Summer School in Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH (LZI)

https://www.dagstuhl.de/en/program/calendar/evhp/?semnr=19383

Development, Deployment, and Runtime of Context-Aware Software Systems

„Role-based Software Infrastructures (RoSI), funded by DFG
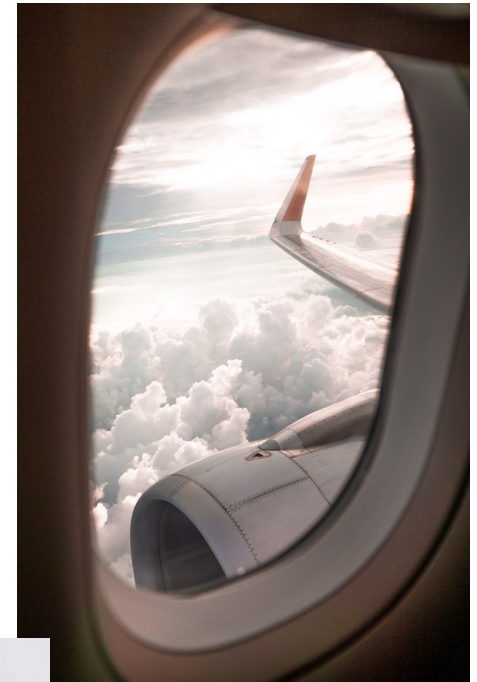https://rosi-project.org

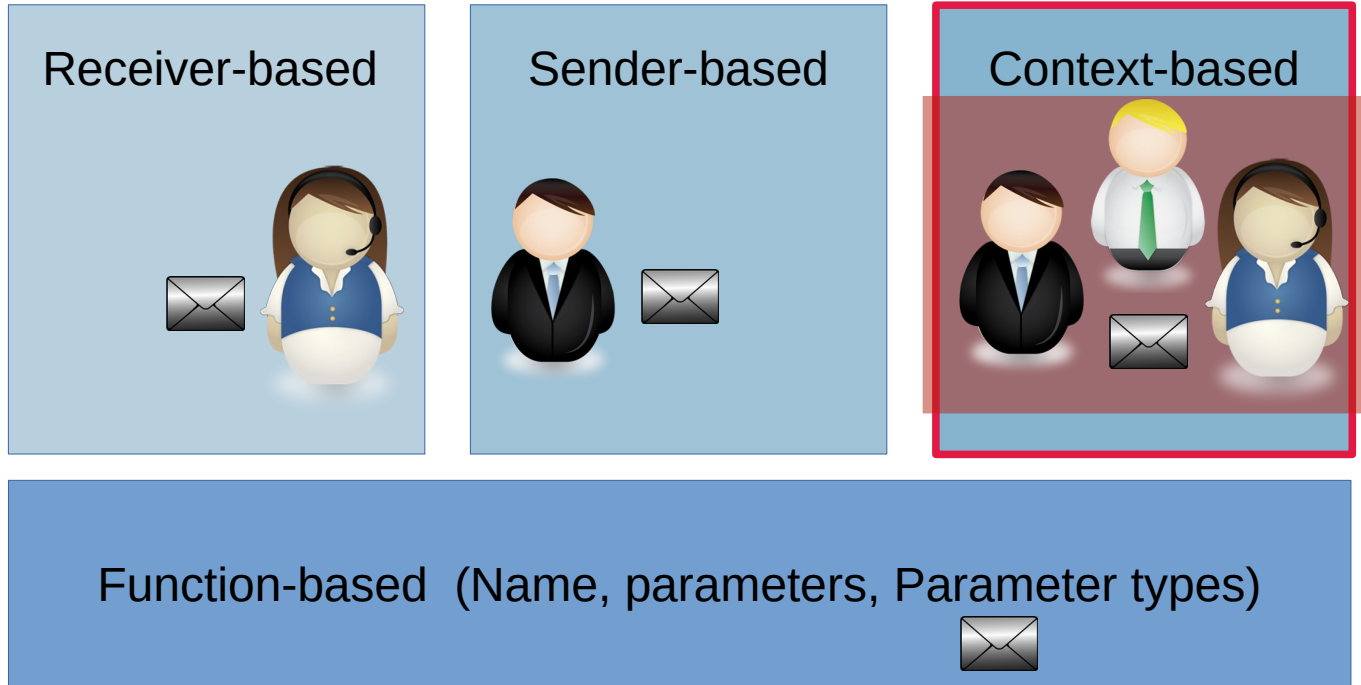Wolfgang Lehner, Uwe Aßmann

Sept 16-21 2019

1.1, After talk

Welcome to RoSI Summer School

- „ever-changing contexts"
  - Mobility
  - Personalization
  - Resource availablity
- How to realize
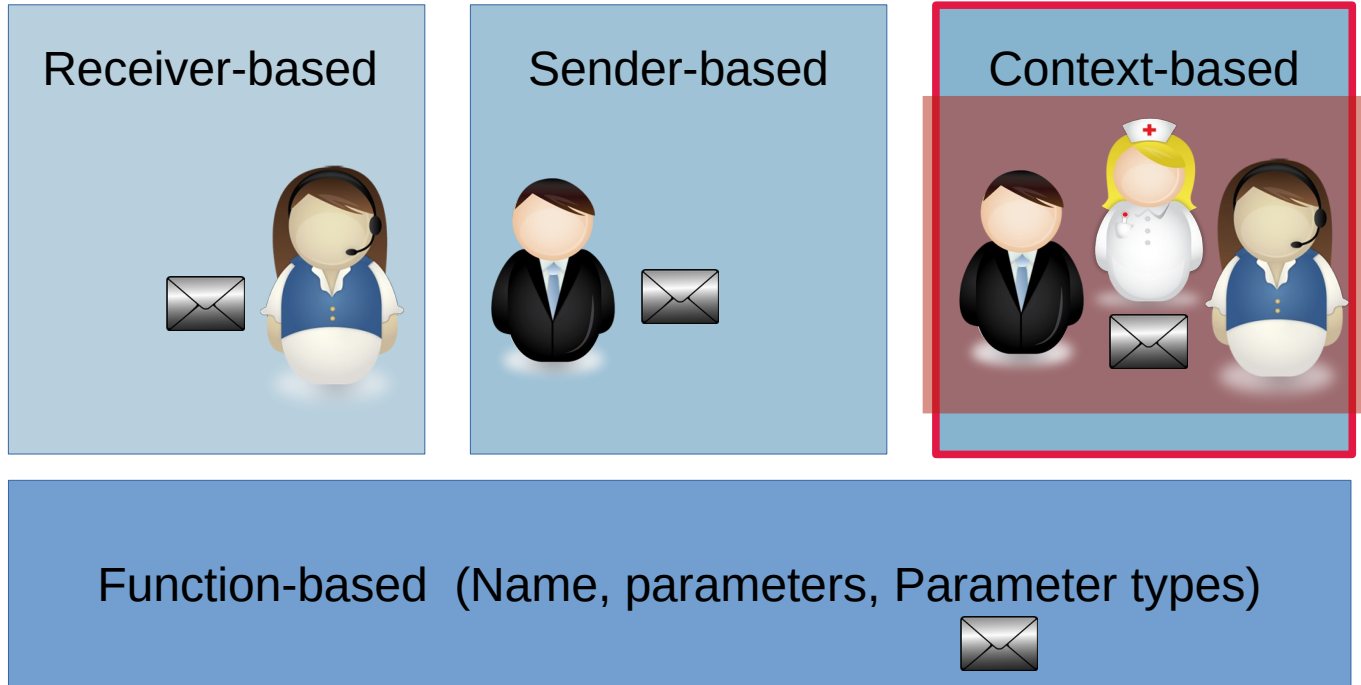 - Adaptation to change of context?
  - Context polymorphism?

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

2 DRESDEN concept

Multi-Dimensional Dispatch for Multi-Polymorphism

- How is the semantics of a function (method) determined?

Receiver-based

Sender-based

Context-based

Function-based (Name, parameters, Parameter types)

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

3 DRESDEN
concept

Multi-Dimensional Dispatch for Multi-Polymorphism

- How is the semantics of a function (method) determined?

Receiver-based

Sender-based

Context-based

Function-based (Name, parameters, Parameter types)

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

4 DRESDEN concept

New Application Areas of ROSI

- Roles for context-sensitive cyber-physical systems (CPS)

    - Hypothesis: Role-contracts for safety and security

- Roles for emergence in Systems-of-Systems (SoS)
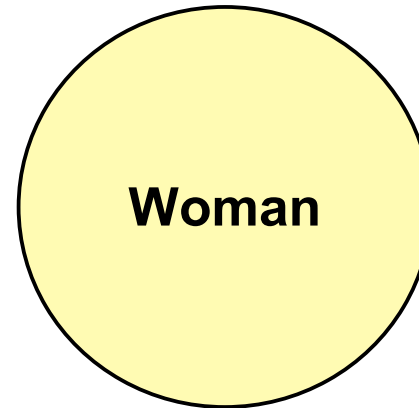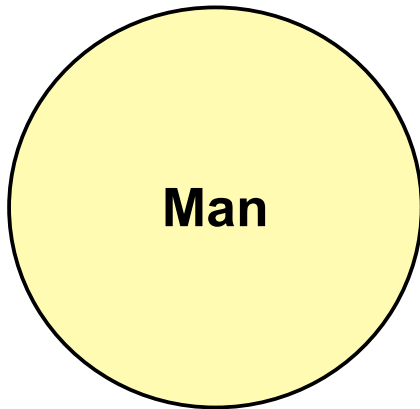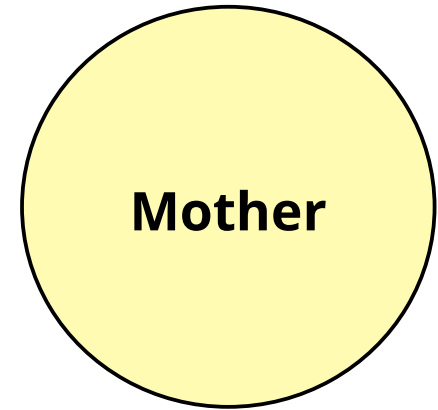
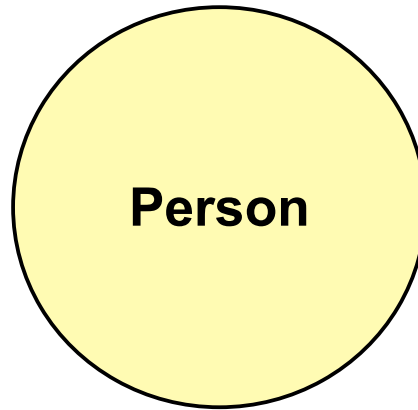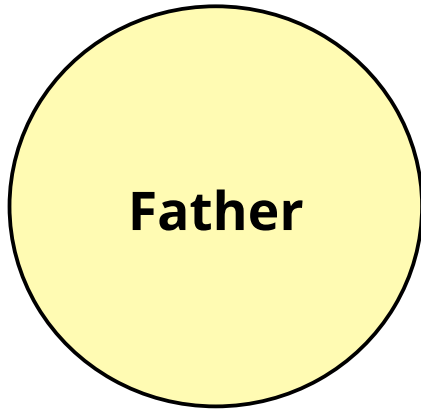   - Hypothesis: Role models for unforeseen emergence

http://commons.wikimedia.org/wiki/File:Traffic_seen_from_top_of_Arc_de_Triomphe.JPG

http://commons.wikimedia.org/wiki/File:Jona_(SG)_-_L%C3%A4ttenhofweg_2011-04-08_14-54-48_ShiftN.jpg

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

5

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

A Riddle..

Father

Person

Mother

Man

Woman

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

6

Another Riddle..

Person

Marriage

Husband

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann
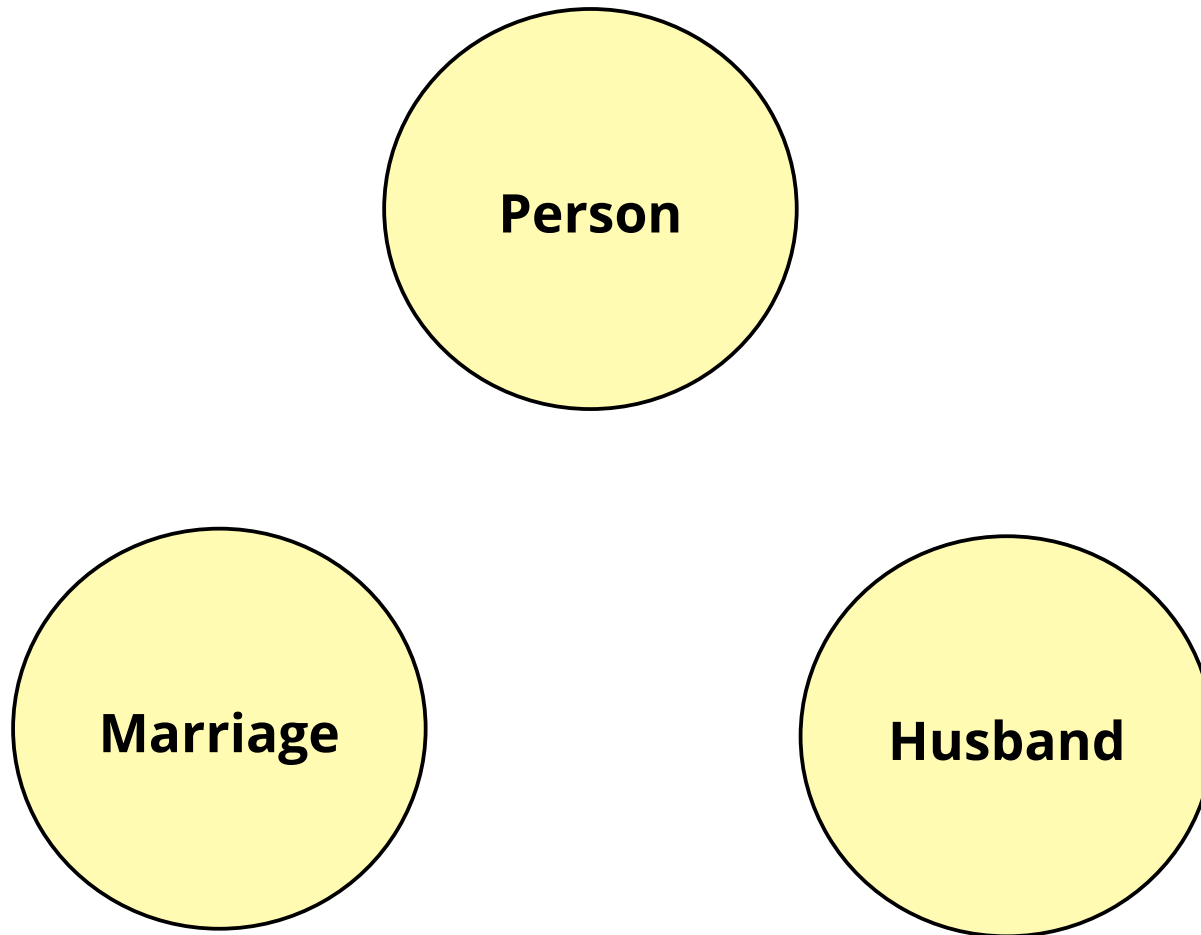
Program

	Summer School venue: Schloss Dagstuhl			Summer School venue: Dresden		
	Mon, 16. Sept	Tue, 17. Sept	Wed, 18. Sept	Thur, 19. Sept	Fri, 20. Sept	Sat, 21. Sept
08:00 / 08:30	Breakfast	Breakfast	Breakfast	Breakfast	Breakfast	Breakfast
09:00 / 09:30 / 10:00	Uwe Assmann, TU Dresden	Christian S. Jensen, Aalborg Universitet	Stefan Marr, University of Kent	Introduction to Role-based Modeling with CROM	Introduction to Role-oriented Programming with SCROLL	Departure from Dresden
10:30	Break	Break	Break	Break	Break	
11:00 / 11:30 / 12:00	Ralf Laemmel, Facebook London/Uni Koblenz-Landau	Christian S. Jensen	Stefan Marr	Designing Reusable Design Patterns with CROM	Developing Context-aware Application with SCROLL	
12:30 / 13:00	Lunch	Lunch	Lunch	Lunch	Lunch	
13:30 / 14:00	Poster-Session	Poster-Session	Transport to Dresden	Introduction to FRaMED 2.0	Role-based Software Engineering	
14:30 / 15:00	Friedrich Steimann	Jonathan Goldstein				
				Break	Break	
15:30	Coffee-Break	Coffee-Break				
16:00 / 16:30 / 17:00 / 17:30	Friedrich Steimann, FernUni Hagen	Jonathan Goldstein, Microsoft Research, Redmond		Tailoring FRaMED to your needs.	Creating a context for your needs	
18:00 / 18:30	Dinner	Dinner		Dinner	Dinner	
19:00 / 19:30 / 20:00 / 20:30		Tutorial "Code Generation", Frank Tetzel, TU Dresden/ SAP		Social Event		

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

8

TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

CS Department in Dresden

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

9

OUTPUT Demo Day of the Department

Fakultät Informatik
Lehrstuhl Softwaretechnologie

Role-Oriented Context-Aware Software Infrastructures (ROSI)

Prof. Uwe Aßmann

16.09.19

Overview of Talk

1) Adaptation problems of the classic OO model

2) Beyond Objects

    1) From Objects to Roles and their Benefit for Separation of Concerns

    2) From Roles to Contexts

    3) The Steimann product-lattice factorization of types and its Kühn extension (Role-oriented Context-Aware Software Infrastructures, ROSI)

3) Advantages of the ROSI: Dynamic Data Adaptability (Extensibility, Variability)...

4) Roles and Contexts for Behavior Abstraction

5) Advantages of the ROSI: Dynamic Behavior Adaptability

6) Roles and their Benefit for Separation of Concerns

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

13 DRESDEN
concept

Role-Oriented Context-Aware Software Infrastructures (ROSI)
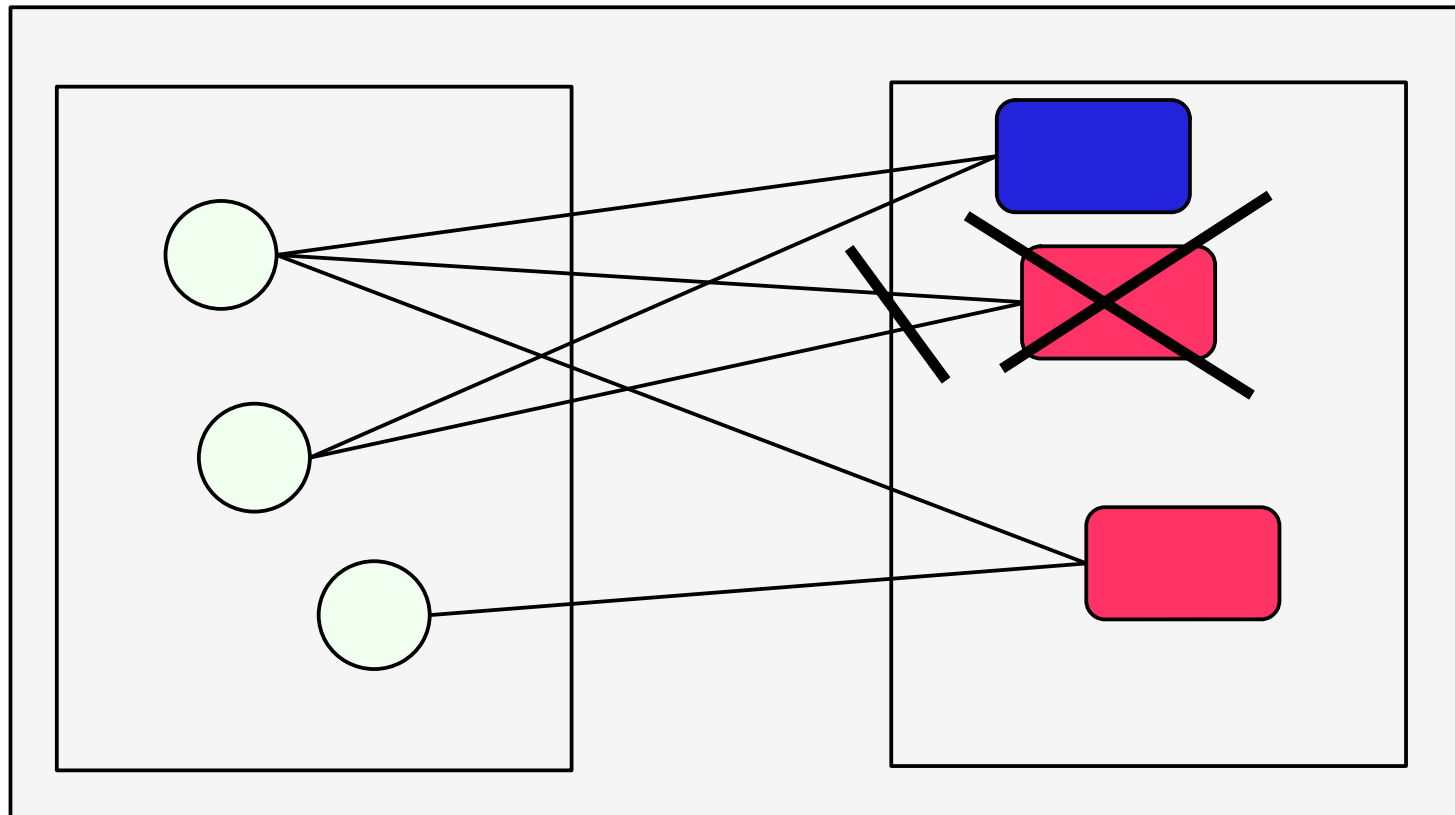
1. Adaptation Problems of the Standard Object Model

The Extensibility Problem

- How to extend software in a fine-grained way? Obliviousness?

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann
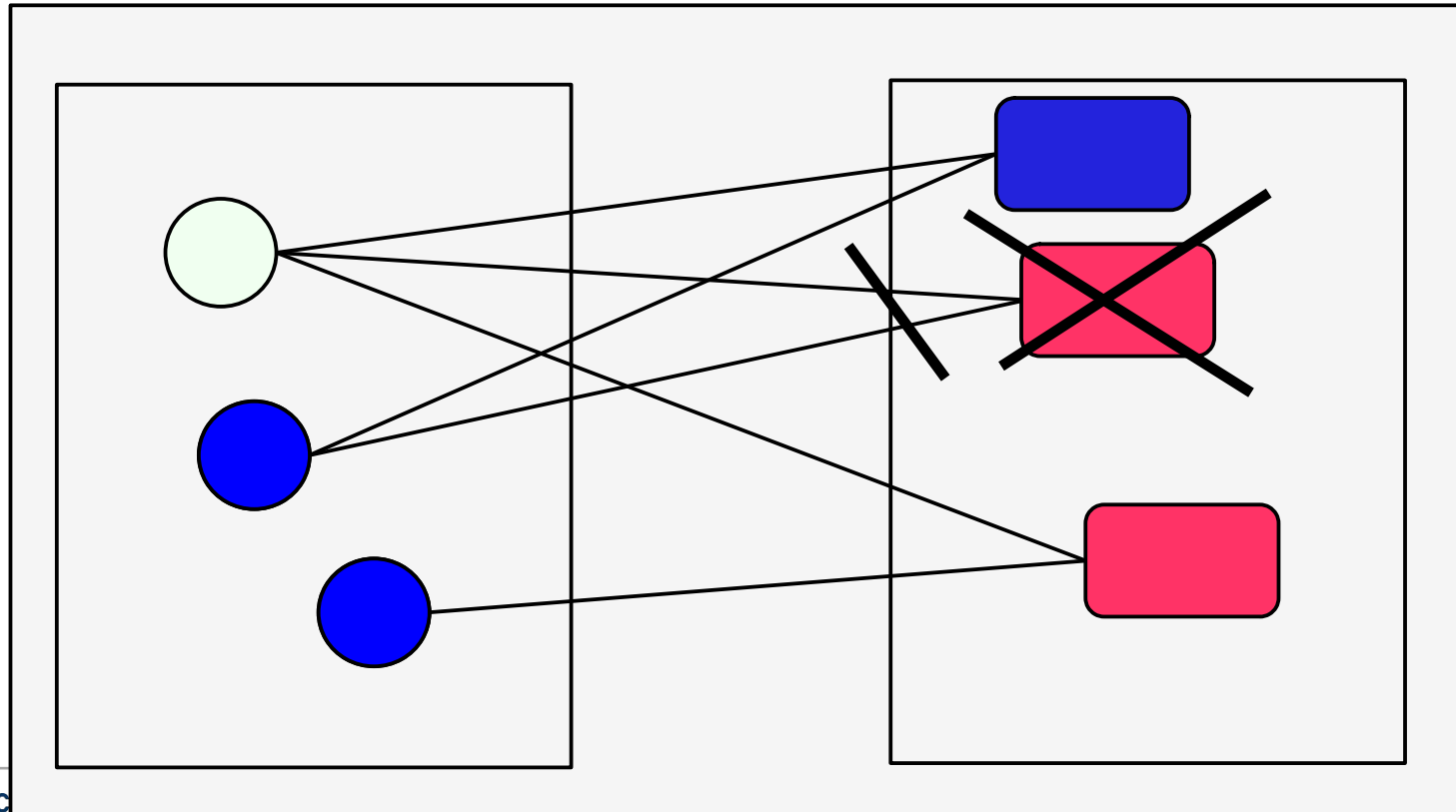
15
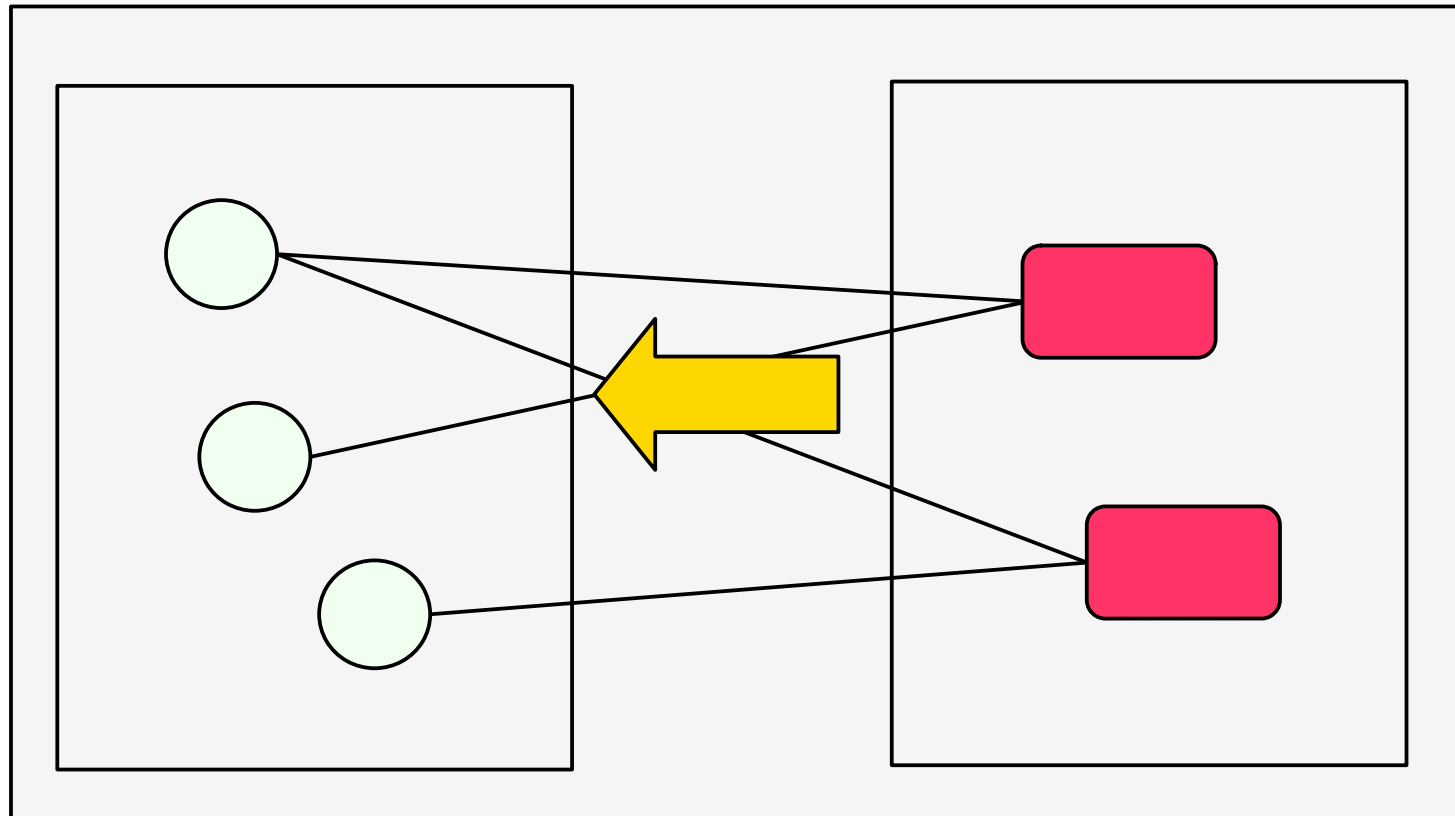
The Substitutability Problem

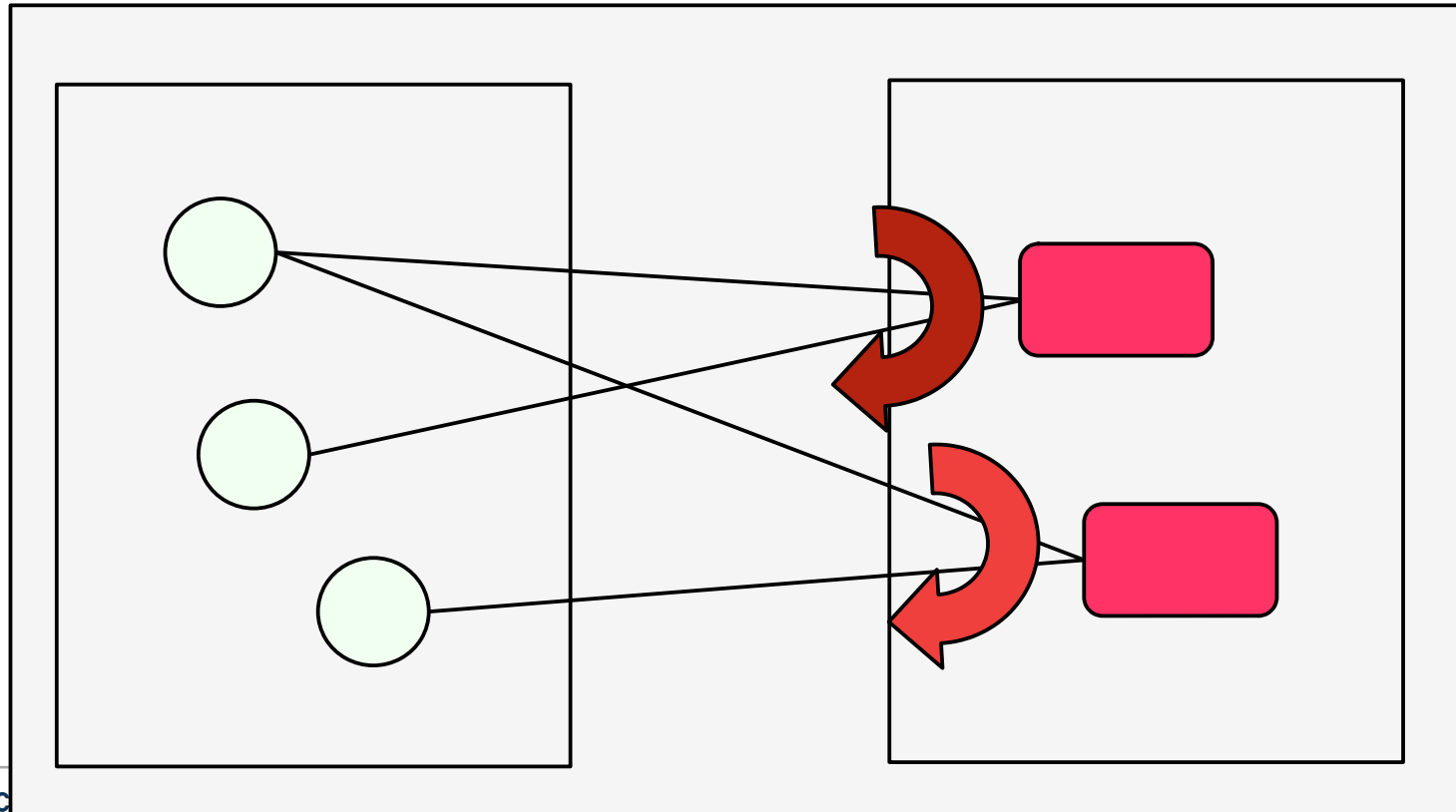- How to substitute a component? (contracts necessary)

The Variability Problem

- How to vary many components, layers, slices, or a subsystem?

The Wrapping and Synchronization Problem

- How to wrap software with code, e.g., for protection or synchronization, transactions?

Big Problem 1) The Synchronization Problem (Inheritance Anomaly)

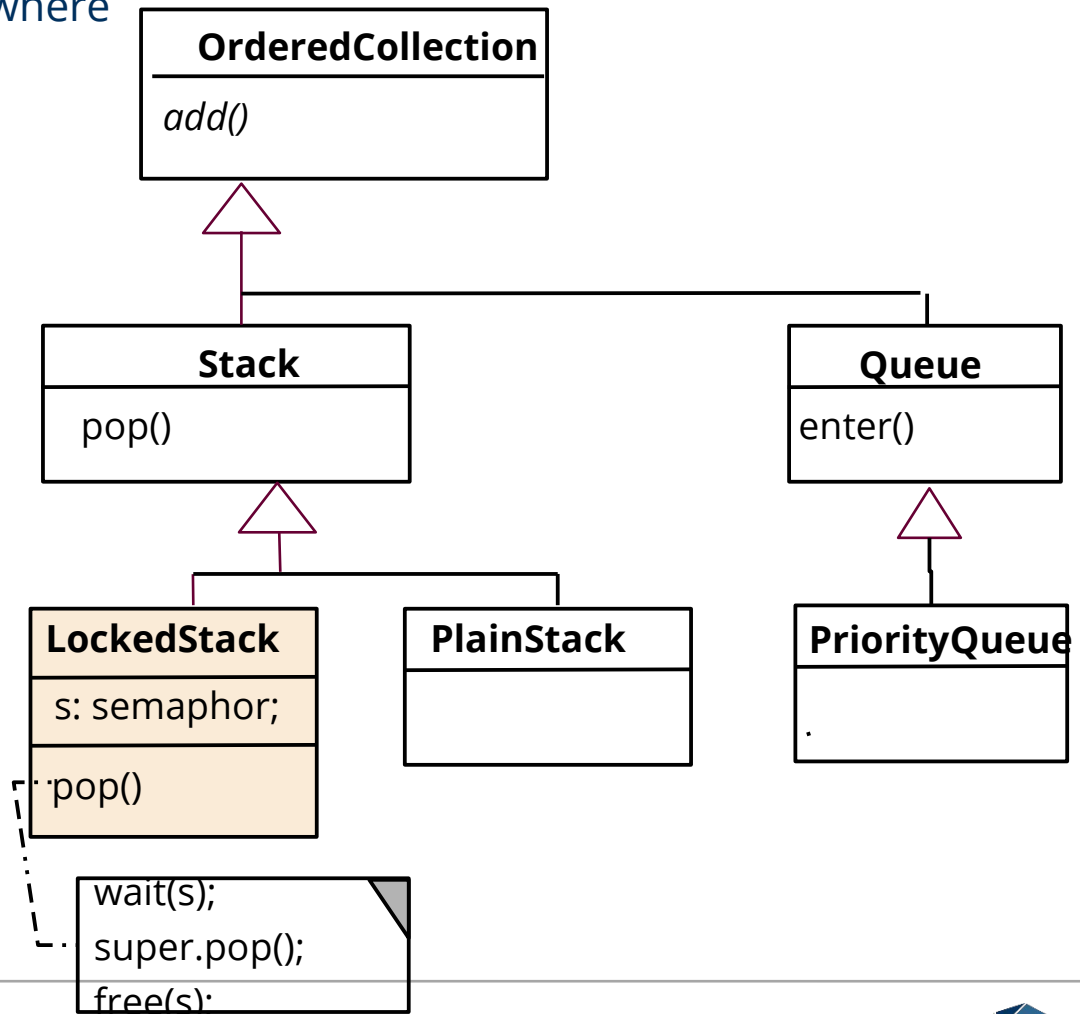1980s: Parallel object-oriented languages (POOL, COOL)

1991: Inheritance Anomaly

1988: Composition Technology (Aksit)

1994: Generic Synchronization Policies (McHale)

1988-94: Composition Filters (Aksit)

1996: Aspect-oriented Programming (AOP)

1996: D+COOL+RIDL (Lopes), 1999: Aspect/J (Kiczales)

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

19

Inheritance Anomaly - Why Concerns are Necessary

- In a parallel program or library, where should synchronization code be inserted?

  - Stack?

  - Queue?

  - OrderedCollection?

  - Collection?

  - Object?

OrderedCollection

add()

Stack

pop()

Queue

enter()

LockedStack

s: semaphor;

pop()

PlainStack

PriorityQueue

.

wait(s);
super.pop();
free(s);

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

20 DRESDEN
concept

TECHNISCHE
UNIVERSITÄT
DRESDEN

The Synchronization Problem (Inheritance Anomaly)

- At the beginning of the 90s, parallel object-oriented languages failed, due to the inheritance anomaly problem

  - *Inheritance anomaly*: In inheritance hierarchies, synchronization code is intermingled with the algorithm and cannot be easily exchanged

  - *Synchronization tangling*: Because synchronization code *braces* code, it is *tangling*

  - *Synchronization crosscut*: Because synchronization code *is reused* code, it is *crosscutting*

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

Algorithm and Synchronization are Two Different Concerns (Core and Aspect)

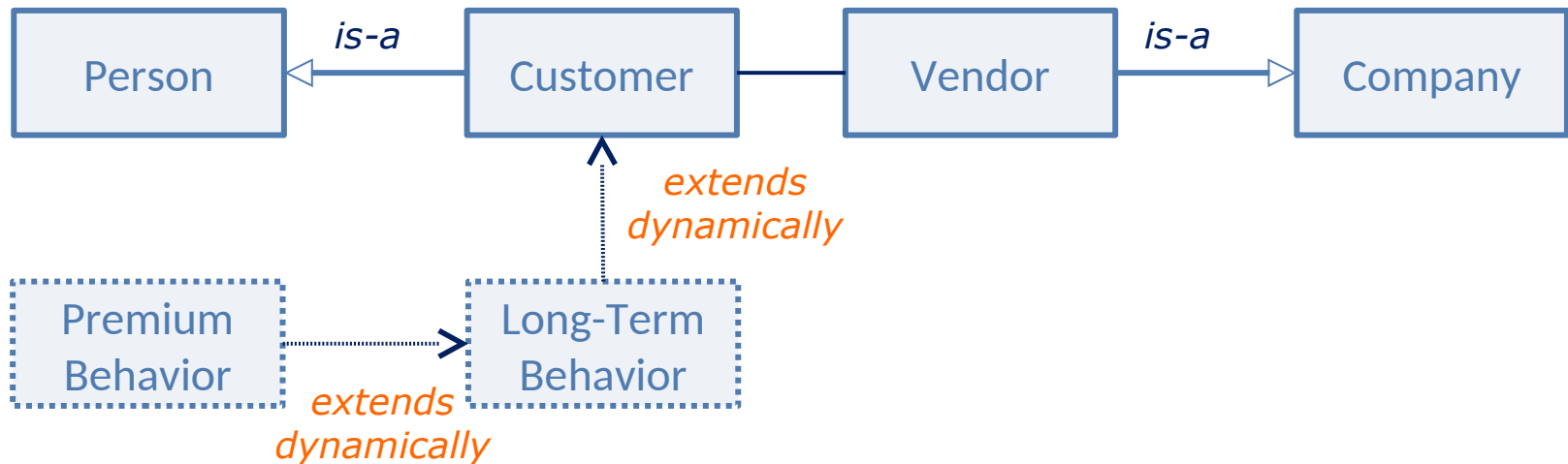- Composition fixes crosscut between core and aspect

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann
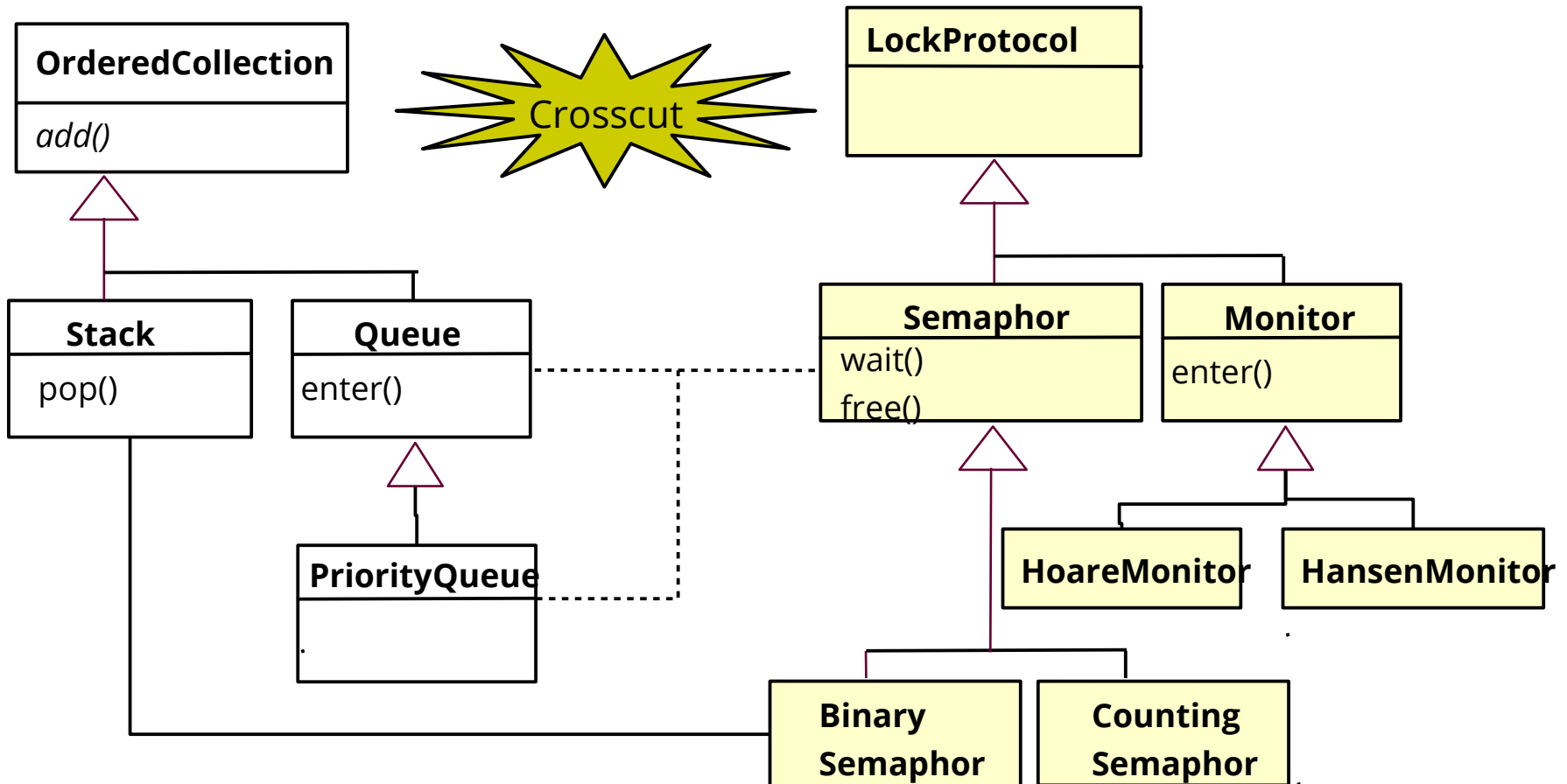
Problems of Aspect-Orientation

- Not well integrated into the standard OO model

- Semantics unclear

- Often only static

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

23   DRESDEN
      concept

TECHNISCHE
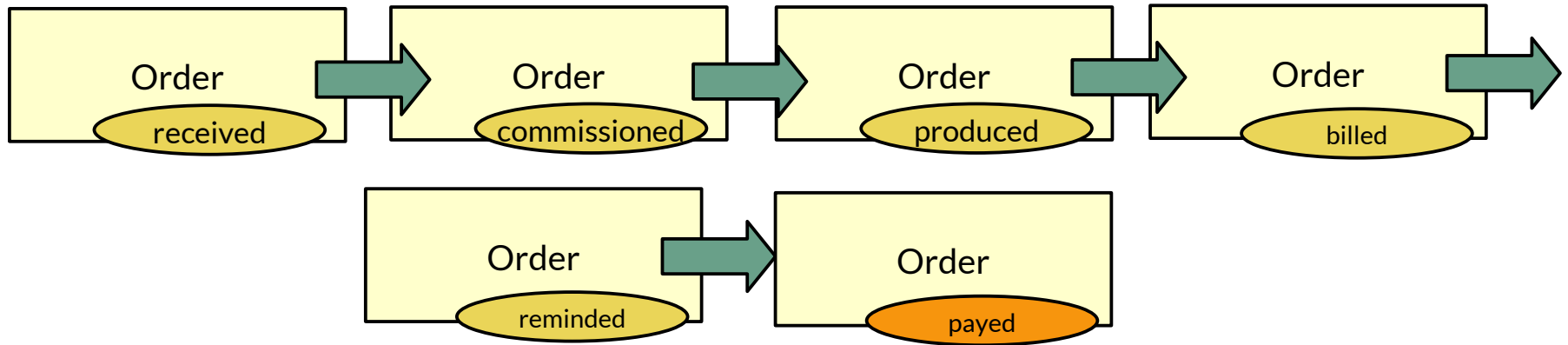UNIVERSITÄT
DRESDEN

Big Problem 2) Run-time Adaptability
Negative Example: "San Francisco"-Framework of IBM

- Enterprise Resource Planning (ERP) in Java, 1995-99
- Dynamic extensions of classes and life-cycle automata
- Classic object-orientation too inflexible
- FAILED

Role-Oriented Context-Aware Software Infrastructures (ROSI)
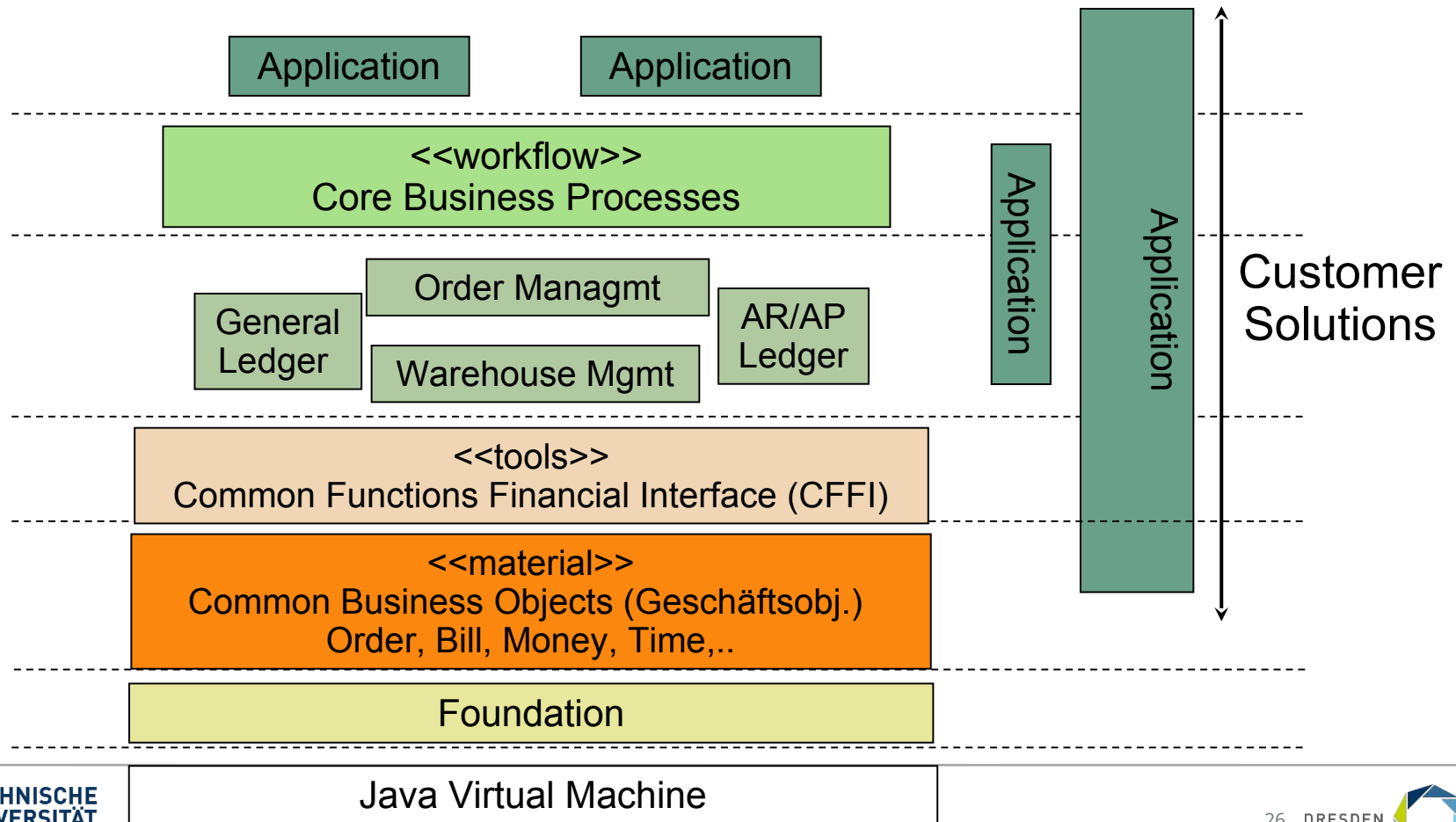Prof. Uwe Aßmann

24   DRESDEN
     concept

Business Objects

- In large ERP frameworks (see SAP) business objects get very complex

- Ex.: **Order**

  - Many phases and collaborators

  - Many states and roles

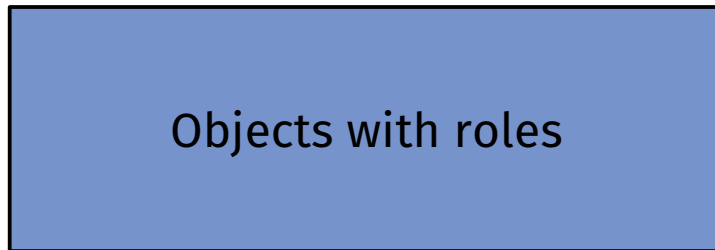- Dynamic Extensibility and Variability (Adaptation) required

Order
received → Order
commissioned → Order
produced → Order
billed →

Order
reminded → Order
payed

Architecture of IBM San Francisco ERP Java-Framework

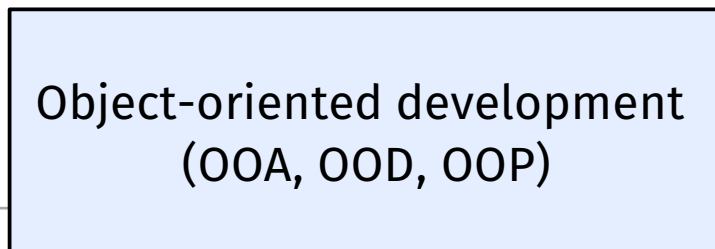- P. Monday, J. Carey, M. Dangler. SanFrancisco Component Framework: an introduction. Addison-Wesley, 2000.
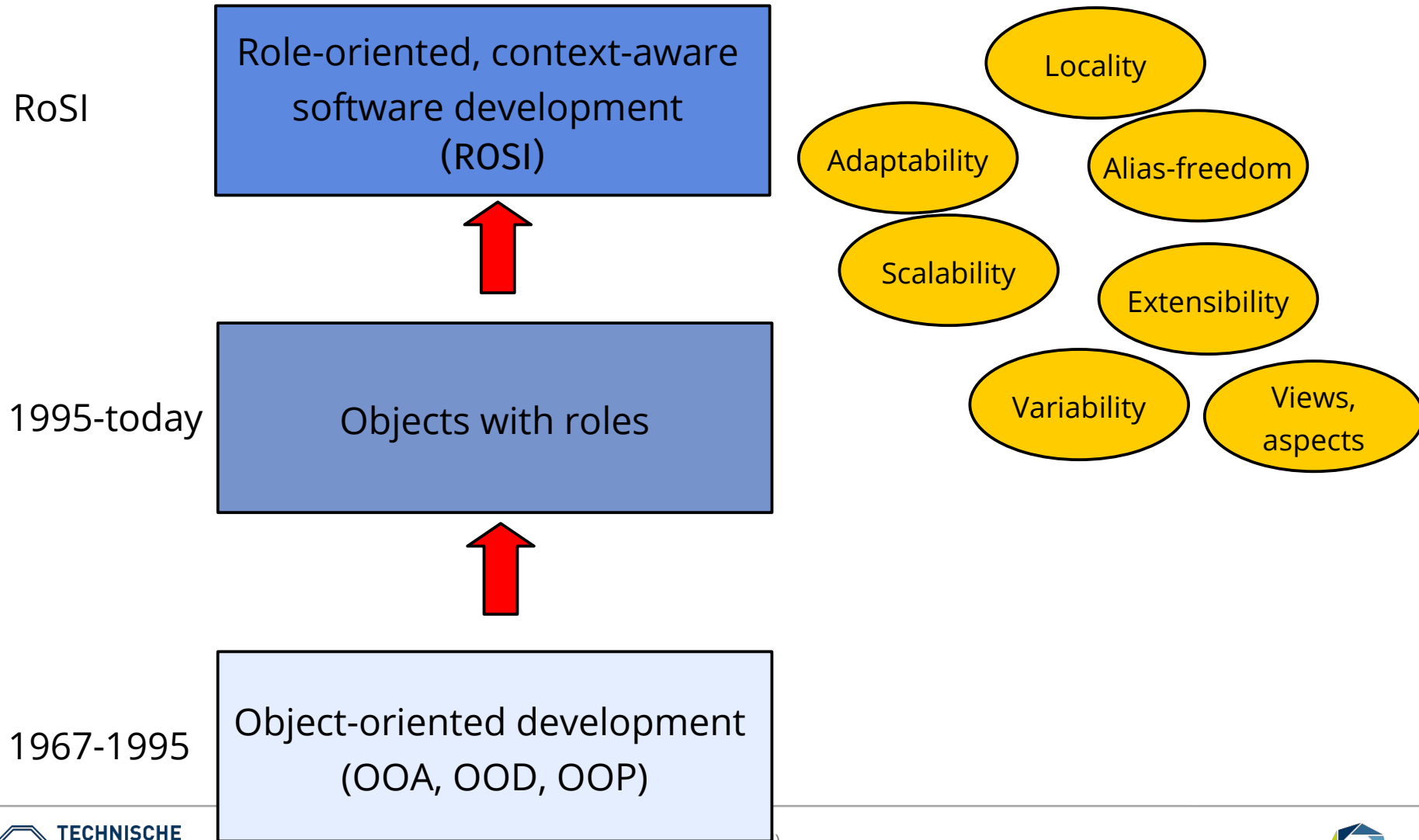
Ladder of Technologies

1995-today	Objects with roles
1967-1995	Object-oriented development (OOA, OOD, OOP)

Ladder of Technologies

RoSI	Role-oriented, context-aware software development (ROSI)
	↑
1995-today	Objects with roles
	↑
1967-1995	Object-oriented development (OOA, OOD, OOP)

- Locality
- Adaptability
- Alias-freedom
- Scalability
- Extensibility
- Variability
- Views, aspects

Role Modelling – a Hope

- Separate the **functional core** of an object of its **context-based a**nd **fluid** features

```
Person ●──────▷ [ Customer ]────[ Vendor ] ◁──────● Company
       plays-a                                plays-a
```
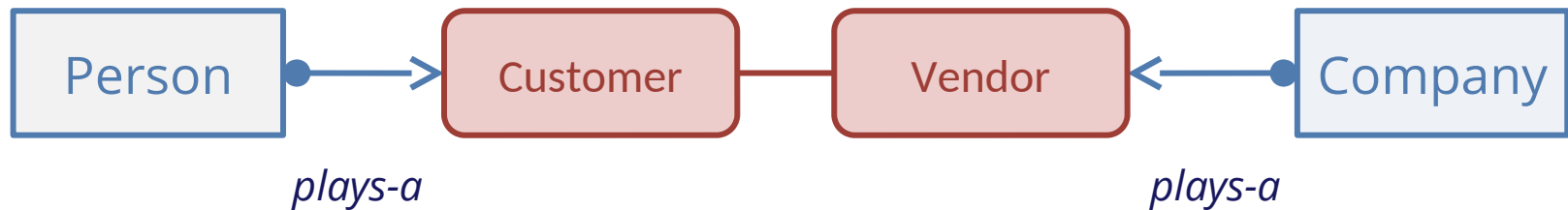
- Restrictions so far:
  - only used in singular fields of Computer Science
  - no cross-layer correspondance
  - no formalization

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

29 DRESDEN concept

TECHNISCHE UNIVERSITÄT DRESDEN

Example: Business Objects

- Extend behavior dynamically by **roles (context-based and fluid types)**

- Convention: Context is expressed by *background boxes* or *color*
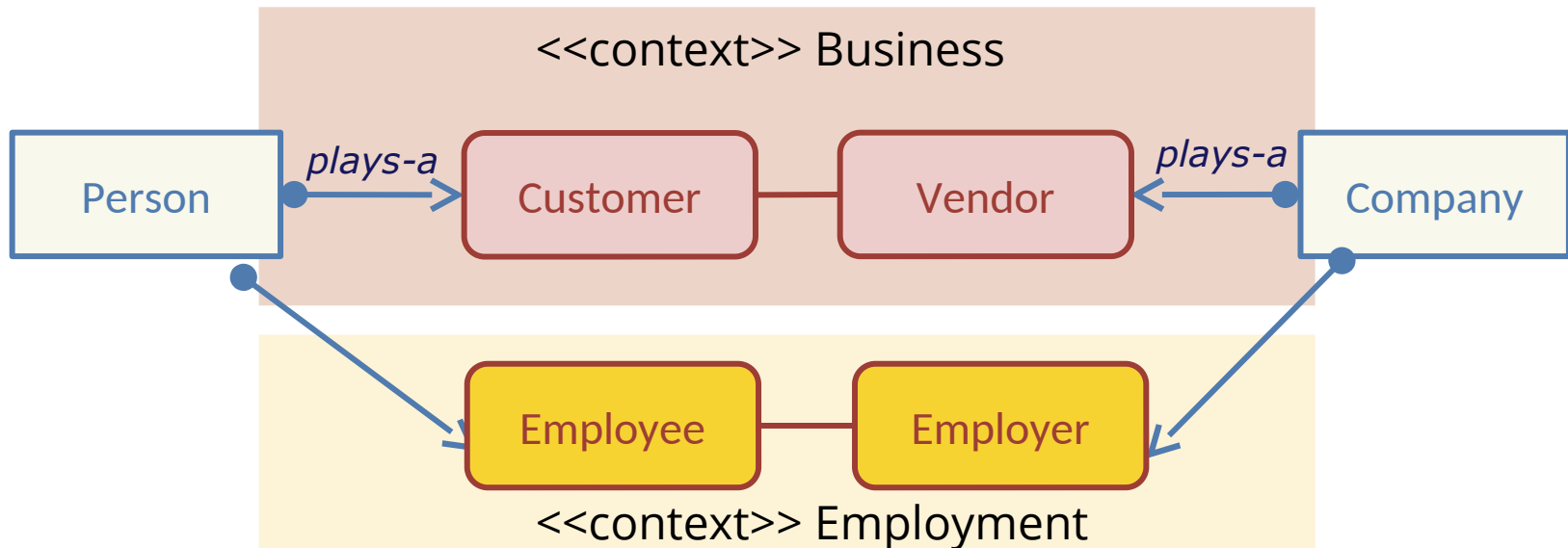
Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

30

Example: Business Objects

- Extend behavior dynamically by **roles (context-based and fluid types)**
- Refinement by role inheritance
-

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

31
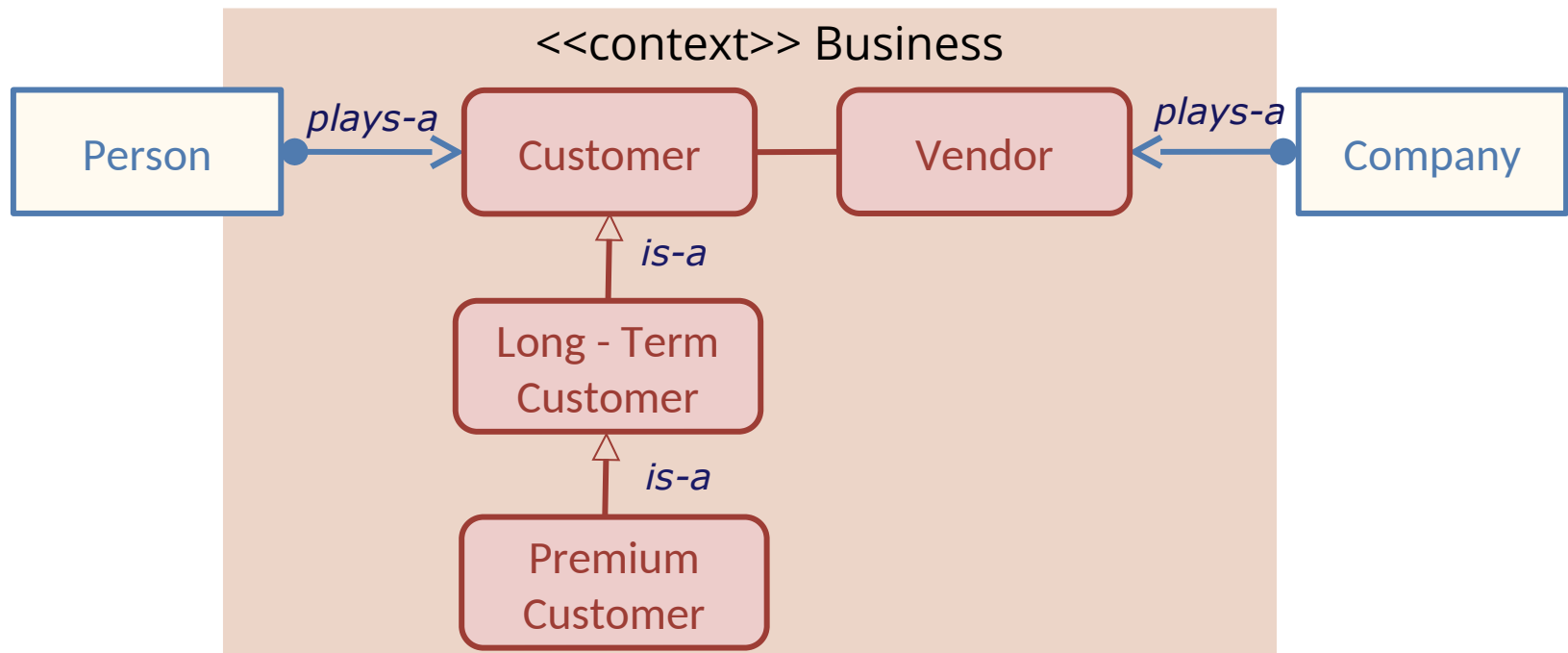
The Hypothesis of Role-Oriented Context-Aware Development

- ...is that **context-based features of objects and systems** can be modeled **with roles, cross-cutting**

  - all phases of the life-cycle
    - requirements, design, implementation, runtime
  - all levels of development
    - Concept modelling in metalanguages,
    - Language modelling,
    - Application modelling and programming,
    - Run-time

- and that this technology is **practically applicable.**

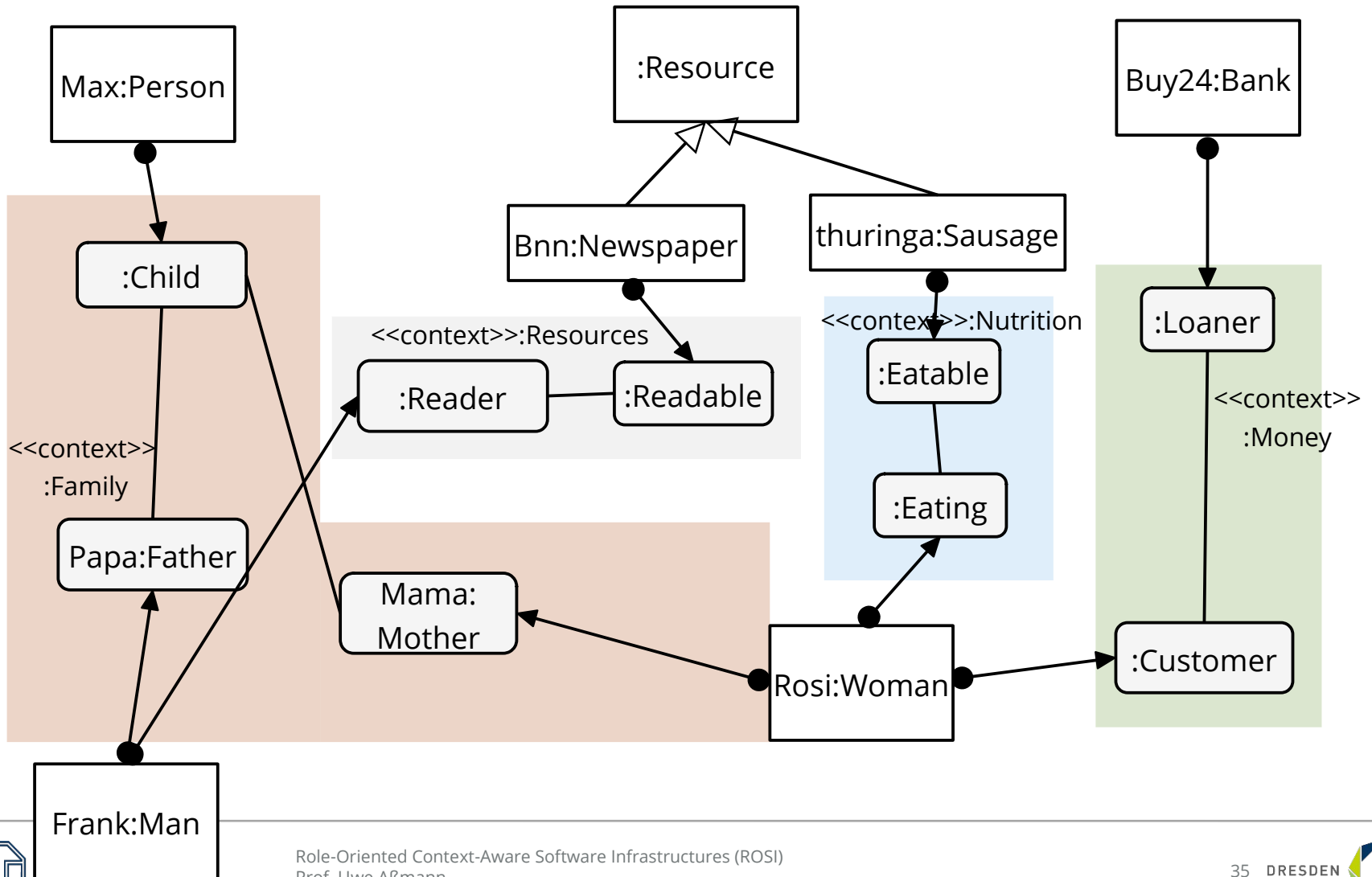Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

32   DRESDEN
     concept

Role-Oriented Context-Aware Software Infrastructures (ROSI)

1.2. Scenario Families and Banks

Complex Objects

A **complex object (subject, compound object)**
is a (logically coherent) object,
represented in modeling and programming level by
one *Core* and several *Subobjects (mixins)*
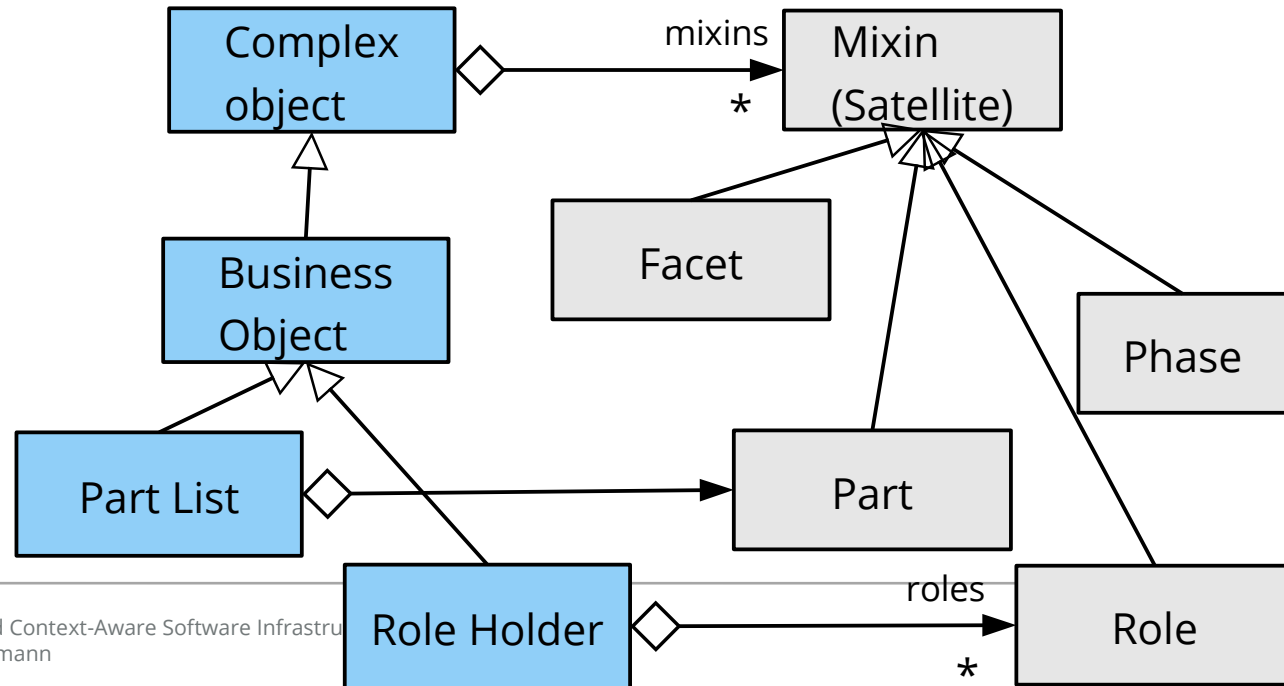
Families, Resources and Banks (Snapshot, Object-Role Model)

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

35

Families and Banks in Natural and Role Types

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann
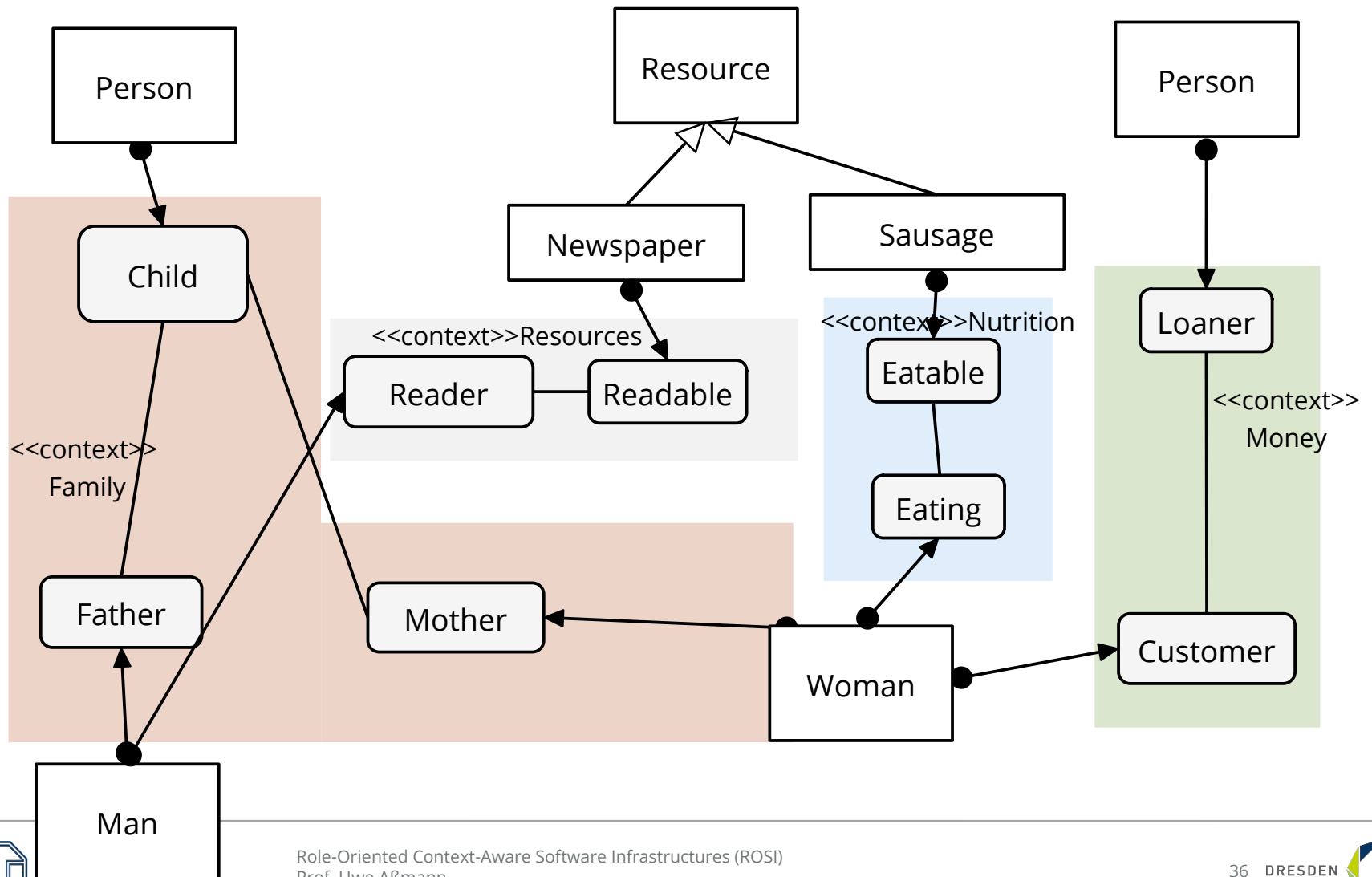
Role-Oriented Context-Aware Software Infrastructures (ROSI)

2. Beyond Objects -
Role Modeling and the Steimann Factorization of Types

Splitting a type into a tuple of natural and founded parts

Roles in the Literature

- Databases (Bachmann 77)

- ER model (Chen 76); though hidden in association ends

- OO Modeling (Reenskaug 95)

    – Design patterns (Riehle 98)

    – Course "Design patterns and frameworks" at TUD

- Product line engineering (Smaragdakis,Batory 02)

- Connectors in architectural languages (Garlan, Shaw 95)

- Security: Role-based Access Control (RBAC)

    – ACL lists in operating systems

- Ontologies (Brachman, description logic)

- … [Steimann DKE 2000] has many more and tries to unify them

- UML has "collaborations" using role types

- [Kühn 2014] defines compartments as structured context objects

TECHNISCHE
UNIVERSITÄT
DRESDEN

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

38   DRESDEN
concept

Rigid and Founded Types

> If an object that has a **rigid** type, it cannot stop being of the type without loosing its identity [Guarino]

- Example:
    - *Book* is a rigid type, *Reader* is a non-rigid type
        - Reader can stop reading, but Book stays Book

- Rigid types are *tied to the identity* of objects
    - A *non-rigid type* is a dynamic type that is indicating a state of the object

> A **founded type** *(relative type)* is a type that exists always in collaboration (association) with another class.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

TECHNISCHE UNIVERSITÄT DRESDEN
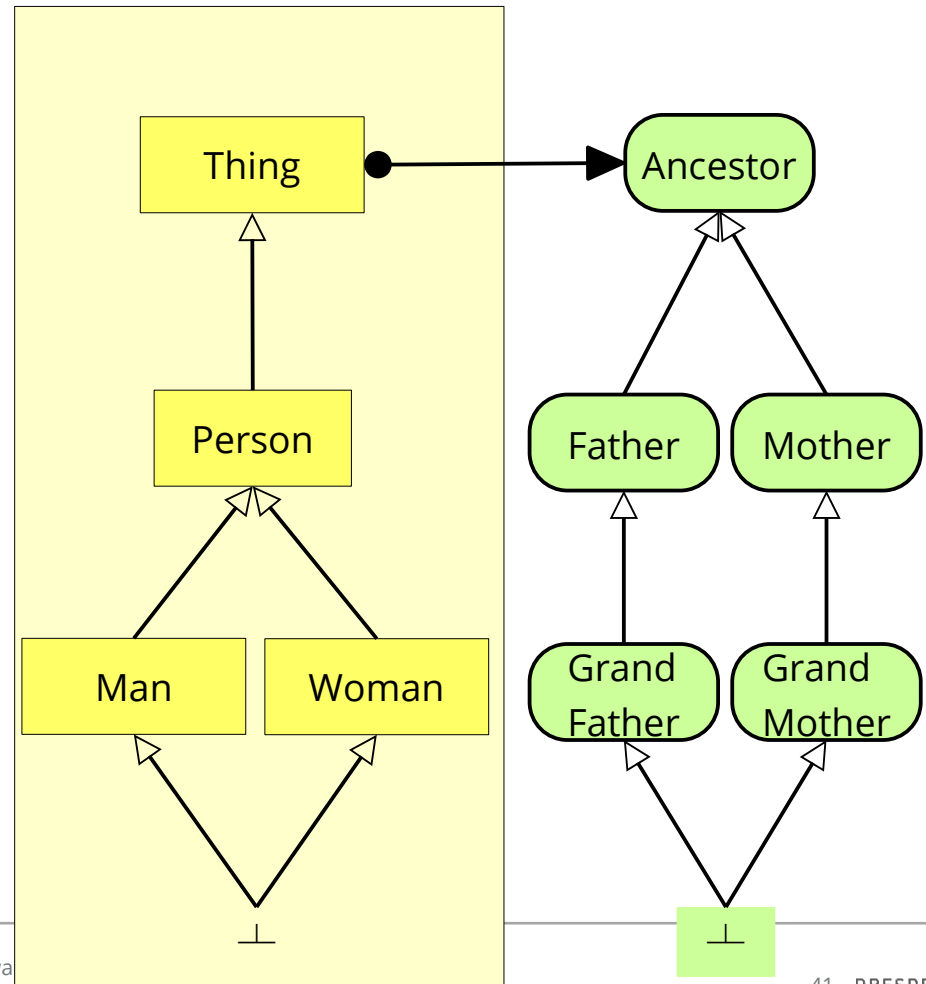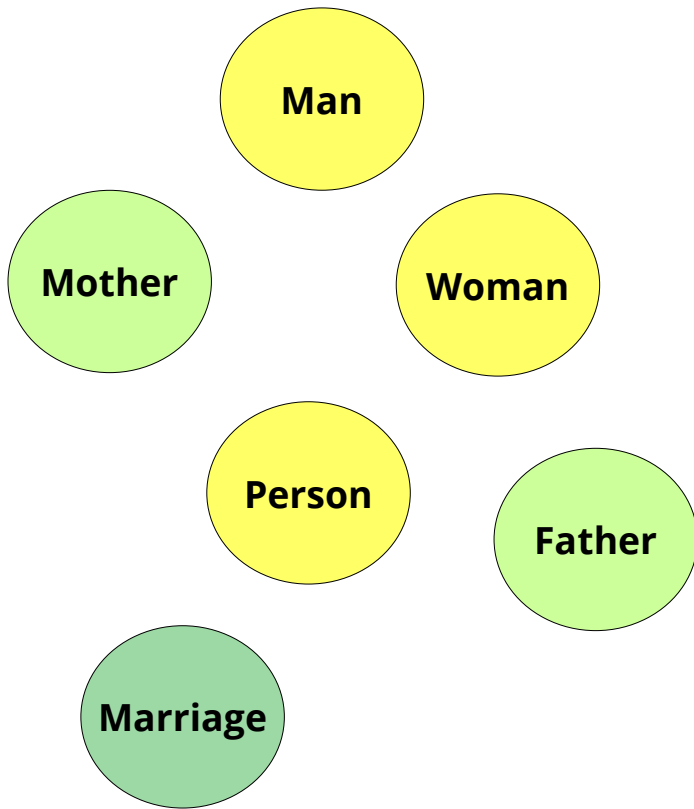
39 DRESDEN concept

Role and Natural Types

> A *role type* is a founded and non-rigid type.

Role types are in collaboration and if the object does no longer play the role type, it does not give up identity.

> A *natural type is* non-founded and rigid.

A natural type is *independent* of a relationship.
The objects cannot leave it.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

40   DRESDEN
      concept

TECHNISCHE
UNIVERSITÄT
DRESDEN

Solution to the Little Riddles..

Role-Oriented Context-Aware Softwa...
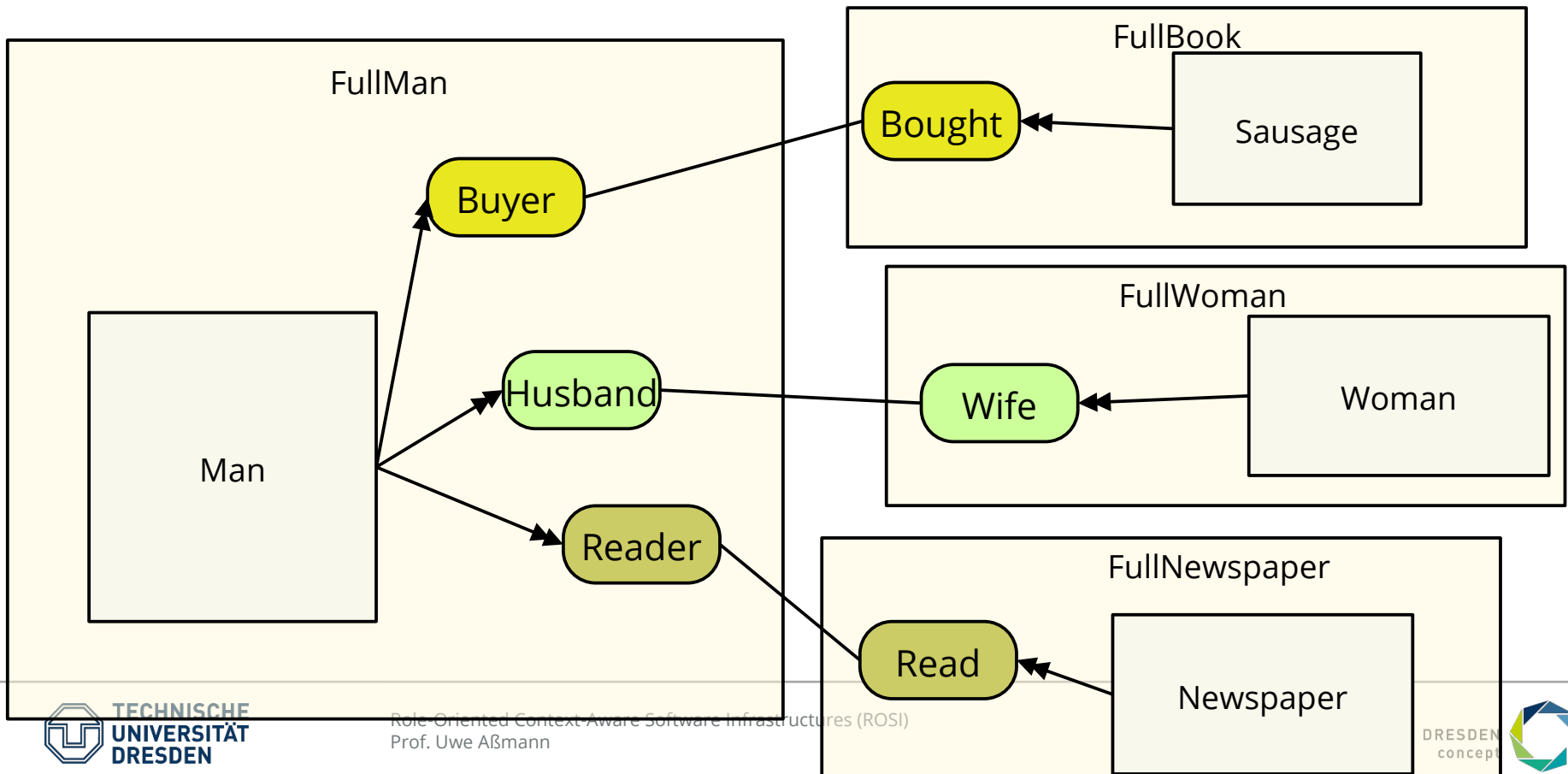Prof. Uwe Aßmann

41

Role Types are Metatypes

- A **metatype** describes a type (is a type of a type)

    - Rigid Type

    - Natural Type

    - Founded Type

    - Role Type

Hypothesis:
The distinction of metatypes promotes
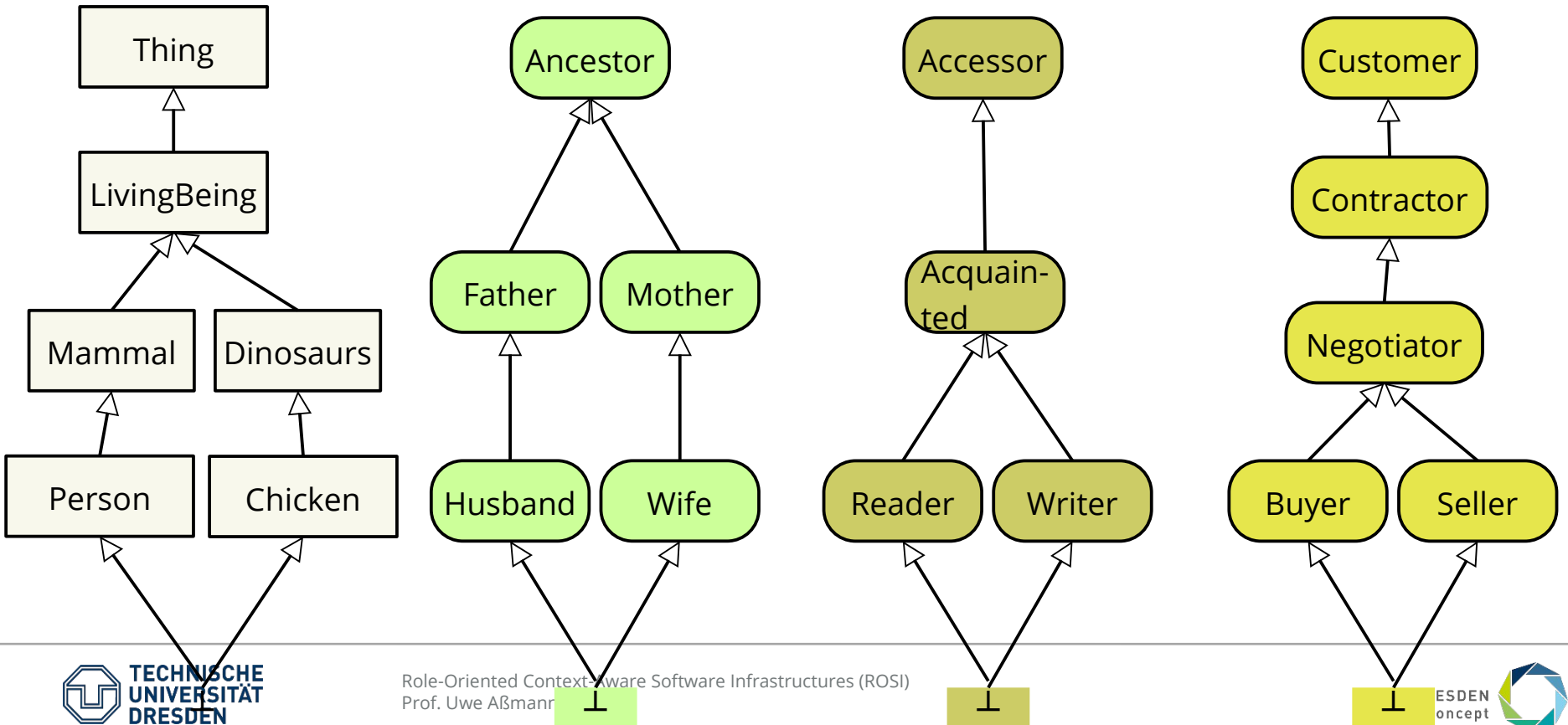Separations of Concerns.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
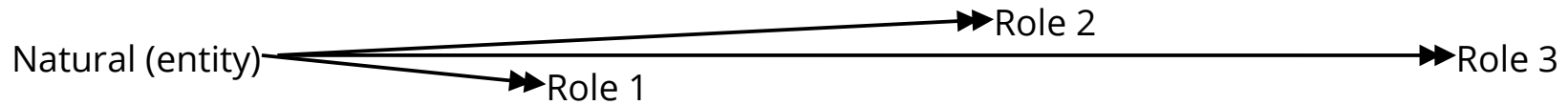Prof. Uwe Aßmann

42

Steimann Factorization [Steimann, DKE 2000]

- Splitting a full type into its *natural* and *role-type* components

  - FullType = Natural x (role-type, role-type, …)

  - FullMan = Man x (Reader, Husband, Customer, ..)

Full Type is from an Inheritance Product Lattice

- What is a reading buying husband person?

Natural (entity) → Role 1, Role 2, Role 3

Thing	Ancestor	Accessor	Customer
LivingBeing	Father / Mother	Acquainted	Contractor
Mammal / Dinosaurs	Husband / Wife	Reader / Writer	Negotiator
Person / Chicken			Buyer / Seller

TECHNISCHE UNIVERSITÄT DRESDEN

ESDEN oncept

The Steimann Factorization

- Simpler, multi-dimensional inheritance hierarchies (product lattice)

Divide (partition) a *type* into a *tuple type* over a product lattice of a core dimension and n-1 role dimensions(Core, Role_1, ..., Role_n)

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

45

Concern-Separated Representation of Object Nets

- Collaborations (Role models) are interprocedural slices and belong to contexts

- Collaboration schemas are schemas for interprocedural slices
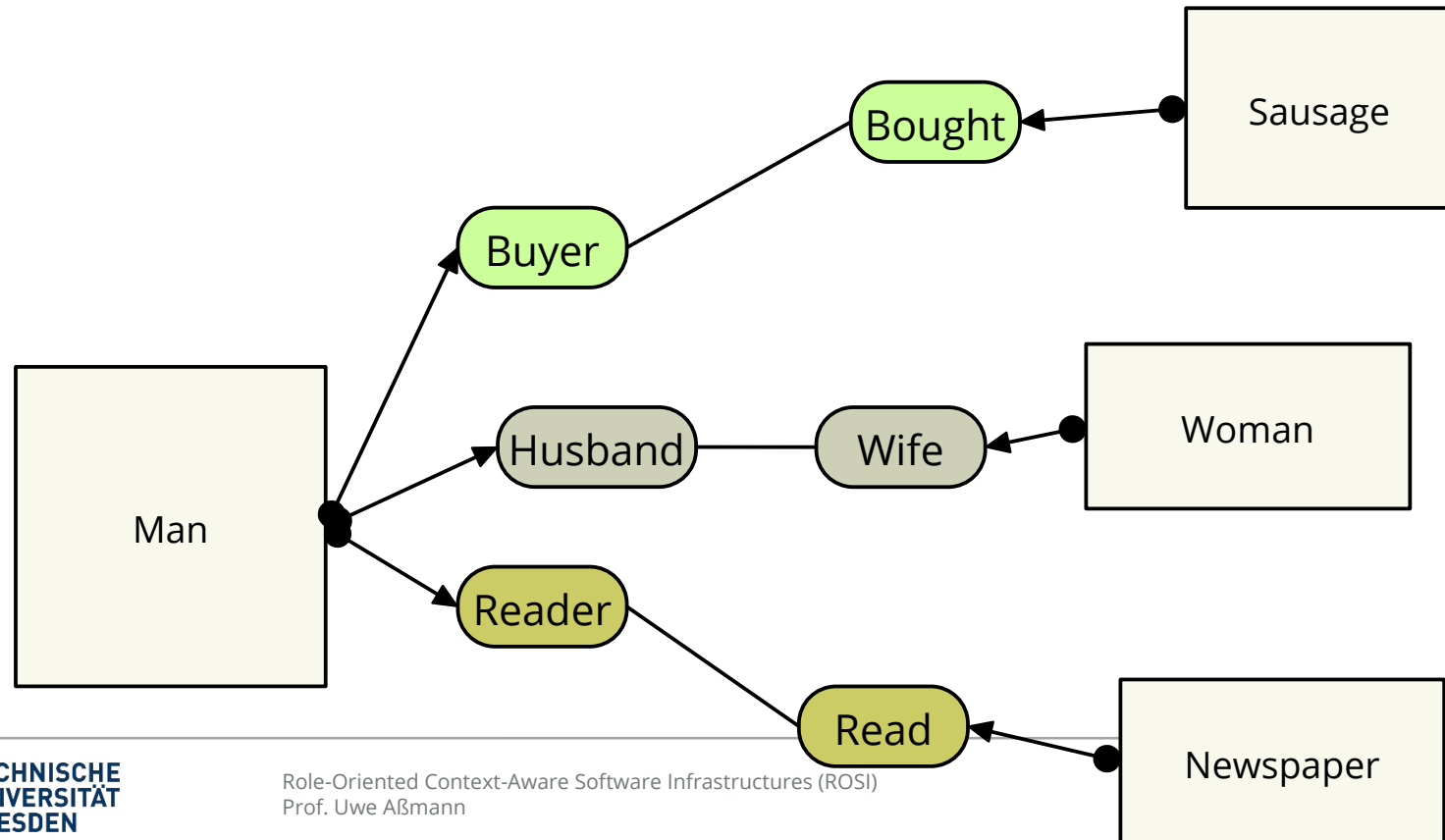
ROSI – Role-Oriented Context-Aware Software Infrastructures

2.1 Contexts and Compartments

[Kühn 2014]

How to Model Contexts

- A ***context*** is an object reifying contextual conditions, *activating* and *deactivating* a set of roles of a set of objects
  - Contexts show that contextual conditions hold
  - *Marriage* (enables Husband and Wife)
  - *Light* (enables reading)
- A ***compartment*** is a structured context *activating* and *deactivating subcontexts*
  - *Marriage*: Mistress (Mätresse) enables lover and lovee during Marriage
  - *Light*: Glasses (enables reading while light is on)
- A ***compartment hierarchy*** is a hierarchy of structured contexts
  - *World model* (town, building, room)
- A ***compartment forest*** is a multi-hierarchy of structured contexts
  - *World model and company model*
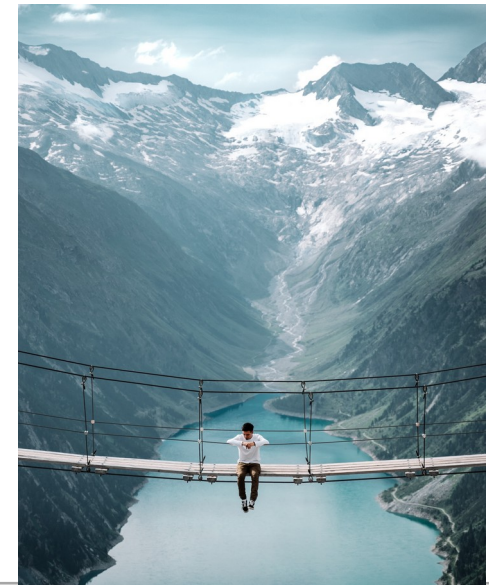
Photo by ROOM on Unsplash

Photo by Alex Azabache on Unsplash

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

48

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept

More on Concern-Separated Representation of Object Nets

Photo by Bruno Kelzer on Unsplash

- Compartments contain collaborations

- Compartments form ***indices*** to interprocedural slices

TECHNISCHE
UNIVERSITÄT
DRESDEN

DRESDEN
concept
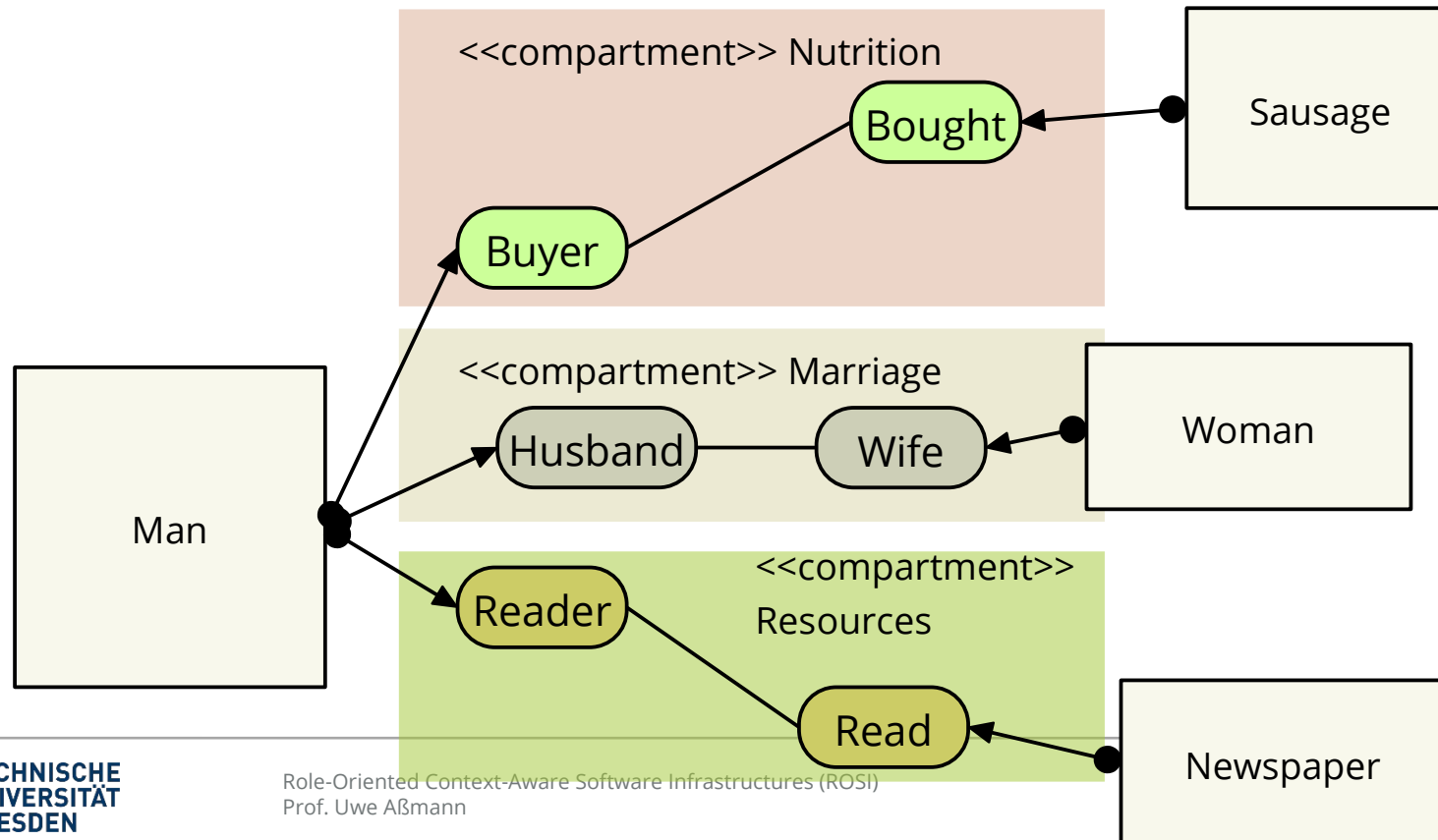
Example of Compartment Multi-Hierarchies

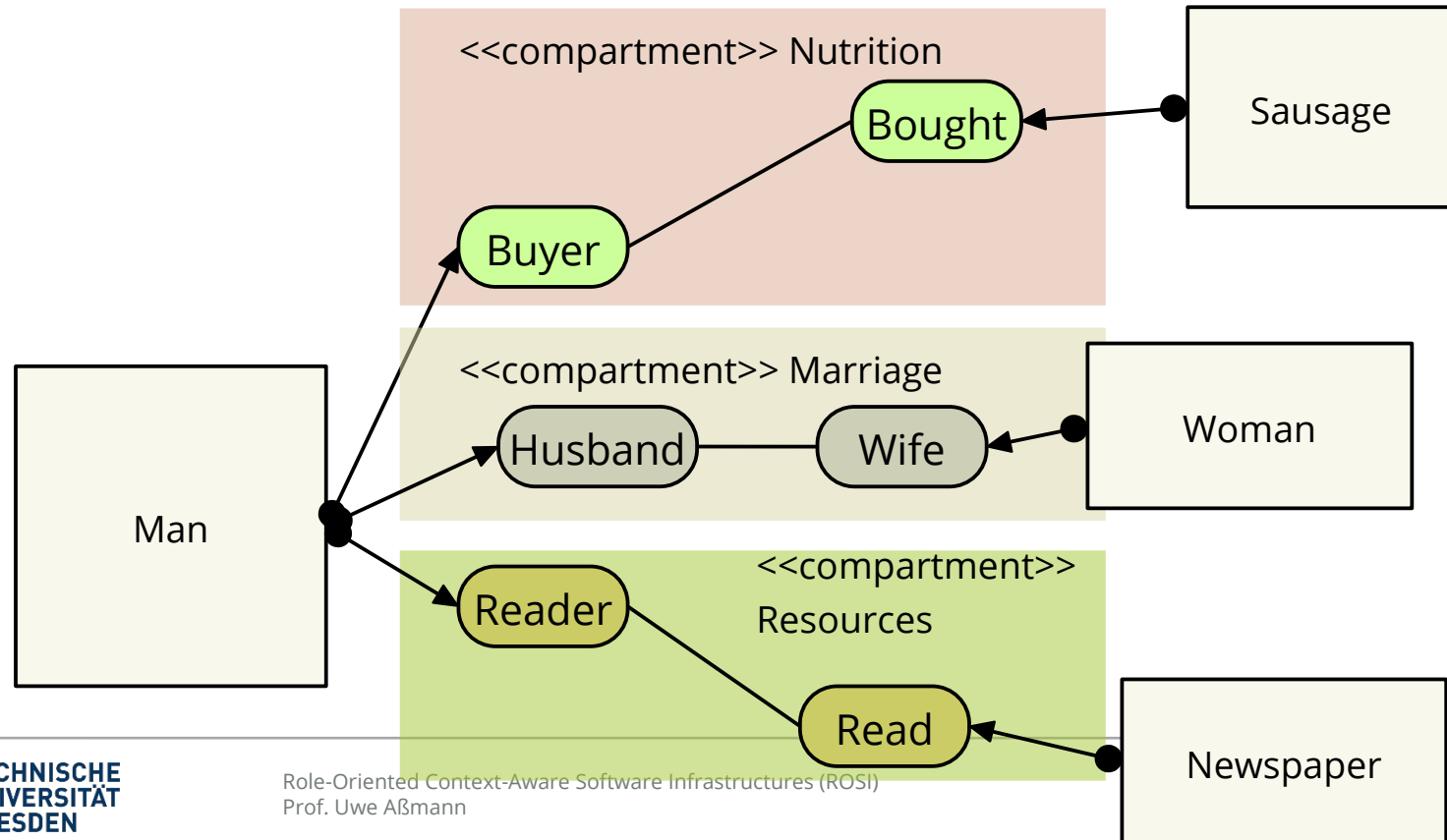Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

50
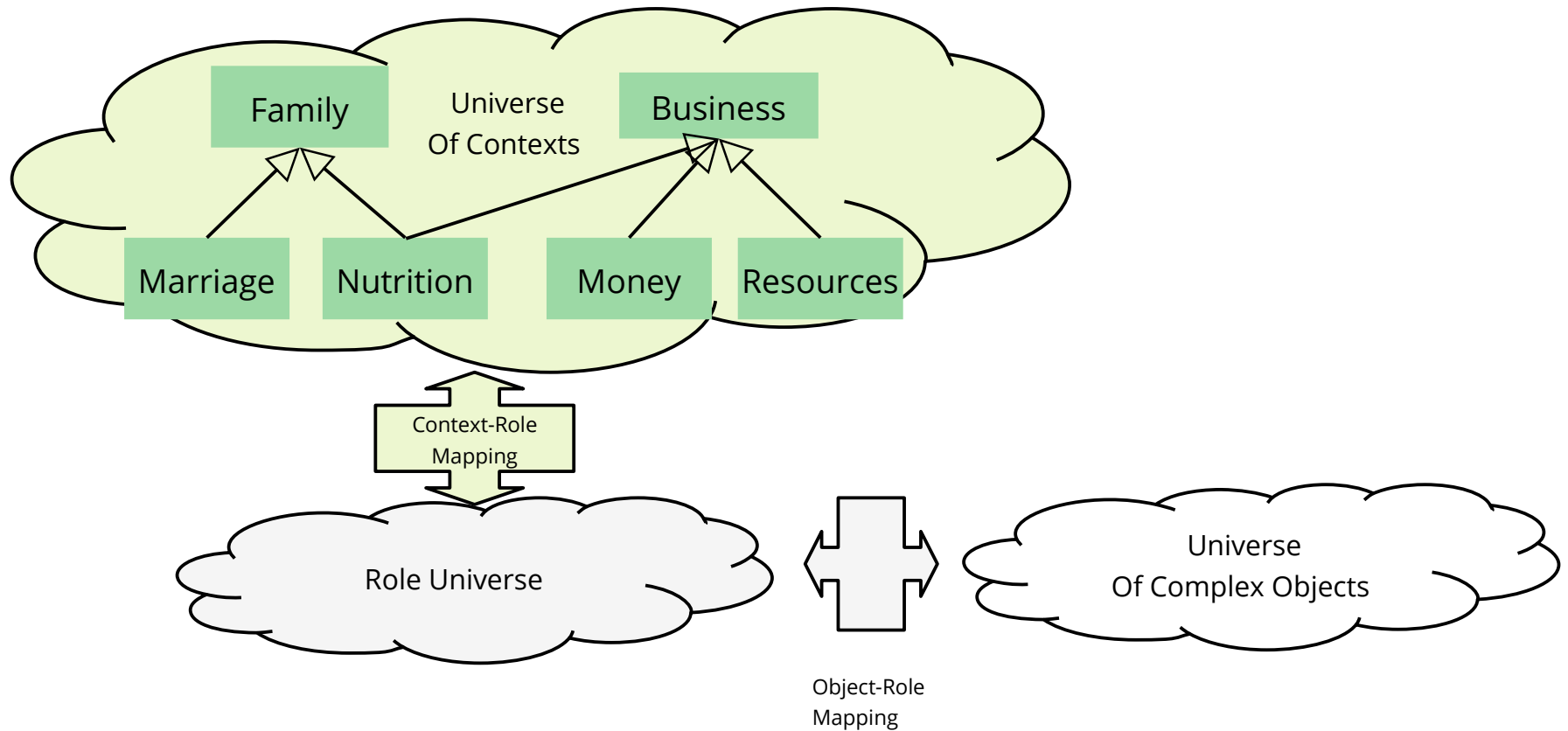
Role-Oriented Context-Aware Software Infrastructures (ROSI)

3. Advantages of Roles:
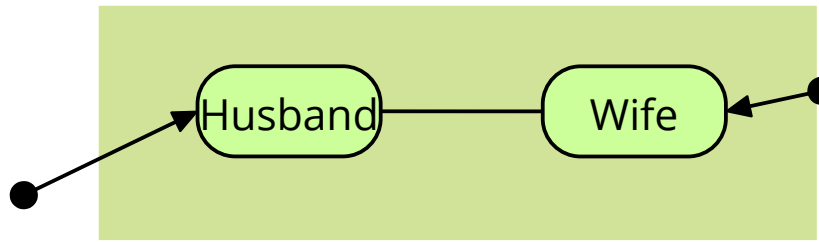## Simple Static and Dynamic Data Extensibility

Simplified Extension with Compartments

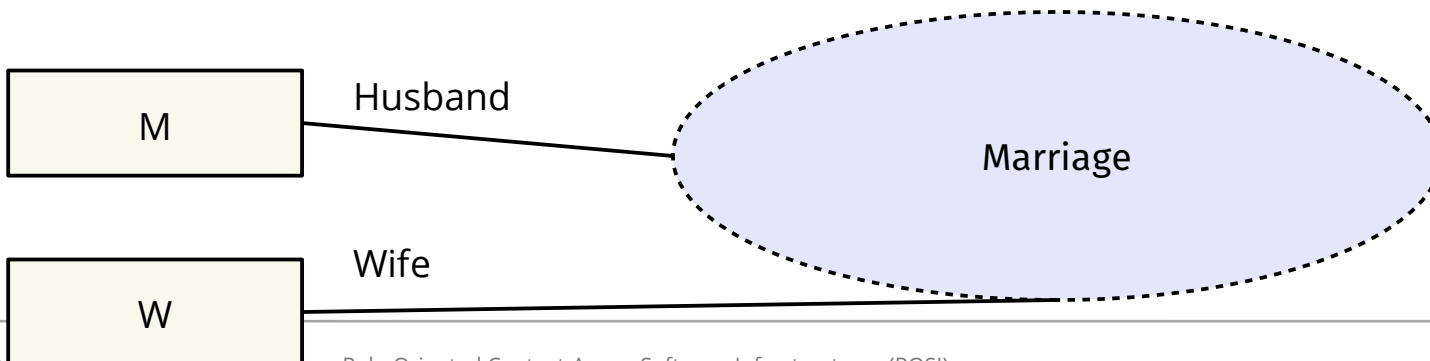- Object-role nets can be *extended by* new compartments with new role models collaborations
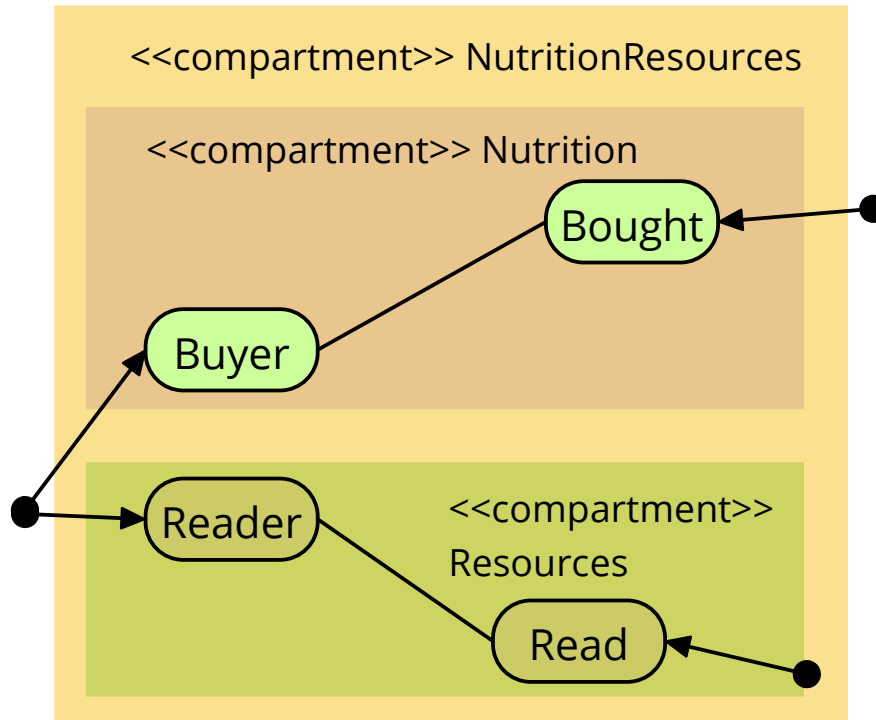
A Compartment is a Relational Module (Collaboration)

- Nets of roles with open ends, open *plays-a* tentacles,

  - to be attached to object cores

  - UML Notation (class level) with *role-type parameter* P:

Role-Oriented Context-Aware Software Infrastructures (ROSI)
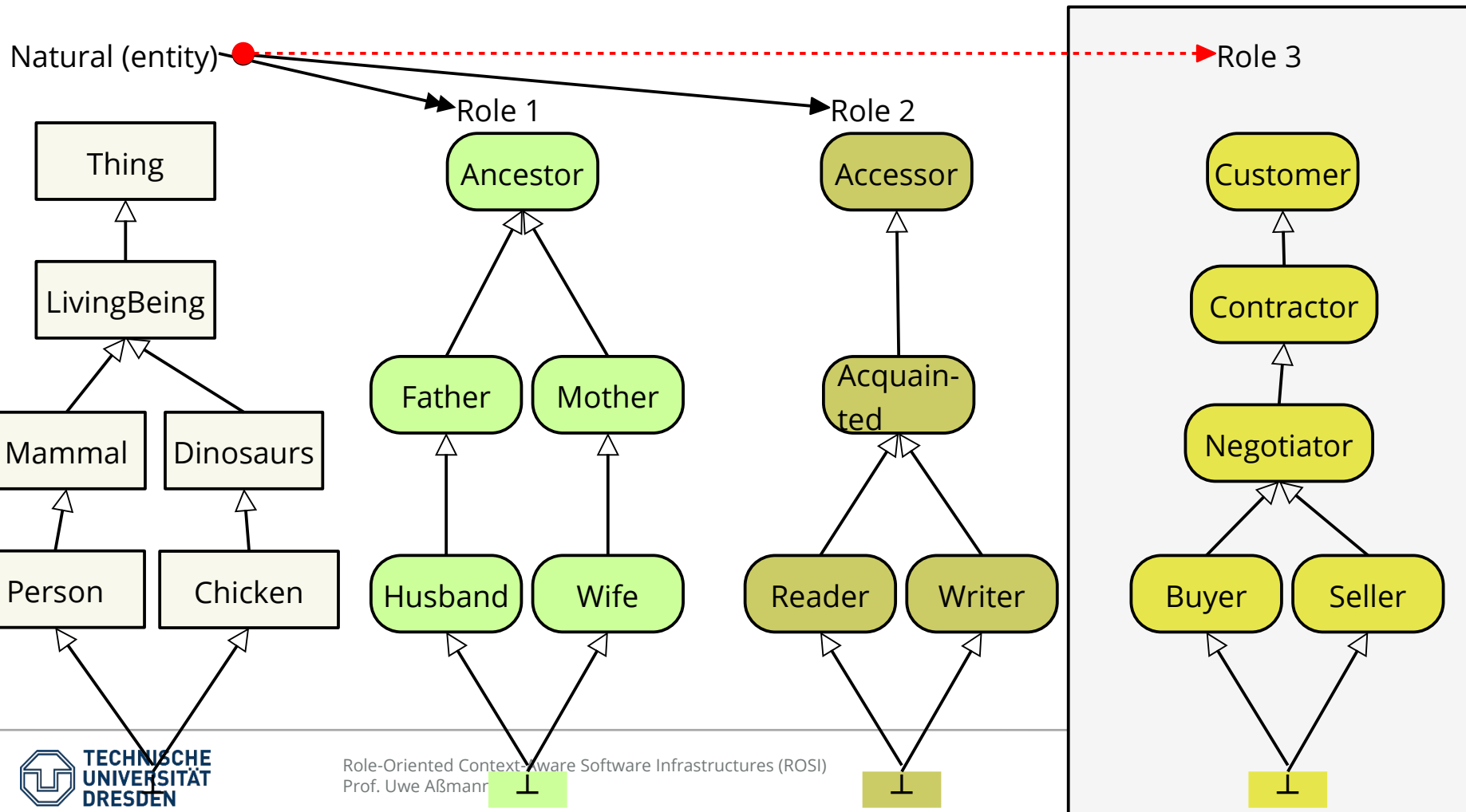Prof. Uwe Aßmann

53

# Structured Compartment: Resources and Nutrition

Extension on the Steimann Product Lattice

- A new role relationship extends the product lattice by another dimension.

Separation of Concerns with Roles:
Identity of Objects is Fixed to Core Facet of Product Lattice

- Role type extensions does not change the name of the core type nor of the full type (polymorphism)

DRESDEN
concept
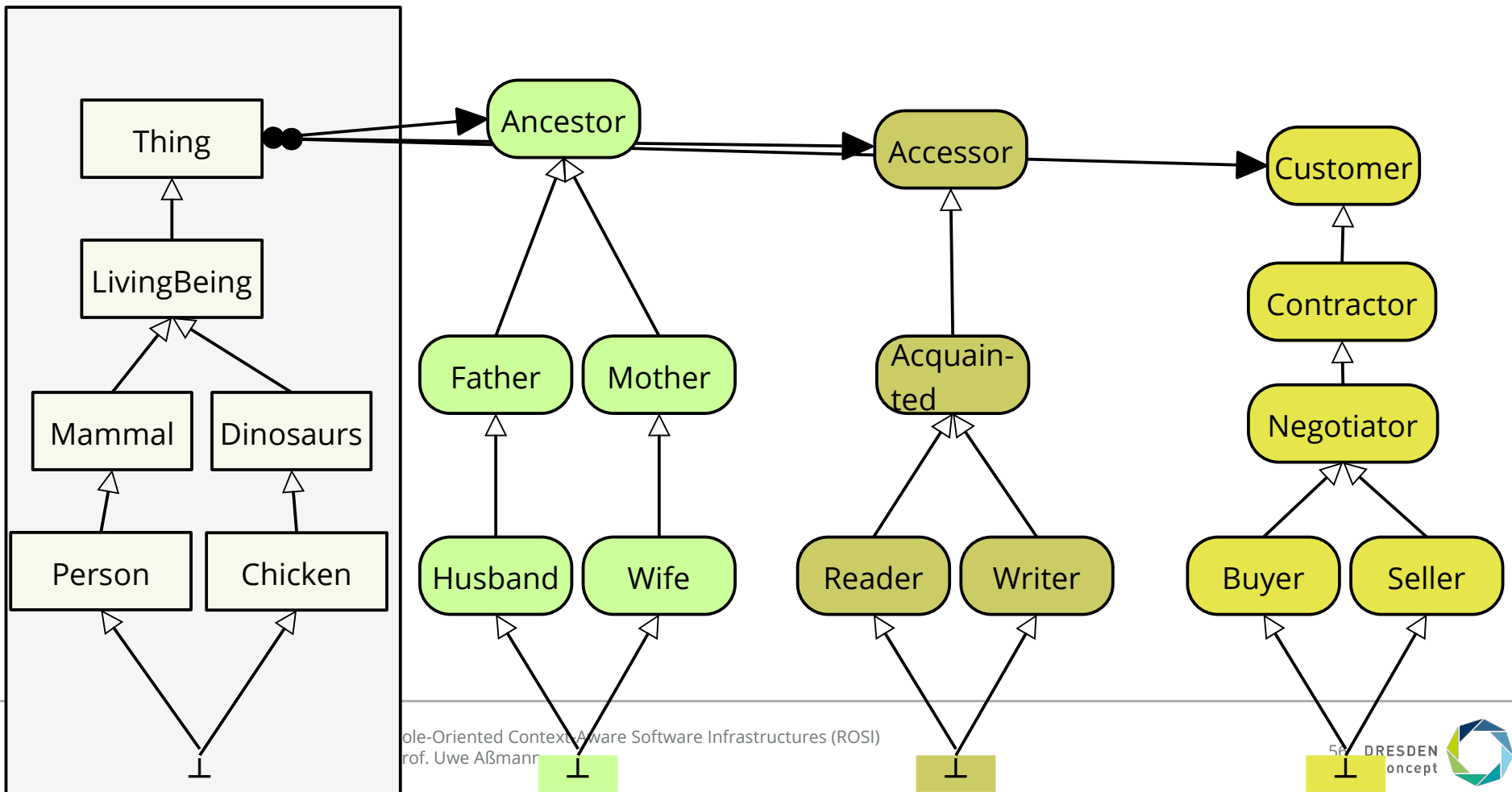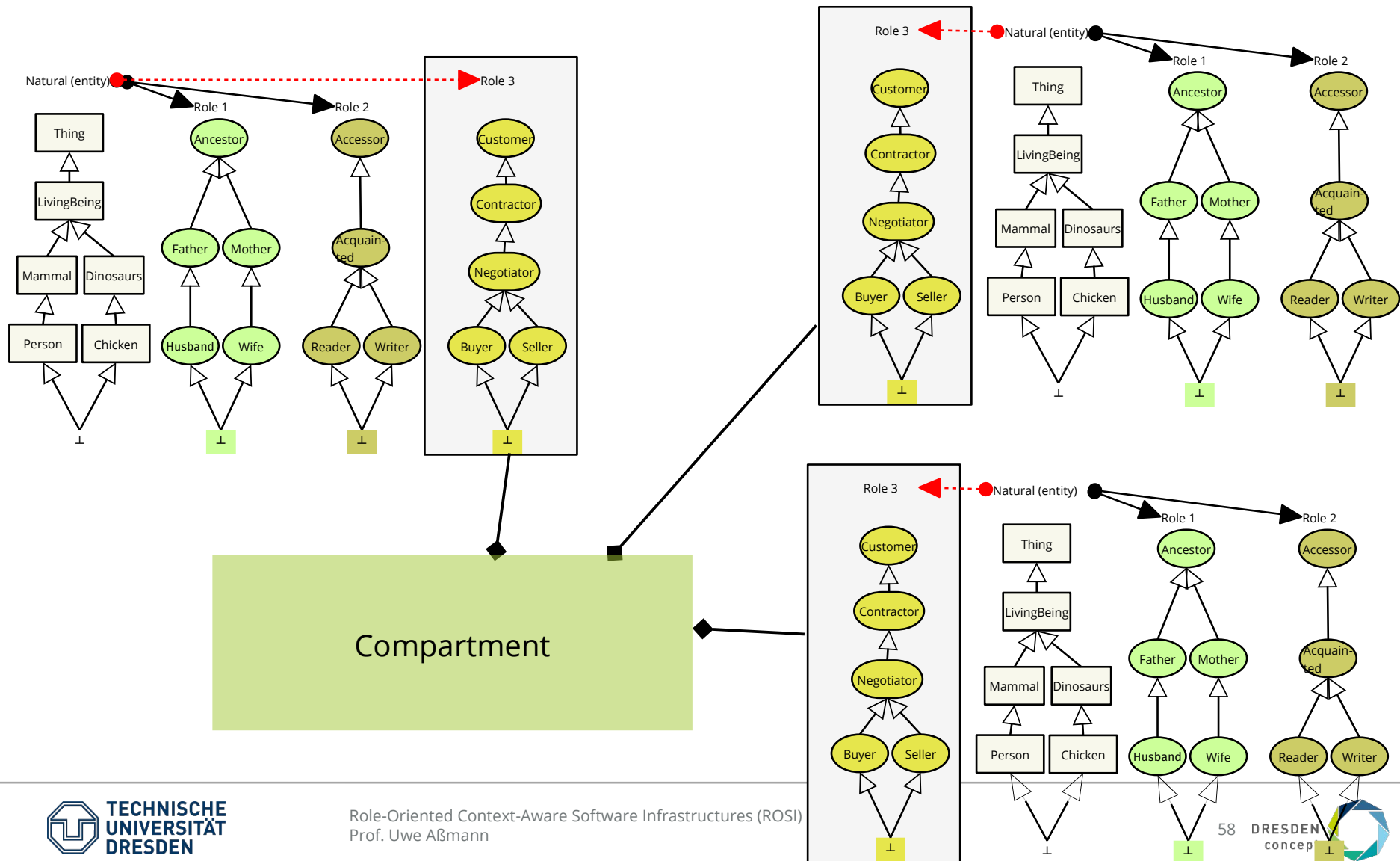
Separation of Concerns with Roles: Simplifies Inheritance Hierarchies

- Role Extension Retains Core Identity

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

57
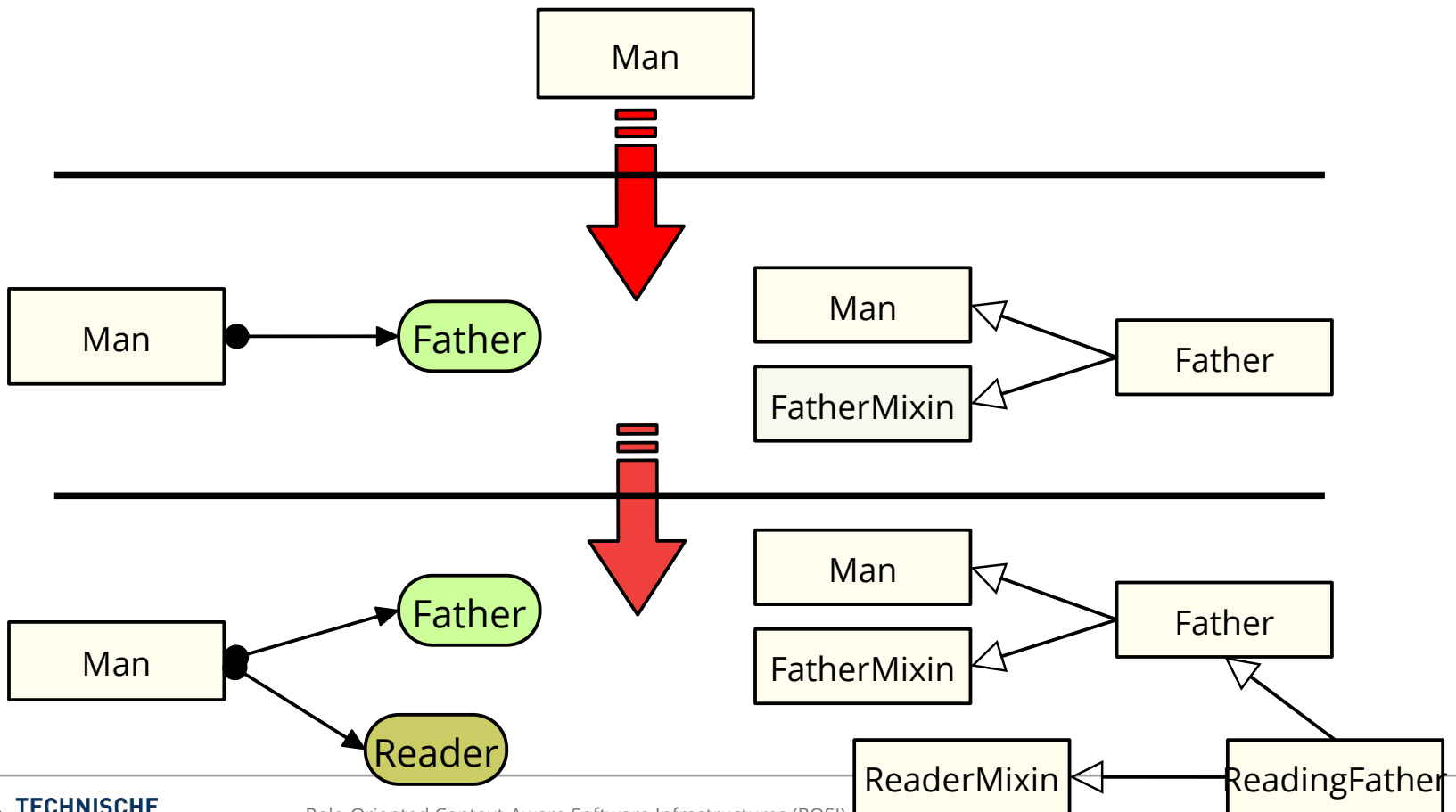
Compartment Superimposition extends the Steimann Lattices of all involved Classes

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

58

Extension and Adaptatation in the Steimann Lattice Retains Inheritance

- Stable entity inheritance hierarchies, if concepts are added *relationally* to a model

    - Otherwise: extension of superclasses necessary (role classes become superclasses of entity classes)

    - Adding of new *concerns* is simple (adding a collaboration)

> Superimposition of compartments to objects in Steimann-factored form retains all inheritance structures
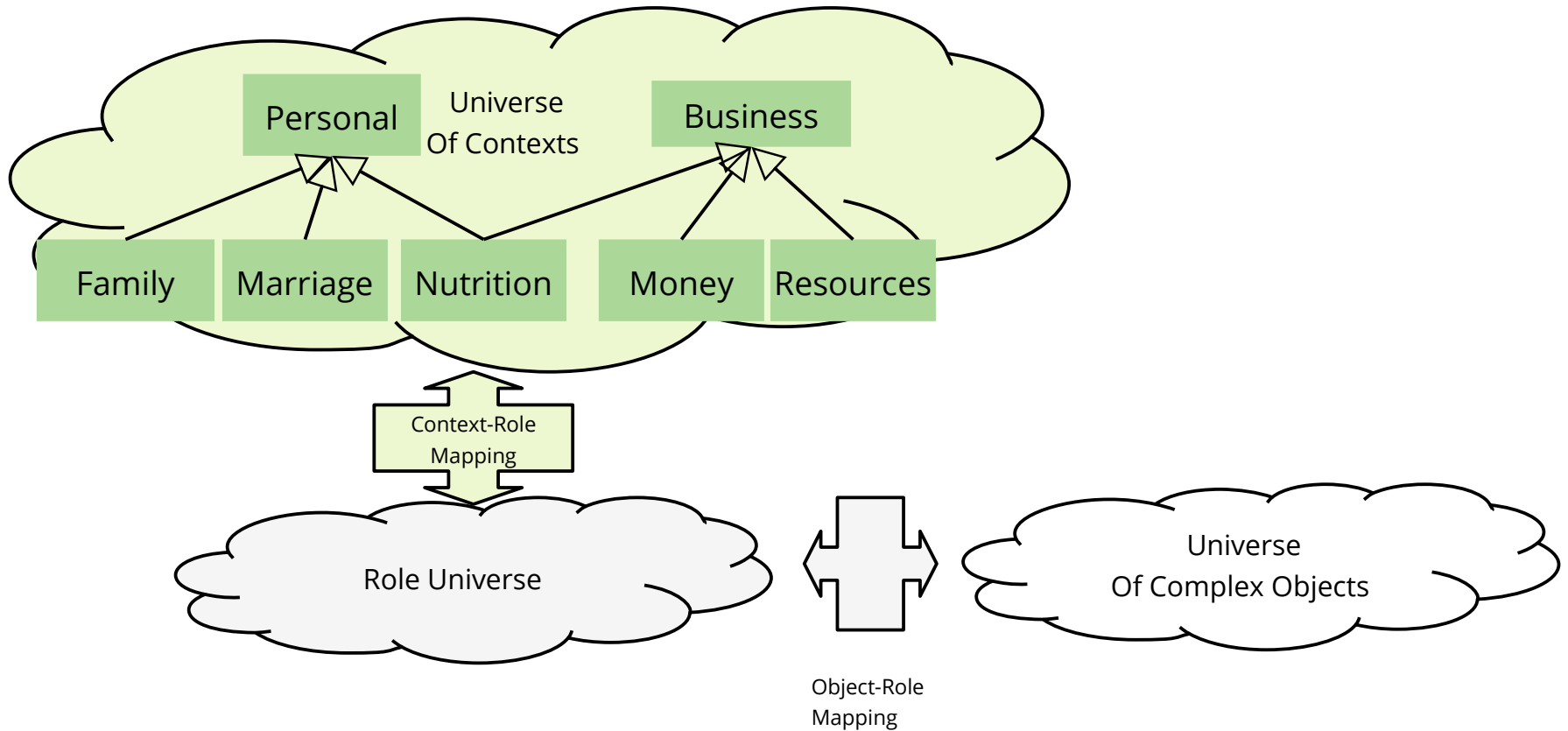
Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

59 DRESDEN
concept

Adaptability with Compartment Multi-Hierarchies

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

60

Adaptability with Compartment Multi-Hierarchies

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

61

Adaptability with Compartment Multi-Hierarchies

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

62

Adaptability with Compartment Multi-Hierarchies

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

63
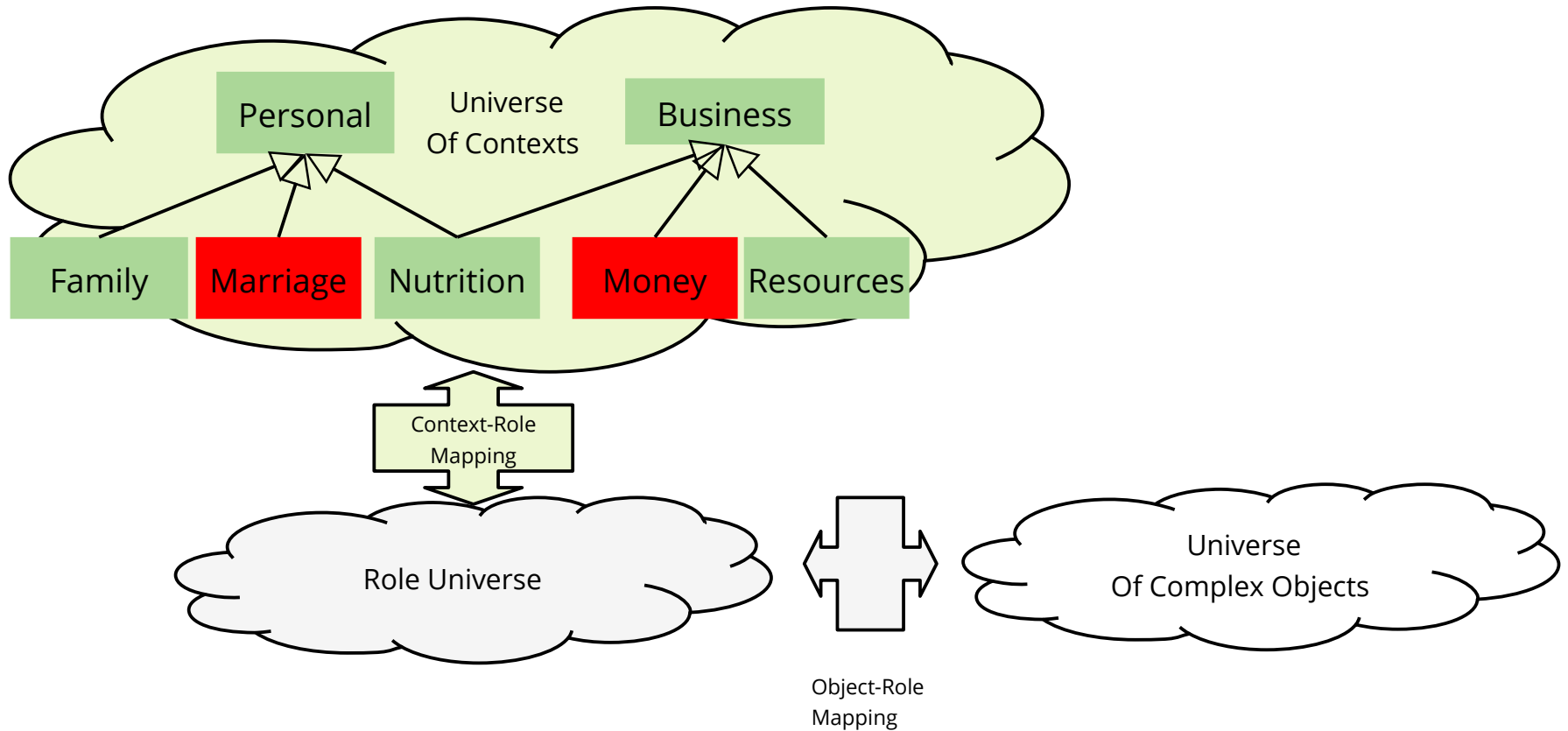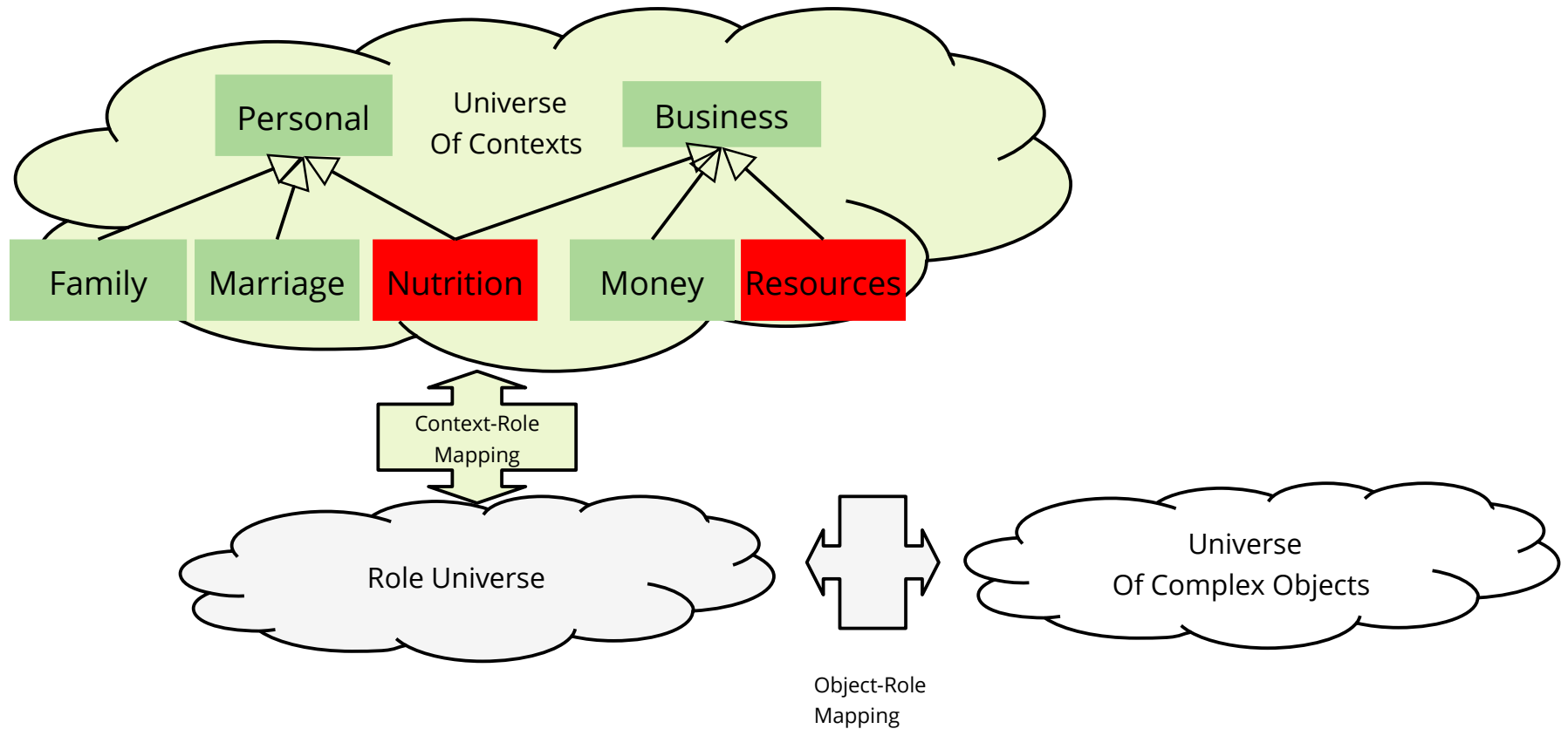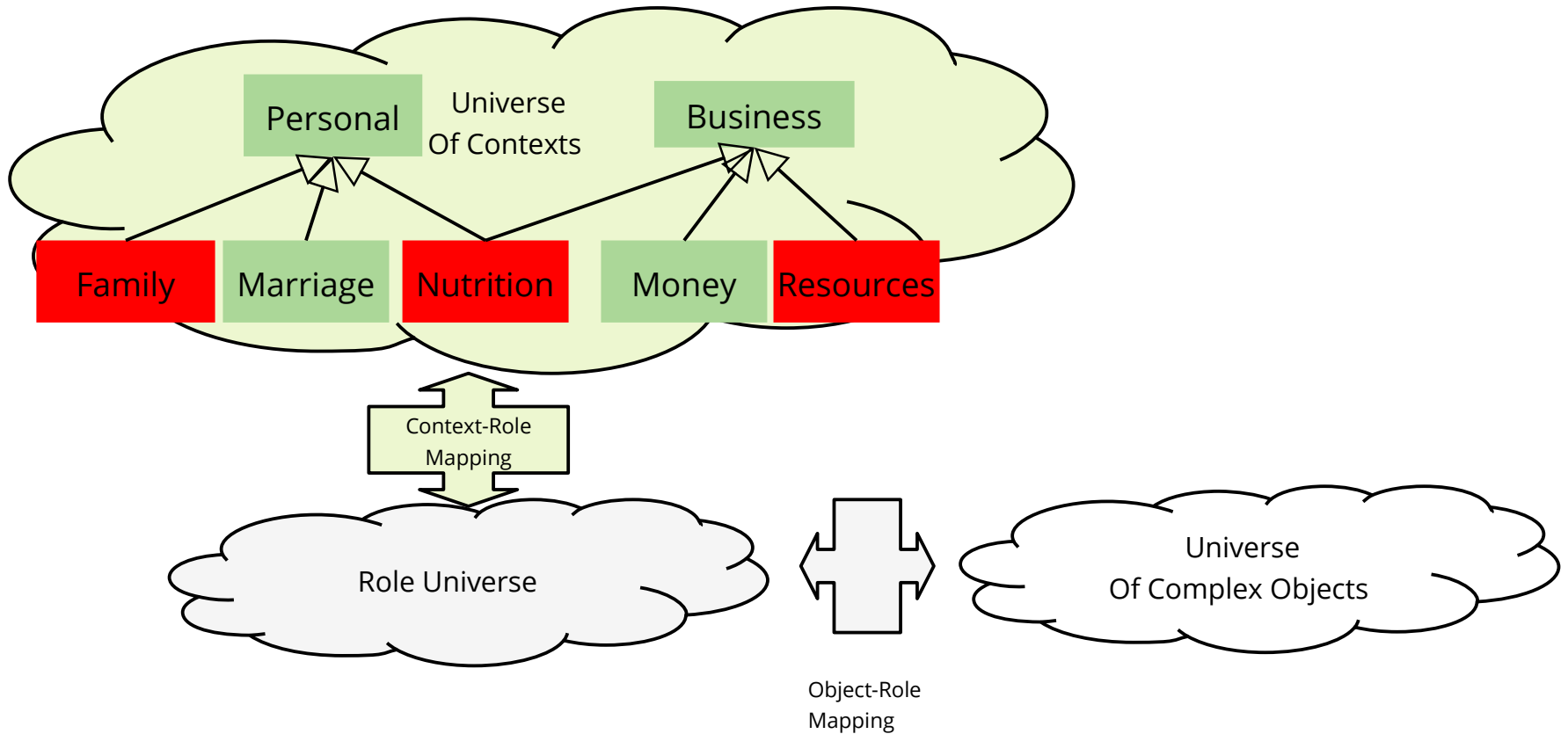
Role-Oriented Context-Aware Software Infrastructures (ROSI)

4. How to Do Object-Oriented Analysis with ROSI

Object-Oriented Analysis with ROSI
Step 1: Ask for the Core Objects with Natural Types

Max:Person

:Resource

Buy24:Bank

Bnn:Newspaper

thuringa:Sausage

Rosi:Woman

Frank:Man

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

65 DRESDEN concept

DRESDEN

Object-Oriented Analysis with ROSI
Step 2: Ask for the Roles with Founded Types

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

66
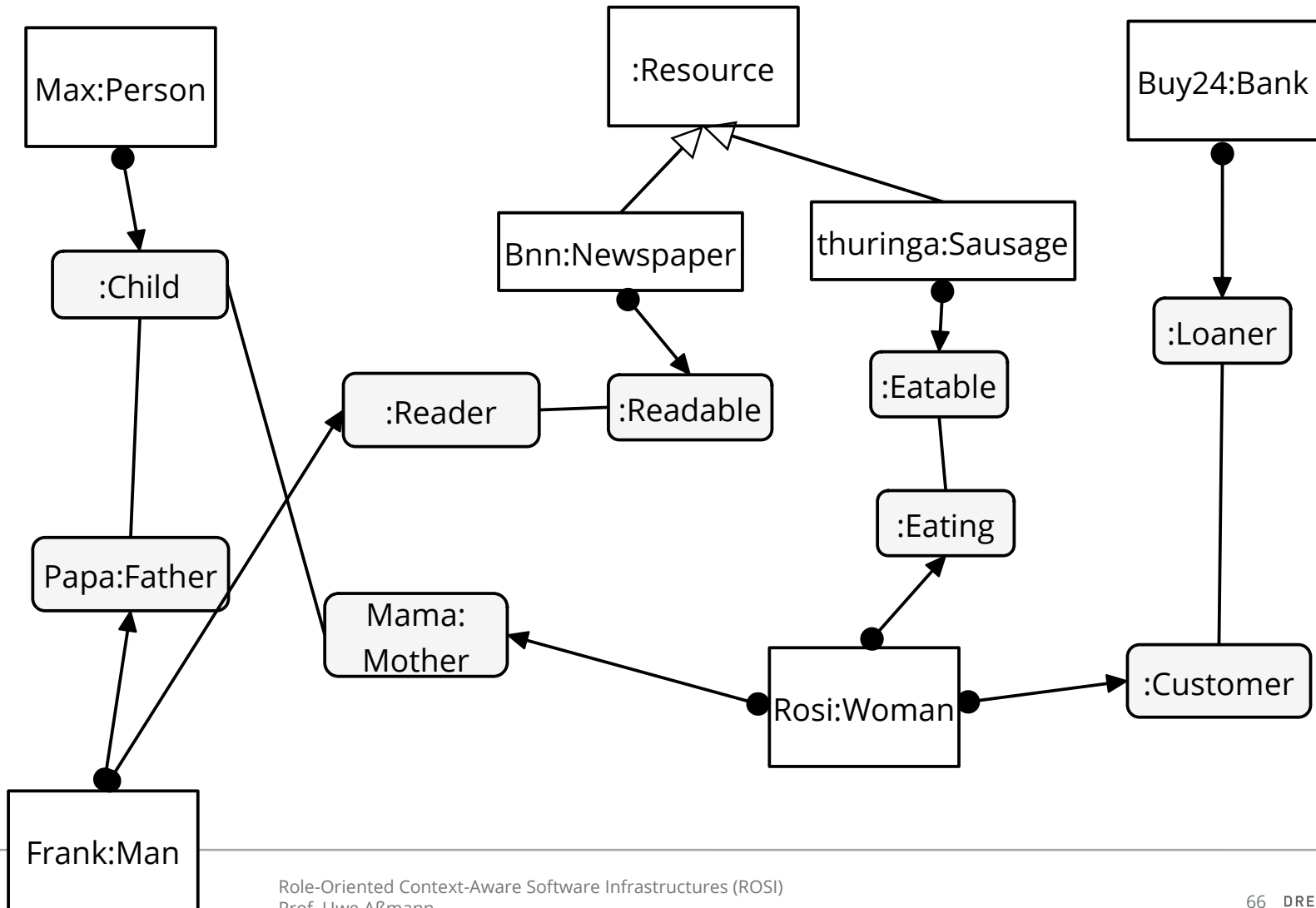
Object-Oriented Analysis with ROSI
Step 3: Ask for the Contexts and
Compartments of the Roles

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

67
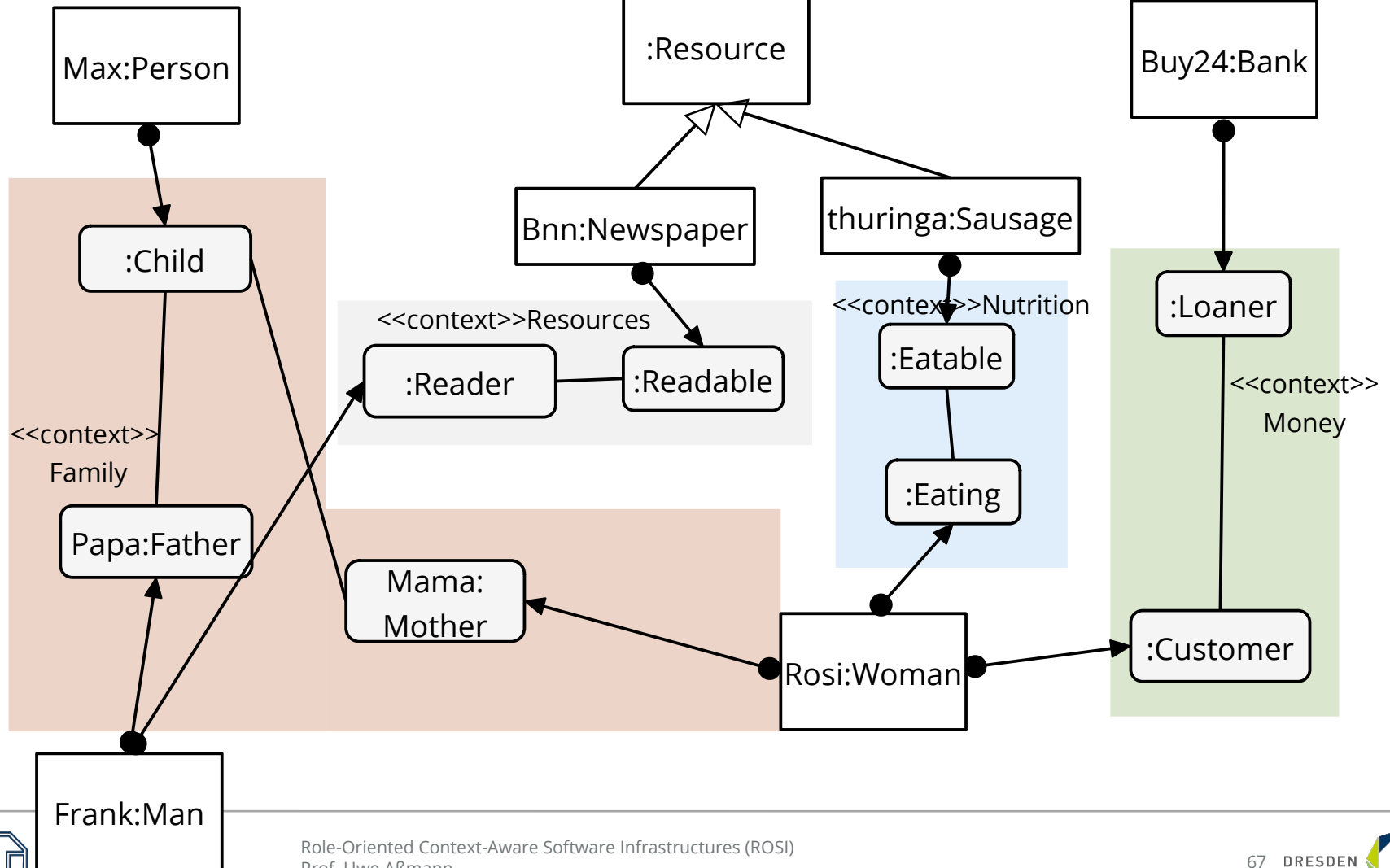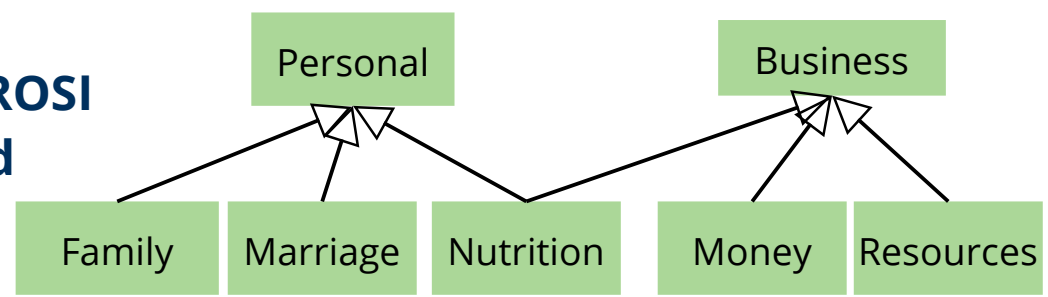
Object-Oriented Analysis with ROSI
Step 4: Dynamic Variation: Variant with Family and Money Contexts

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

68

DRESDEN

DRESDEN concept
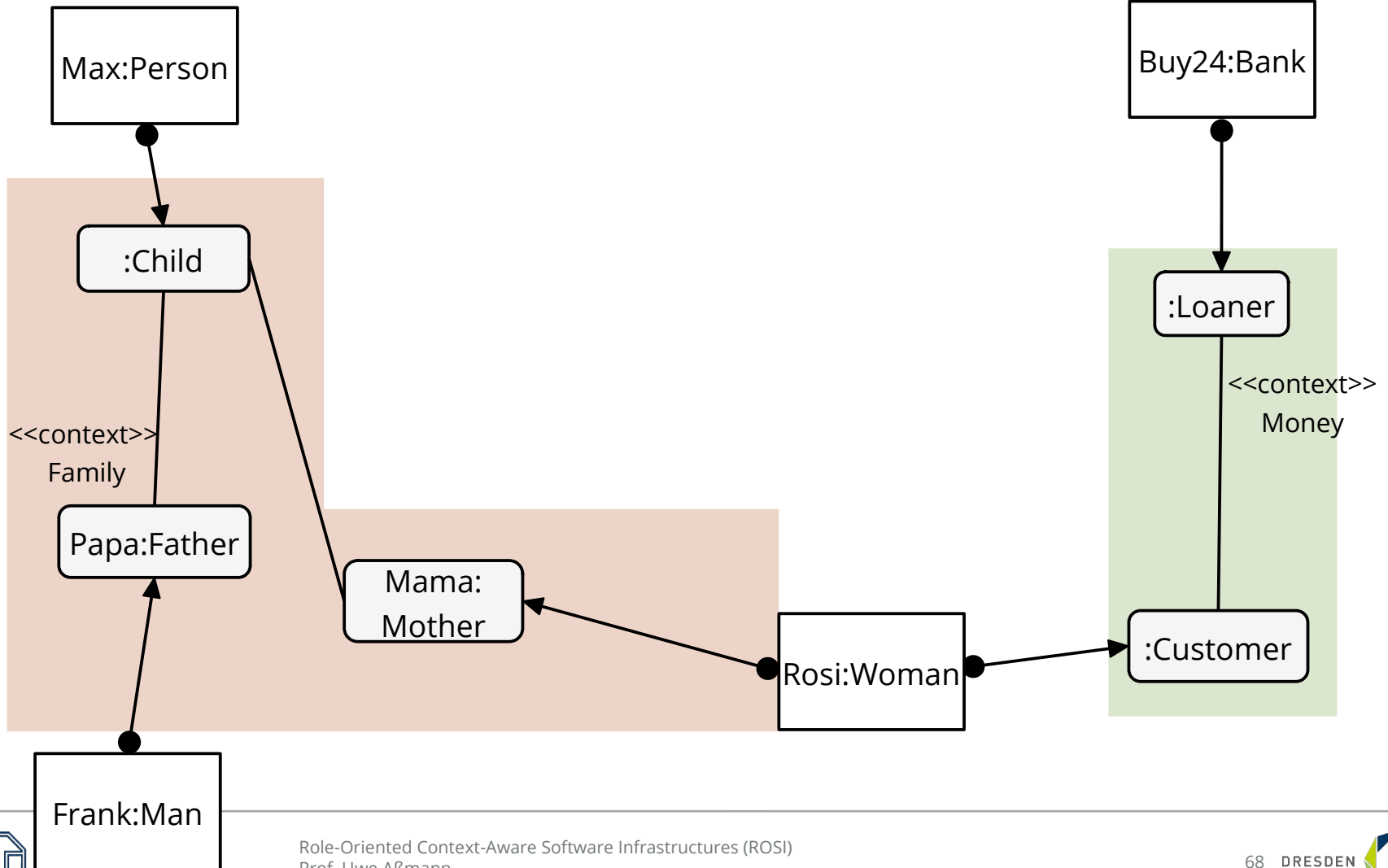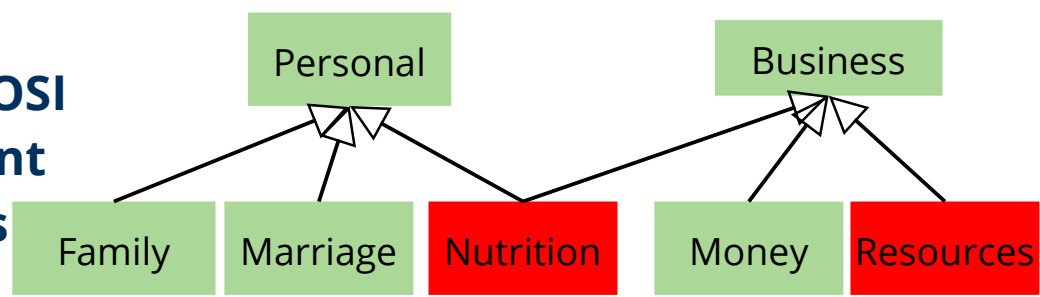
Object-Oriented Analysis with ROSI
Step 4: Dynamic Variation:
Variant with
Contexts Nutrition and Family

Personal

Business

Family | Marriage | Nutrition

Money | Resources

Max:Person

:Resource

thuringa:Sausage

<<context>>Nutrition

:Eatable

:Eating

:Child

<<context>>
Family

Papa:Father

Mama:
Mother

Rosi:Woman

Frank:Man

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

69

DRESDEN
concept

DRESDEN
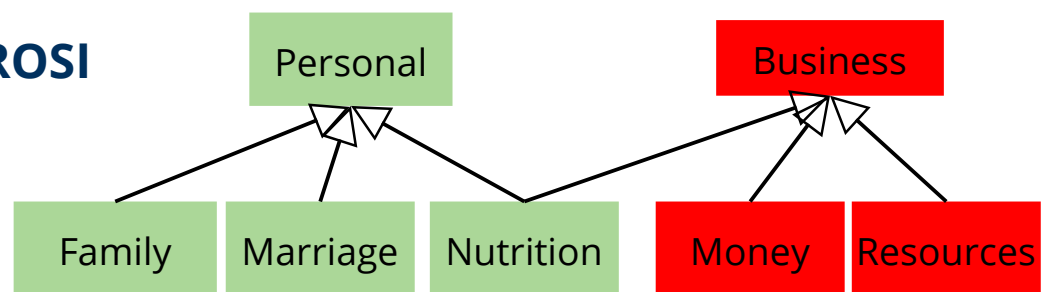
**Object-Oriented Analysis with ROSI
Step 4: Dynamic Variation: Variant
with Compartment Hierarchy
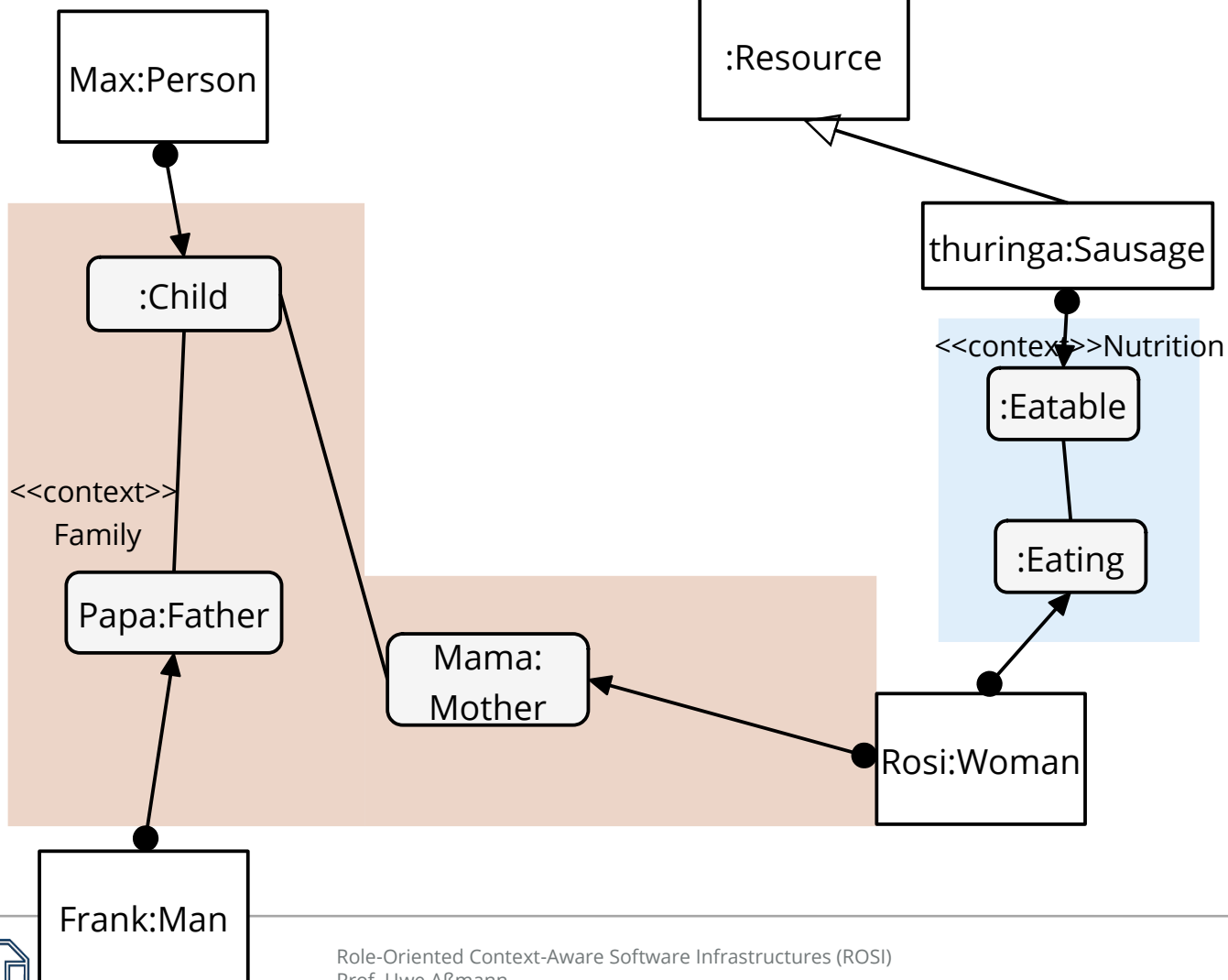(Money, Resources) < Business | Family)**

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

70
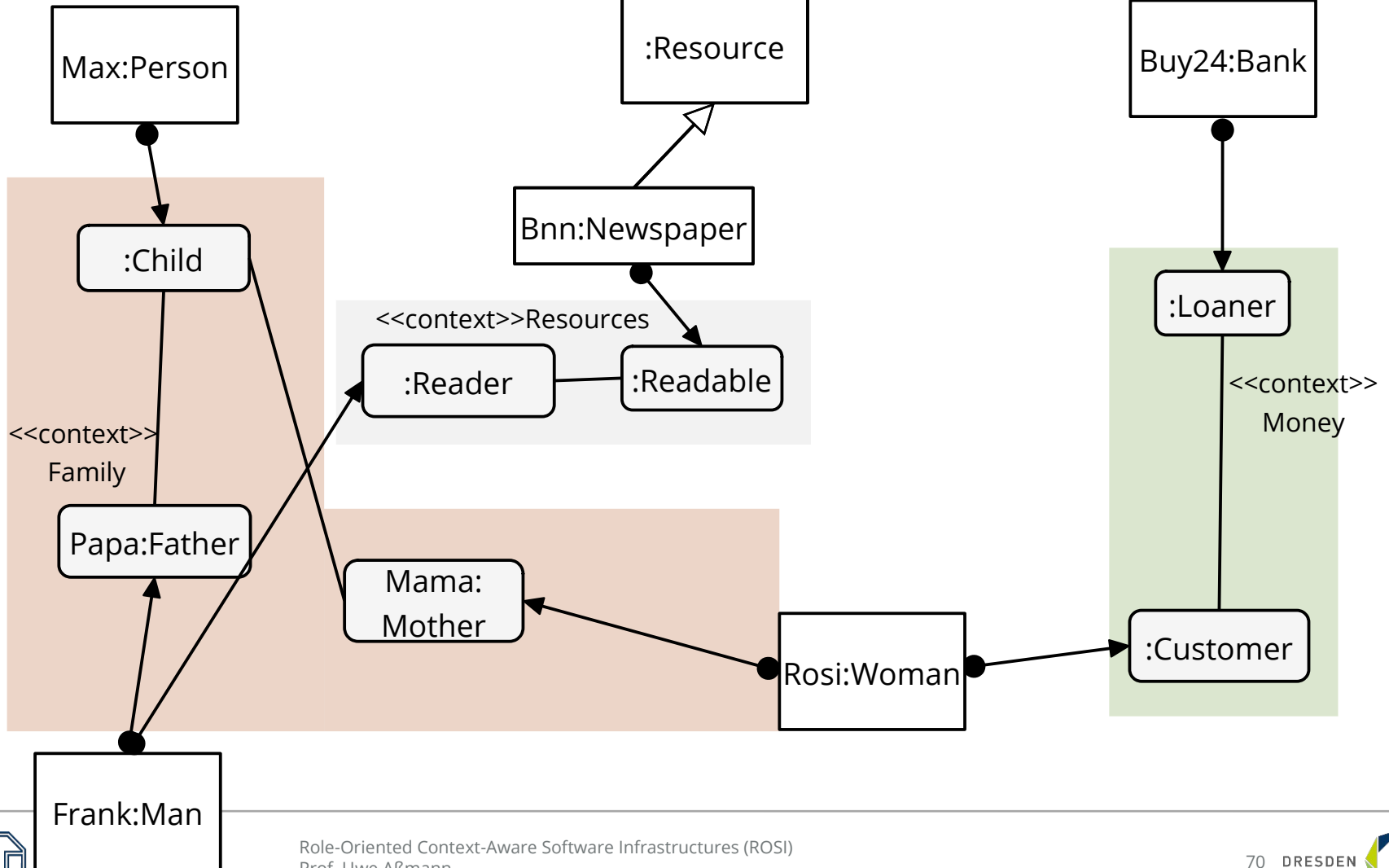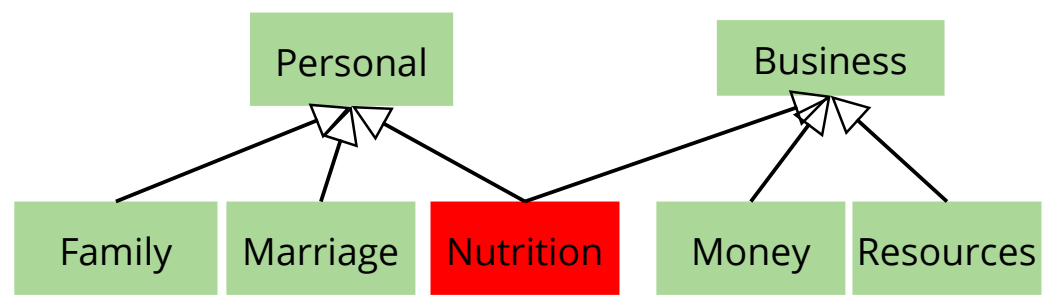
Fakultät Informatik
Lehrstuhl Softwaretechnologie

4.2. Scenario Fire Alarm – in the CROM Modeling Language

Context-Dependent Runtime Models
Compartment Role Object Model (CROM) *[Kühn2015]*

FireDetector (FD)
- place : coord
- fireDetected() : void

AnnouncementProcess (AP)
- logs : string
- triggerAlarm(t:boolean) : void

Announcer (A)
- volume : int
- announceFire() : void

FeedBackSensor (FBS)
- state : boolean
- turnOn() : void
- turnOff() : void

SmokeDetector (SD)
- ID : int
- detect() : void

Camera (C)
- mode : int
- switchMode(m:Int) : void

Phone (P)
- number : string
- call(num:string):void

Speaker (S)
- maxVolume : int
- activate() : void

Legend

Natural Type
- Set of attributes
- Set of methods

OccurenceConstraint

Role Type
- Set of attributes
- Set of methods

Fills relation

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

72

Context-Dependent Runtime Models
Compartment Role Object Model (CROM) *[Kühn2015]*

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

73

Context-Dependent Runtime Models
Compartment Role Object Model (CROM) *[Kühn2015]*

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

Context-Dependent Runtime Models
Compartment Role Object Model (CROM) *[Kühn2015]*

Key properties
- Roles and Relationships depend on the compartments (contexts)
- Roles change over time
- Compartments, "players" and roles have their own identity
- Formal definition of *well-formedness*, *compliance*, and *validity*

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

75 DRESDEN
concept

Context-Dependent Runtime Models
Compartment Role Object Instance (CROI) *[Kühn2015]*

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

76
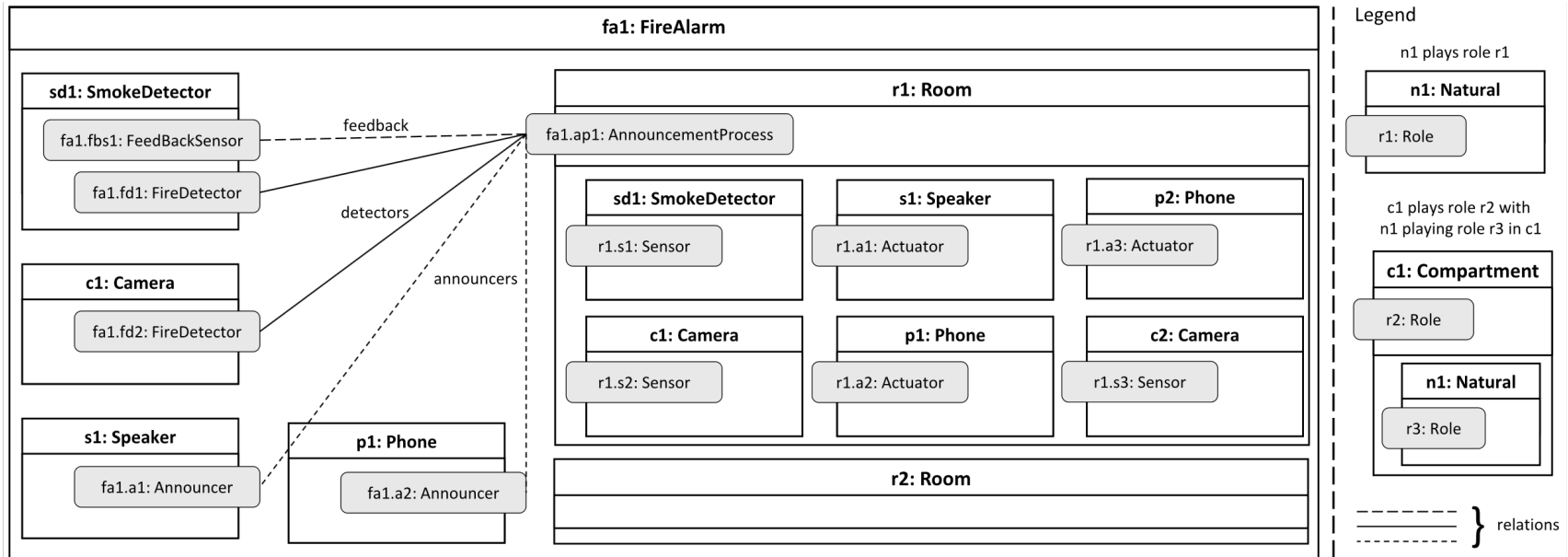
Role-Oriented Context-Aware Software Infrastructures (ROSI)

5. Compartments and Role Playing

Advantages of Roles: Abstraction of Behavior
The Role-Play Automaton
The Role-Play Petri Net

Aquisition and Loss of Roles
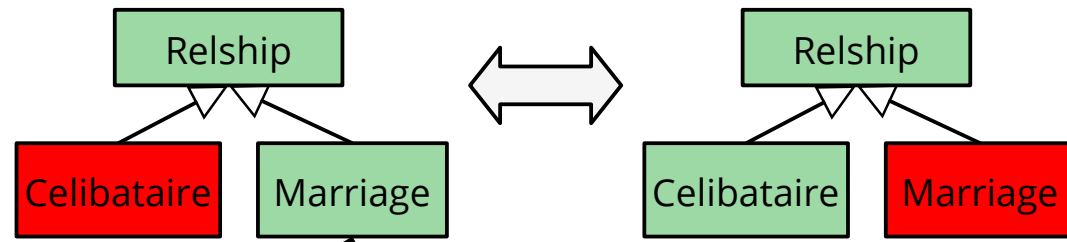
- Aquisition and Loss of Roles creates an **Role-Play Automaton** *abstracting the behavior of a class of objects*

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

78
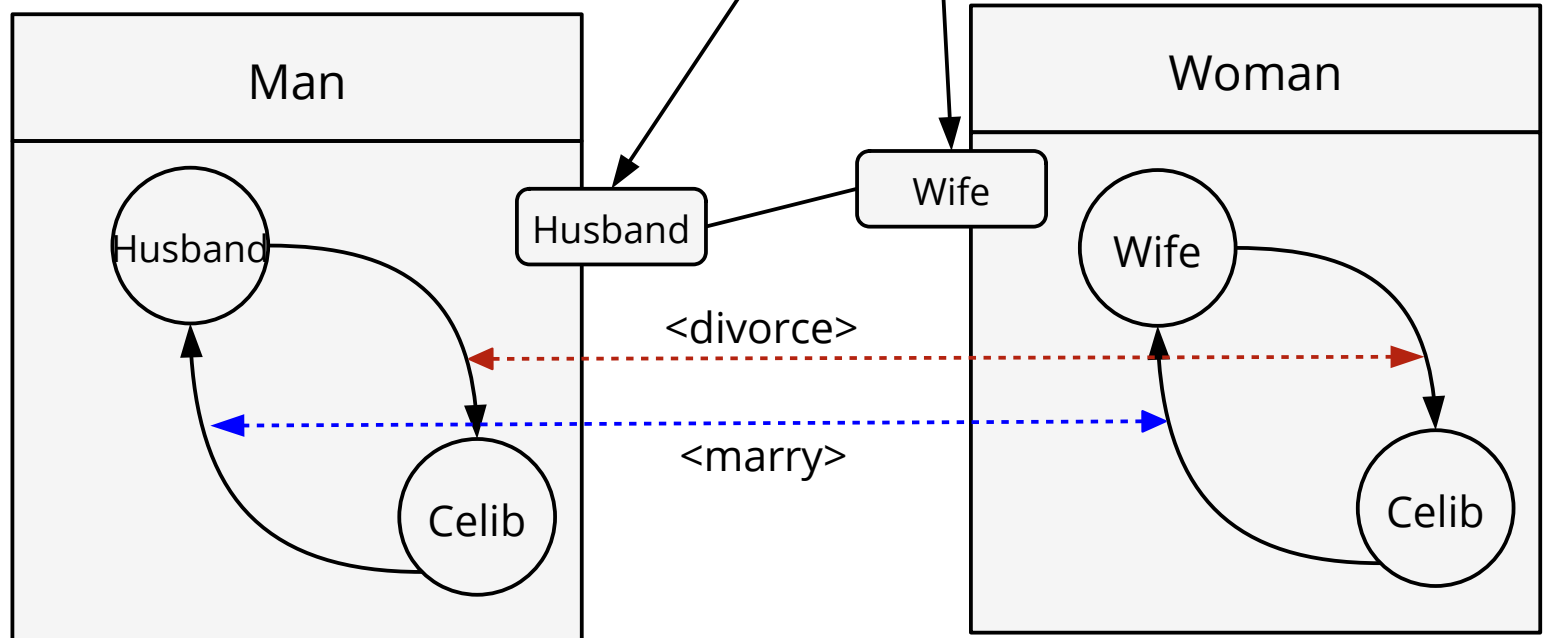
Parallel Aquisition and Loss of Roles

- Parallel Aquisition and Loss of Roles in a parallel class creates an **Role-Play Petrinet**
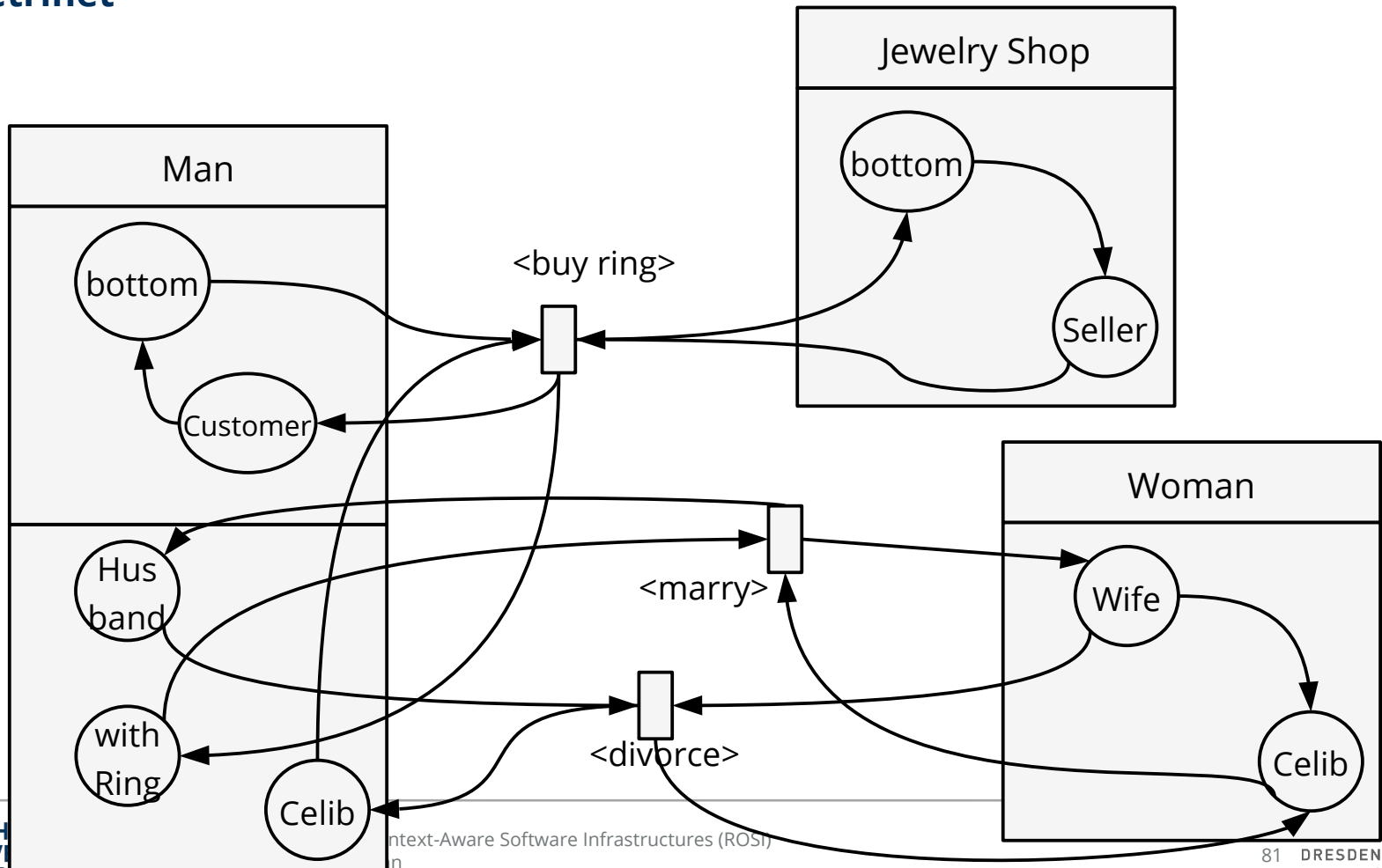
Regular Adaptability

- Many applications have a restricted form of adapability

- A **regularly adaptable class** has a finite role-play automaton with n compartments as states

    – Infinitely many adaptations, but regularly many

> The role-play petrinet of a regularly adaptable class is k-bounded.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

82 DRESDEN
concept

Role-Oriented Context-Aware Software Infrastructures (ROSI)

5. Advantages of Roles: Behavioral Extensibility

Extending Workflow Nets (Petri Nets)

- With an appropriate behavioral specification language, classes can be extended with regard to behavior

- Example: CPN, Workflow Nets, Reference Attribute Grammars

  - for extension and variation of behavior of classes:

  - Extension of AND, OR, XOR split and join *open operators*

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

84   DRESDEN
concept

Complex Transition Operators in Workflow Nets: Join and Split Open Operators of YAWL

- All incoming places are ready (conjunctive input, AND-join)

  AND
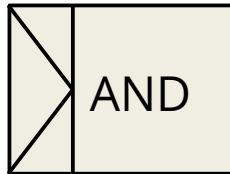
- One out of n incoming places are ready (disjunctive input)

  XOR
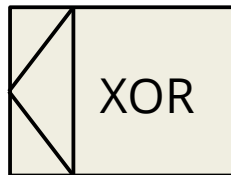
- Some out of n incoming places are ready (selective input)

  OR
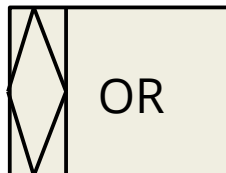
- All outgoing places are filled (conjunctive output, AND-split)

  AND
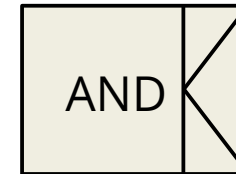
- One out of n outgoing places are filled (disjunctive output, XOR split)

  XOR
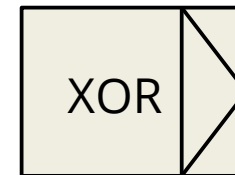
- Some out of n outgoing places are filled (selective output, OR-split)

  OR

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept
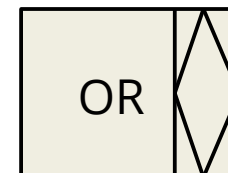
Extension of Workflows

- Behavior can be added in *slices* to *open* split and join operators

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

86   DRESDEN
      concept
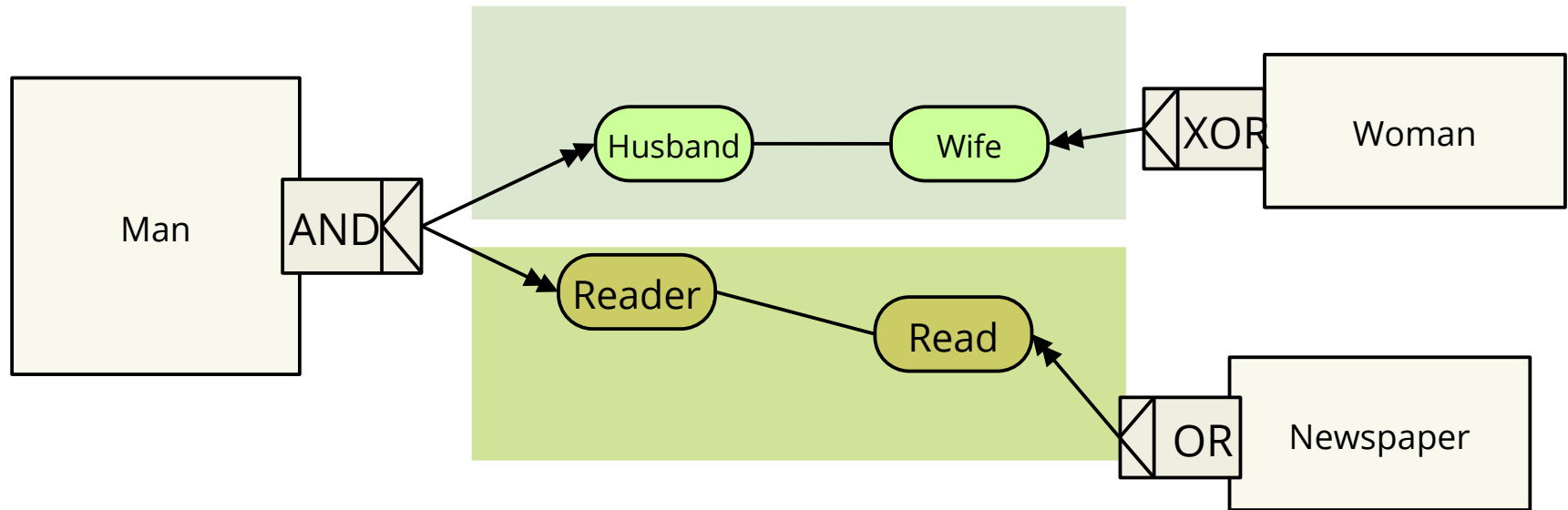
Extension of Workflows

- with AND semantics
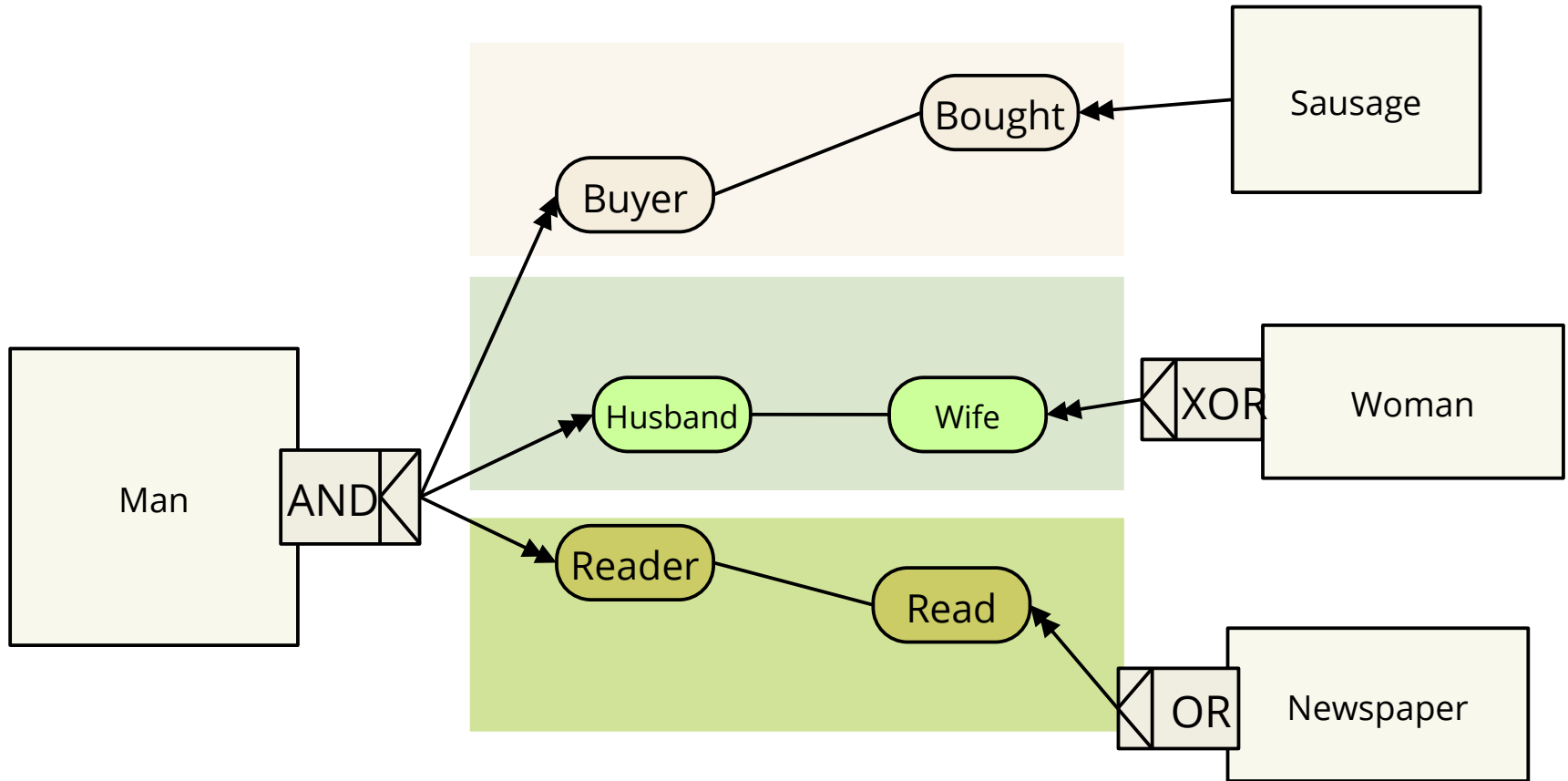
Extension of Workflows

- with OR semantics
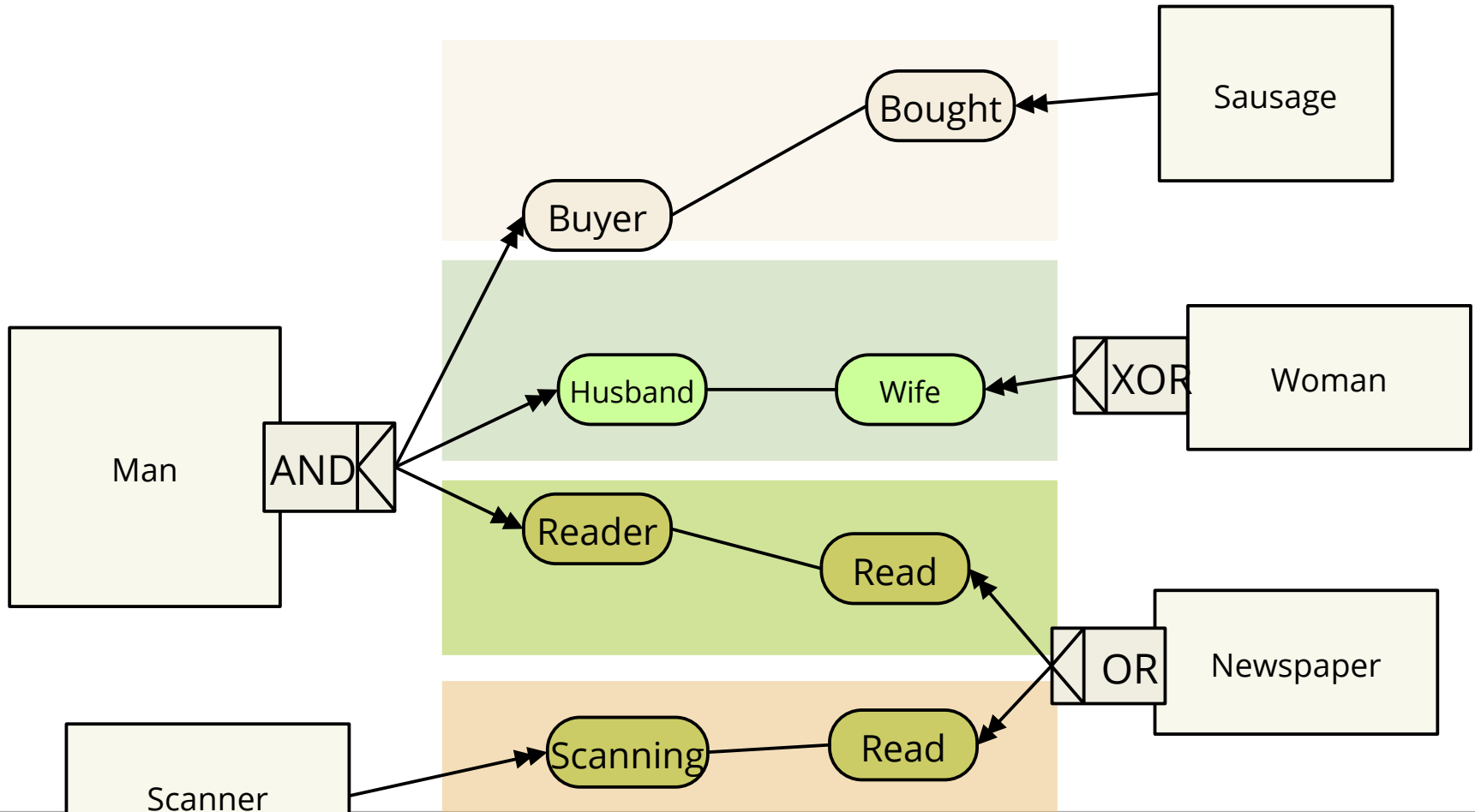
Separation of Concerns with Roles:
Role-Based Contracts are Context-Based

- Contracts describe conditions for *substitutability*

- The Role-Play Petrinet determines which contracts are active

    - in which compartment

Roles can improve contract theory for object-oriented programs

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

89 DRESDEN concept

TECHNISCHE
UNIVERSITÄT
DRESDEN

ROSI Programming with SCROLL

- Compartment and Role Classes

- Dynamic Role Playing with *deep roles*

- SCROLL Scala Library https://github.com/max-leuthaeuser

Roles and context are ready for programming in SCROLL

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

90

Role-Oriented Context-Aware Software Infrastructures (ROSI)

6. Roles and their Benefit for Separation of Concerns

Business Objects with Roles and Contexts

- In large ERP frameworks (see SAP) business objects get very complex

- Ex.: **Order** gets different contexts, with roles
    - Every phase defines a context with different collaborators

- Dynamic Extensibility and Variability (Adaptation) by activation of new contexts
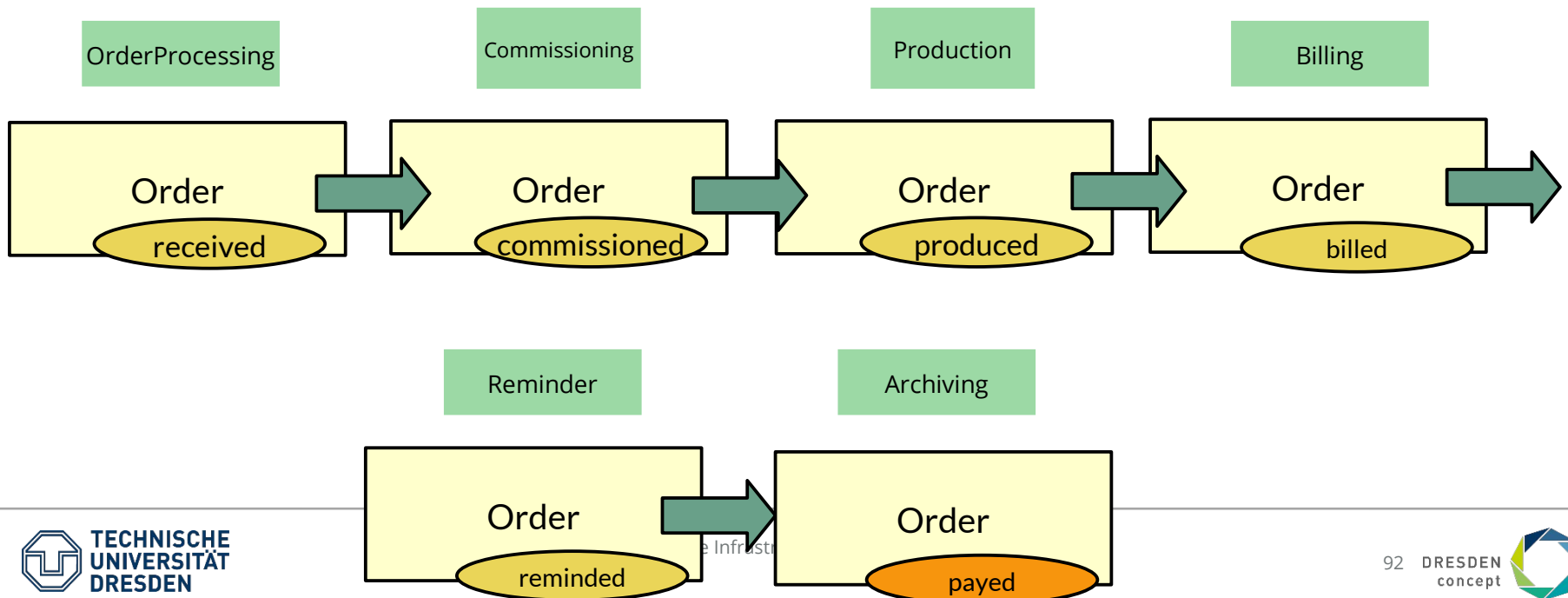
Parallel Objects with Roles and Contexts

- Selection of synchronisationprotocol by activation of new contexts
    -

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

93
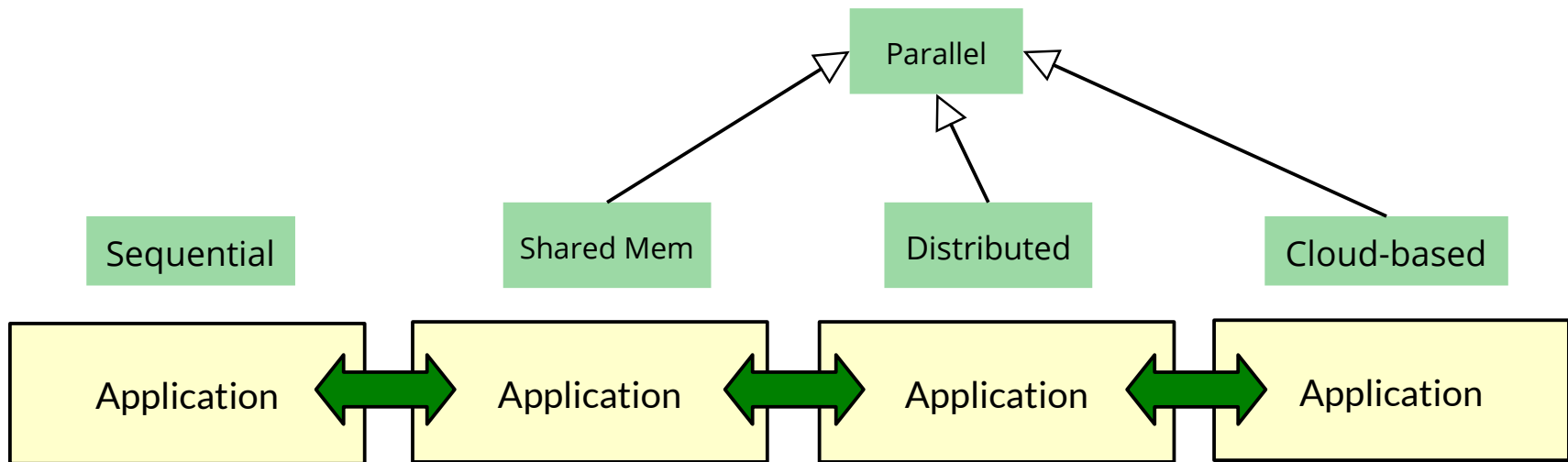
Advantages of ROSI for System Construction

- Separation of Concerns

    - Natural features – Context-dependent features

    - Dynamic features – static features

- Representation of roles as interprocedural graph slices

- Adaptability

    - Extensibility

    - Aspect Orientation (behavioral extensibility)

    - Variability (delayed role embedding decisions)

    - Substitutability (of roles and role models)

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

94

ROSI supports Roles and Contexts for Multi-Dimensional Dispatch for Multi-Polymorphism

- How is the semantics of a function (method) determined?

Receiver-based

Sender-based

Context-based

Function-based (Name, parameters, Parameter types)
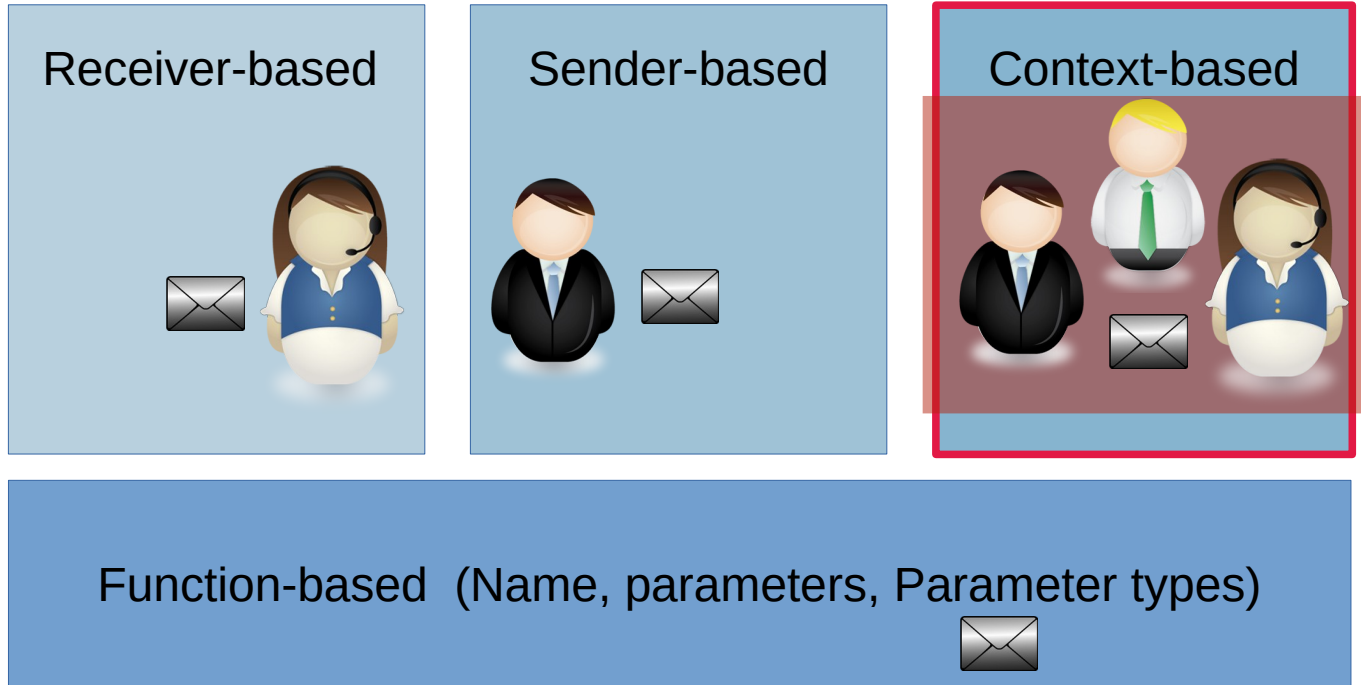
Dijkstra on Separation of Concerns

E. W. Dijkstra "On the Role of Scientific Thought", EWD 447 Selected Writings on Computing: A Personal Perspective, pages 60–66, 1982.

"Let me try to explain to you, what to my taste is **characteristic for all intelligent thinking.**

It is, that one is willing to study in depth **an aspect of one's subject matter in isolation** for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects.

We know that a program must be correct and we can study it from that viewpoint only; we also know that it should be efficient and we can study its efficiency on another day, so to speak. In another mood we may ask ourselves whether, and if so: why, the program is desirable. But nothing is gained --on the contrary!-- by tackling these various aspects simultaneously.

Roles and contexts introduce separations of concerns.

Intelligent thinking and scientific thought

It is what I sometimes have called **"the separation of concerns"**, which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of.

This is what I mean by **"focussing one's attention upon some aspect"**: it does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant. It is being one- and multiple-track minded simultaneously.

Scientific thought comprises "intelligent thinking" as described above. A scientific discipline emerges with the --usually rather slow!-- discovery of which aspects can be meaningfully **"studied in isolation for the sake of their own consistency"**, in other words: with the discovery of useful and helpful concepts. Scientific thought comprises in addition the conscious search for the useful and helpful concepts.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

97 DRESDEN
concept

TECHNISCHE
UNIVERSITÄT
DRESDEN

The End

https://rosi-project.org

Important References

- T. Reenskaug, P. Wold, and O. Lehne. Working with Objects, The OOram Software Engineering Method. Manning Publications, 1996.

- Friedrich Steimann. On the representation of roles in object-oriented and conceptual modelling. Data Knowl. Eng, 35(1):83-106, 2000.

- Friedrich Steimann. A radical revision of UML's role concept". UML 2000, 3rd International Conference, Springer LNCS, 194–209.

    – and many more, see his home page at U Hagen

- Charles W. Bachman and Manilal Daya. The role concept in data models. In VLDB '1977: Proceedings of the third int.l conf. on Very large data bases, pages 464–476. VLDB Endowment, 1977.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

99 DRESDEN concept

Ontological Foundations of Metatypes

- Giancarlo Guizzardi, Heinrich Herre, and Gerd Wagner. On the general ontological foundations of conceptual modeling. 21st Int. Conf. on Conceptual Modeling (ER 2002), LNCS 2503, pages 65-78, 2002.

- Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. PhD thesis, University of Twente.

- [Guariono] Nicola Guarino Chris Welty. Supporting ontological analysis of taxonomic relationships. Data and Knowledge Engineering, 39:51--74, 2001.

- Paul Lorenzen, Oswald Schwemmer. Konstruktive Logik, Ethik und Wissenschaftstheorie. BI Hochschultaschenbücher, Band 700, 1973.

- Paul Lorenzen. Lehrbuch der konstruktiven Wissenschaftstheorie, Metzler Reprint, 2000.

- H. Wedekind, E. Ortner, R. Inhetveen. Informatik als Grundbildung. Informatik Spektrum, Springer, April 2004

- H. v. Braun, W. Hesse, H.B. Kittlaus,G. Scheschonk. Ist die Welt objektorientiert? Von der natürlich-sprachlichen Weltsicht zum OO-Modell. Uni Marburg.

TECHNISCHE UNIVERSITÄT DRESDEN

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

100   DRESDEN concept

Programming Languages

- S. Herrmann. Object teams: Improving modularity for crosscutting collaborations. In Proc. Net Object Days 2002, 2002.

- S. Herrmann. A precise model for contextual roles: The programming language objectteams/java. Applied Onthology, (to appear), 2007.

- www.objectteams.org: a programming lanugage with roles

- Max Leuthäuser. A Pure Embedding of Roles - Exploring 4-dimensional Dispatch for Roles in Structured Contexts. PhD thesis, Technische Universität Dresden, August 2017.

  - This PhD thesis developes a programming language for contexts and roles, based on some implementation patterns and the base language Scala. " http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-227624

- [SCROLL] SCROLL Library https://github.com/max-leuthaeuser

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

101 DRESDEN concept

TECHNISCHE UNIVERSITÄT DRESDEN

Important References

- D. Bäumer, D. Riehle, W. Silberski, and M. Wulf. Role object. In Conf. On Pattern Languages of Programming (PLOP), 1997.

- Dirk Riehle and Thomas Gross. Role model based framework design and integration. ACM SIGPLAN Notices, 33(10):117-133, October 1998.

- Dirk Riehle. Framework Design - A Role Modelling Approach. PhD thesis, ETH Zürich, 2000. No. 13509. www.riehle.org.

- Y. Smaragdakis and D. Batory. Mixin layers: an object-oriented implementation technique for refinements and collaboration-based designs. ACM Transactions on Software Engineering and Methodology, 11(2):215–255, 2002.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

102

Works at TU Dresden

- Thomas Kühn. A Family of Role-Based Languages. PhD thesis, Technische Universität Dresden, March 2017. http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-228027

- U. Aßmann, S. Zschaler, and G. Wagner. Ontologies, Meta-Models, and the Model-Driven Paradigm, Handbook on Ontologies and Software Engineering. pages 249–273. Springer, 2006.

- U Aßmann, J Johannes, J Henriksson, and Ilie Savga. Composition of rule sets and ontologies. In F. Bry, editor, Reasoning Web, Second Int. Summer School 2006, number 4126 in LNCS, pages 68-92, Sept 2006. Springer.

- M. Pradel, J. Henriksson, and U. Aßmann. A good role model for ontologies: Collaborations. Int. Workshop on Semantic-Based Software Development. at OOPSLA'07, Montreal, Oct 22, 2007.

- Christian Piechnick, Sebastian Richly, Sebastian Götz, Claas Wilke, and Uwe Aßmann. Using Role-Based Composition to Support Unanticipated, Dynamic Adaptation - Smart Application Grids. In Proceedings of ADAPTIVE 2012, The Fourth International Conference on Adaptive and Self-adaptive Systems and Applications, pages 93-102, 2012.

Works at TU Dresden

- J. Reimann, M. Seifert, U. Aßmann. Role-based generic model refactoring. MODELS Okt. 2010

- Thomas Kühn, Max Leuthäuser, Sebastian Götz, Christoph Seidl, and Uwe Aßmann. A metamodel family for role-based modeling and programming languages. In Benoit Combemale, David J. Pearce, Olivier Barais, and Jurgen J. Vinju, editors, SLE, volume 8706 of Lecture Notes in Computer Science, pages 141--160. Springer, 2014.

- Thomas Kühn, Stephan Böhme, Sebastian Götz, and Uwe Aßmann. A combined formal model for relational context-dependent roles. In Richard F. Paige, Davide Di Ruscio, and Markus Völter, editors, SLE, pages 113--124. ACM, 2015.

- Johannes Mey, René Schöne, Görel Hedin, Emma Söderberg, Thomas Kühn, Niklas Fors, Jesper Öqvist, and Uwe Aßmann. Continuous model validation using reference attribute grammars. In Proceedings of the 11th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2018, pages 70--82, New York, NY, USA, 2018. ACM.

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

104   DRESDEN
       concept

Other PhD Theses (all available via www.qucosa.de)

- Mirko Seifert. Designing Round-Trip Systems by Model Partitioning and Change Propagation. PhD thesis, Dresden University of Technology, June 2011.

    – Shows how roles simplify round-trip engineering by partitioning data

- Sebastian Richly. Autonom rekonfigurierbare Workflows. PhD thesis, Dresden University of Technology, December 2011.

    – shows how roles can be used to provide an extensible tool platform

- Christian Wende. Language Family Engineering. PhD thesis, Dresden University of Technology, March 2012.

    – shows how roles can be used to do context-based language composition

TECHNISCHE UNIVERSITÄT DRESDEN

Role-Oriented Context-Aware Software Infrastructures (ROSI)
Prof. Uwe Aßmann

105 DRESDEN concept