TECHNISCHE UNIVERSITÄT DRESDEN

Department of Computer Science Institute for Software and Multimedia Technology, Software Technology Group

Extending FRaMED
Kickoff

Komplex Praktikum *(SS2014)*

Thomas Kühn

Outline

Looking Back

Task

Graphical Notation

Ecore Model

Next Steps

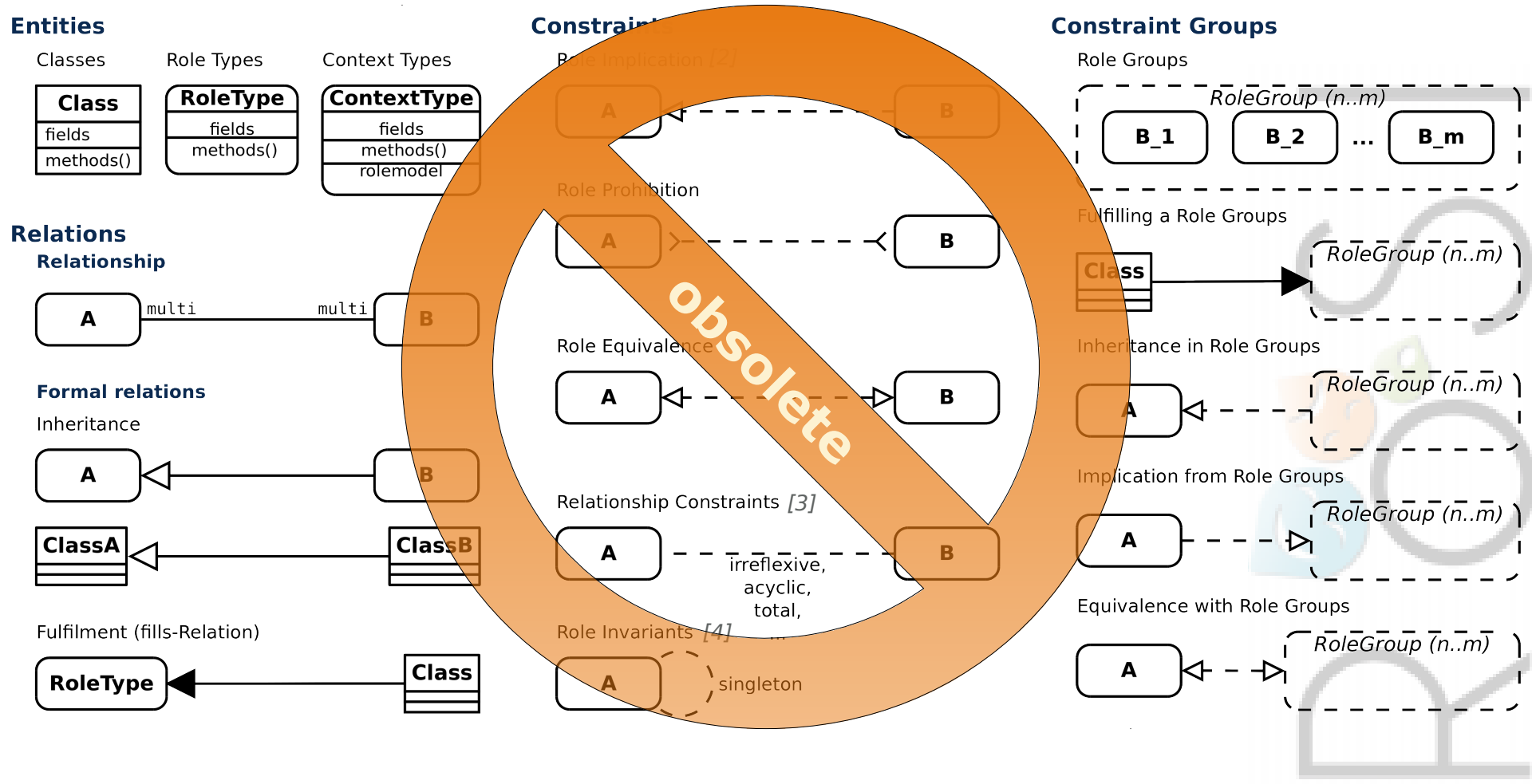
First Role Modelling Editor (FraMED)

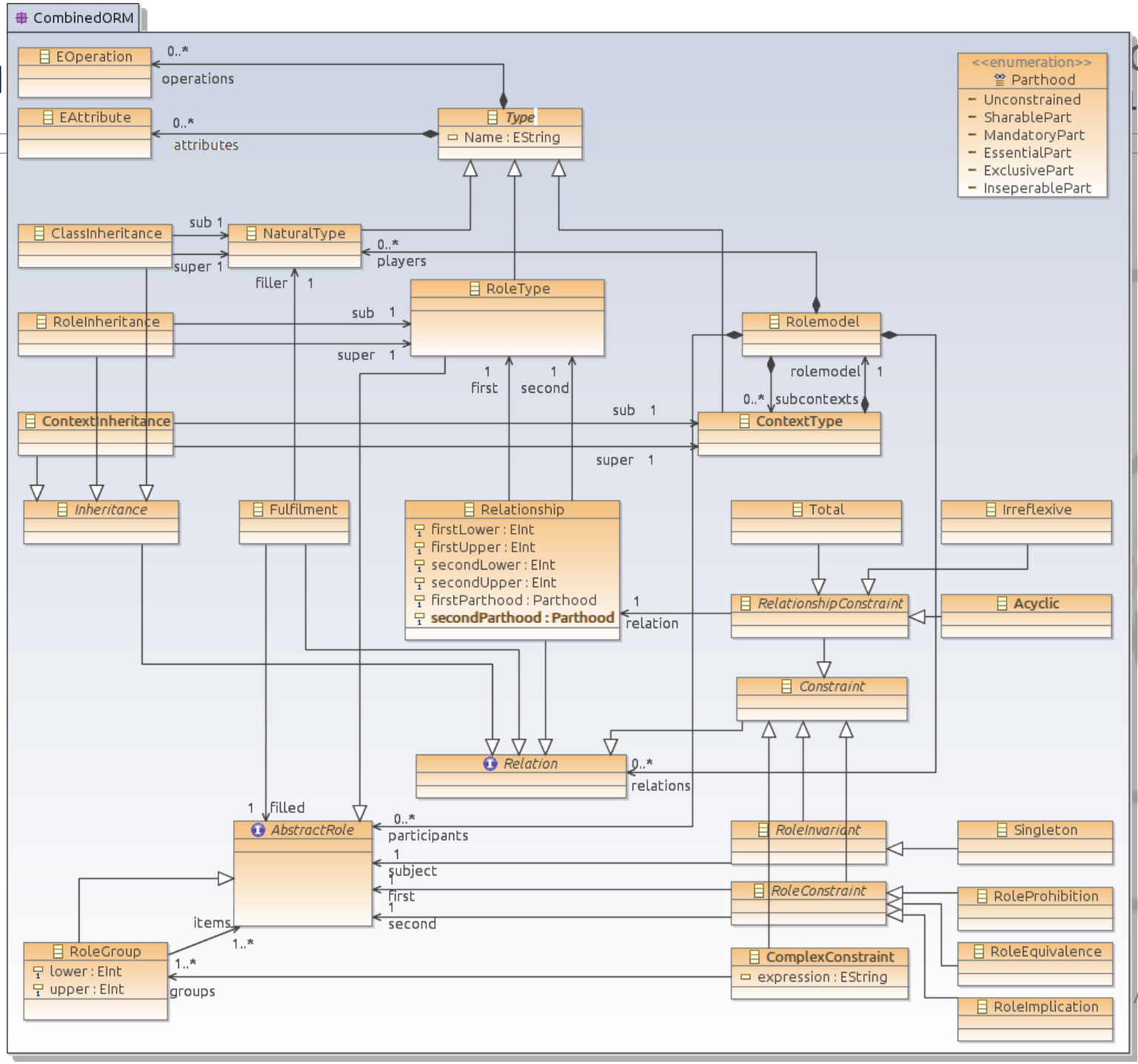
- Development of an Eclipse-based diagram editor
  - Graphical specification of *Object Role Models*
  - Generate coherent Models
  - Enable the usage of common concepts and notations
- ~~Validate the consistancy of *Object Role Models*~~

Prerequisites

- Knowledge about Eclipse plugin development, meta modelling with Ecore, and GEF diagram framework
- Basic understanding of the concept of roles and compartments
- An established Ecore Metamodel

Looking Back
Syntax of Object Role Model[1]




CombinedORM
EOperation
0..*
operations
EAttribute
0..*
attributes
Type
Name : EString
<<enumeration>>
Parthood
Unconstrained
SharablePart
MandatoryPart
EssentialPart
ExclusivePart
InseperablePart
ClassInheritance
sub 1
super 1
NaturalType
0..*
players
filler 1
RoleType
RoleInheritance
sub 1
super 1
Rolemodel
rolemodel 1
0..*
subcontexts
1
first
1
second
ContextInheritance
sub 1
super 1
ContextType
Inheritance
Fulfilment
Relationship
firstLower : EInt
firstUpper : EInt
secondLower : EInt
secondUpper : EInt
firstParthood : Parthood
secondParthood : Parthood
1
relation
Total
Irreflexive
RelationshipConstraint
Acyclic
Constraint
Relation
0..*
relations
1
filled
AbstractRole
0..*
participants
1
subject
1
first
1
second
RoleInvariant
Singleton
RoleConstraint
RoleProhibition
RoleEquivalence
RoleImplication
ComplexConstraint
expression : EString
items
1..*
RoleGroup
lower : EInt
upper : EInt
1..*
groups

Demo Failed

Extending First Role Modelling Editor (FRaMED)

- Stabilize the current release
  - Refactor and Cleanup current code base
  - Document the current release
  - Find and fix bugs
- Extension of FRaMED towards a new release
  - Incorperate the new *CROM Meta Model*(Version: 2.0)
  - Validate the consistancy of *CROM Models*
- Establish a new release of FRaMED
  - Including UnitTests, Documentation, Tutorials, Examples

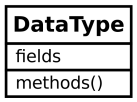
Prerequisites

- Knowledge about Eclipse plugin development, meta modelling with Ecore, and GEF diagram framework
- A new Ecore Metamodel

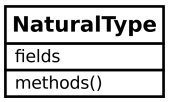
Graphical Notation
Syntax of CROM

Entities

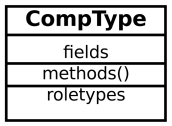
Data Types



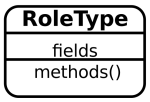
Natural Types



Compartment Types

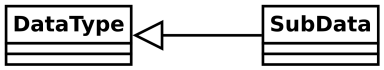


Role Types

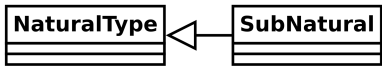


Static Relations

Data Type Inheritance

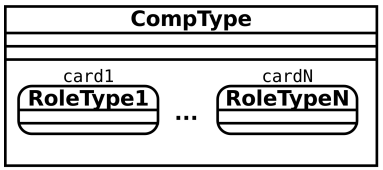


Natural Type Inheritance

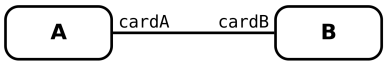


Dynamic Relations

Participation (participates-Relation)



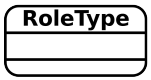
Binary Relationship



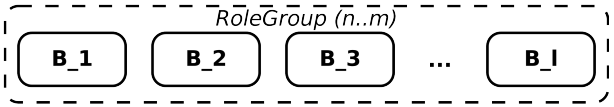
Fulfilment (fills-Relation)

Constraint Atomes

Role Types



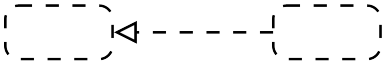
Role Groups



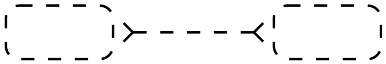
Note: Role Groups may contain other Role Groups with Role Types. However, all the relationships between them have to be retained!

Constraints

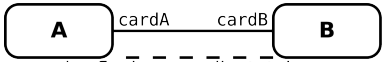
Role Implication



Role Prohibition

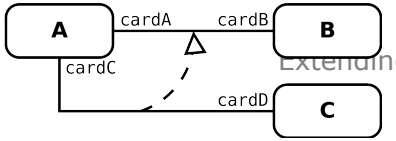


Relationship Constraints



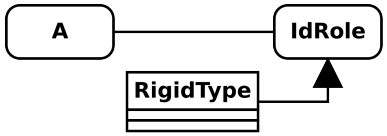
irreflexive, acyclic, total, ...

Relationship Implication

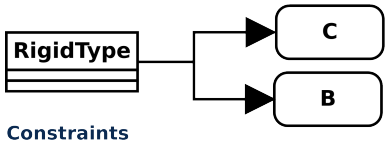


Dynamic Relations

Classes with Relationships

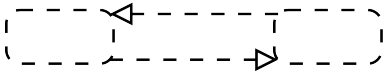


Fullfillment (fills-Relation)



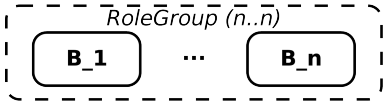
Constraints

Role Equivalence

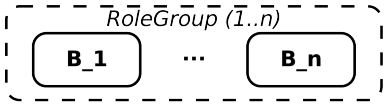


Constraint Groups

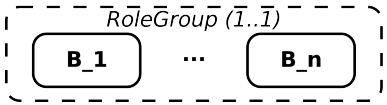
Grouping Roles with AND



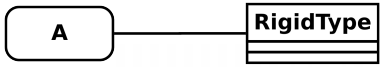
Grouping Roles with OR



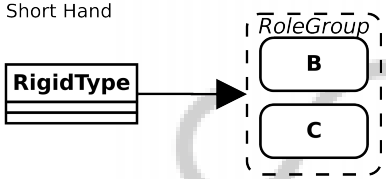
Grouping Roles with XOR



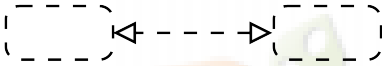
Short Hand (with Identity Role)



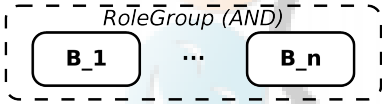
Short Hand



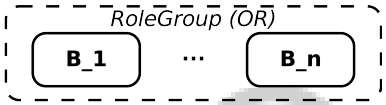
Short Hand



Short Hand

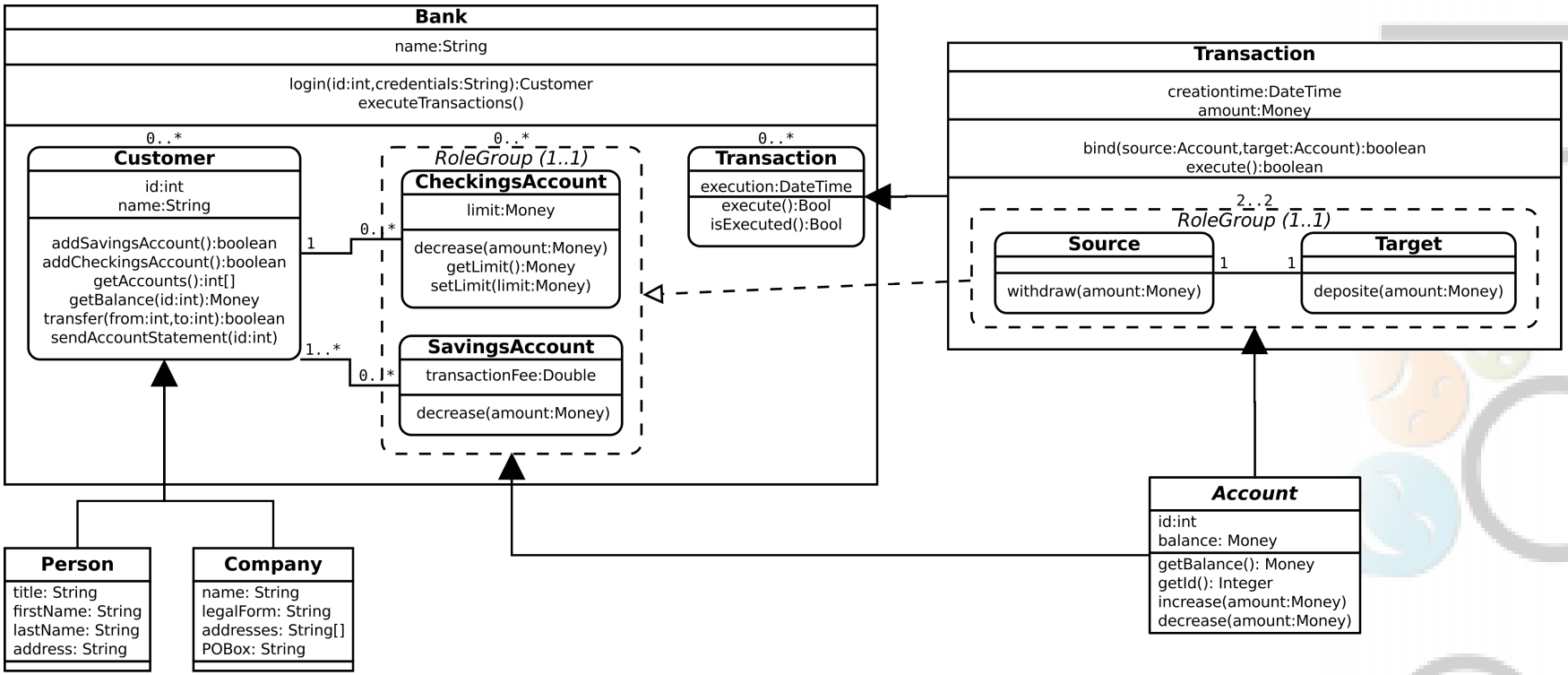


Short Hand

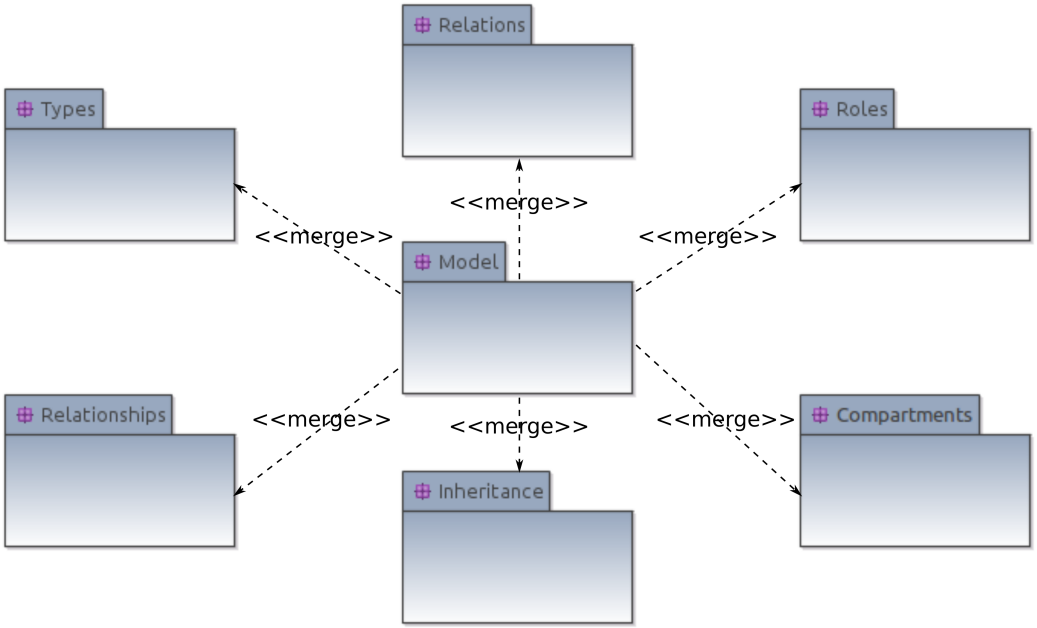


Short Hand

Graphical Notation
Usage Scenario



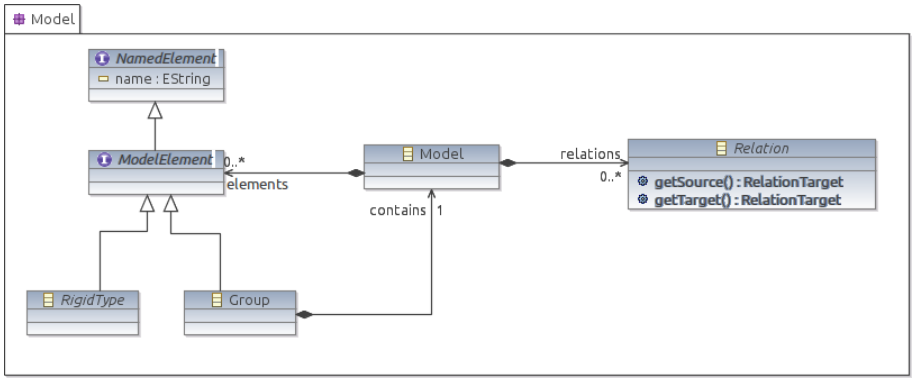
Ecore Model

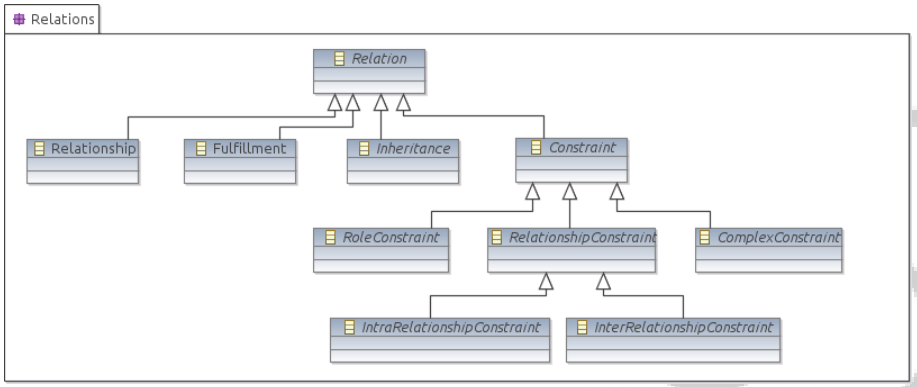


Aspects of the meta model

- Types
- Relations
- Kinds of Inheritance
- Relationships
- Concept of Roles
- Concept of Compartments

Ecore Model
Model and Relations





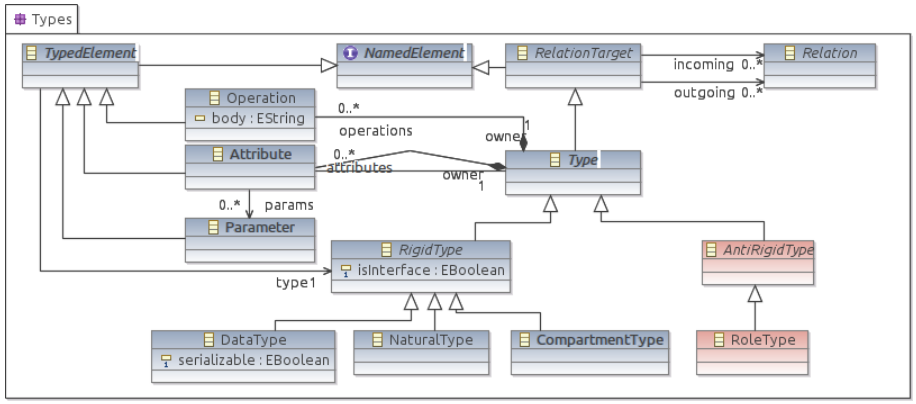
Model

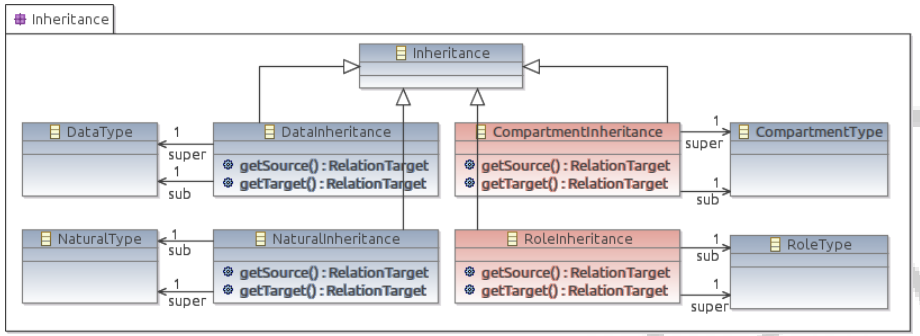
- Base of each CROM Model
- Contains *ModelElements* and *Relations*
- *ModelElements* are either *RigidTypes* or *Groups*

Relations

- Discerns all the different kinds of relations
- Contains formal, material, and constraint relations

Ecore Model
Types and Inheritance





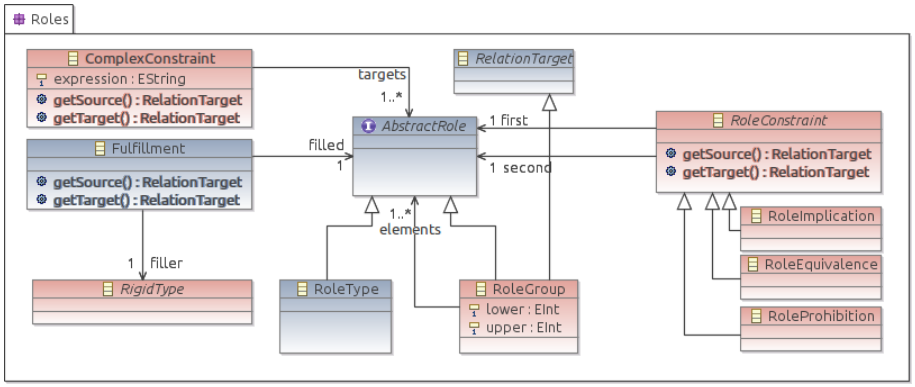
Types

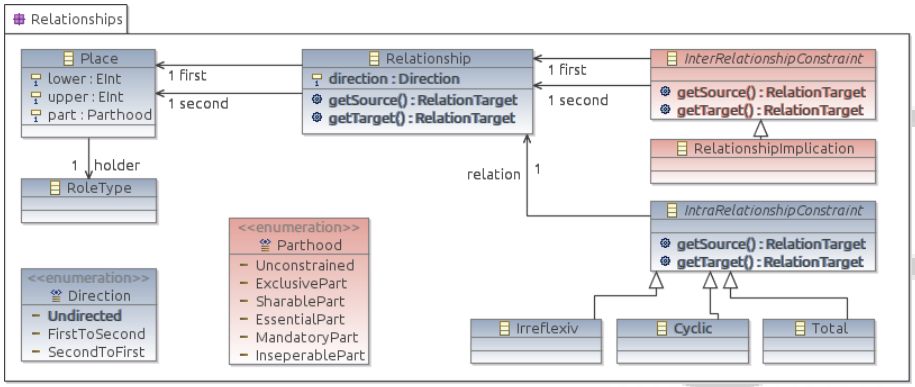
- Defines the fundamental Types
- Types have a name, attributes, and operations

Inheritance

- Defines the various inheritance relations for each concept
- Inheritance is prohibited between different concepts

Ecore Model
Roles and Relationships





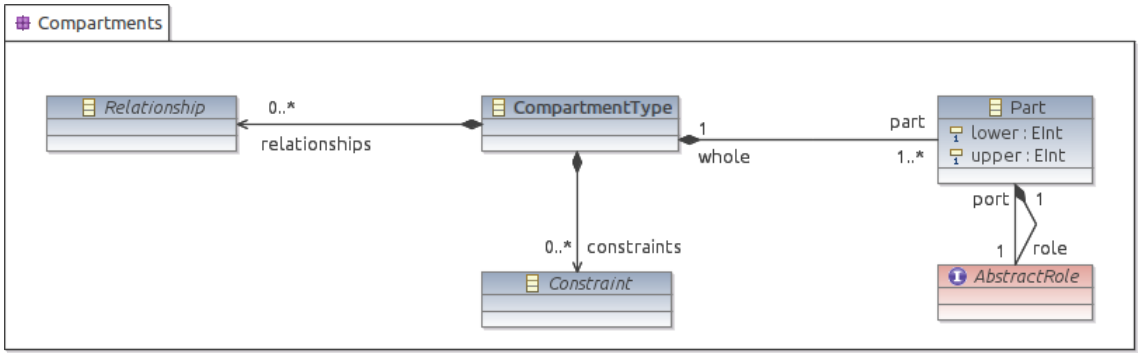
Roles

- Fulfilment relation specifies which *RigidType* can play (fill) an *AbstractRole*
- *AbstractRoles* can be further constrained
- An *AbstractRole* is either a RoleGroup or a RoleType
- *RoleGroups* contain several (at least on) Abstract Role

Relationships

- Relationships are defined between two *RoleTypes (via Place)*
- Relationships have multiplicities, parthood and relational constraints

Ecore Model
Compartments

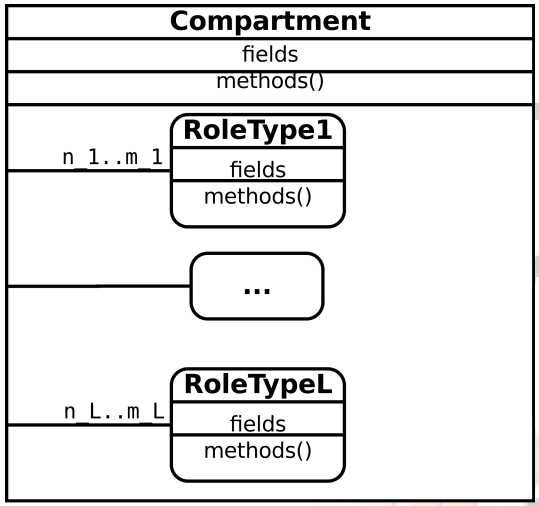


Compartments

- Each Compartments contains
  - at least one *AbstractRoles* (via Part),
  - Various Relationships and Constraints,
  - *But no Fullfillment relation.*
- Each *Part* of the *Compartment* carries a cardinality limiting the number of roles within this *AbstractRole*

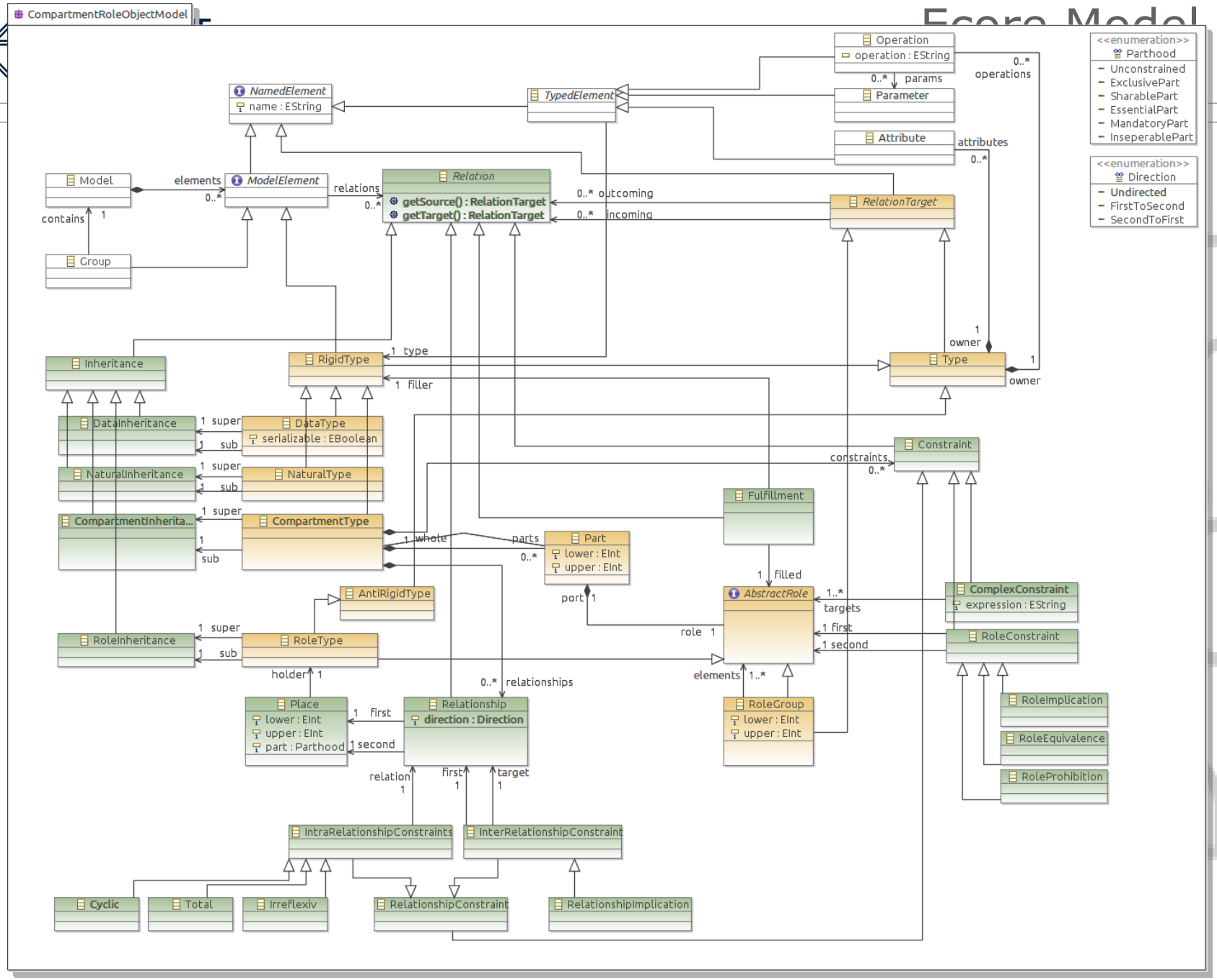
Ecore Model
Formal Definition of Compartments

Let
- *N*... set of all NaturalTypes
- *C*... set of all CompartmentTypes

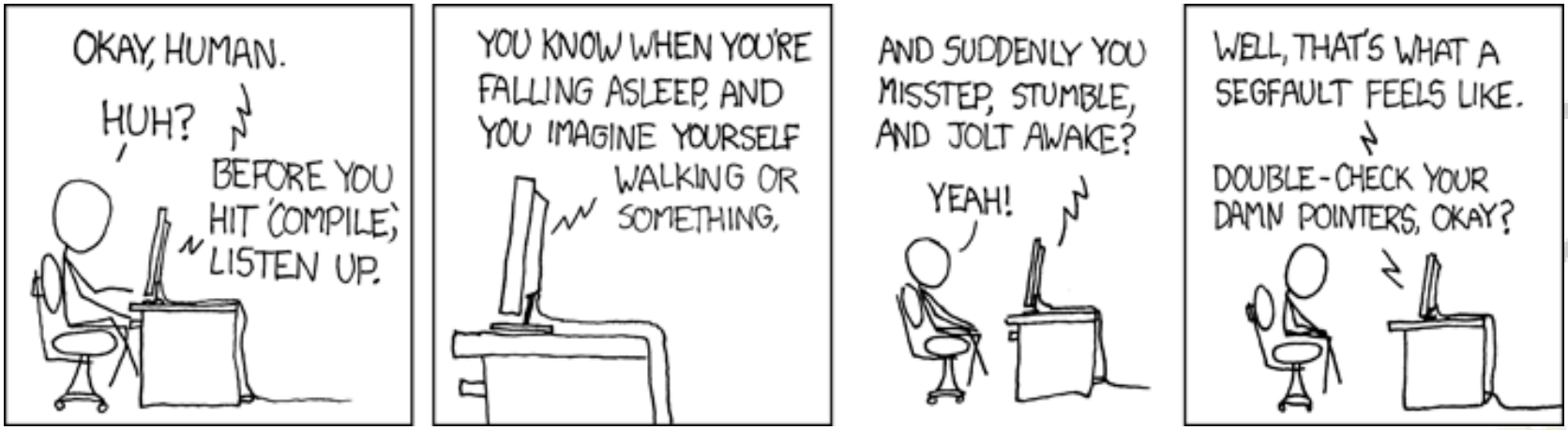


CompartmentType *C=(F,M,R,fills,Rel,card)* with:
- *F*... set of fields
- *M*... set of methods
- *R*... set of contained RoleTypes
- Fills: $R \rightarrow N \cup C$
- $Rel \subseteq R \times R$... set of Relationships
- *card:* $(R \rightarrow N \times N) \cup (Rel \rightarrow N \times N \times N \times N)$
  - *card(r)=(l,u)* with $r \in R$ and *l* and *u* denotes the lower bound and upper bound
  - *card(rel)=(l1,u1,l2,u2)* with $rel \in Rel$ and *l1,u1,l2,u2* cardinalities of the relationship

Ecore Model

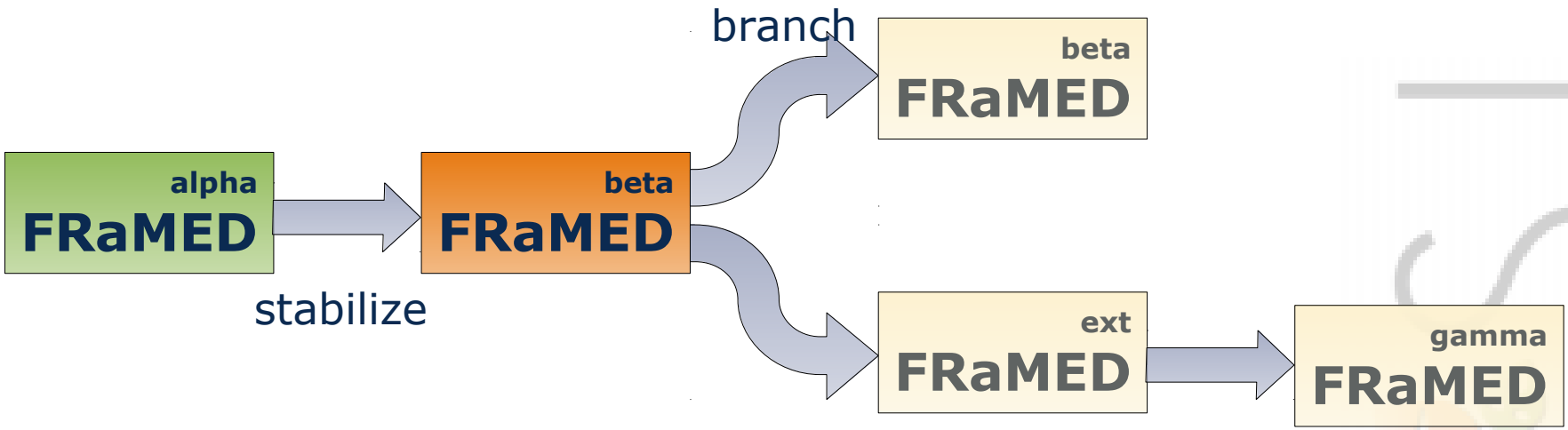

CompartmentRoleObjectModel
Operation
operation : EString
0..* operations
0..* params
Parameter
Attribute
attributes
0..*
NamedElement
name : EString
TypedElement
<<enumeration>>
Parthood
Unconstrained
ExclusivePart
SharablePart
EssentialPart
MandatoryPart
InseperablePart
<<enumeration>>
Direction
Undirected
FirstToSecond
SecondToFirst
Model
elements
0..*
ModelElement
relations
0..*
Relation
getSource() : RelationTarget
getTarget() : RelationTarget
0..* outcoming
0..* incoming
RelationTarget
contains
1
Group
1 owner
Inheritance
RigidType
1 type
1 filler
Type
1 owner
DataInheritance
1 super
1 sub
DataType
serializable : EBoolean
Constraint
constraints
0..*
NaturalInheritance
NaturalType
CompartmentInherita...
CompartmentType
1 whole
parts
0..*
Part
lower : EInt
upper : EInt
Fulfillment
1 filled
AntiRigidType
port 1
AbstractRole
1..* targets
ComplexConstraint
expression : EString
RoleInheritance
RoleType
role 1
1 first
1 second
RoleConstraint
holder 1
0..* relationships
elements 1..*
Place
lower : EInt
upper : EInt
part : Parthood
1 first
1 second
Relationship
direction : Direction
RoleGroup
lower : EInt
upper : EInt
RoleImplication
RoleEquivalence
RoleProhibition
relation 1
first 1
target 1
IntraRelationshipConstraints
InterRelationshipConstraint
Cyclic
Total
Irreflexiv
RelationshipConstraint
RelationshipImplication

Next Steps



- Download FRaMED from github.com
  https://github.com/leondart/FRaMED
- Get used to work with GEF / Ecore Development Tools
- Investigate the structure and behavior of FRaMED
- Find bugs and issue them
- Fix bugs and refactor the code base
- Document each class you find and understood

Next Steps
Project Plan



Milestone 1: 15th May 2014

- Stable FRaMED^{beta} Editor
- Refectored, Documented and Tested
- Runnable Version on Github and at least two examples

References

[1] **Explizite Rollenbindung mit Story-Boards**
T. Kühn
Großer Beleg, TU-Dresden (2011)

[2] **Role model based framework design and integration**
D. Riehle and T. Gross
In Proceedings OOPSLA '98, ACM SIGPLAN Notices (1998)

[3] **A relational model of object collaborations and its use in reasoning about relationships**
S. Balzer, T. Gross, and P. Eugster
ECOOP, vol. 4609 of Lecture Notes in Computer Science (2007)

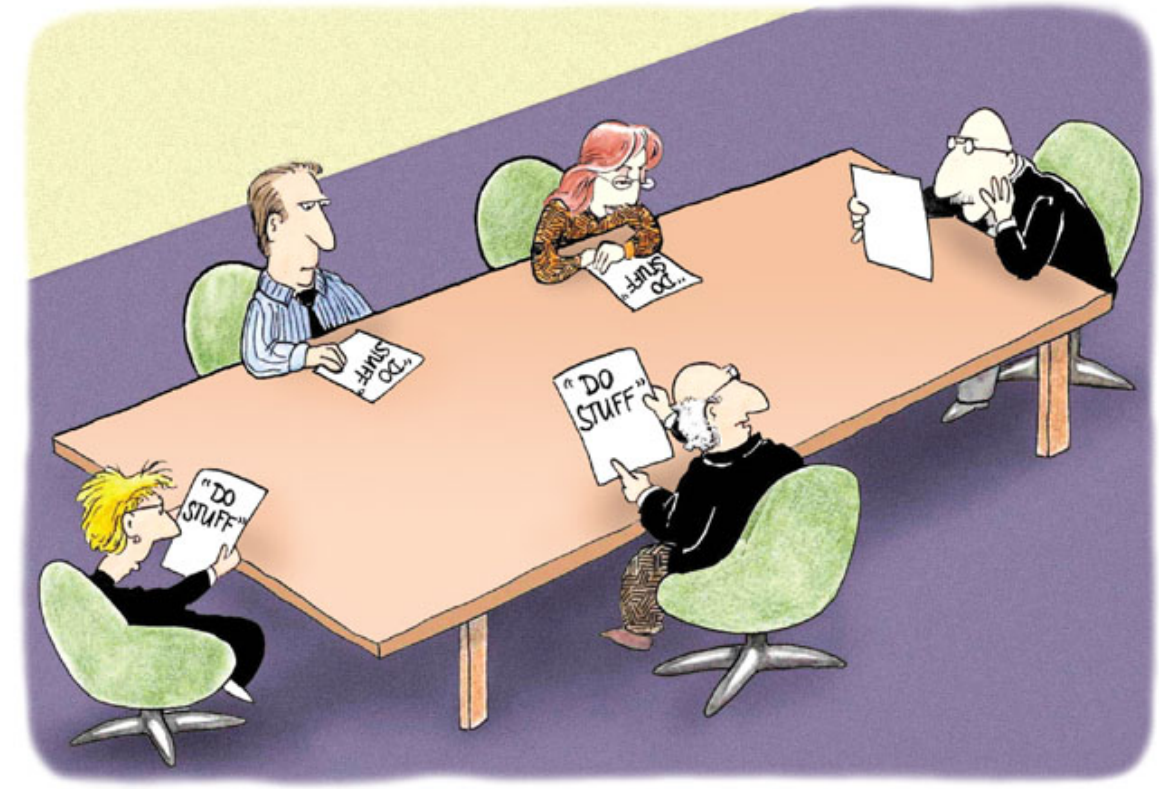
[4] **ORM 2**
T. A. Halpin
OTM Workshops, vol. 3762 of Lecture Notes in Computer Science (2005)

TECHNISCHE UNIVERSITÄT DRESDEN

Extending FRaMED
Kickoff

ROSI

Department of Computer Science Institute for Software and Multimedia Technology, Software Technology Group



"I'm beginning to wonder if our Terms of Reference may be a little too broad..?"
Cartoon by Nick D Kim strange-matter.net. Used by permission.

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur