41. Role-Based (Meta-)Modeling

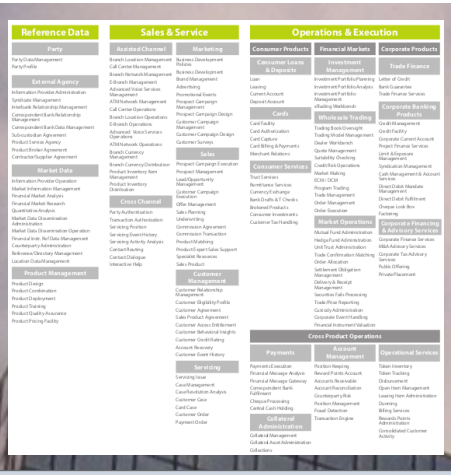
in the Research Training School on
Role-oriented Software Infrastructures (RoSI)

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

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Challenges of Software Systems

Complexity **Change** **Longevity**



Roles increase separation of concerns

Roles allow for dynamic changes of the system

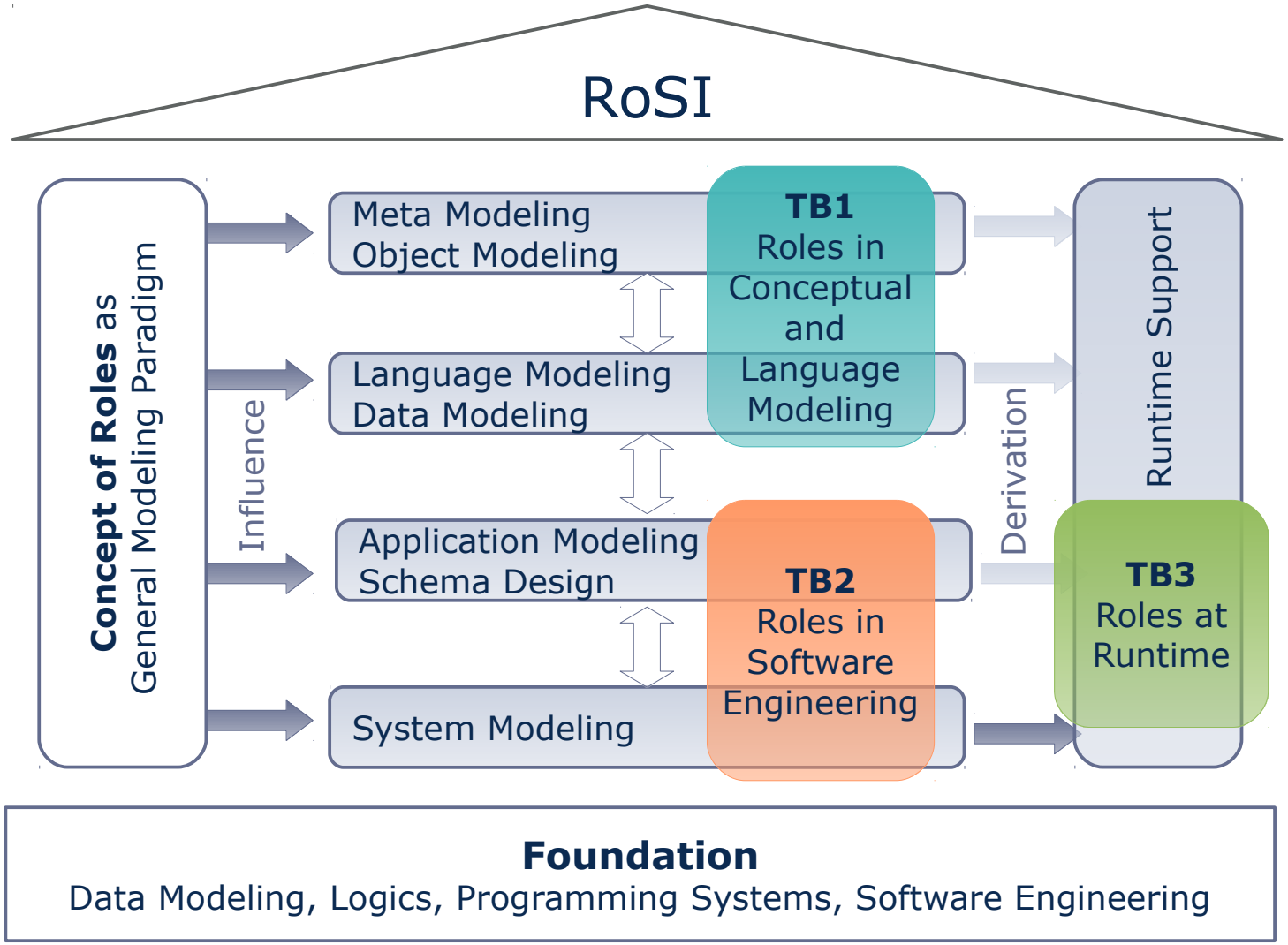
Roles enable updating running applications

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

Markus Bernet (CC-SA 2.5) http://en.wikipedia.org/wiki/File:Frankfurt_Deutsche_Bank.jpg
Katrina Tuliao (CC-SA 2.0) http://www.flickr.com/photos/thewalkingirony/3051500551/

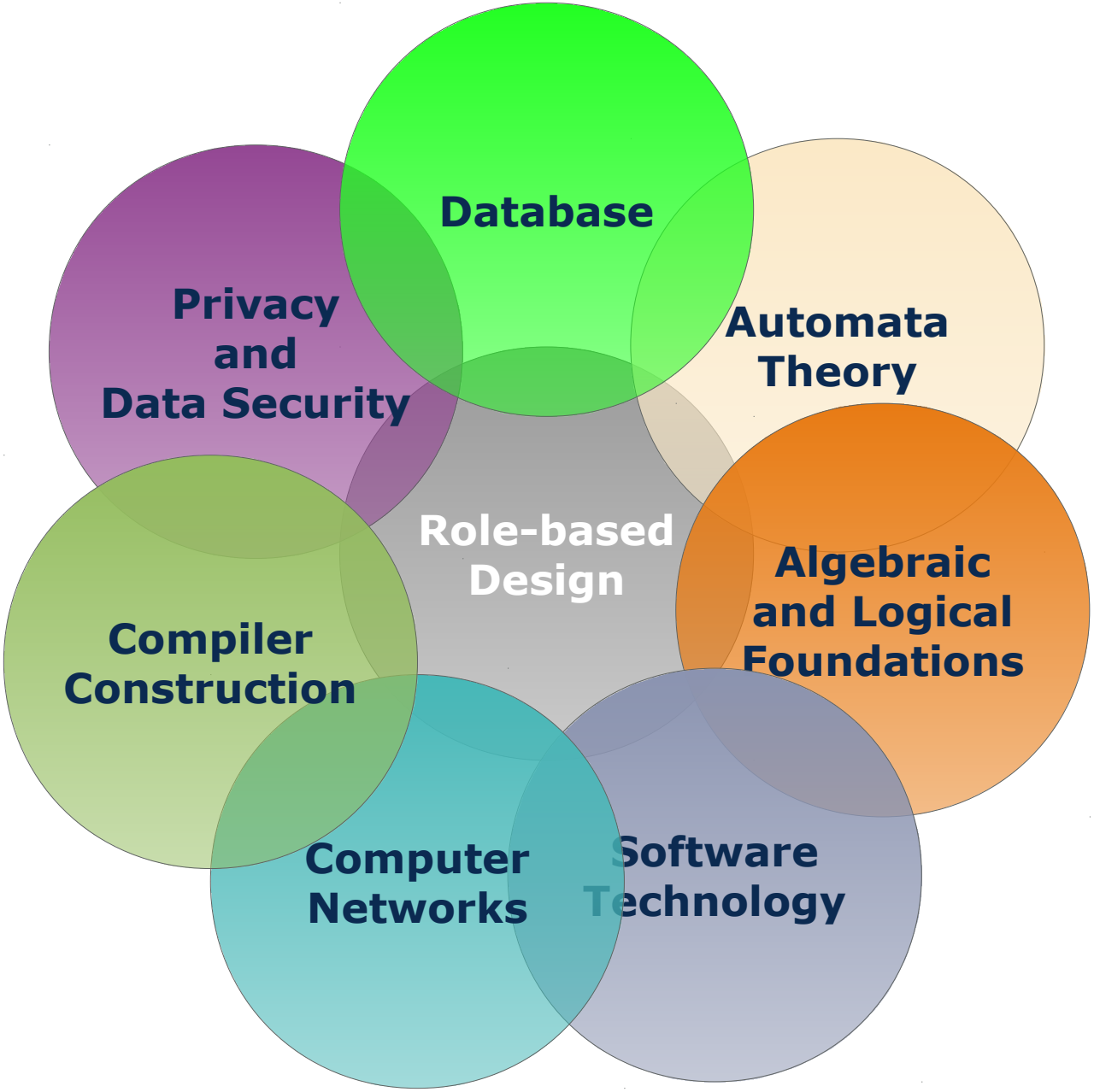
The RoSI Research Training Group

Software Development for continuous-context-sensitive Systems



The RoSI Research Training Group

Research Areas



41.1. A Primer on Roles

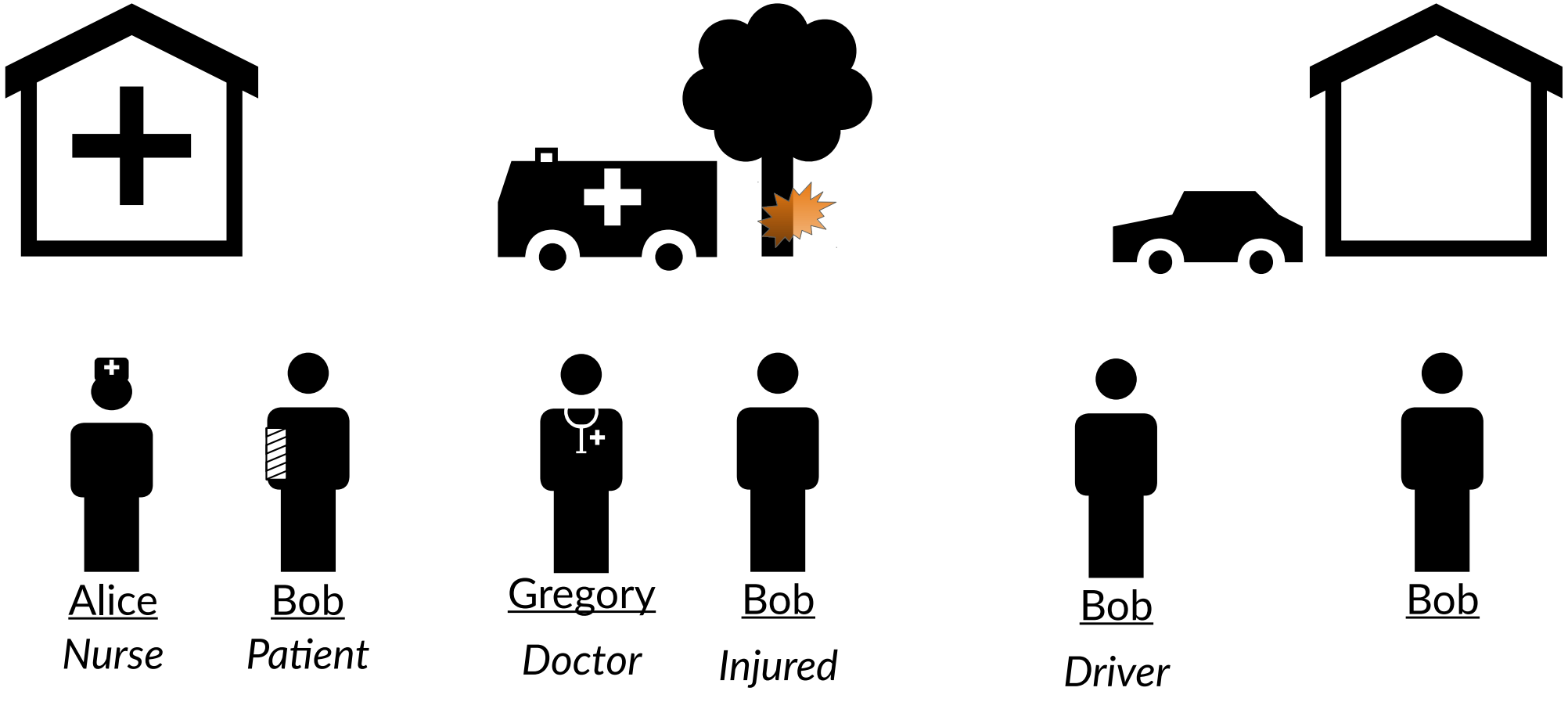
Prof. Dr. Uwe Aßmann
Dr.-Ing. Thomas Kühn
Technische Universität Dresden
Institut für Software- und Multimediatechnik
http://st.inf.tu-dresden.de/teaching/most
Version 16-1.0, 27.03.18

A Primer on Roles

Basic Roles

- Entities **play** multiple Roles during their lifetime

  Examples: *Driver*, *Doctor*, *Patient*, *Student*, ...

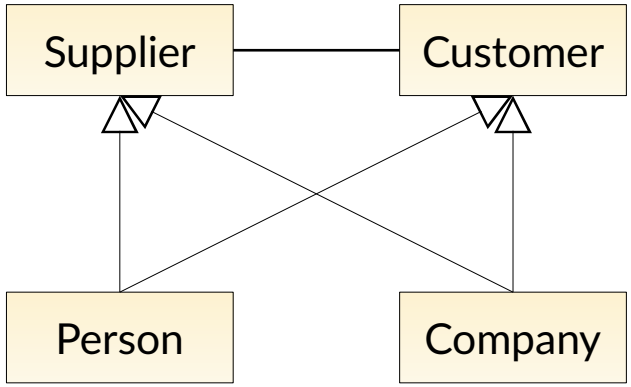


A Primer on Roles

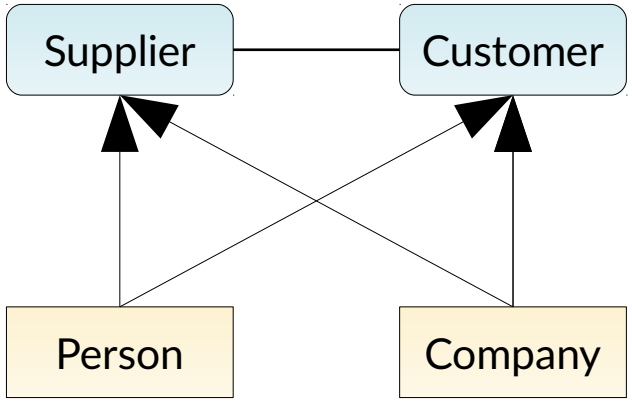
Limitations of Object-Oriented Design

Supplier/Customer Problem

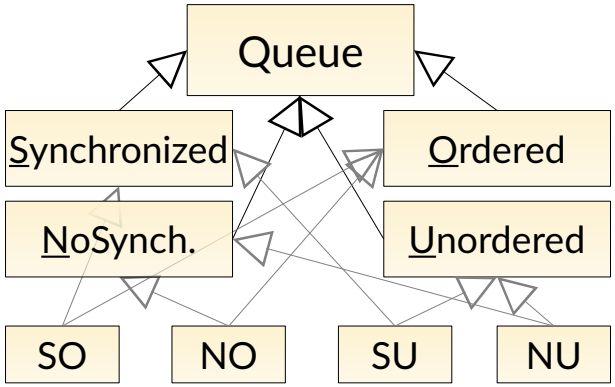
[Steimann2000]



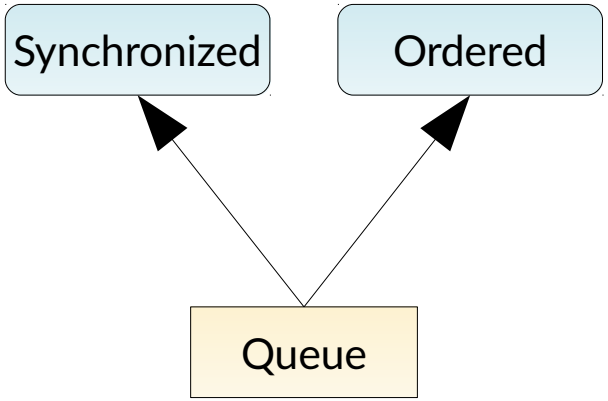
- Multiple entities fulfill the same roles



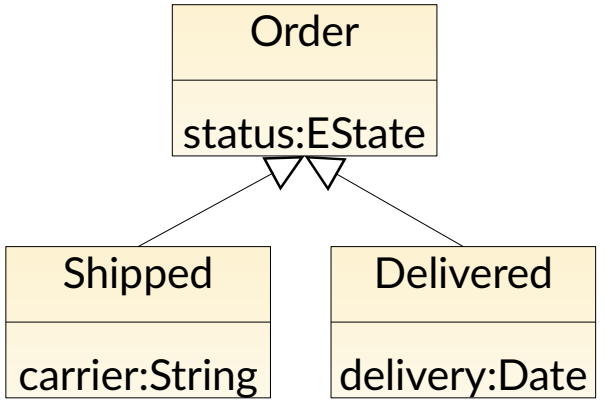
Multiple Classification



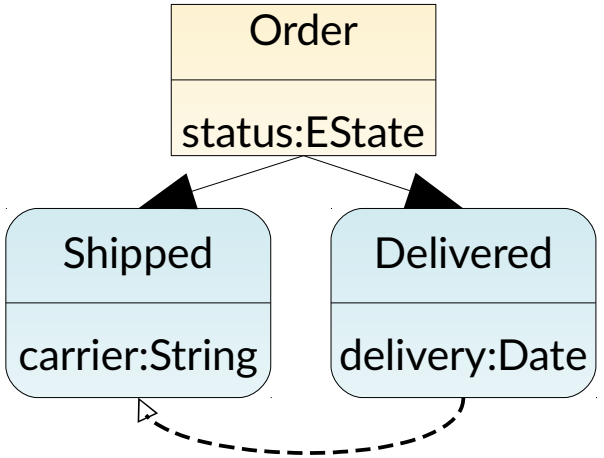
- Entity subject to multiple classifying features



State-Dependence



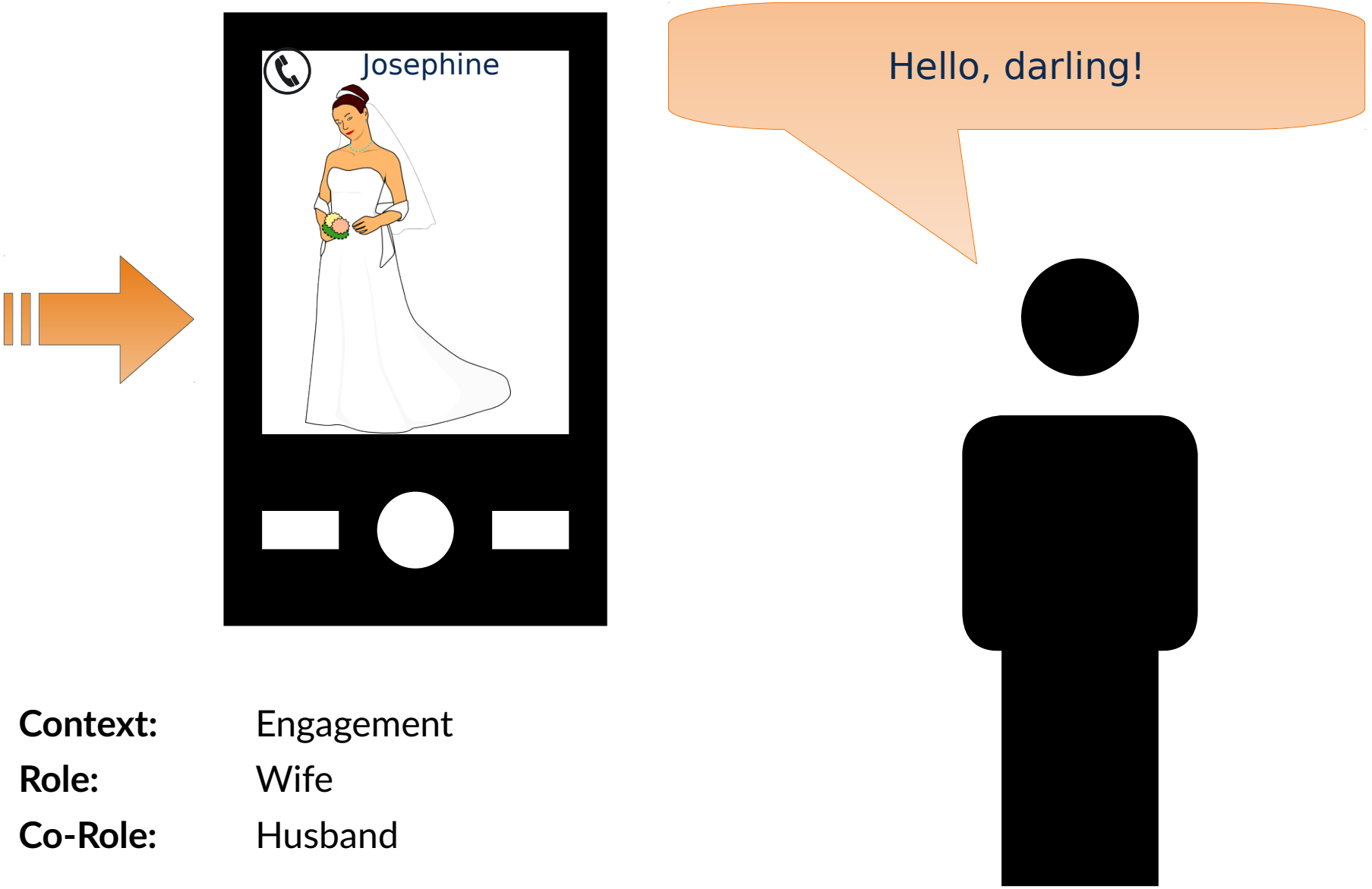
- Specialization of entity depends on state



Class — inheritance — RoleType — can play — role implication

A Primer on Roles

Context-Dependent Roles



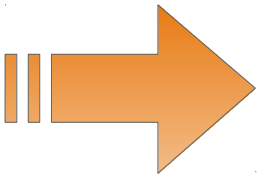
Context: Engagement

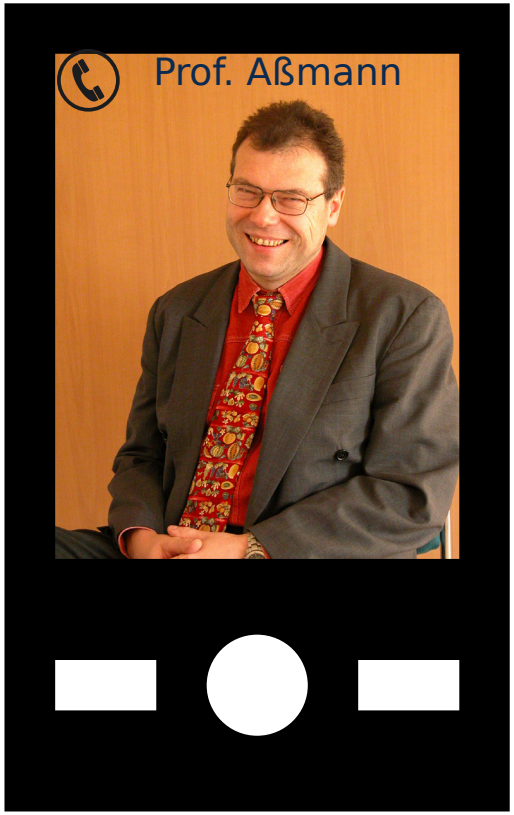
Role: Wife

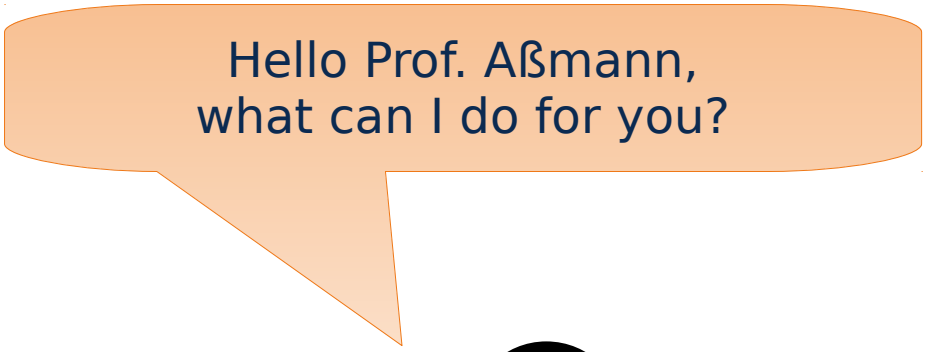
Co-Role: Husband

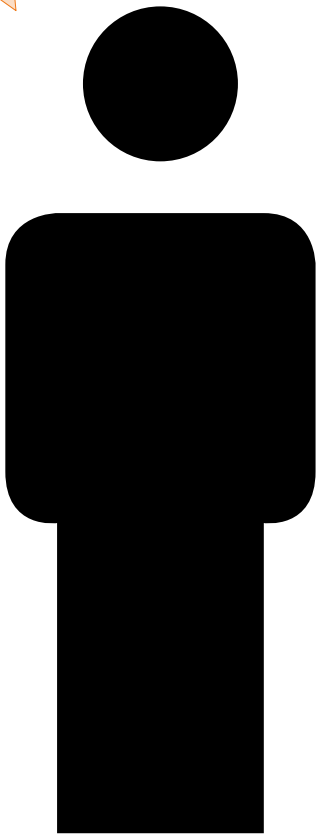
A Primer on Roles

Context-Dependent Roles









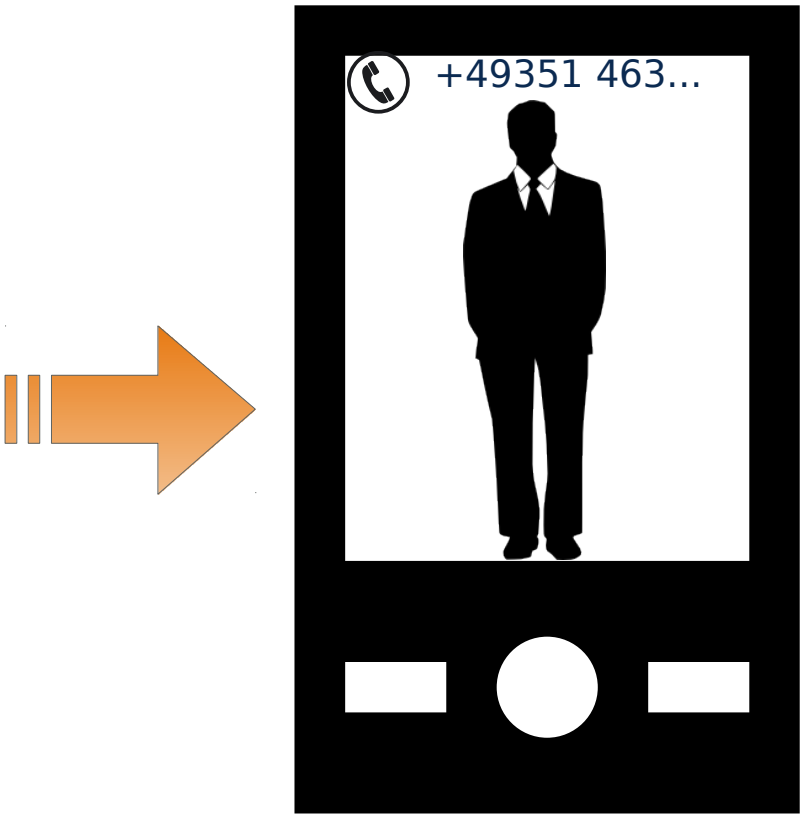
Context: Chair of Software Technology

Role: Research Assistant

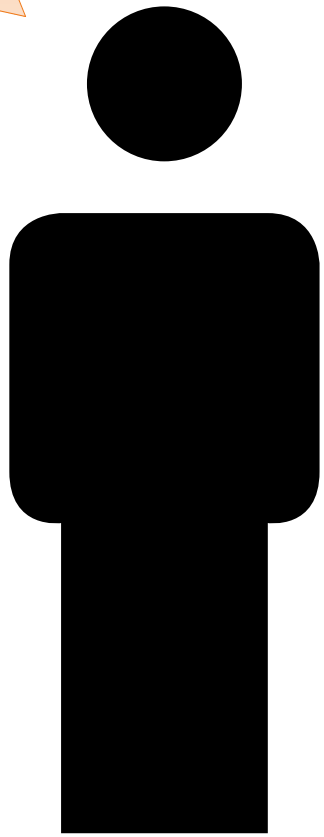
Co-Role: Professor

A Primer on Roles

Context-Dependent Roles



Hello, this is chair of software technology, Thomas Kühn speaking.
How may I help you?



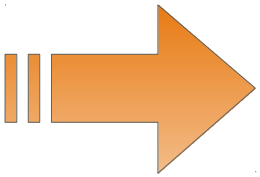
Context: Chair of Software Technology

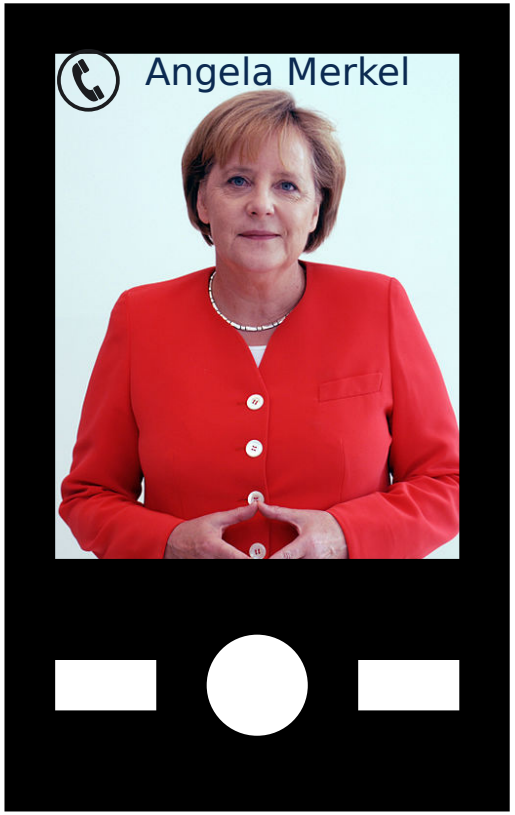
Role: Called

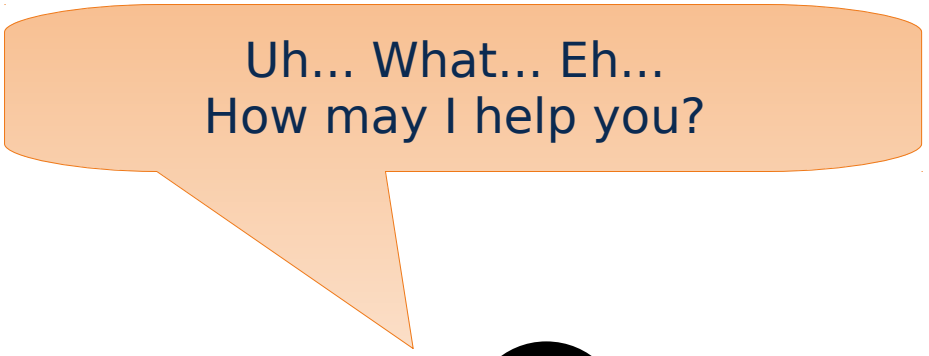
Co-Role: *(Unkown)*Caller

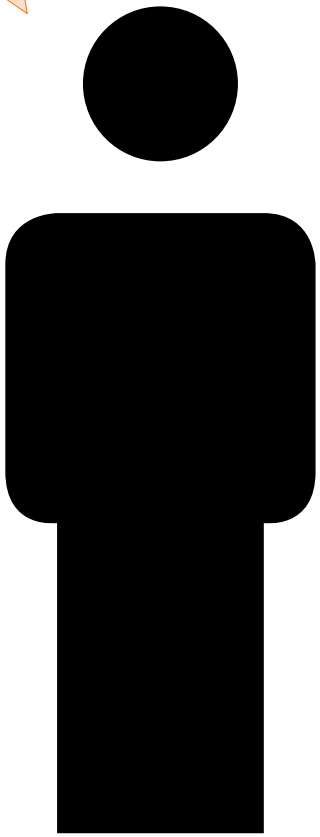
A Primer on Roles

Context-Dependent Roles









Context: Federal Republic of Germany

Role: Federal Citizen

Co-Role: Chancellor of Germany

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 Multimediatechnik
http://st.inf.tu-dresden.de/teaching/most
Version 16-1.0, 27.03.18

Literature

[Steimann2000] On the Representation of Roles in Object-Oriented and Conceptual Modelling.

Friedrich Steimann

Data & Knowledge Engineering, Elsevier, (2000)

[Kühn2014] A Metamodel Family for Role-based Modelling and Programming Languages

T. Kühn, M. Leuthäuser, S. Götz, C. Seidl, and U. Aßmann

Software Language Engineering SLE'14, Springer (2014)

[Kühn2017] A Family of Role-Based Languages

T. Kühn

Dissertation, Technische Universität Dresden, Fakultät Informatik (2017)

References

[Bachman1977] **The Role Concept in Data Models.** Charles W. Bachman and Manilal Daya. *VLDB (1977)*

[Balzer2007] **A Relational Model of Object Collaborations and its Use in Reasoning about Relationships.** S. Balzer, T. Gross, and P. Eugster. *ECOOP, vol. 4609 of LNCS (2007)*

[Barbosa2012] **Modeling and Programming with Roles: introducing JavaStage.** F.S. Barbosa and A. Aguiar. *Tech.Rrep., Instituto Polit´cnico de Castelo Branco (2012)*

[Burmester2005] **Model-Driven Development of Reconfigurable Mechatronic Systems with Mechatronic UML.** S. Burmester, G. Holger, and M. Tichy. *Model Driven Architecture, Springer, 2005*

[Carrington2004] **Using Integrated Metamodeling to Define OO Design Patterns with Object-Z and UML.** S.-K. Kim and D. Carrington. *11th Asia-Pacific Software Engineering Conference, IEEE (2004)*

[Dahchour2002] **A Generic Role Model for Dynamic Objects.** M. Dahchour et al. *Advanced Information Systems Engineering, Springer (2002)*

[Genovese2007] **A meta-model for roles: Introducing sessions.** V. Genovese. *Roles' 07 (2007)*

[Graversen2003] **Implementation of a Role Language for Object-Specific Dynamic Separation of Concerns.** K.B. Graversen and K. Østerbye. *AOSD03 Workshop (2003)*

[Halpin2005] **ORM 2.** T. A. Halpin. *OTM Workshops, vol. 3762 of LNCS (2005)*

[He2006] **Rava: Designing a Java Extension with Dynamic Object Roles.** Chengwan He, et al. *13th Annual IEEE International Symposium and Workshop (2006)*

[Hennicker2014] **Foundations for Ensemble Modeling The HELENA Approach.** *R. Hennicker and A. Klarl. Specification, Algebra, and Software, Springer (2014)*

References

[Kamina2009] **Towards Safe and Flexible Object Adaptation.** Tetsuo Kamina and Tetsuo Tamai. *International Workshop on COP (2009)*

[Kim2002] **Using Role-Based Modeling Language (RBML) to Characterize Model Families.** D.-K. Kim, R. France, S. Ghosh, and E. Song. *Engineering of Complex Computer Systems, IEEE, 2002*

[Klarman2010] **$ALC_{ALC:}$ A Context Description Logic.** S. Klarman and V. Gutiérrez-Basulto. *Workshop on Logics in Artificial Intelligence, Springer (2010)*

[Liu2009] **Information Networking Model.** M. Liu and J. Hu. *Conceptual Modeling - ER (2009)*

[Reenskaug2009] **The DCI Architecture: A New Vision of Object-oriented Programming.** T. Reenskaug and J. O. Coplien. *http://www. artima. com/articles/dci vision. Html (2009)*

[Selçuk2004} **JAWIRO: Enhancing Java with Roles.** Y. E. Selçuk and N. Erdogan. *Symposium on Computer and Information Sciences, Springer (2004)*

[Silva2003] **Taming Agents and Objects in Software Engineering.** V. Da Silva, A. Garcia, A. Brandão, C. Chavez, C. Lucena, and P. Alencar. *Workshop on SE for Large-Scale MAS, Springer (2003)*

[Zhu2006] **Role-Based Collaboration and its Kernel Mechanisms.** H. Zhu and M. Zhou. *IEEE Transactions 36(4) (2006)*

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

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

– Friedrich Steimann [Steimann2000]

Roles in Modeling and Programming Languages

Additional Classifying Features of Roles

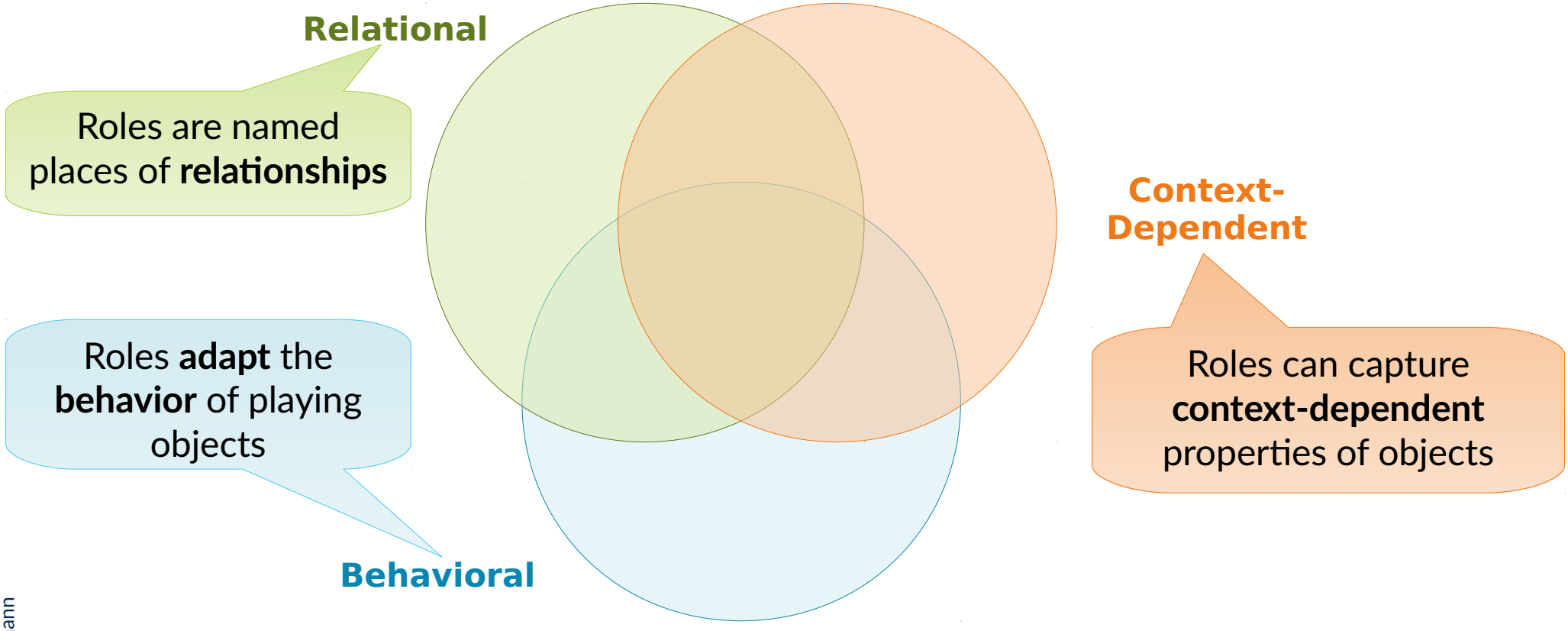
Feature	*Metalevel*
(16)Relationships between Roles can be constrained	*(M1)*
(17)There may be constraints between relationship	*(M1)*
(18)Roles can be grouped and constrained together	*(M1)*
(19)Roles depend on contexts	*(M1,M0)*
(20)Contexts have properties and behaviors	*(M1,M0)*
(21)A role can be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

Context-Dependent

– *Kühn et al.* [Kühn2000]

Roles in Modeling and Programming Languages

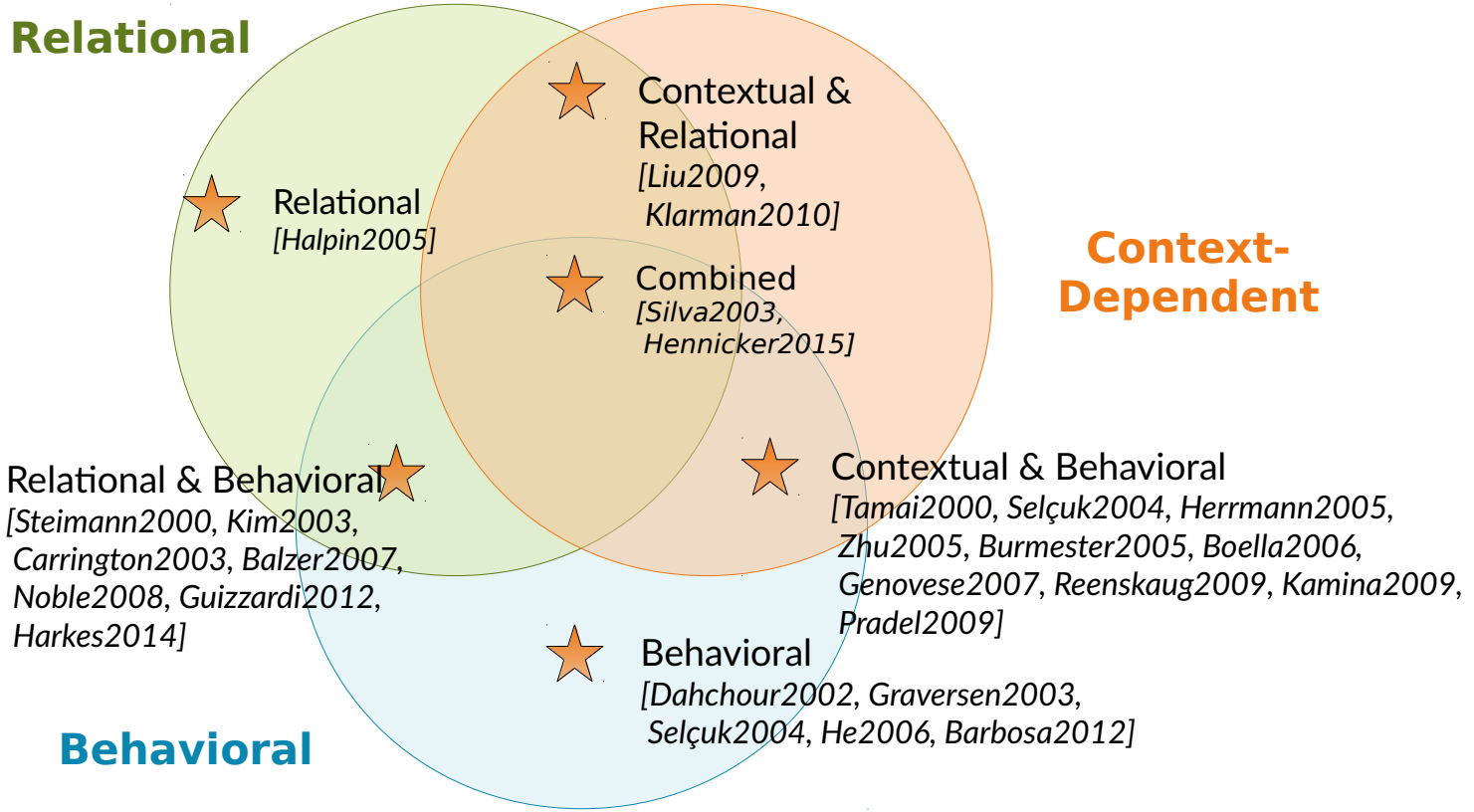
Natures of Roles



Roles in Modeling and Programming Languages

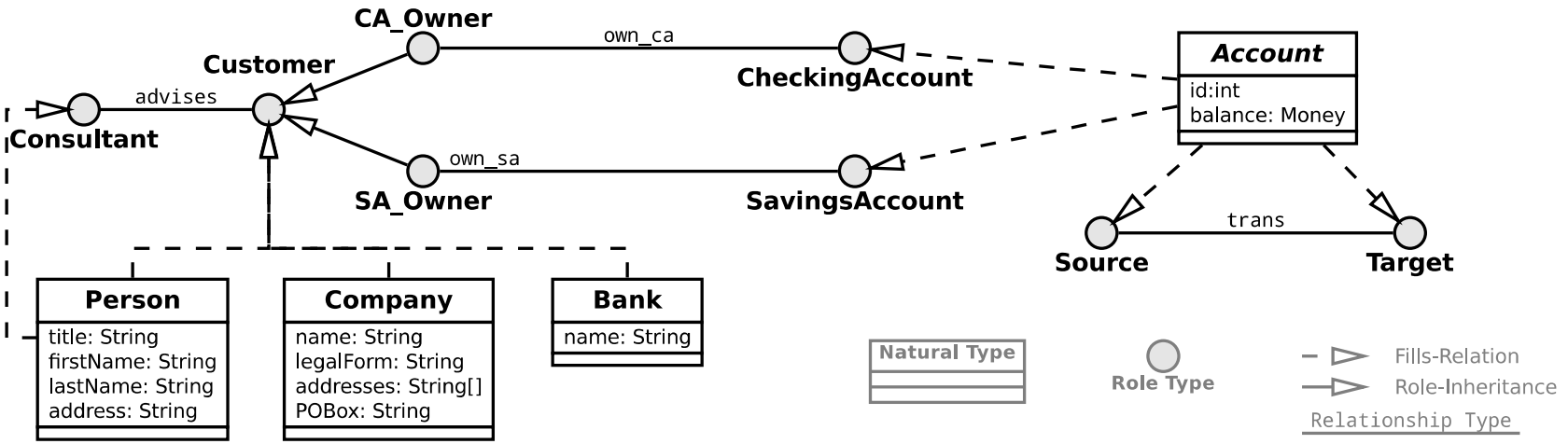
Literature Survey *[Kühn2014,Kühn2017]*

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

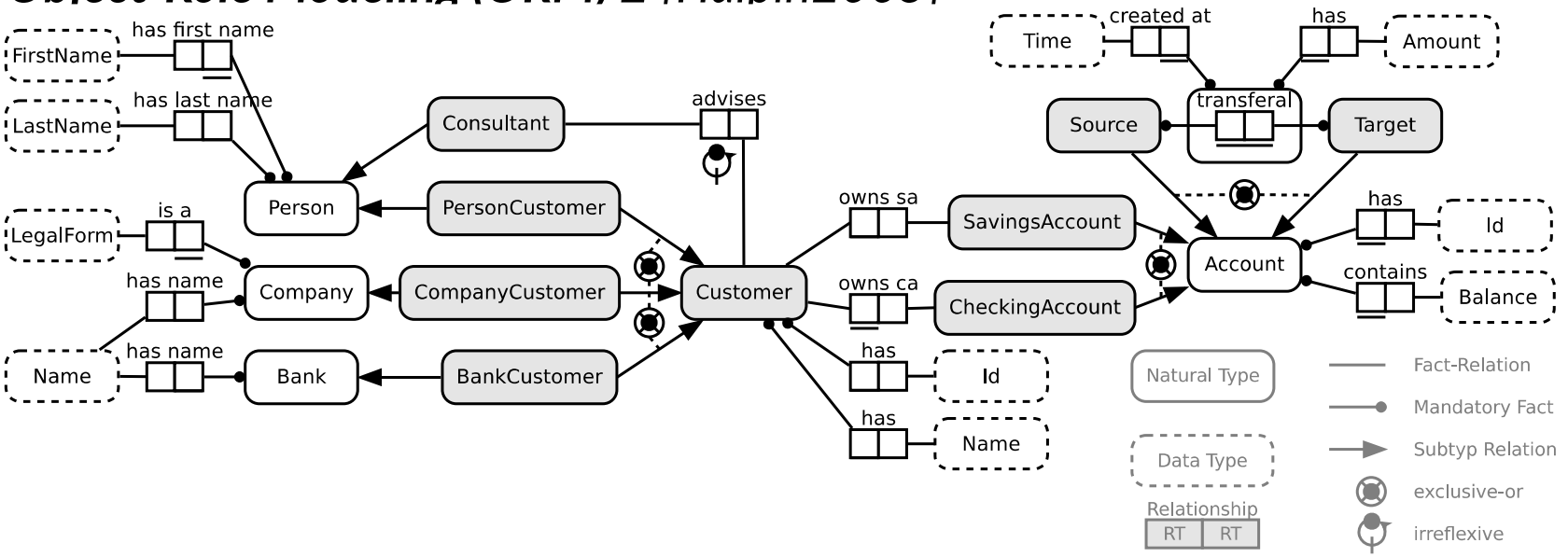


Research Field suffers from *fragmentation* and *discontinuity*

LODWICK's UML Notation *[Steimann2000]*



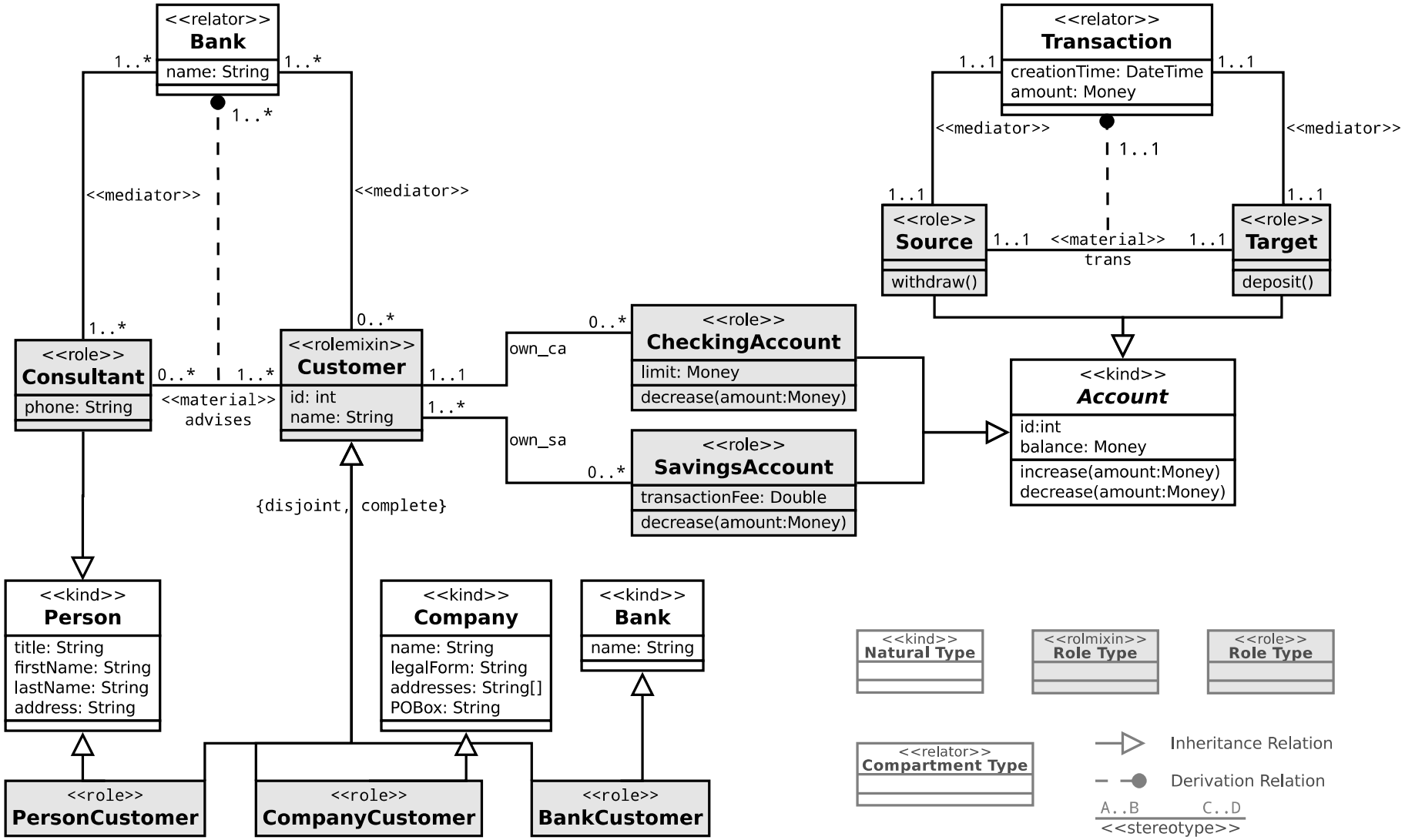
Object-Role Modeling (ORM) 2 [Halpin2005]



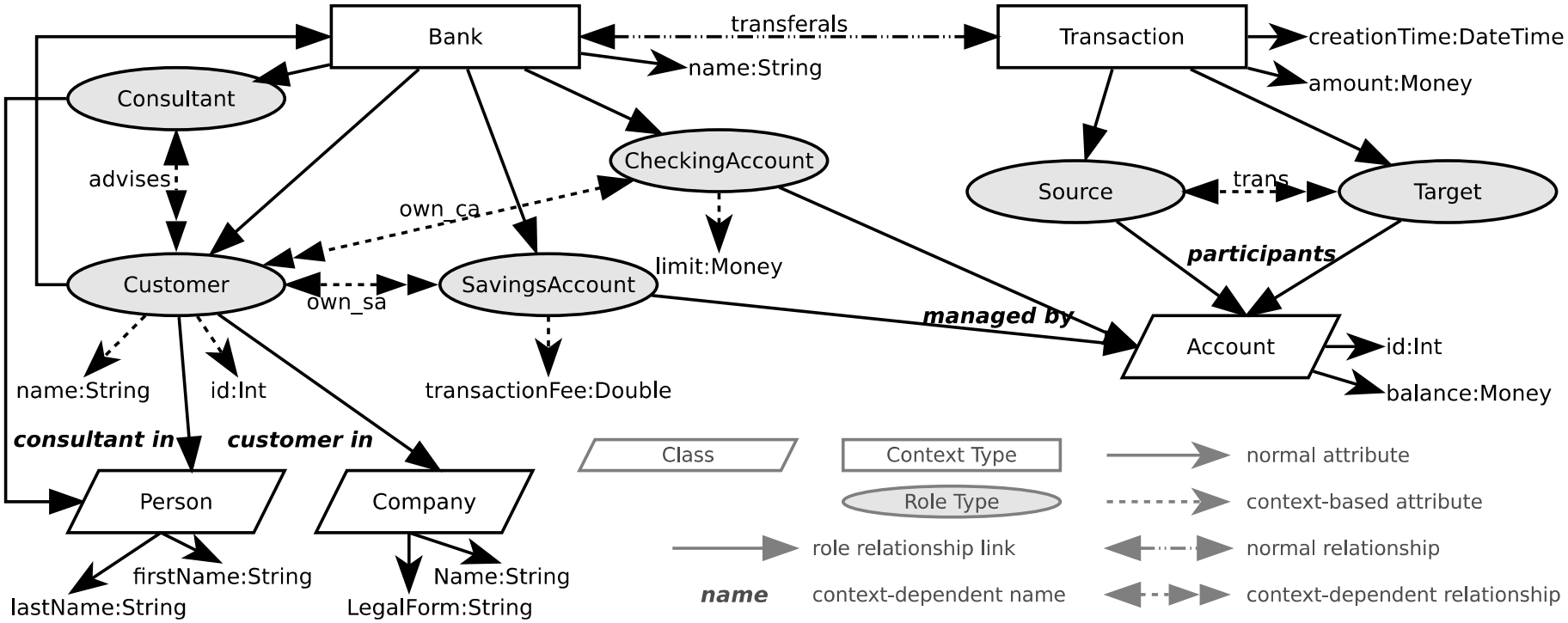
Roles in Modeling and Programming Languages

Selected Relational and Behavioral Modeling Languages

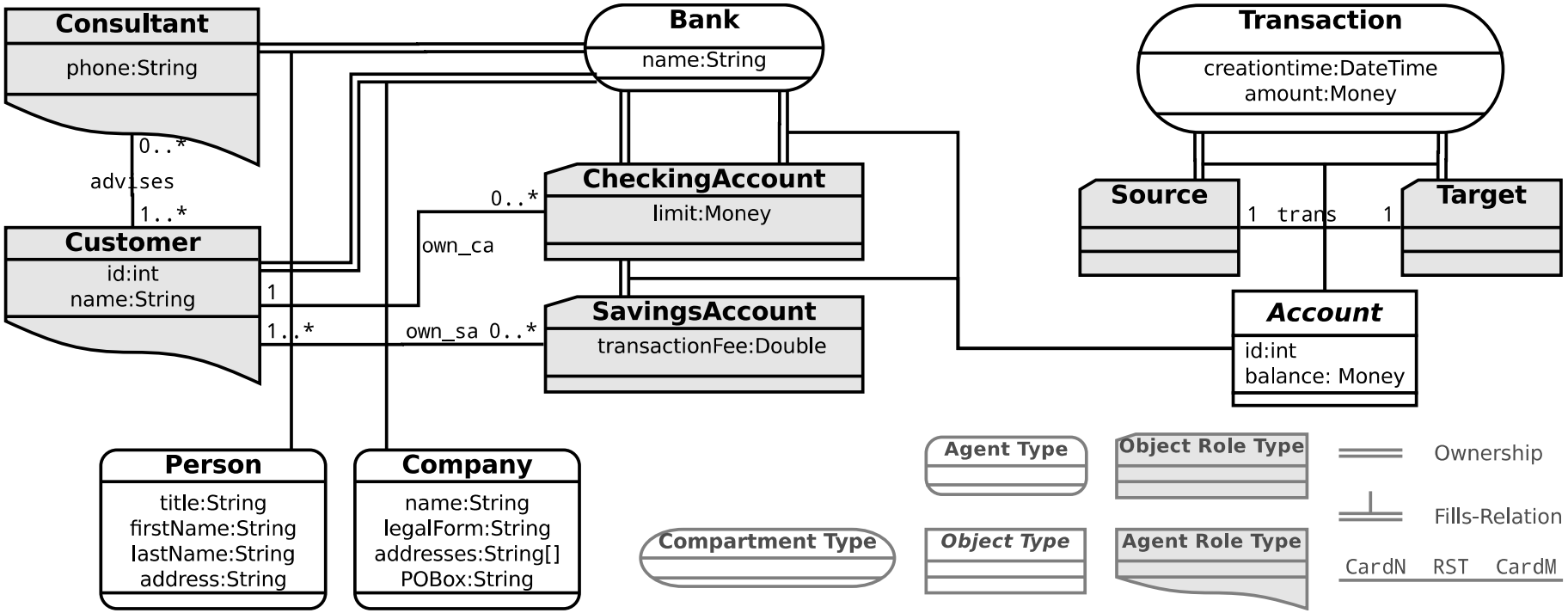
OntoUML *[Guizzardi2012]*



Information Network Model (INM) [Liu2009]



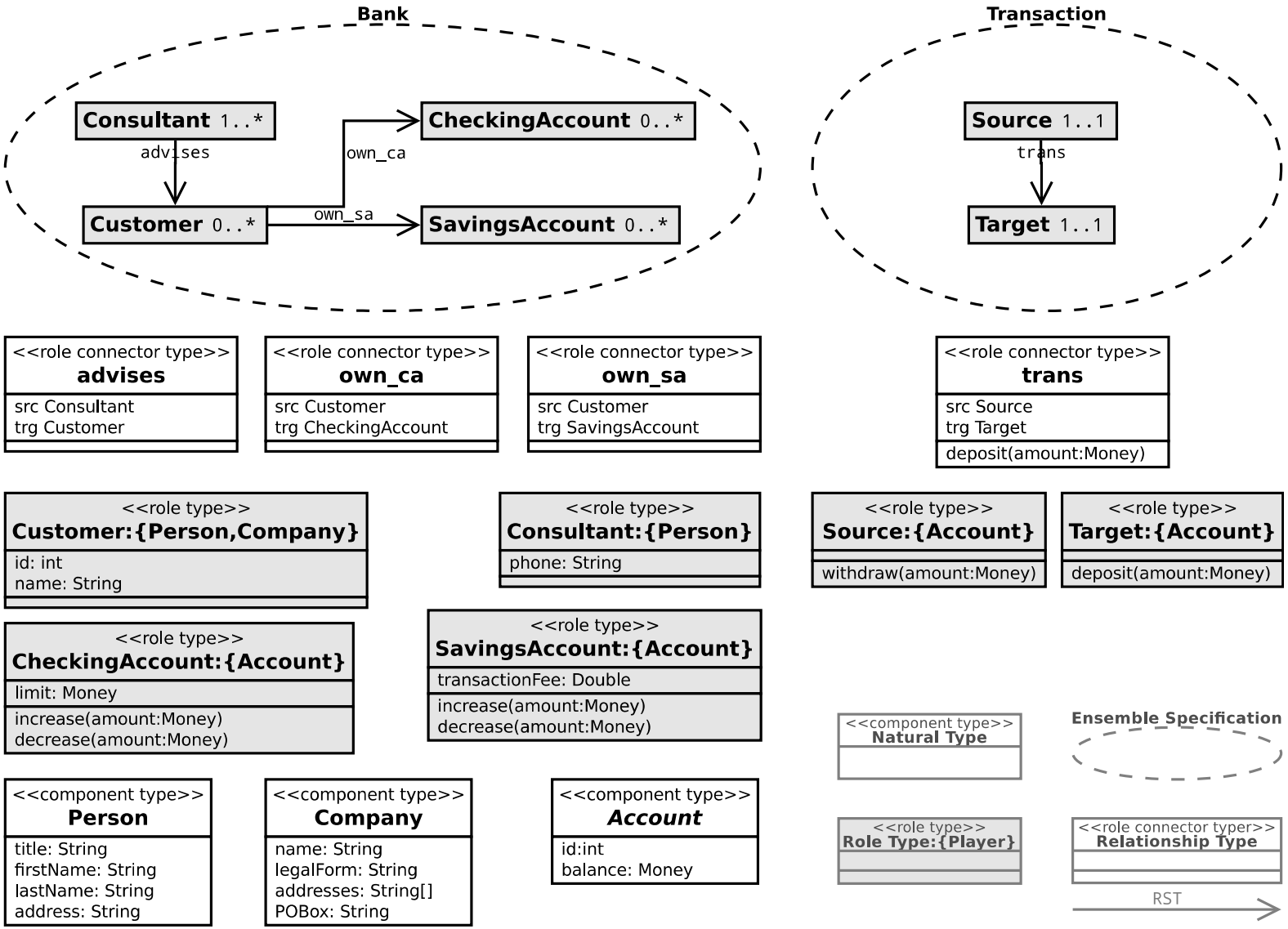
Taming Agents and Objects (TAO) *[Silva2003]*



Roles in Modeling and Programming Languages

Selected Combined Modeling Languages

The HELENA Approach *[Hennicker2015]*



Roles in Modeling and Programming Languages

Comparison (1)

Features 2014	Lodwick 2000	Generic Role Model 2002	TAO 2003	RBML 2003	Role-Based Patterns 2004	ORM 2 2005	E-CARGO 2006	Metamodel for Roles 2007	INM 2009	DCI 2009	OntoUML 2012	Helena Approach 2014
1	■	■	■	■	■	⊞	⊞	■	■	■	■	■
2	■	□	■	■	■	■	□	⊞	■	⊞	■	■
3	■	■	■	■	■	■	■	⊞	■	■	⊞	■
4	■	■	■	■	□	■	■	■	■	□	⌀	■
5	■	■	⌀	⌀	⌀	■	■	■	⌀	⊞	⌀	■
6	⊞	■	⌀	⌀	■	⊞	⊞	⌀	□	□	□	□
7	■	□	■	□	□	□	⊞	■	□	■	⊞	■
8	□	■	□	□	□	□	□	■	■	□	■	□
9	⊞	□	⊞	⌀	⌀	⌀	■	⊞	⌀	□	⌀	□
10	⊞	■	■	⌀	□	⌀	⊞	⊞	■	■	■	■
11	⊞	■	■	□	■	□	□	⊞	■	■	■	■
12	⌀	⊞	⊞	■	⌀	⌀	■	⌀	⌀	■	⌀	■
13	⊞	■	■	■	■	□	□	■	■	⊞	⊞	□
14	■	□	□	■	■	■	■	⊞	■	⊞	■	□
15	□	■	■	□	□	□	⊞	⊞	□	⊞	□	■

■: yes, ⊞: possible, □: no, ⌀: not applicable

Role-Based Modeling Languages

Features 2014	EpsilonJ 2001	Chameleon 2003	RICA-J 2004	JAWIRO 2004	OT/J 2005	Rava 2006	powerJava 2006	Rumer 2007	First-Class Relationships 2008	Scala Roles 2009	NextEJ 2009	JavaStage 2012	Relations 2014
1	■	■	■	■	■	■	■	■	■	■	■	■	□
2	⊞	□	□	□	⊞	□	⊞	■	■	⊞	⊞	□	■
3	■	■	■	■	■	■	■	■	■	■	■	■	■
4	■	■	□	■	■	□	■	■	■	■	■	⊞	■
5	■	■	■	■	■	■	■	■	■	■	■	□	⌀
6	□	□	⊞	⊞	■	□	⊞	■	□	□	□	■	⊞
7	■	□	■	■	□	■	□	⊞	□	■	■	■	□
8	■	□	■	■	■	□	■	□	□	■	■	■	□
9	■	■	□	■	□	□	■	□	□	■	■	□	□
10	■	■	■	■	■	■	■	■	■	■	■	■	□
11	■	■	■	■	■	■	■	■	■	■	■	■	□
12	■	■	■	■	■	■	■	■	□	■	■	■	⌀
13	□	□	■	■	■	■	■	□	□	■	□	■	□
14	⊞	⊞	□	⊞	⊞	□	□	■	■	■	⊞	□	■
15	■	■	■	■	■	■	■	□	□	■	■	■	□

■: yes, ⊞: possible, □: no, ⌀: not applicable

Role-Based Programming Languages

Roles depend on Relationships

Roles played by unrelated Objects

Roles in Modeling and Programming Languages

Comparison (2)

Roles depend on Contexts

Features 2014	Lodwick 2000	Generic Role Model 2002	TAO 2003	RBML 2003	Role-Based Patterns 2004	ORM 2 2005	E-CARGO 2006	Metamodel for Roles 2007	INM 2009	DCI 2009	OntoUML 2012	Helena Approach 2014
16	■	□	■	■	■	■	□	□	□	□	■	□
17	□	□	□	□	□	■	□	□	□	□	■	□
18	□	□	□	□	□	□	⊞	□	□	□	□	□
19	⊞	□	■	□	□	⊞	■	■	■	■	⊞	■
20	□	□	■	□	□	⊞	□	■	■	■	■	■
21	⊞	□	□	□	□	□	■	■	⊞	□	■	■
22	□	□	■	□	□	■	□	■	■	■	□	□
23	□	□	□	□	□	□	□	■	□	□	□	□
24	□	□	■	□	□	□	□	□	■	□	□	□
25	□	□	■	□	□	□	□	■	■	⊞	□	□
26	□	□	■	□	□	■	■	⊞	■	■	□	■
27	□	□	□	■	■	□	■	⊞	□	□	⊞	■

■: yes, ⊞: possible, □: no, ∅: not applicable

Role-Based Modeling Languages

Features 2014	EpsilonJ 2001	Chameleon 2003	RICA-J 2004	JAWIRO 2004	OT/J 2005	Rava 2006	powerJava 2006	Rumer 2007	First-Class Relationships 2008	Scala Roles 2009	NextEJ 2009	JavaStage 2012	Relations 2014
16	□	□	□	□	□	□	□	■	□	□	□	□	□
17	□	□	□	□	□	□	□	□	□	□	□	□	□
18	□	□	□	□	■	□	□	⊞	□	□	□	□	□
19	■	□	■	□	■	□	■	⊞	⊞	■	■	□	■
20	■	□	■	□	■	□	■	■	■	■	■	□	■
21	□	□	□	□	□	□	⊞	□	□	□	□	□	□
22	■	□	□	□	■	□	■	■	□	■	■	□	■
23	□	□	□	□	■	□	□	□	□	⊞	□	□	□
24	⊞	□	□	□	■	□	■	■	□	■	■	□	⊞
25	□	□	■	□	■	□	⊞	□	■	■	□	□	□
26	■	□	■	□	■	□	■	■	■	■	■	□	■
27	⊞	□	□	□	⊞	□	□	□	□	□	⊞	□	■

■: yes, ⊞: possible, □: no, ∅: not applicable

Role-Based Programming Languages

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

[Kühn2015] A Combined Formal Model for Relational Context-Dependent Roles
T. Kühn, S. Böhme, S. Götz and U. Aßmann
Software Language Engineering *SLE'15, ACM (2015)*

[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

[Leuthäuser2015] Enabling View-based Programming with SCROLL: Using Roles and Dynamic Dispatch for Establishing View-based Programming. Max Leuthäuser and Uwe Aßmann. *MORSE/VAO '15, ACM (2015)*

[Jäkel2016] **Towards a Contextual Database.** T. Jäkel, T. Kühn, H. Voigt, and W. Lehner. *ADBIS (2016)*

[Böhme2017] ***Reasoning on Context-Dependent Domain Models.*** *S Böhme, T. Kühn.Proceedings of the JIST (2017)*

The Compartment Role Object Model (CROM)

Design Goals

Design a role-based modeling language for RoSI

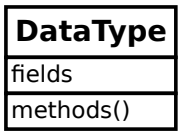
- Incorporate **all** natures of roles and model constraints
- Develop a *graphical* role-based modeling language
- Provide a *formal foundation* for the modeling language
- Offer readily *applicable tools* for modeling and code generation
- Support both *formal* and *automatic verification* of role models

The Compartment Role Object Model (CROM)

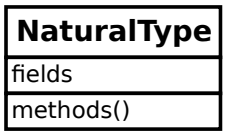
Graphical Notation

Entities

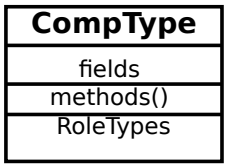
Data Types



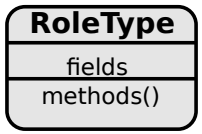
Natural Types



Compartment Types

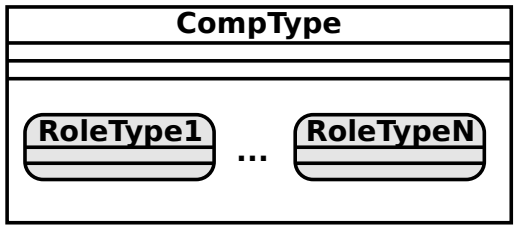


Role Types

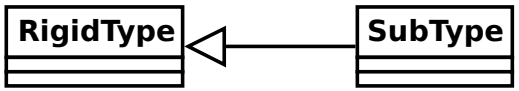


Relations

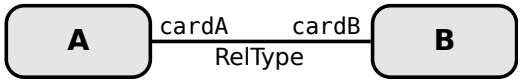
Participation (participates-Relation)



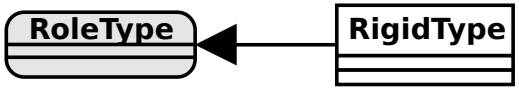
Rigid Type Inheritance



Binary Relationship

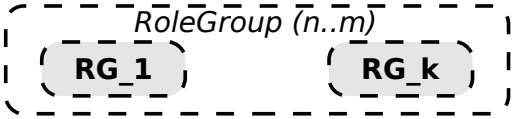


Fulfilment (fills-Relation)

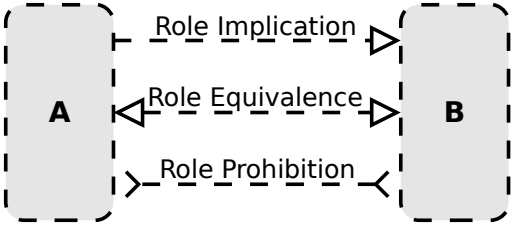


Local Role Constraints

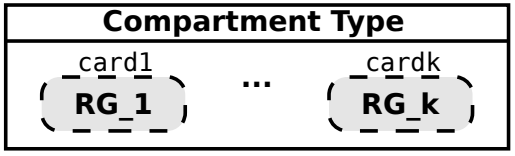
Role Groups



Role Constraints

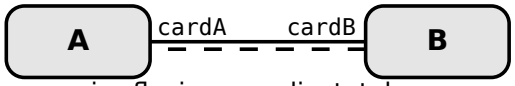


Occurence Constraints



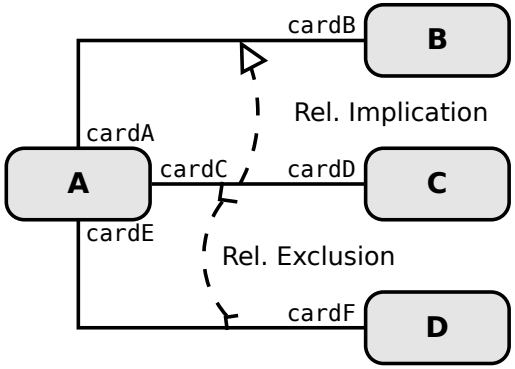
Relationship Constraints

Intra-Relationship Constraints



irreflexive, acyclic, total, ...

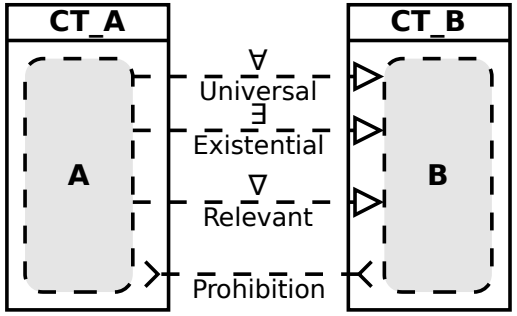
Inter-Relationship Constraints



Global Role Constraints

Universal Existential Relevant

Global Implications / Prohibition

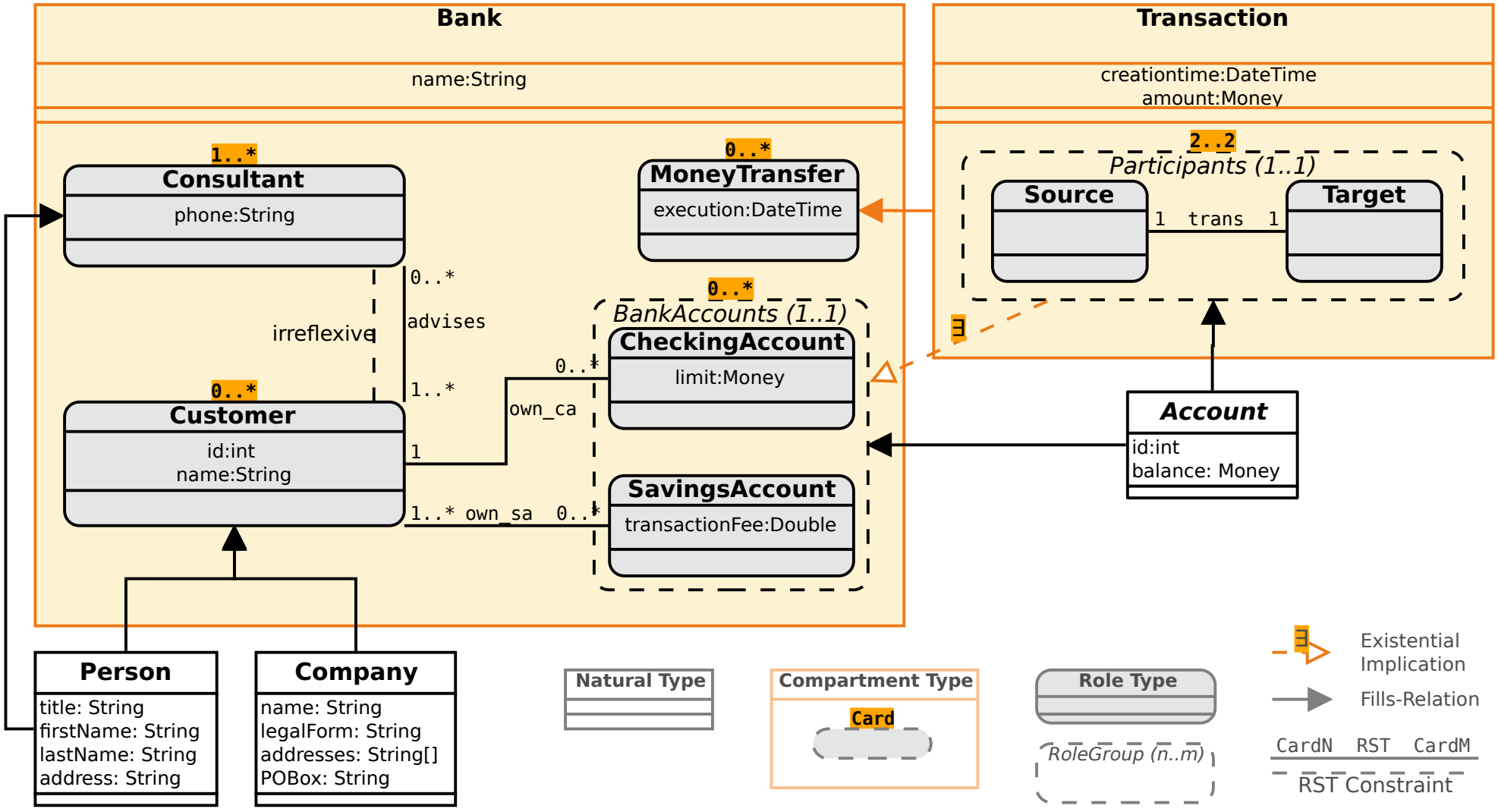



```
card = (n...m)
```
where n is lower and m upper bound

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

Concept	Rigid	Founded	Identity	Example
Natural Types	yes	no	unique	*Person, Company*
Data Types	yes	no	derived	*Money*
Role Types	no[1]	yes	derived	*Consultant, Customer*
Compartment Types	yes	yes	unique	*Bank, Transaction*
Relationship Types	yes	yes	composite	*advises, owns*

[1]) Actual classified as **anti-rigid** by Guizzardi et.al. *[Guizzardi2005]*

Ontological Foundation

Questions to classify domain concepts:

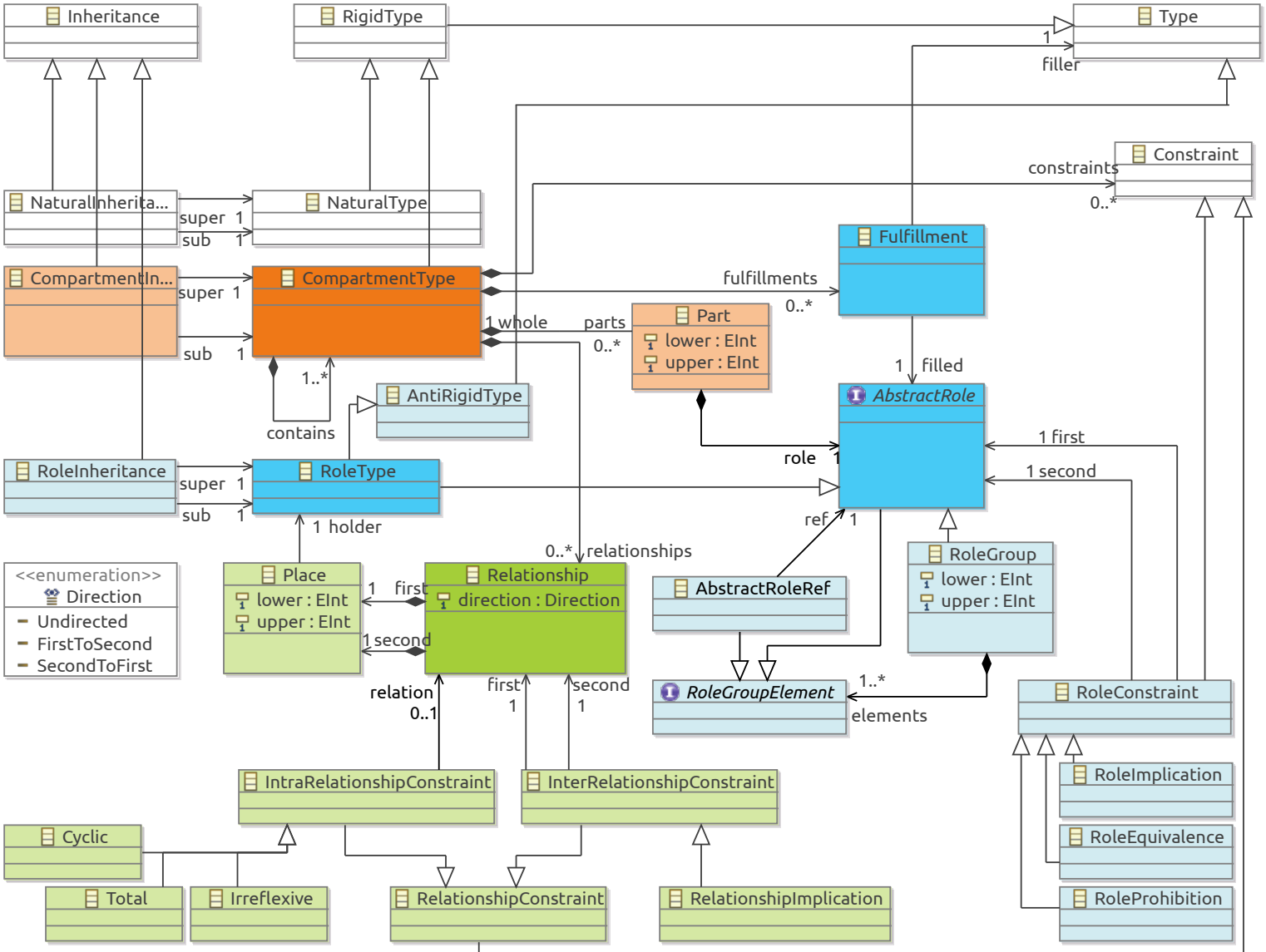
- **Rigidity**
  - Belong instances of the concept to this type throughout their lifetime?
- **Foundedness *(Dependence)***
  - *Depend instances of the concept on existence of another instance?*
- **Identity**
  - *Carry instances of the concept a unique, derived or composite identity criterion?*

Answers are still domain-dependent

The Compartment Role Object Model (CROM)

Formal Foundation

CROM EMOF (Ecore) Metamodel[2]



2) https://github.com/Eden-06/CROM

Formal Model

Definition (Compartment Role Object Model)

$\mathcal{M} = (NT, RT, CT, RST, \text{fills}, \text{parts}, \text{rel})$ 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)

$\mathfrak{i} = (N, R, C, \text{type}, \text{plays}, \text{links})$ 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.

Constraint Model

Definition (Constraint Model)

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

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

Definition (Syntax of Role Groups)

Role Groups RG are defined inductively over RT:

$$\frac{rt \in RT}{rt \in RG} \qquad \frac{B \subseteq RG \quad n..m \in \text{Card}}{(B, n..m) \in RG}$$

Definition (Syntax of Quantified Role Groups)

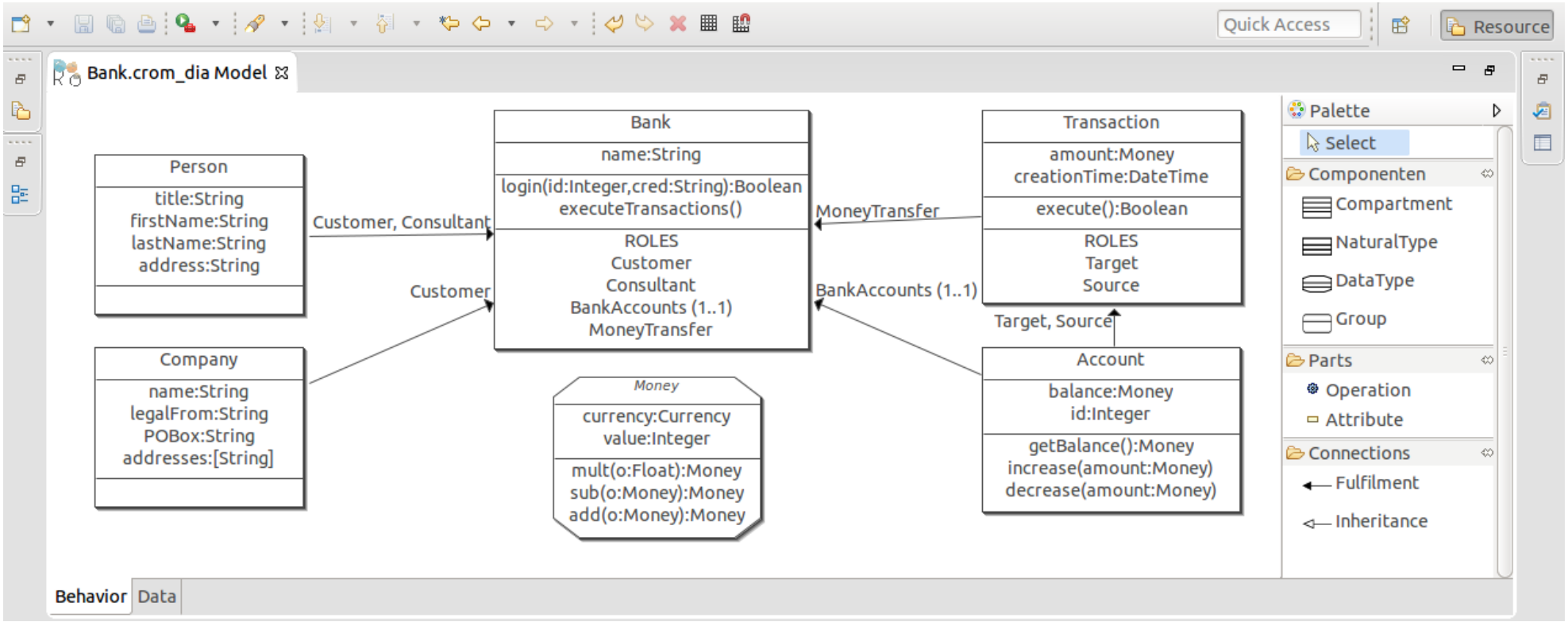
Quantified Role Groups QRG are defined inductively over RT, CT and RG:

$$\frac{a \in RG \quad ct \in CT \quad n..m \in \text{Card}}{\mathbb{Q}\langle ct, n..m\rangle.a \in QRG} \qquad \frac{B \subseteq QRG \quad n..m \in \text{Card}}{\langle B, n..m\rangle \in QRG}$$

[1] $\mathbb{E}$ is the set of functions $e: 2^O \times 2^O \times 2^{O \times O} \to \{0, 1\}$.

The Compartment Role Object Model (CROM)

Tool Support



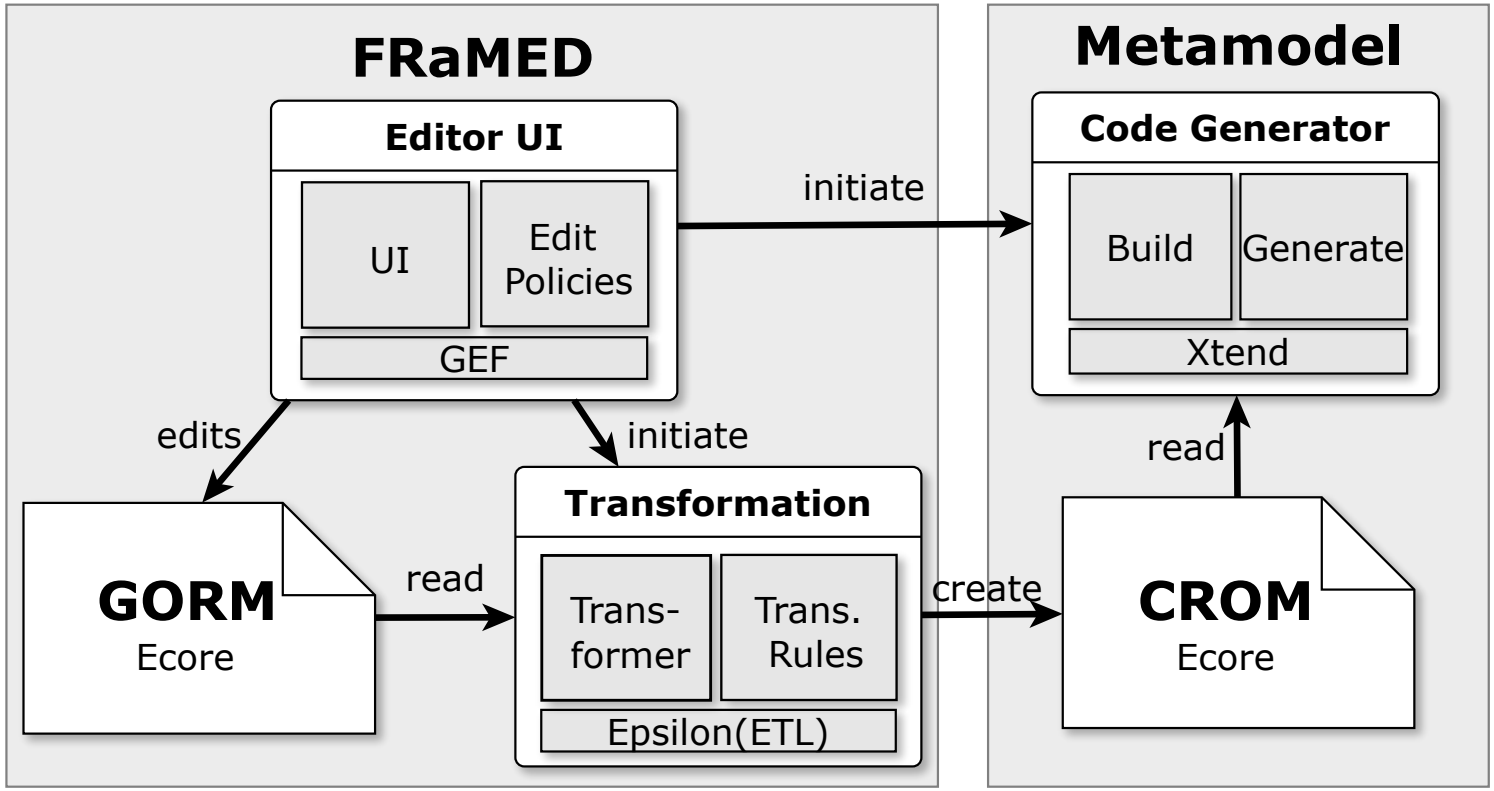
Full-fledged Role Modeling Editor (FRaMED)[3]

- 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]

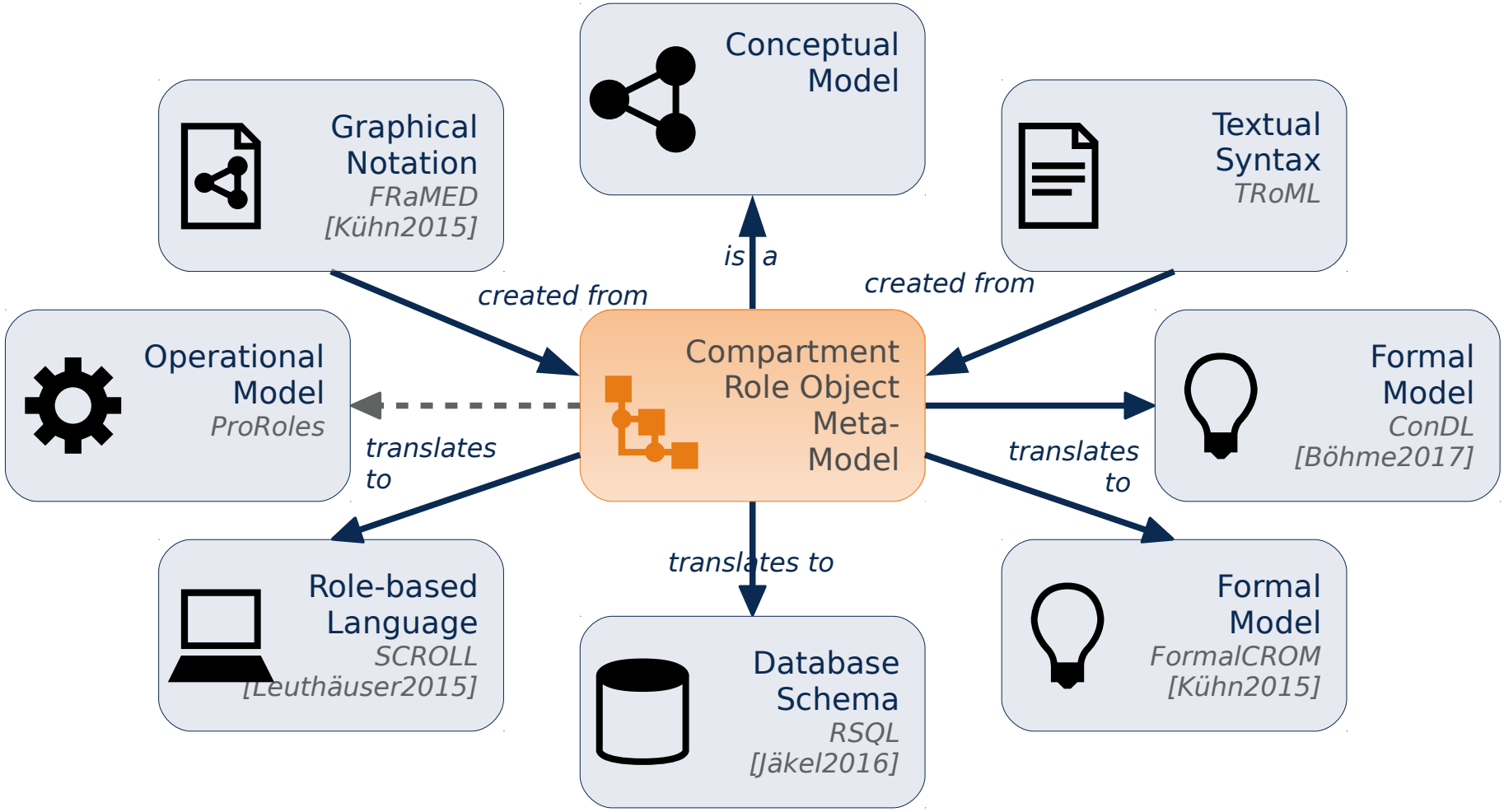
- Fully model-driven Eclispe-based editor based on:
  - *Eclipse Modeling Framework* (EMF), *Graphical Editing Framework* (GEF), *Epsilon* (ETL)
- Separation of *Graphical Model* (GORM) and *Semantic Model* (CROM)

3) https://github.com/leondart/FRaMED

The Compartment Role Object Model (CROM)

Tool Support

Additional tools supported by FRaMED



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

The End

- 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?