41. Role-Based (Meta-)Modeling

in the Research Training School on Role-oriented Software Infrastructures (RoSI)

1. A Primer on Roles
2. Role-based Modeling and Programming Languages
3. The Compartment Role Object Model (CROM)
Challenges of Software Systems

Complexity

Roles increase separation of concerns

Change

Roles allow for dynamic changes of the system

Longevity

Roles enable updating running applications

1) https://bian.org/assets/bian-standards/bian-service-landscape-3-0/

Katrina Tuliao (CC-SA 2.0) http://www.flickr.com/photos/thewalkingirony/3051500551/
The RoSI Research Training Group
Software Development for continuous-context-sensitive Systems

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

Foundation
Data Modeling, Logics, Programming Systems, Software Engineering
The RoSI Research Training Group

Research Areas

- Database
- Privacy and Data Security
- Role-based Design
- Automata Theory
- Algebraic and Logical Foundations
- Compiler Construction
- Computer Networks
- Software Technology
41.1. A Primer on Roles

Prof. Dr. Uwe Aßmann
Dr.-Ing. Thomas Kühn
Technische Universität Dresden
Institut für Software- und Multimediatechnik
http://st.inf.tu-dresden.de
/teaching/most
Version 16-1.0, 27.03.18
A Primer on Roles

Basic Roles

- Entities **play** multiple Roles during their lifetime
  
  Examples: *Driver, Doctor, Patient, Student, ...*
A Primer on Roles
Limitations of Object-Oriented Design

Supplier/Customer Problem

[Steimann2000]

- Multiple entities fulfill the same roles

Multiple Classification

- Entity subject to multiple classifying features

State-Dependence

- Specialization of entity depends on state

Supplier

Customer

Person

Company

Queue

Synchronized

NoSynch.

Ordered

Unordered

SO

NO

SU

NU

Status: EState

Order

Shipped

Delivered

carrier: String

delivery: Date

Class

inheritance

RoleType

can play

role implication
A Primer on Roles

Context-Dependent Roles

Context: Engagement
Role: Wife
Co-Role: Husband

Hello, darling!
Hello Prof. Aßmann,
what can I do for you?

Context: Chair of Software Technology
Role: Research Assistant
Co-Role: Professor
A Primer on Roles
Context-Dependent Roles

Context: Chair of Software Technology
Role: Called
Co-Role: (Unknown)Caller

Hello, this is chair of software technology, Thomas Kühn speaking. How may I help you?
A Primer on Roles
Context-Dependent Roles

Context: Federal Republic of Germany
Role: Federal Citizen
Co-Role: Chancellor of Germany

Uh... What... Eh... How may I help you?
A Primer on Roles

Summary

- Role activation **depends** on context of both **Caller** and **Called**
- Roles can denote **places** in a relationship
- Each role is **bound** to context (instance)
- Contexts are hierarchically decomposable
  - May **contain contexts**, but
  - May **overlap**
- In the literature a context can be:
  - Relationship,
  - Process,
  - Social Individual,
  - Social Institution or
  - Ontology
41.2. Roles in Modeling and Programming Languages

Prof. Dr. Uwe Aßmann
Dr.-Ing. Thomas Kühn
Technische Universität Dresden
Institut für Software- und Multimedia-technik
http://st.inf.tu-dresden.de/teaching/most
Version 16-1.0, 27.03.18
Friedrich Steimann

[Kühn2014] A Metamodel Family for Role-based Modelling and Programming Languages
Software Language Engineering SLE'14, Springer (2014)

[Kühn2017] A Family of Role-Based Languages
T. Kühn
References


References


di vision.html](http://www.artima.com/articles/dci
di vision.html) (2009)


Roles in Modeling and Programming Languages

History

“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 and Daya [Bachmann1977]
- Since then many different approaches emerged [Kühn2017]
- No common understanding (or formalism) for roles

Each approach can be classified along design decisions
### Roles in Modeling and Programming Languages

#### Initial Classifying Features of Roles

<table>
<thead>
<tr>
<th>Feature</th>
<th>Metalevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Roles have properties and behaviors</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(2) Roles depend on relationships</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(3) An object may play different roles simultaneously</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(4) An object may play the same role (type) several times</td>
<td>(M0)</td>
</tr>
<tr>
<td>(5) An object may acquire and abandon roles dynamically</td>
<td>(M0)</td>
</tr>
<tr>
<td>(6) Sequence of role acquisition and removal may be restricted</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(7) Unrelated objects can play the same role</td>
<td>(M1)</td>
</tr>
<tr>
<td>(8) Roles can play roles</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(9) Roles can be transferred between objects</td>
<td>(M0)</td>
</tr>
<tr>
<td>(10) The state of an object can be role-specific</td>
<td>(M0)</td>
</tr>
<tr>
<td>(11) Features of an object can be role-specific</td>
<td>(M1)</td>
</tr>
<tr>
<td>(12) Roles restrict access</td>
<td>(M0)</td>
</tr>
<tr>
<td>(13) Different roles may share structure and behavior</td>
<td>(M1)</td>
</tr>
<tr>
<td>(14) An object and its roles share identity</td>
<td>(M0)</td>
</tr>
<tr>
<td>(15) An object and its roles have different identities</td>
<td>(M0)</td>
</tr>
</tbody>
</table>

– Friedrich Steimann [Steimann2000]
## Roles in Modeling and Programming Languages

### Additional Classifying Features of Roles

<table>
<thead>
<tr>
<th>Feature</th>
<th>Metalevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>(16) Relationships between Roles can be constrained</td>
<td>(M1)</td>
</tr>
<tr>
<td>(17) There may be constraints between relationship</td>
<td>(M1)</td>
</tr>
<tr>
<td>(18) Roles can be grouped and constrained together</td>
<td>(M1)</td>
</tr>
<tr>
<td>(19) Roles depend on contexts</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(20) Contexts have properties and behaviors</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(21) A role can be part of several contexts</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(22) Contexts may play roles like objects</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(23) Contexts may play roles which are part of themselves</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(24) Contexts can contain other contexts</td>
<td>(M1,M0)</td>
</tr>
<tr>
<td>(25) Different contexts may share structure and behavior</td>
<td>(M1)</td>
</tr>
<tr>
<td>(26) Contexts have their own identity</td>
<td>(M0)</td>
</tr>
<tr>
<td>(27) The number of roles occurring in a context can be constrained</td>
<td>(M1)</td>
</tr>
</tbody>
</table>

– Kühn et al. [Kühn2000]
Roles in Modeling and Programming Languages

Natures of Roles

- Relational
  - Roles are named places of relationships
- Behavioral
  - Roles adapt the behavior of playing objects
- Context-Dependent
  - Roles can capture context-dependent properties of objects
Roles in Modeling and Programming Languages

Literature Survey [Kühn2014, Kühn2017]

- **Structured Literature Review** of publications since 2000
- Published by the big four (i.e., Springer, IEEE, ACM, Science Direct)

Research Field suffers from **fragmentation and discontinuity**
Roles in Modeling and Programming Languages

Selected Relational Modeling Languages

LODWICK’s UML Notation \cite{Steimann2000}

Object-Role Modeling (ORM) 2 \cite{Halpin2005}
Roles in Modeling and Programming Languages
Selected Relational and Behavioral Modeling Languages

OntoUML [Guizzardi2012]
Roles in Modeling and Programming Languages

Selected Contextual and Relational Modeling Languages

**Information Network Model (INM) [Liu2009]**

- **Class**
  - **Person**
    - firstName: String
    - lastName: String
  - **Company**
    - LegalForm: String
  - **Account**
    - id: Int
    - balance: Money
  - **CheckingAccount**
    - limit: Money
  - **SavingsAccount**
    - transactionFee: Double
  - **Customer**
    - name: String
    - id: Int
  - **Consultant**
  - **Bank**
    - transferals
    - name: String
  - **Transaction**
    - creationTime: Date Time
    - amount: Money
  - **Source**
  - **Target**

- **Role Type**
  - **consultant in**
  - **customer in**

- **Context Type**
  - normal attribute
  - context-based attribute
  - normal relationship
  - context-dependent relationship

- **Name**
  - context-dependent name

- **Managed by**
  - participants
  - trans
Roles in Modeling and Programming Languages
Selected Combined Modeling Languages

Taming Agents and Objects (TAO) [Silva2003]
Roles in Modeling and Programming Languages

Selected Combined Modeling Languages

The HELENA Approach [Hennicker2015]
## Roles in Modeling and Programming Languages

### Comparison (1)

<table>
<thead>
<tr>
<th>Features</th>
<th>2014</th>
<th>Lodwick</th>
<th>Generic Role Model</th>
<th>TAO</th>
<th>RBML</th>
<th>Role-Based Patterns</th>
<th>ORM 2</th>
<th>E-CARGO</th>
<th>Metamodel for Roles</th>
<th>INM</th>
<th>DCI</th>
<th>OntoUML</th>
<th>Helena Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</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>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>
</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>
</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>
</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>
</tr>
<tr>
<td>6</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>7</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>8</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>9</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>10</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>11</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>12</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>13</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>14</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>15</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>

<table>
<thead>
<tr>
<th>Features</th>
<th>2011</th>
<th>EpsilonJ</th>
<th>Chameleon</th>
<th>RICA-J</th>
<th>JAWIRO</th>
<th>OT/J</th>
<th>Runer</th>
<th>Scala Roles</th>
<th>NextEJ</th>
<th>JavaStage</th>
<th>Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</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>2</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>
</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>
</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>
</tr>
<tr>
<td>6</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>7</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>8</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>9</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>10</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>11</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>12</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>13</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>14</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>15</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>

**Role-Based Modeling Languages**

**Role-Based Programming Languages**

Roles depend on Relationships

Roles played by unrelated Objects

| 1: yes, 2: possible, □: no, ø: not applicable | 1: yes, 2: possible, □: no, ø: not applicable |
Roles in Modeling and Programming Languages

Comparison (2)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16</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>17</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>18</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>19</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>20</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>21</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>22</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>23</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>24</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>25</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>26</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>27</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, ☐: possible, ☐: no, ☒: not applicable

Role-Based Modeling Languages

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16</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>17</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>18</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>19</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>20</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>21</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>22</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>23</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>24</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>25</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>26</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>27</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, ☐: possible, ☐: no, ☒: not applicable

Role-Based Programming Languages

Roles depend on Contexts
Roles in Modeling and Programming Languages

Summary

- Discontinuity and fragmentation of research field
- Insufficient formal foundation for role-based languages
- No language supports all features of roles and modeling constraints
- Only few languages provide tool support, most rely on UML stereotypes
- No family of role-based language for all language variants
41.3. The Compartment Role Object Model (CROM)

Prof. Dr. Uwe Aßmann
Dr.-Ing. Thomas Kühn
Technische Universität Dresden
Institut für Software- und Multimediatechnik
http://st.inf.tu-dresden.de/teaching/most
Version 16-1.0, 27.03.18
Literature

T. Kühn, S. Böhme, S. Götz and U. Aßmann

[Kühn2016] FRaMED: Full-Fledge Role Modeling Editor (Tool Demo)
T. Kühn, K. Bierzynski, S. Richly, and U. Aßmann
Software Language Engineering SLE’16, ACM (2016)
References


The Compartment Role Object Model (CROM)  
Design Goals

Design a role-based modeling language for RoSI

- Incorporate all natures of roles and model constraints
- Develop a graphical role-based modeling language
- Provide a formal foundation for the modeling language
- Offer readily applicable tools for modeling and code generation
- Support both formal and automatic verification of role models
The Compartment Role Object Model (CROM)

Graphical Notation

<table>
<thead>
<tr>
<th>Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Types</td>
</tr>
<tr>
<td>Natural Types</td>
</tr>
<tr>
<td><strong>DataType</strong></td>
</tr>
<tr>
<td>fields</td>
</tr>
<tr>
<td>methods()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participation (participates-Relation)</td>
</tr>
<tr>
<td><strong>CompType</strong></td>
</tr>
<tr>
<td><strong>RoleType1</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Role Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compartment Type</strong></td>
</tr>
<tr>
<td><strong>RoleType</strong></td>
</tr>
<tr>
<td>fields</td>
</tr>
<tr>
<td>methods()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Local Role Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Groups</td>
</tr>
<tr>
<td><strong>RoleGroup (n..m)</strong></td>
</tr>
<tr>
<td><strong>RG_1</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Role Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Implication</td>
</tr>
<tr>
<td>Role Equivalence</td>
</tr>
<tr>
<td>Role Prohibition</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Occurrence Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compartment Type</strong></td>
</tr>
<tr>
<td>card1</td>
</tr>
<tr>
<td><strong>RG_1</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relationship Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra-Relationship Constraints</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>irreflexive, acyclic, total, ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inter-Relationship Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>cardC</td>
</tr>
<tr>
<td>cardE</td>
</tr>
<tr>
<td>Rel. Implication</td>
</tr>
<tr>
<td>Rel. Exclusion</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rigid Type Inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RigidType</strong></td>
</tr>
<tr>
<td><strong>SubType</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fulfilment (fills-Relation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td><strong>RoleType</strong></td>
</tr>
<tr>
<td><strong>RigidType</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global Role Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal</td>
</tr>
<tr>
<td>Existential</td>
</tr>
<tr>
<td>Relevant</td>
</tr>
<tr>
<td>∀CT</td>
</tr>
<tr>
<td>∃CT</td>
</tr>
<tr>
<td>∀CT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global Implications / Prohibition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CT_A</strong></td>
</tr>
<tr>
<td>∀ Universal</td>
</tr>
<tr>
<td>∃ Existential</td>
</tr>
<tr>
<td>∀ Relevant</td>
</tr>
<tr>
<td>Prohibition</td>
</tr>
<tr>
<td><strong>CT_B</strong></td>
</tr>
</tbody>
</table>

card = (n...m)
where n is lower and m upper bound
The Compartment Role Object Model (CROM)
Graphical Notation

Example: Banking Application
The Compartment Role Object Model (CROM)

Graphical Notation

**Context**
- Prescriptive (Bottom Up)
- Have (so far) no identity
- Have no intrinsic behavior
- Indefinite lifetime
- Can not play roles
- Has no existential part

**Compartments**
- Descriptive (Top Down)
- Instances carry identity (*Feature 26*)
- Have behavior and state (*Feature 19*)
- Have a defined lifetime
- Can play roles
- Has roles as parts (*Feature 20*)

**Compartment Types**
- Denote an *objectified collaboration* between participants
- Declare a class of compartments (instances) with
  - Properties, behavior, role types, and relationships
- Represent *processes, teams, institutions*, or “context“ [*Kühn2014*]
The Compartment Role Object Model (CROM)

Formal Foundation

Ontological Foundation

Distinction of concepts by meta-properties:

- **Rigidity** [Steimann2000, Guizzardi2005]
  - Type is *rigid*, if its instances have this type until they die
- **Foundedness (Dependence)** [Steimann2000, Guizzardi2005]
  - Type is *founded*, if its instances depend on existence of other instances
- **Identity** [Guizzardi2005]
  - Whether identity of an instance is *unique, derived* or *composed* from others

<table>
<thead>
<tr>
<th>Concept</th>
<th>Rigid</th>
<th>Founded</th>
<th>Identity</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural Types</td>
<td>yes</td>
<td>no</td>
<td>unique</td>
<td>Person, Company</td>
</tr>
<tr>
<td>Data Types</td>
<td>yes</td>
<td>no</td>
<td>derived</td>
<td>Money</td>
</tr>
<tr>
<td>Role Types</td>
<td>no(^1)</td>
<td>yes</td>
<td>derived</td>
<td>Consultant, Customer</td>
</tr>
<tr>
<td>Compartment Types</td>
<td>yes</td>
<td>yes</td>
<td>unique</td>
<td>Bank, Transaction</td>
</tr>
<tr>
<td>Relationship Types</td>
<td>yes</td>
<td>yes</td>
<td>composite</td>
<td>advises, owns</td>
</tr>
</tbody>
</table>

\(^1\) Actual classified as anti-rigid by Guizzardi et.al. [Guizzardi2005]
The Compartment Role Object Model (CROM)

Formal Foundation

Ontological Foundation

Questions to classify domain concepts:

- **Rigidity**
  - Belong instances of the concept to this type throughout their lifetime?

- **Foundedness (Dependence)**
  - Depend instances of the concept on existence of another instance?

- **Identity**
  - Carry instances of the concept a unique, derived or composite identity criterion?

Answers are still domain-dependent
The Compartment Role Object Model (CROM)
Formal Foundation

CROM EMOF (Ecore) Metamodel²

2) https://github.com/Eden-06/CROM
The Compartment Role Object Model (CROM)

Formal Foundation

Formal Model

Definition (Compartment Role Object Model)
\[ \mathcal{M} = (NT, RT, CT, RST, \text{fills, parts, rel}) \]
is a Compartment Role Object Model (CROM) with:

- \( NT, RT, CT, \) and \( RST \) are mutual disjoint sets
- \( \text{fills} \subseteq T \times CT \times RT \) is a relation (with \( T := NT \cup CT \)) and
- \( \text{rel} : RST \times CT \to (RT \times RT) \) is a partial function.

Definition (Compartment Role Object Instance)
\[ i = (N, R, C, \text{type, plays, links}) \]
is a Compartment Role Object Instance (CROI) of a well-formed CROM \( \mathcal{M} \) with:

- \( N, R, \) and \( C \) are mutual disjoint sets
- \( \text{type} : (N \to NT') \cup (R \to RT') \cup (C \to CT) \) is a labeling function,
- \( \text{plays} \subseteq O \times C \times R \) a relation (with \( O := N \cup C \)), and
- \( \text{links} : RST \times C \to 2^{R \times R} \) is a total function.
The Compartment Role Object Model (CROM)

Formal Foundation

Constraint Model

Definition (Constraint Model)

\[ C = (\text{rolec}, \text{card}, \text{intra}, \text{inter}, \text{grolec}) \] is a Constraint Model over \( \mathcal{M} \) with:\(^1\)

- \text{rolec}: \( CT \rightarrow 2^{\text{Card} \times RG} \), and
- \text{card}: \( RST \times CT \rightarrow (\text{Card} \times \text{Card}) \) are partial functions, as well as
- \text{intra} \subseteq RST \times CT \times E and
- \text{inter} \subseteq RST \times CT \times IRC \times RST (with IRC := \{\leq, 
\otimes\}) are relations.
- Additionally, \( \text{grolec} \subseteq QRG \) is a finite set of quantified role groups.

Definition (Syntax of Role Groups)

Role Groups \( RG \) are defined inductively over \( RT \):

\[
\frac{rt \in RT}{rt \in RG} \quad \frac{B \subseteq RG \quad n..m \in Card}{(B, n..m) \in RG}
\]

Definition (Syntax of Quantified Role Groups)

Quantified Role Groups \( QRG \) are defined inductively over \( RT, CT \) and \( RG \):

\[
\frac{a \in RG \quad ct \in CT \quad n..m \in Card}{\exists \langle ct, n..m \rangle. a \in QRG} \quad \frac{B \subseteq QRG \quad n..m \in Card}{\langle B, n..m \rangle \in QRG}
\]

\(^1\)\( E \) is the set of functions \( e: 2^O \times 2^O \times 2^{O \times O} \rightarrow \{0, 1\} \).
Full-fledged Role Modeling Editor (FRaMED)³

- Zoomable editor with Top-Level and Focus view for Compartment Types
- Top-Level view for specifying:
  - Natural, Data and Compartment Types, as well as inheritance and fulfillment
- Focus view for specifying
  - Role and Relationship Type, as well as Role Groups and Constraints

³) https://github.com/leondart/FRaMED
The Compartment Role Object Model (CROM)

Tool Support

Full-fledged Role Modeling Editor (FRaMED)³

- Fully model-driven Eclipse-based editor based on:
  - Eclipse Modeling Framework (EMF), Graphical Editing Framework (GEF), Epsilon (ETL)
- Separation of Graphical Model (GORM) and Semantic Model (CROM)

³ https://github.com/leondart/FRaMED
The Compartment Role Object Model (CROM)

Tool Support

Additional tools supported by FRaMED

- Graphical Notation: FRaMED [Kühn2015]
- Operational Model: ProRoles
- Role-based Language: SCROLL [Leuthäuser2015]
- Conceptual Model: FormalCROM [Kühn2015]
- Textual Syntax: TRoML [Jäkel2016]
- Database Schema: RSQL [Jäkel2016]
- Formal Model: ConDL [Böhme2017]
The Compartment Role Object Model (CROM)

Conclusion

- Incorporating all *natures of roles* and various *modeling constraints*
- Modeling language (formal CROM) fulfilled 22 (19) features of roles
- Introduce common *graphical notation* for role-based modeling languages
- CRO(meta-)Model provides its *abstract syntax*
- FRaMED as eclipse-based editor for *modeling* and *code generation*
- Propose CROM as formal foundation for roles

Still no common role-based modeling language supporting all language variants
Why is it hard to unify the role concept?
Why are compartments necessary to group roles in metamodels?
What was crucial for providing tool support for RoSI?