Reutlingen University

06.06.2019, DBKDA 2019, Athens

Data Modelling using the Labeled Property Graph

Fritz Laux
Prof. emeritus
Reutlingen University
Dept. of Informatics
Reutlingen, Germany

fritz.laux@reutlingen-university.de

© F. Laux

Reutlingen University

Aim of the Talk

Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

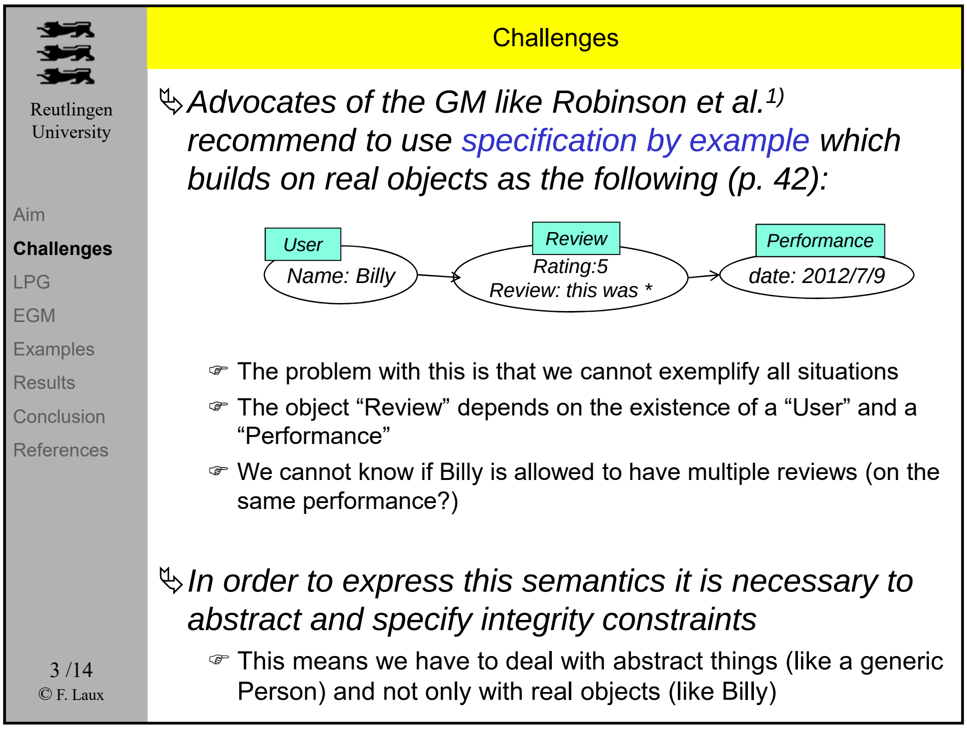
↳ *Enhance the Graph Model (GM) for data modelling and answer the following questions:*

- ☞ Is the GM suitable for data schemas?
  - ⇨ Which enhancements to the GM are needed?
  - ⇨ Is it better matching the way we communicate reality?
  - ⇨ What is the semantic expressiveness of the GM?
  - ⇨ Is there support for multiple abstraction levels?

↳ *Contents*

- ☞ Present the GM with some enhancements for our purpose
  - ⇨ Formally compact, yet sufficient for the target aim
- ☞ Apply and compare the GM to prevailing data models
  - ⇨ Show and discuss the results (benefits and pitfalls)

2 /14
© F. Laux



Reutlingen University

- Aim
- **Challenges**
- LPG
- EGM
- Examples
- Results
- Conclusion
- References

Challenges

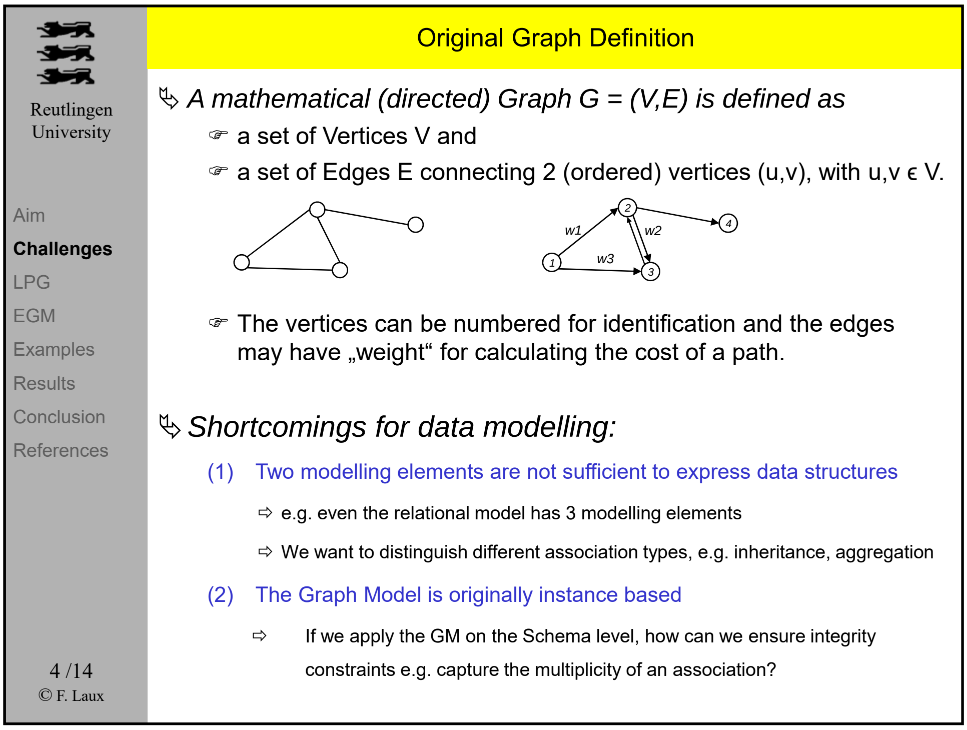
- *Advocates of the GM like Robinson et al.[1)] recommend to use specification by example which builds on real objects as the following (p. 42):*

User
Name: Billy
Review
Rating:5
Review: this was *
Performance
date: 2012/7/9

  - The problem with this is that we cannot exemplify all situations
  - The object "Review" depends on the existence of a "User" and a "Performance"
  - We cannot know if Billy is allowed to have multiple reviews (on the same performance?)

- *In order to express this semantics it is necessary to abstract and specify integrity constraints*
  - This means we have to deal with abstract things (like a generic Person) and not only with real objects (like Billy)

3 /14
© F. Laux



Reutlingen University

- Aim
- **Challenges**
- LPG
- EGM
- Examples
- Results
- Conclusion
- References

Original Graph Definition

- *A mathematical (directed) Graph G = (V,E) is defined as*
  - a set of Vertices V and
  - a set of Edges E connecting 2 (ordered) vertices (u,v), with u,v ϵ V.

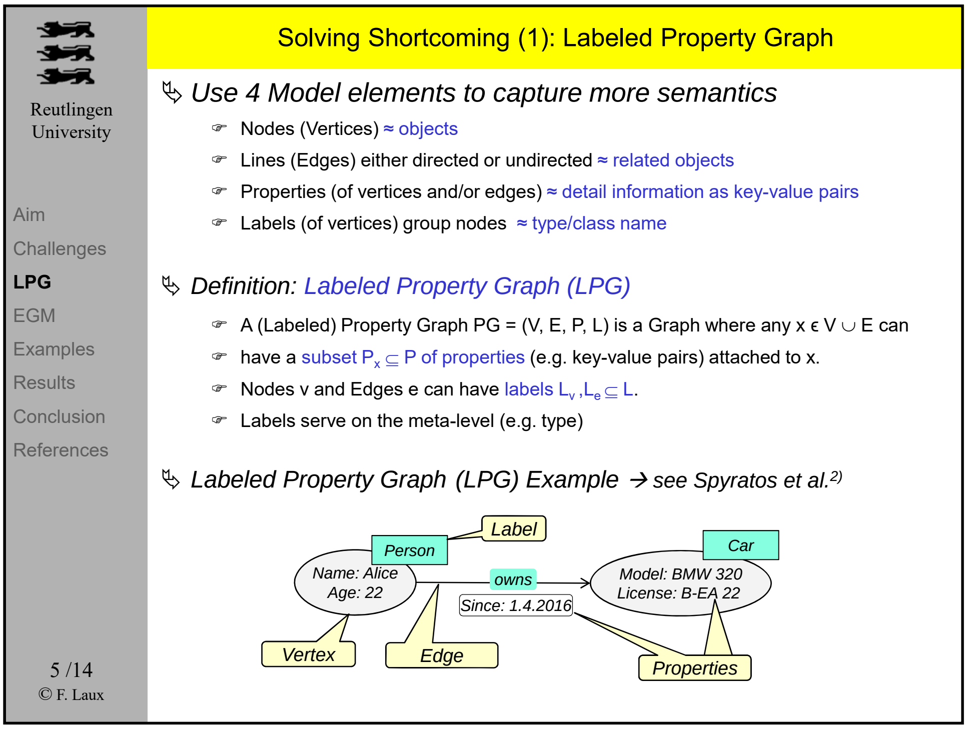
w1
w2
w3
1
2
3
4

  - The vertices can be numbered for identification and the edges may have „weight" for calculating the cost of a path.

- *Shortcomings for data modelling:*
  (1) Two modelling elements are not sufficient to express data structures
  - ⇨ e.g. even the relational model has 3 modelling elements
  - ⇨ We want to distinguish different association types, e.g. inheritance, aggregation

  (2) The Graph Model is originally instance based
  - ⇨ If we apply the GM on the Schema level, how can we ensure integrity constraints e.g. capture the multiplicity of an association?

4 /14
© F. Laux



Reutlingen University

Aim

Challenges

LPG

EGM

Examples

Results

Conclusion

References

5 /14

© F. Laux

Solving Shortcoming (1): Labeled Property Graph

Use 4 Model elements to capture more semantics

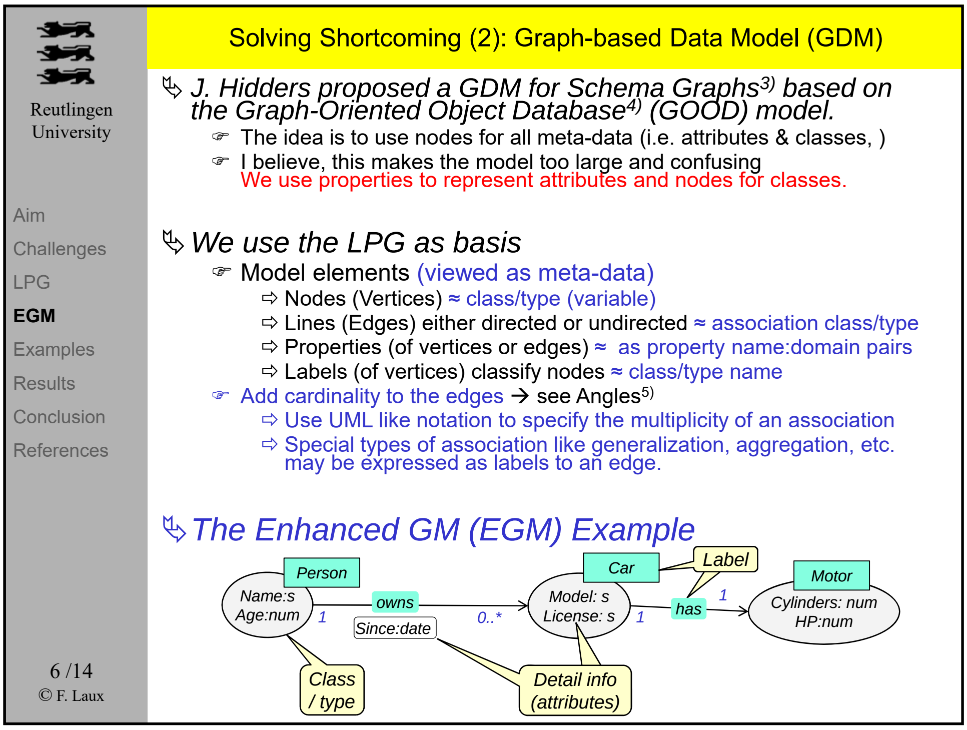
- ☞ Nodes (Vertices) ≈ objects
- ☞ Lines (Edges) either directed or undirected ≈ related objects
- ☞ Properties (of vertices and/or edges) ≈ detail information as key-value pairs
- ☞ Labels (of vertices) group nodes ≈ type/class name

Definition: Labeled Property Graph (LPG)

- ☞ A (Labeled) Property Graph PG = (V, E, P, L) is a Graph where any x ϵ V ∪ E can
- ☞ have a subset $P_x \subseteq P$ of properties (e.g. key-value pairs) attached to x.
- ☞ Nodes v and Edges e can have labels $L_v, L_e \subseteq L$.
- ☞ Labels serve on the meta-level (e.g. type)

Labeled Property Graph (LPG) Example → see Spyratos et al.[2]

Label
Person
Name: Alice
Age: 22
owns
Since: 1.4.2016
Car
Model: BMW 320
License: B-EA 22
Vertex
Edge
Properties



Reutlingen University

Aim

Challenges

LPG

EGM

Examples

Results

Conclusion

References

6 /14

© F. Laux

Solving Shortcoming (2): Graph-based Data Model (GDM)

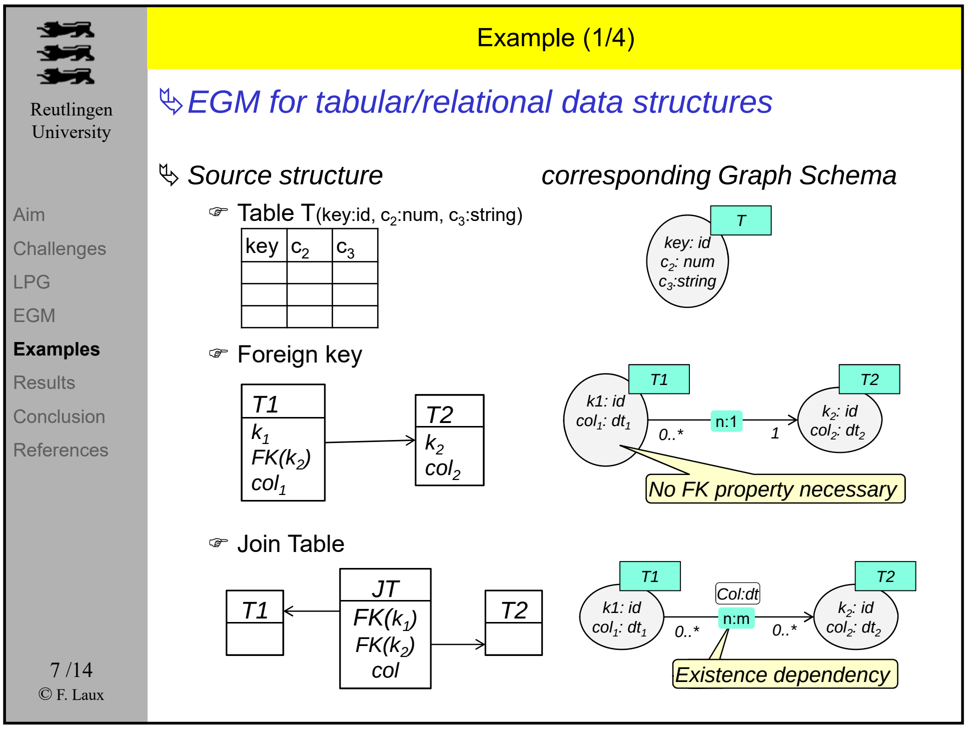
J. Hidders proposed a GDM for Schema Graphs[3] based on the Graph-Oriented Object Database[4] (GOOD) model.

- ☞ The idea is to use nodes for all meta-data (i.e. attributes & classes,)
- ☞ I believe, this makes the model too large and confusing
  We use properties to represent attributes and nodes for classes.

We use the LPG as basis

- ☞ Model elements (viewed as meta-data)
  - ⇨ Nodes (Vertices) ≈ class/type (variable)
  - ⇨ Lines (Edges) either directed or undirected ≈ association class/type
  - ⇨ Properties (of vertices or edges) ≈ as property name:domain pairs
  - ⇨ Labels (of vertices) classify nodes ≈ class/type name
- ☞ Add cardinality to the edges → see Angles[5]
  - ⇨ Use UML like notation to specify the multiplicity of an association
  - ⇨ Special types of association like generalization, aggregation, etc. may be expressed as labels to an edge.

The Enhanced GM (EGM) Example



Reutlingen University

Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

7 /14
© F. Laux

Example (1/4)

EGM for tabular/relational data structures

Source structure | corresponding Graph Schema

- Table T(key:id, c_2:num, c_3:string)

key	c_2	c_3

T
key: id
c_2: num
c_3:string

- Foreign key

T1
k_1
FK(k_2)
col_1

T2
k_2
col_2

T1
k1: id
col_1: dt_1
0..*
n:1
1
T2
k_2: id
col_2: dt_2

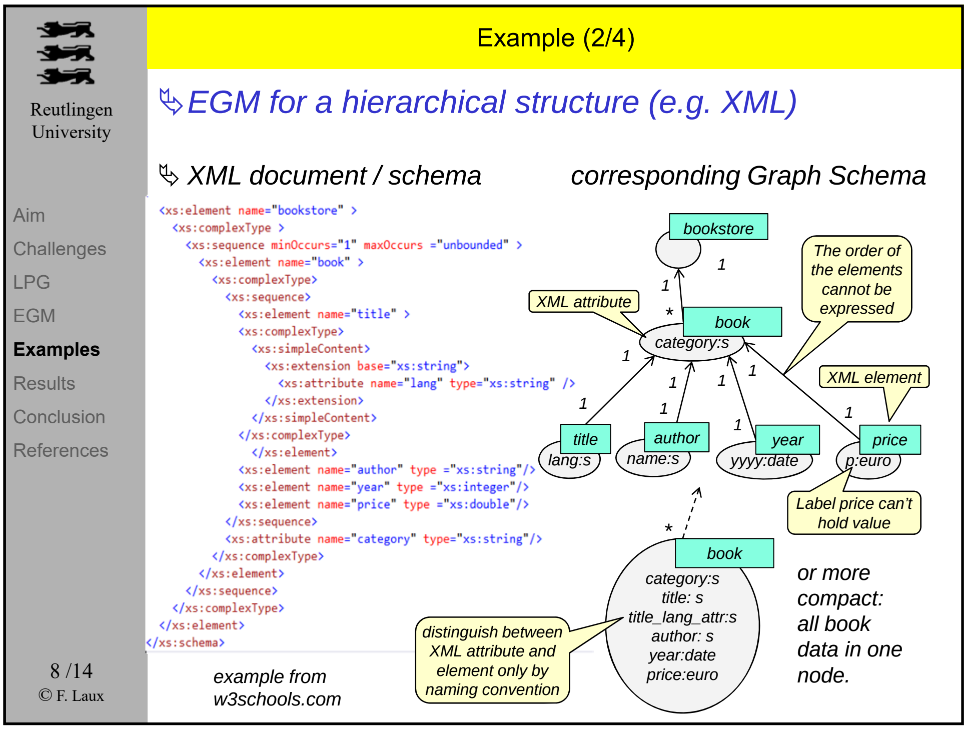
No FK property necessary

- Join Table

T1
JT
FK(k_1)
FK(k_2)
col
T2

T1
k1: id
col_1: dt_1
0..*
Col:dt
n:m
0..*
T2
k_2: id
col_2: dt_2

Existence dependency



Reutlingen University

Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

8 /14
© F. Laux

Example (2/4)

EGM for a hierarchical structure (e.g. XML)

XML document / schema | corresponding Graph Schema

```
  <xs:element name="bookstore" >
    <xs:complexType >
      <xs:sequence minOccurs="1" maxOccurs ="unbounded" >
        <xs:element name="book" >
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title" >
              <xs:complexType>
                <xs:simpleContent>
                  <xs:extension base="xs:string">
                    <xs:attribute name="lang" type="xs:string" />
                  </xs:extension>
                </xs:simpleContent>
              </xs:complexType>
                </xs:element>
              <xs:element name="author" type ="xs:string"/>
              <xs:element name="year" type ="xs:integer"/>
              <xs:element name="price" type ="xs:double"/>
            </xs:sequence>
            <xs:attribute name="category" type="xs:string"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

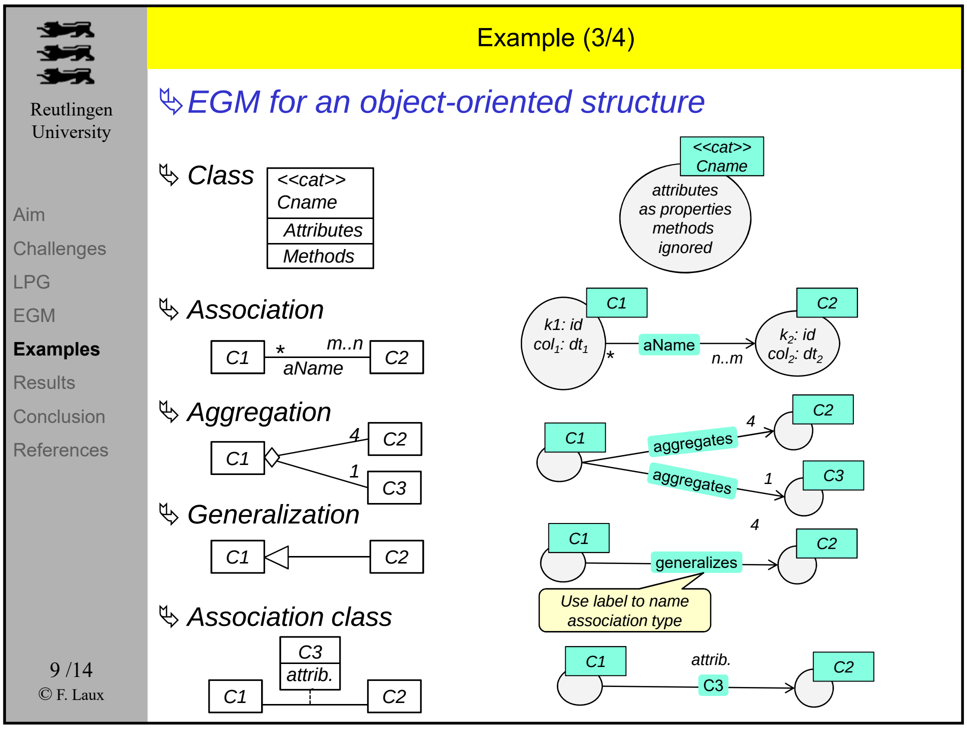
example from
w3schools.com

bookstore
1
1
*
book
category:s
XML attribute
The order of the elements cannot be expressed
XML element
1
1
1
1
1
1
1
1
title
lang:s
author
name:s
year
yyyy:date
price
p:euro
Label price can't hold value

*
book
category:s
title: s
title_lang_attr:s
author: s
year:date
price:euro

distinguish between XML attribute and element only by naming convention

or more compact:
all book data in one node.



Reutlingen University

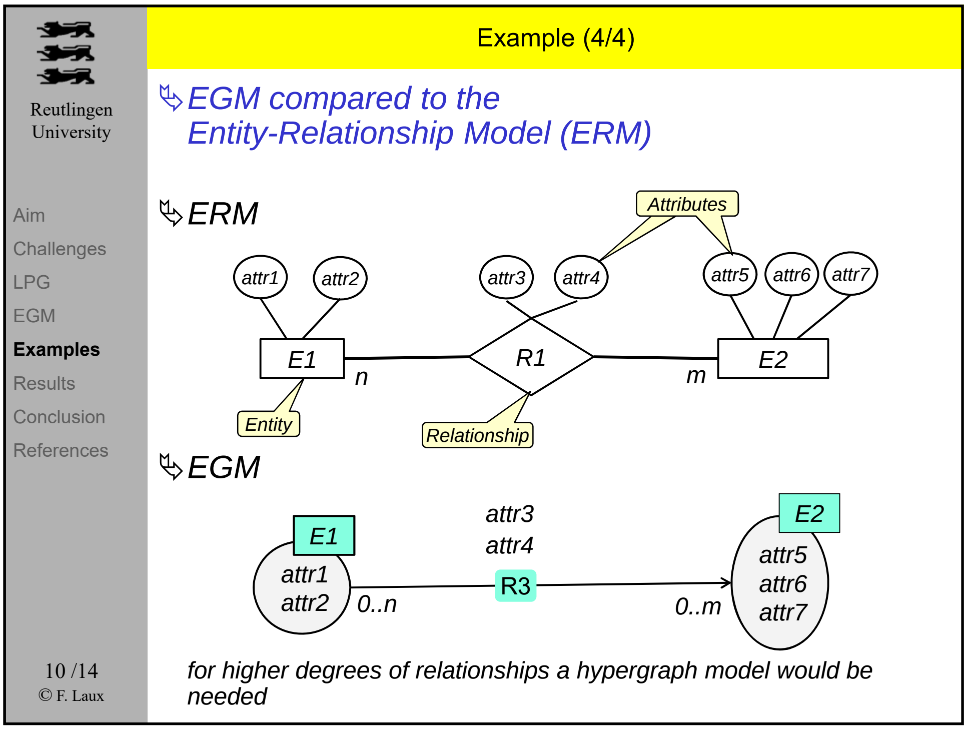
Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

9 /14
© F. Laux

Example (3/4)

EGM for an object-oriented structure

- Class
- Association
- Aggregation
- Generalization
- Association class



Reutlingen University

Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

10 /14
© F. Laux

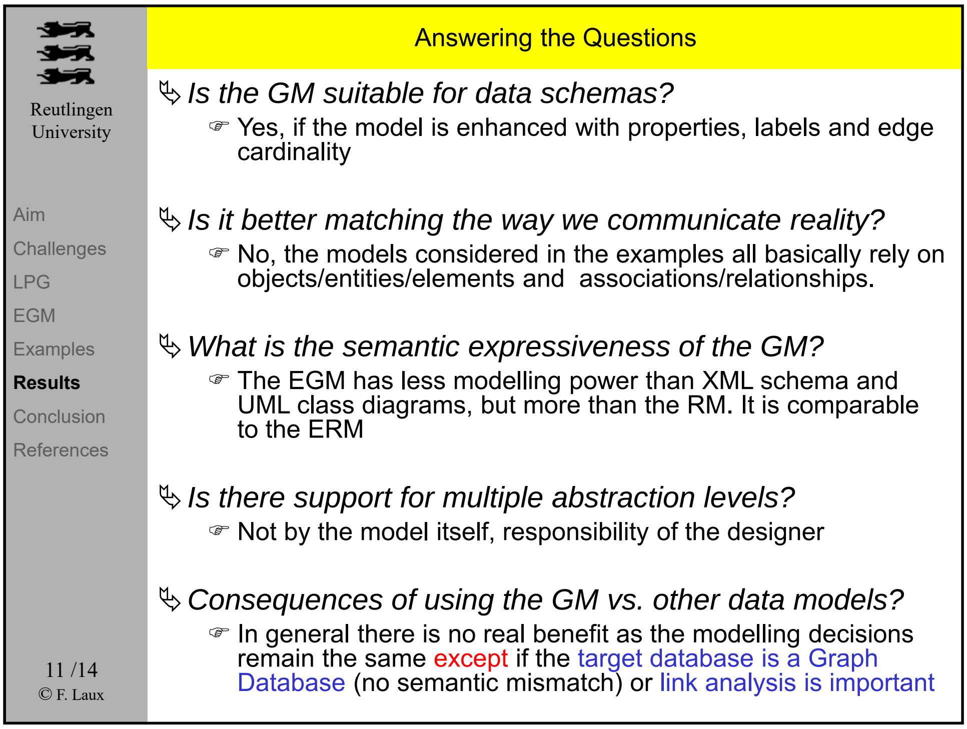
Example (4/4)

EGM compared to the Entity-Relationship Model (ERM)

- ERM

- EGM

for higher degrees of relationships a hypergraph model would be needed



Reutlingen University

Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

11 /14
© F. Laux

Answering the Questions

Is the GM suitable for data schemas?

- Yes, if the model is enhanced with properties, labels and edge cardinality

Is it better matching the way we communicate reality?

- No, the models considered in the examples all basically rely on objects/entities/elements and associations/relationships.

What is the semantic expressiveness of the GM?

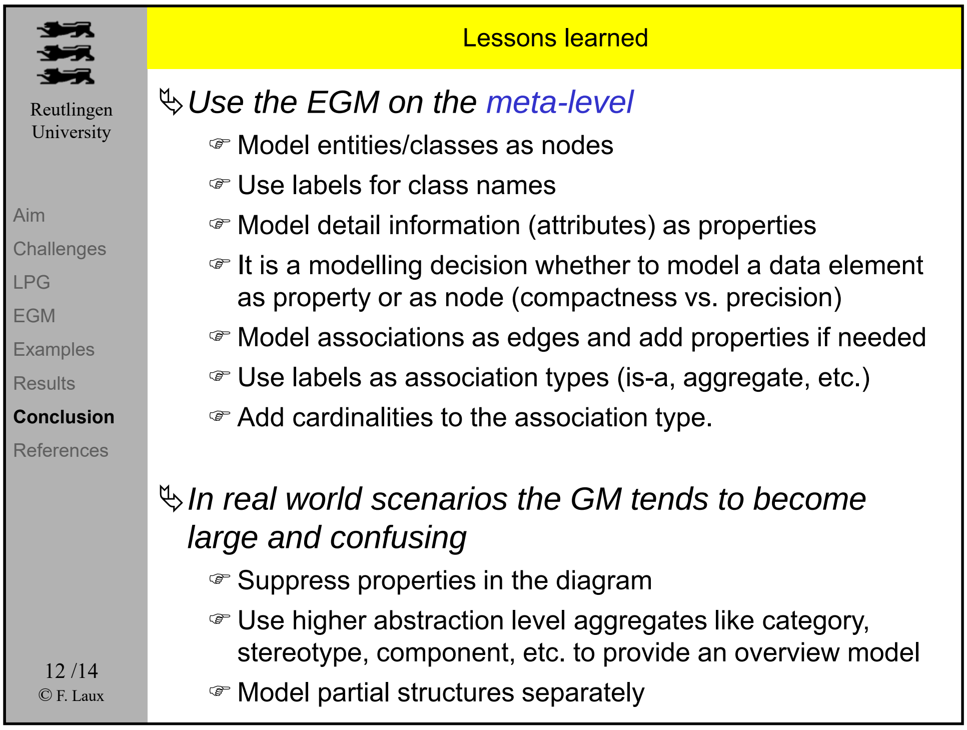
- The EGM has less modelling power than XML schema and UML class diagrams, but more than the RM. It is comparable to the ERM

Is there support for multiple abstraction levels?

- Not by the model itself, responsibility of the designer

Consequences of using the GM vs. other data models?

- In general there is no real benefit as the modelling decisions remain the same except if the target database is a Graph Database (no semantic mismatch) or link analysis is important



Reutlingen University

Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

12 /14
© F. Laux

Lessons learned

Use the EGM on the meta-level

- Model entities/classes as nodes
- Use labels for class names
- Model detail information (attributes) as properties
- It is a modelling decision whether to model a data element as property or as node (compactness vs. precision)
- Model associations as edges and add properties if needed
- Use labels as association types (is-a, aggregate, etc.)
- Add cardinalities to the association type.

In real world scenarios the GM tends to become large and confusing

- Suppress properties in the diagram
- Use higher abstraction level aggregates like category, stereotype, component, etc. to provide an overview model
- Model partial structures separately

Reutlingen University

Aim
Challenges
LPG
EGM
Examples
Results
Conclusion
References

References

1) *I. Robinson et al.: Graph Databases, 2nd ed., O'Reilly Media, 2015*

2) *N. Spyratos and T. Sugibuchi: PROPER - A Graph Data Model Based on Property Graphs,* ISIP – 10th International Workshop, Communications in Computer and Information Science, vol.622, Springer, 2015, pp. 23-35

3) *J. Hidders: "Typing Graph-Manipulation Operations", Proc. 9th International Conference on Database Theory (ICDT), 2003, pp. 391-406*

4) *M. Gyssens et al.: "A graph-oriented object database model", IEEE Transactions on Knowledge and Data Engineering, Vol. 6, Num. 4, 1994, pp. 572–586*

5) *R. Angles: The Property Graph Database Model,* Proc. 12th Alberto Mendelzon International Workshop on Foundations of Data Management, CEUR WS Proc., 2018, URL: http://ceur-ws.org/Vol-2100/paper26.pdf

13 /14
© F. Laux

Reutlingen University

Discussion

14 /14
© F. Laux