31.7. More Details of the CROM Metamodel (from GRK Role-based Software Infrastructures RoSI)

Prof. Dr. rer. nat. habil. 
Uwe Aßmann 
Dipl. Inf. Thomas Kühn 
Technische Universität Dresden 
Institut für Software- und Multimediatechnik 
http://st.inf.tu-dresden.de 
/teaching/most 
Version 15-1.4, 16.01.16
References

[1] The Role Concept in Data Models
Charles W. Bachman and Manilal Daya
VLDB (1977)

Friedrich Steimann

[3] Role model based framework design and integration
D. Riehle and T. Gross

[4] A relational model of object collaborations and its use in reasoning about relationships
S. Balzer, T. Gross, and P. Eugster

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

[6] Rava: Designing a Java extension with dynamic object roles
Chengwan He, et al.

Stephan Herrmann
Applied Ontology 2.2 (2007)

[8] powerJava: ontologically founded roles in object oriented programming languages
Matteo Baldoni, Guido Boella, and Leendert Van Der Torre

[9] Towards safe and flexible object adaptation
Tetsuo Kamina and Tetsuo Tamai
Literature

  - http://www.easst.org/eceasst/
The RoSI Research Training Group

RoSI

Concept of Roles as General Modeling Paradigm

- Meta Modeling
- Object Modeling
- Language Modeling
- Data Modeling
- Application Modeling
- Schema Design
- System Modeling

TB1
Roles in Conceptual and Language Modeling

TB2
Roles in Software Engineering

TB3
Roles at Runtime

Runtime Support

Influence

Derivation

Foundation
Data Modeling, Logics, Programming Systems, Software Engineering
The RoSI Project Areas of Research

- Business Informatics
- Software Engineering of Ubiquitous Systems
- Role-based Design
- Database
- Automata Theory
- Algebraic and Logical Foundations
- Computer Networks
- Software Technology

Model-Driven Software Development in Technical Spaces (MOST)
Motivation

„All the world’s a stage, and all the men and women merely players: they have their exits and their entrances; and one man in his time plays many parts, his acts being seven ages.“

– William Shakespeare

The Role Concept

► Relatively old, e.g. Bachman 1977 [1]
► Since then many different approaches emerged
► They do not share a common understanding (or formalism)
► There might be no common universal role concept

Each approach can be classified along design decisions
Classification
Asking 15 Questions

Classification of Roles
(1) Roles have properties and behaviours
(2) Roles depend on relationships
(3) An Object may play different roles simultaneously
(4) An Object may play the same role (type) several times
(5) An Object may acquire and abandon roles dynamically
(6) Sequence of role acquisition and removal may be restricted
(7) Unrelated objects can play the same role
(8) Roles can play roles
(9) Roles can be transferred between objects
(10) The State of an object can be role-specific
(11) Features of an object can be role-specific
(12) Roles restrict access
(13) Different roles may share structure and behaviour
(14) An Object and its roles share identity
(15) An Object and its roles have different identities

Evaluation of Current Approaches

1. Lodwick [1]
2. Rava [6]
4. powerJava [8]
5. NextEJ [9]

<table>
<thead>
<tr>
<th>Approach</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>3.</td>
<td>+</td>
<td>?</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>/</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>?</td>
</tr>
<tr>
<td>4.</td>
<td>+</td>
<td>?</td>
<td>+</td>
<td>+</td>
<td>?</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>5.</td>
<td>+</td>
<td>?</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>?</td>
</tr>
</tbody>
</table>

+ yes  - no  ? possible / not applicable
Classification
Asking 15 Questions

Evaluation of a Small Poll within the Research Group (n=7)
Classification
Asking 15 Questions

Results
- Concensus in several questions (1-7)
- Controversial parts start from question (8)
- Maybe the following questions will become Variation Points

Debated Questions
(8) Roles can play roles
(10) The State of an object can be role-specific
(11) Features of an object can be role-specific
(13) Different roles may share structure and behaviour
(14) An Object and its roles share identity
(15) An Object and its roles have different identities
Classification
10 Additional Questions

Additional Questions retrieved from the Literature [3-9]

(16) Relationships between Roles can be constrained
(17) There may be constraints between relationships
(18) Roles can be grouped and constrained together
(19) Roles depend on Compartments
(20) Compartments have properties and behaviors
(21) A Role can be part of several Compartments
(22) Compartments may play roles like objects
(23) Compartments may play roles which are part of themselves
(24) Compartments can contain other compartments
(25) Different compartments may share structure and behavior
(26) Compartments have their own identity

– Thomas Kühn (2014)
Classification

Conclusion

„[...] there is no single definition of roles integrating all of [the classifying Questions]“


A Metamodel for RoSI must

- Provide a common ground for role-based modeling
- Identify the constituents of the role concepts
- Capture the structure of the various role concepts
- Reflect the design decisions w.r.t. to the 26 questions
- Be a family of similar Metamodels
A Metamodel for RoSI

- Graphical Notation
- Textual Syntax

Compartment Role Object Metamodel

- Conceptual Model
- Database Schema
- Formal Model

Models based on a common Metamodel
A Metamodel for RoSI

Graphical Syntax for RoSI

Entities

Data Types
- fields
- methods()

Natural Types
- fields
- methods()

Compartments Types
- fields
- methods()

Role Types
- roles

Static Relations

Data Type Inheritance

Dynamic Relations

Participation (participates-Relation)

Binary Relationship

Constraint Atomes

Role Types
- RoleGroup (n.m)
- RoleGroup (n..m)

Role Groups
- B_1
- B_2
- B_3
- ...
- B_n

Dynamic Relations

Classes with Relationships

IdRole

Fullfillment (fills-Relation)

Constraints

Role Equivalence

Constraint Groups

Grouping Roles with AND

Grouping Roles with OR

Grouping Roles with XOR

Relationship Constraints

[1] Irreflexive, acyclic, total, ...

Relationship Implication

[3] Role Implication

[4] Role Prohibition

[5] Role Implication
A Metamodel for RoSI

Usage Scenario
A Metamodel for RoSI
EMOF/Ecore Model

Aspects of the meta model

- Types
- Relations
- Relation
- Concept of Roles
- Kinds of Inheritance
- Concept of Compartments
A Metamodel for RoSI
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
A Metamodel for RoSI
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
A Metamodel for RoSI
Roles and Relationships

Roles
- Fulfillment 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 one) AbstractRole

Relationships
- Relationships are defined between two RoleTypes (via Place)
- Relationships have multiplicities, parthood and relational constraints
A Metamodel for RoSI

Compartments

- Each Compartments contains
  - at least one AbstractRoles (via Part),
  - Various Relationships and Constraints,
  - *But no Fulfillment relation.*

- Each *Part* of the *Compartment* carries a cardinality limiting the number of roles within this *AbstractRole*
A Metamodel for RoSI
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: $R \to N \cup C$
- $Rel \subseteq R \times R$... set of Relationships
- $card: (R \to N \times N) \cup (Rel \to N \times N \times N \times N)$
  - $card(r)=(l,u)$ 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
The Metamodel Family

- The CROM Metamodel contains various Variation Points (VP)
- Each VP corresponds to one of the 26 classifying questions
- A Variant of the CROM Metamodel can be derived from answering these questions
- It becomes possible to define a common metamodel family for the various role-based modeling approaches
The End

► Thanks to Thomas Kühn for CROM and the slides
► Why is it hard to unify the role concept?
► Why are compartments necessary to group roles in metamodels?