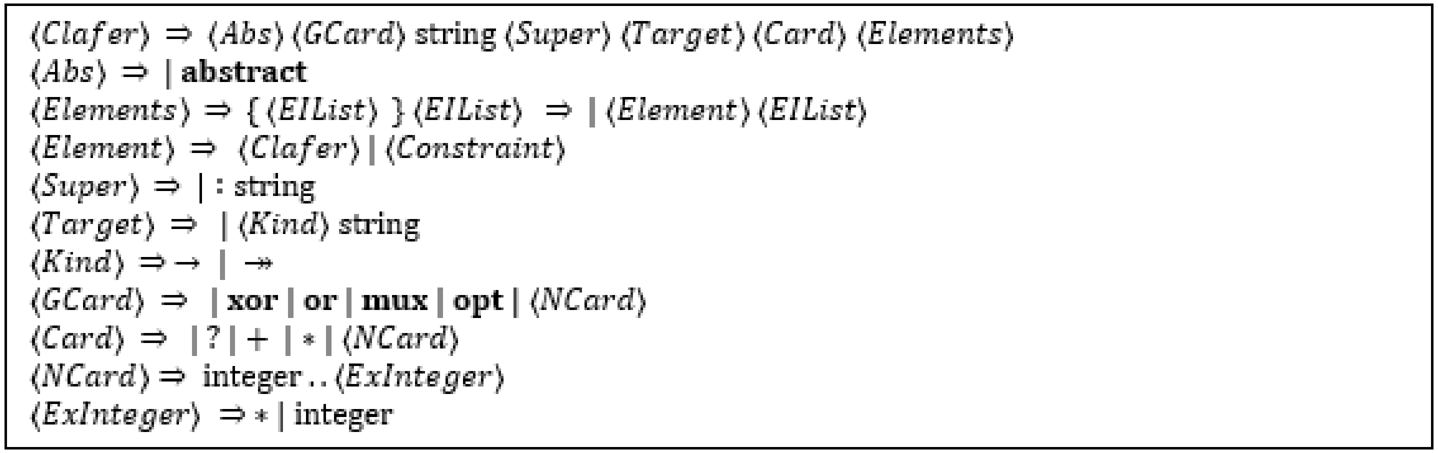
State of the practice: EMF Modeling in Eclipse

Scenario: Develop Clafer Metamodel to be used in model transformation (validated and verified)

UNIT

Validate a Metamodel in EMF

simplified Abstract Syntax (no FOL constructs)



$\langle Clafer \rangle \Rightarrow \langle Abs \rangle\, \langle GCard \rangle\, \text{string}\, \langle Super \rangle\, \langle Target \rangle\, \langle Card \rangle\, \langle Elements \rangle$

$\langle Abs \rangle \Rightarrow \;|\; \textbf{abstract}$

$\langle Elements \rangle \Rightarrow \{\, \langle ElList \rangle \,\}\; \langle ElList \rangle \Rightarrow \;|\; \langle Element \rangle\, \langle ElList \rangle$

$\langle Element \rangle \Rightarrow \langle Clafer \rangle \;|\; \langle Constraint \rangle$

$\langle Super \rangle \Rightarrow \;|\; :\, \text{string}$

$\langle Target \rangle \Rightarrow \;|\; \langle Kind \rangle\, \text{string}$

$\langle Kind \rangle \Rightarrow \rightarrow \;|\; \twoheadrightarrow$

$\langle GCard \rangle \Rightarrow \;|\; \textbf{xor} \;|\; \textbf{or} \;|\; \textbf{mux} \;|\; \textbf{opt} \;|\; \langle NCard \rangle$

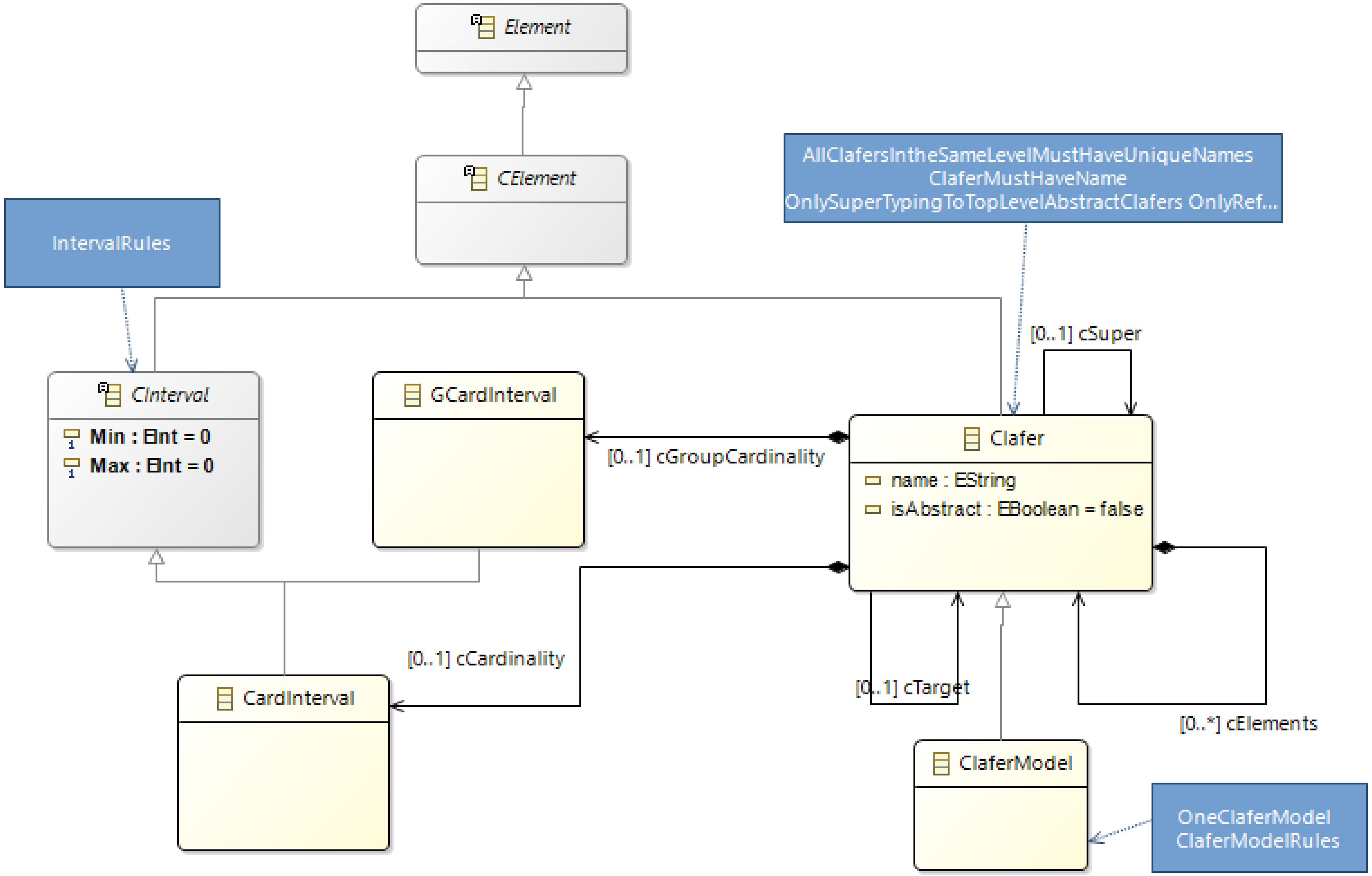
$\langle Card \rangle \Rightarrow \;|\; ? \;|\; + \;|\; * \;|\; \langle NCard \rangle$

$\langle NCard \rangle \Rightarrow \text{integer} \,..\, \langle ExInteger \rangle$

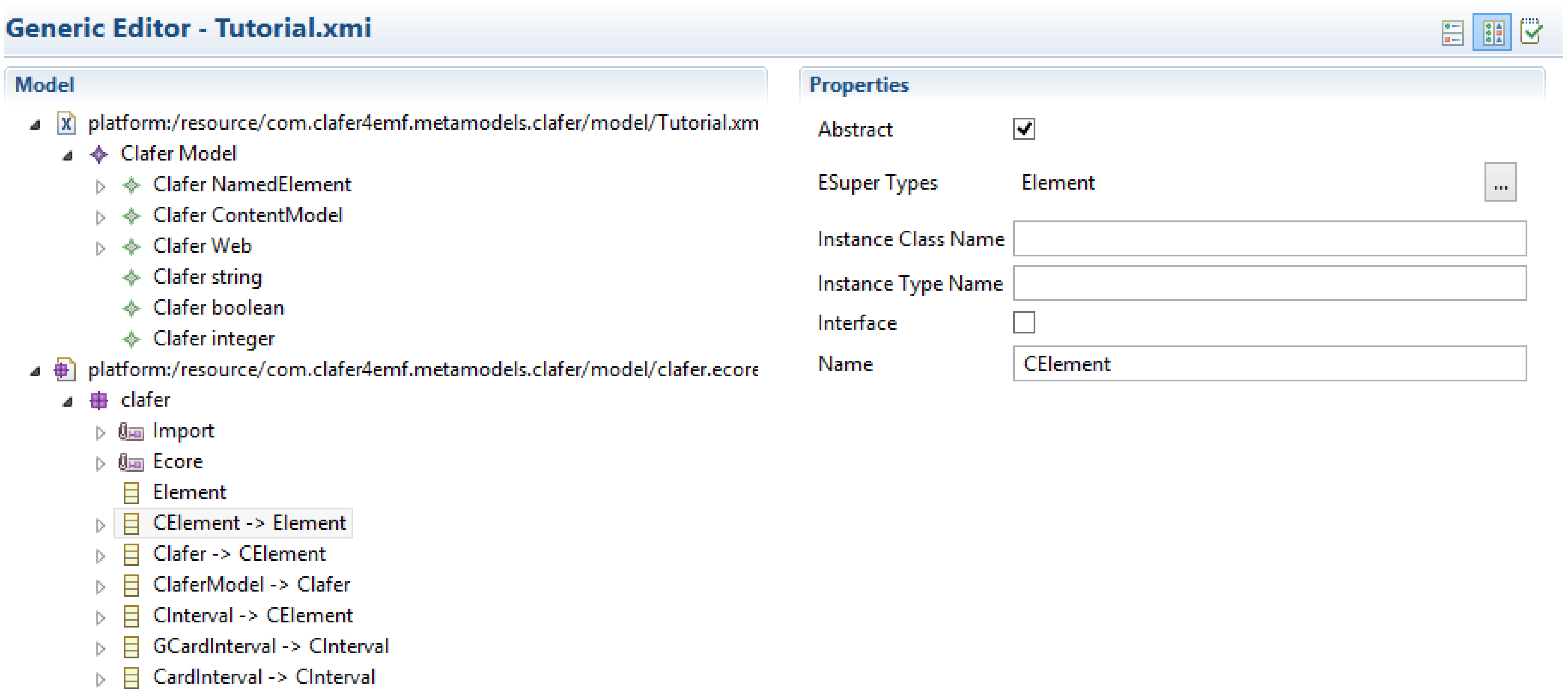
$\langle ExInteger \rangle \Rightarrow * \;|\; \text{integer}$

Model Verification in Eclipse
SotA I (Sirius Editor - Luna)

UNIT



Model Verification in Eclipse
SotA II (Generic EMF Form Editor – Eclipse Luna)



Generic Editor - Tutorial.xmi

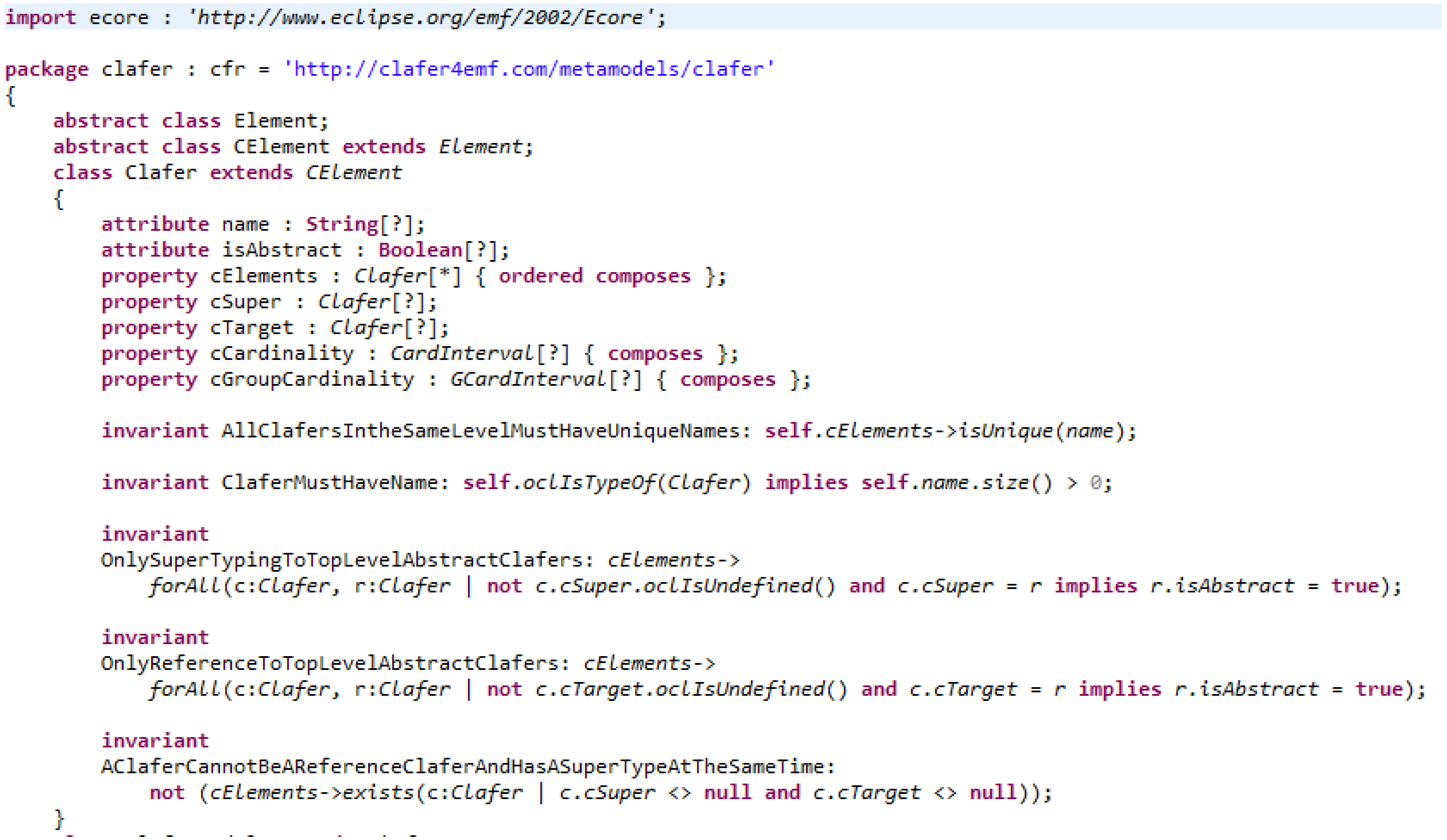
Model

- platform:/resource/com.clafer4emf.metamodels.clafer/model/Tutorial.xm
  - Clafer Model
    - Clafer NamedElement
    - Clafer ContentModel
    - Clafer Web
    - Clafer string
    - Clafer boolean
    - Clafer integer
- platform:/resource/com.clafer4emf.metamodels.clafer/model/clafer.ecore
  - clafer
    - Import
    - Ecore
    - Element
    - CElement -> Element
    - Clafer -> CElement
    - ClaferModel -> Clafer
    - CInterval -> CElement
    - GCardInterval -> CInterval
    - CardInterval -> CInterval

Properties

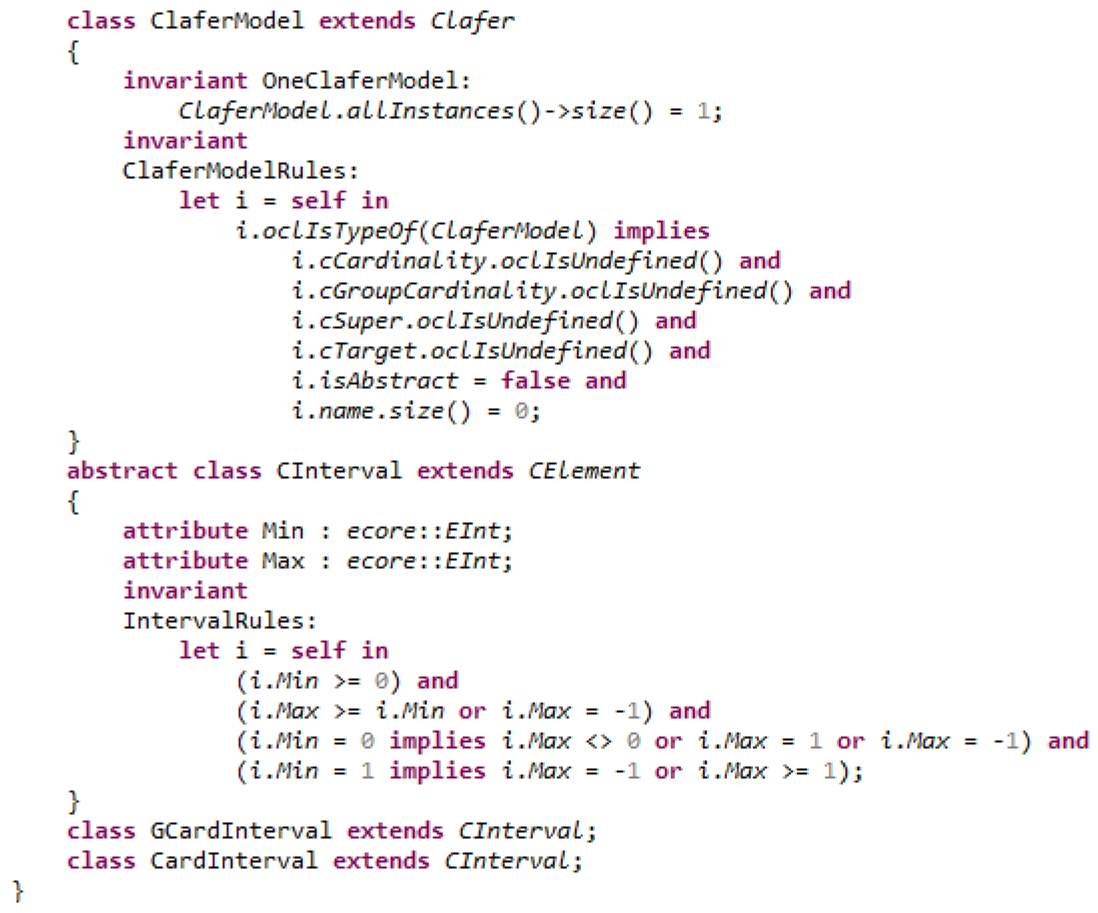
Abstract	✓
ESuper Types	Element
Instance Class Name	
Instance Type Name	
Interface	☐
Name	CElement

Model Verification in Eclipse
SotA III (OCLInEcore Part I)

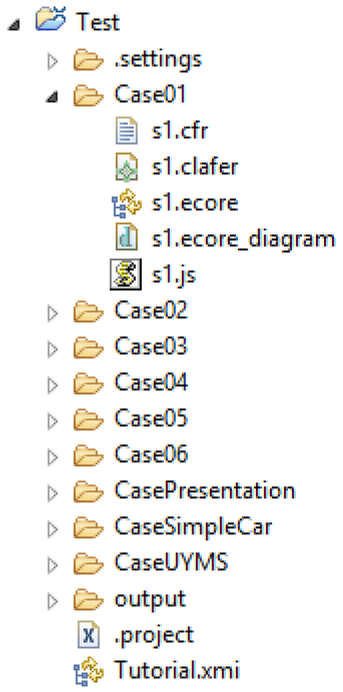


```
import ecore : 'http://www.eclipse.org/emf/2002/Ecore';

package clafer : cfr = 'http://clafer4emf.com/metamodels/clafer'
{
    abstract class Element;
    abstract class CElement extends Element;
    class Clafer extends CElement
    {
        attribute name : String[?];
        attribute isAbstract : Boolean[?];
        property cElements : Clafer[*] { ordered composes };
        property cSuper : Clafer[?];
        property cTarget : Clafer[?];
        property cCardinality : CardInterval[?] { composes };
        property cGroupCardinality : GCardInterval[?] { composes };

        invariant AllClafersIntheSameLevelMustHaveUniqueNames: self.cElements->isUnique(name);

        invariant ClaferMustHaveName: self.oclIsTypeOf(Clafer) implies self.name.size() > 0;

        invariant
        OnlySuperTypingToTopLevelAbstractClafers: cElements->
            forAll(c:Clafer, r:Clafer | not c.cSuper.oclIsUndefined() and c.cSuper = r implies r.isAbstract = true);

        invariant
        OnlyReferenceToTopLevelAbstractClafers: cElements->
            forAll(c:Clafer, r:Clafer | not c.cTarget.oclIsUndefined() and c.cTarget = r implies r.isAbstract = true);

        invariant
        AClaferCannotBeAReferenceClaferAndHasASuperTypeAtTheSameTime:
            not (cElements->exists(c:Clafer | c.cSuper <> null and c.cTarget <> null));
    }
```

Model Verification in Eclipse
SotA III (OCLInEcore Part II)



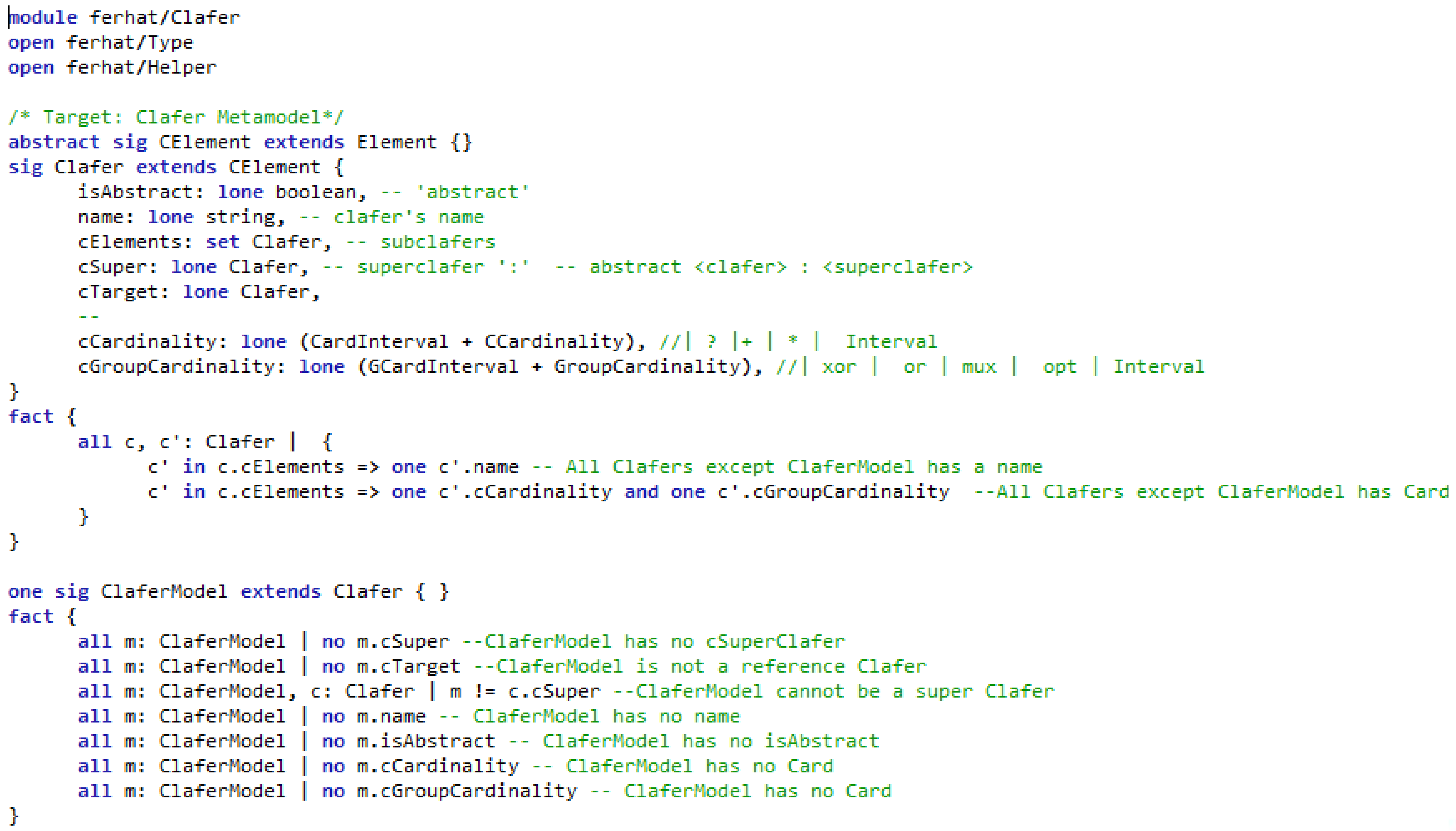
```
    class ClaferModel extends Clafer
    {
        invariant OneClaferModel:
            ClaferModel.allInstances()->size() = 1;
        invariant
        ClaferModelRules:
            let i = self in
                i.oclIsTypeOf(ClaferModel) implies
                    i.cCardinality.oclIsUndefined() and
                    i.cGroupCardinality.oclIsUndefined() and
                    i.cSuper.oclIsUndefined() and
                    i.cTarget.oclIsUndefined() and
                    i.isAbstract = false and
                    i.name.size() = 0;
    }
    abstract class CInterval extends CElement
    {
        attribute Min : ecore::EInt;
        attribute Max : ecore::EInt;
        invariant
        IntervalRules:
            let i = self in
                (i.Min >= 0) and
                (i.Max >= i.Min or i.Max = -1) and
                (i.Min = 0 implies i.Max <> 0 or i.Max = 1 or i.Max = -1) and
                (i.Min = 1 implies i.Max = -1 or i.Max >= 1);
    }
    class GCardInterval extends CInterval;
    class CardInterval extends CInterval;
}
```



```
◢ Test
    ▷ .settings
    ◢ Case01
          s1.cfr
          s1.clafer
          s1.ecore
          s1.ecore_diagram
          s1.js
    ▷ Case02
    ▷ Case03
    ▷ Case04
    ▷ Case05
    ▷ Case06
    ▷ CasePresentation
    ▷ CaseSimpleCar
    ▷ CaseUYMS
    ▷ output
      .project
      Tutorial.xmi
```

Model Verification in Alloy (by MIT)

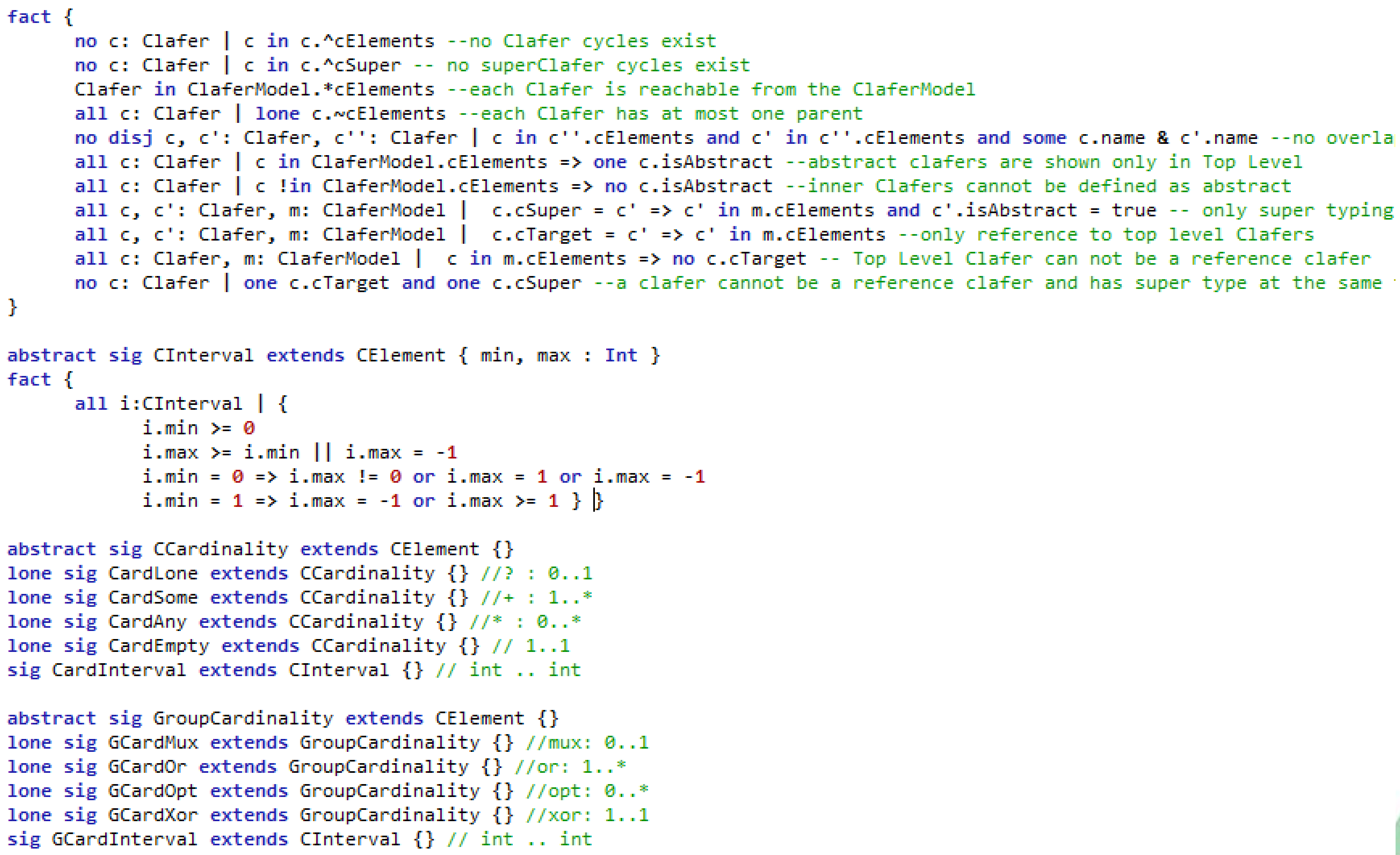
SotA IV (Alloy Part I – separate editor)



```
module ferhat/Clafer
open ferhat/Type
open ferhat/Helper

/* Target: Clafer Metamodel*/
abstract sig CElement extends Element {}
sig Clafer extends CElement {
	isAbstract: lone boolean, -- 'abstract'
	name: lone string, -- clafer's name
	cElements: set Clafer, -- subclafers
	cSuper: lone Clafer, -- superclafer ':'  -- abstract <clafer> : <superclafer>
	cTarget: lone Clafer,
	--
	cCardinality: lone (CardInterval + CCardinality), //| ? |+ | * |  Interval
	cGroupCardinality: lone (GCardInterval + GroupCardinality), //| xor |  or | mux |  opt | Interval
}
fact {
	all c, c': Clafer |  {
		c' in c.cElements => one c'.name -- All Clafers except ClaferModel has a name
		c' in c.cElements => one c'.cCardinality and one c'.cGroupCardinality  --All Clafers except ClaferModel has Card
	}
}

one sig ClaferModel extends Clafer { }
fact {
	all m: ClaferModel | no m.cSuper --ClaferModel has no cSuperClafer
	all m: ClaferModel | no m.cTarget --ClaferModel is not a reference Clafer
	all m: ClaferModel, c: Clafer | m != c.cSuper --ClaferModel cannot be a super Clafer
	all m: ClaferModel | no m.name -- ClaferModel has no name
	all m: ClaferModel | no m.isAbstract -- ClaferModel has no isAbstract
	all m: ClaferModel | no m.cCardinality -- ClaferModel has no Card
	all m: ClaferModel | no m.cGroupCardinality -- ClaferModel has no Card
}
```

Model Verification in Alloy (by MIT)

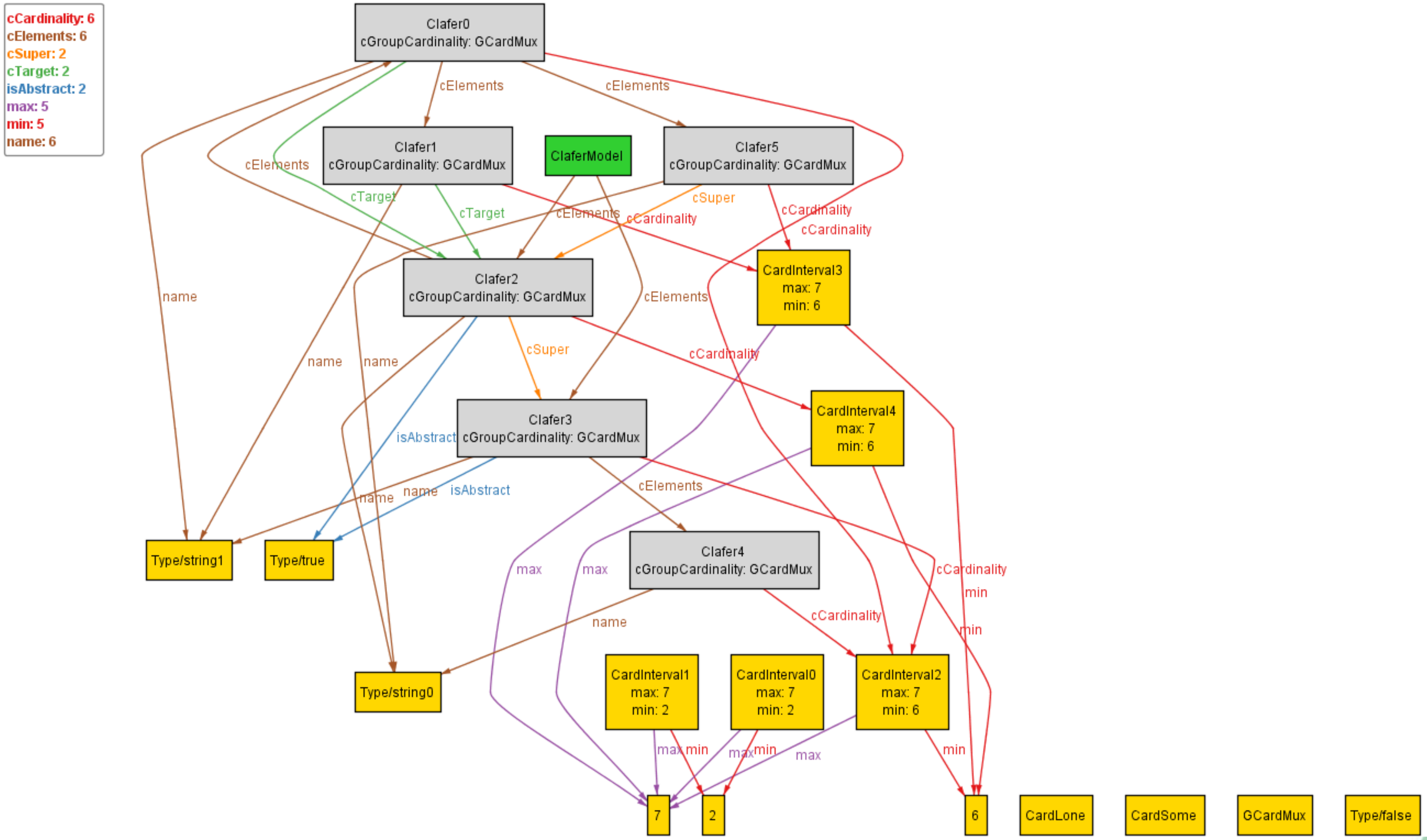
SotA IV (Alloy Part II)



```
fact {
      no c: Clafer | c in c.^cElements --no Clafer cycles exist
      no c: Clafer | c in c.^cSuper -- no superClafer cycles exist
      Clafer in ClaferModel.*cElements --each Clafer is reachable from the ClaferModel
      all c: Clafer | lone c.~cElements --each Clafer has at most one parent
      no disj c, c': Clafer, c'': Clafer | c in c''.cElements and c' in c''.cElements and some c.name & c'.name --no overla
      all c: Clafer | c in ClaferModel.cElements => one c.isAbstract --abstract clafers are shown only in Top Level
      all c: Clafer | c !in ClaferModel.cElements => no c.isAbstract --inner Clafers cannot be defined as abstract
      all c, c': Clafer, m: ClaferModel |  c.cSuper = c' => c' in m.cElements and c'.isAbstract = true -- only super typing
      all c, c': Clafer, m: ClaferModel |  c.cTarget = c' => c' in m.cElements --only reference to top level Clafers
      all c: Clafer, m: ClaferModel |  c in m.cElements => no c.cTarget -- Top Level Clafer can not be a reference clafer
      no c: Clafer | one c.cTarget and one c.cSuper --a clafer cannot be a reference clafer and has super type at the same
}

abstract sig CInterval extends CElement { min, max : Int }
fact {
      all i:CInterval | {
            i.min >= 0
            i.max >= i.min || i.max = -1
            i.min = 0 => i.max != 0 or i.max = 1 or i.max = -1
            i.min = 1 => i.max = -1 or i.max >= 1 } }

abstract sig CCardinality extends CElement {}
lone sig CardLone extends CCardinality {} //? : 0..1
lone sig CardSome extends CCardinality {} //+ : 1..*
lone sig CardAny extends CCardinality {} //* : 0..*
lone sig CardEmpty extends CCardinality {} // 1..1
sig CardInterval extends CInterval {} // int .. int

abstract sig GroupCardinality extends CElement {}
lone sig GCardMux extends GroupCardinality {} //mux: 0..1
lone sig GCardOr extends GroupCardinality {} //or: 1..*
lone sig GCardOpt extends GroupCardinality {} //opt: 0..*
lone sig GCardXor extends GroupCardinality {} //xor: 1..1
sig GCardInterval extends CInterval {} // int .. int
```

Model Verification in Alloy

SotA IV (Alloy Part III)



cCardinality: 6
cElements: 6
cSuper: 2
cTarget: 2
isAbstract: 2
max: 5
min: 5
name: 6
Clafer0 cGroupCardinality: GCardMux
Clafer1 cGroupCardinality: GCardMux
ClaferModel
Clafer5 cGroupCardinality: GCardMux
Clafer2 cGroupCardinality: GCardMux
Clafer3 cGroupCardinality: GCardMux
Clafer4 cGroupCardinality: GCardMux
CardInterval3 max: 7 min: 6
CardInterval4 max: 7 min: 6
CardInterval1 max: 7 min: 2
CardInterval0 max: 7 min: 2
CardInterval2 max: 7 min: 6
Type/string1
Type/true
Type/string0
7
2
6
CardLone
CardSome
GCardMux
Type/false
cElements
cTarget
cSuper
cCardinality
name
isAbstract
max
min

Thank You!
Questions/Comments?

Model Everything! ***Prof. Hans Vangheluwe***

UNIT

Contact

- Ferhat ERATA
  - Email: ferhat@computer.org
  - Tel: +90-539-5661271
  - LinkedIn: http://www.linkedin.com/in/ferhaterata
  - Mdd4cca - www.mdd4cca.com
  - ModelWriter – www.modelwriter.eu