Generic vs. Domain-specific Modeling Languages

Knut Hinkelmann

Generic vs. Domain-specific Modeling Languages

- Domain-specific languages are notation which are defined to model knowledge about a specific domain
- Generic modeling languages can be used to represent any kind of knowledge

Domain-specific Modeling Languages

- Domain-specific languages have modeling elements for representing concepts and relations of a domain of discourse
- Examples:
  - **BPMN** is a domain-specific language for business processes
    - Modeling elements: task, event, gateway, ….
    - relations: sequence flow, message flow, data association, …
  - **BMM** is a domain-specific language for business motivation
    - Modeling elements: vision, mission, goal, strategy, influencer, …
    - relations: judges, channels efforts, …
  - **ArchiMate** is a domain-specific language for enterprise architectures
    - Modeling elements: process, actor, role, business object, …
    - relations: uses, realizes, …

Generic Modeling Languages

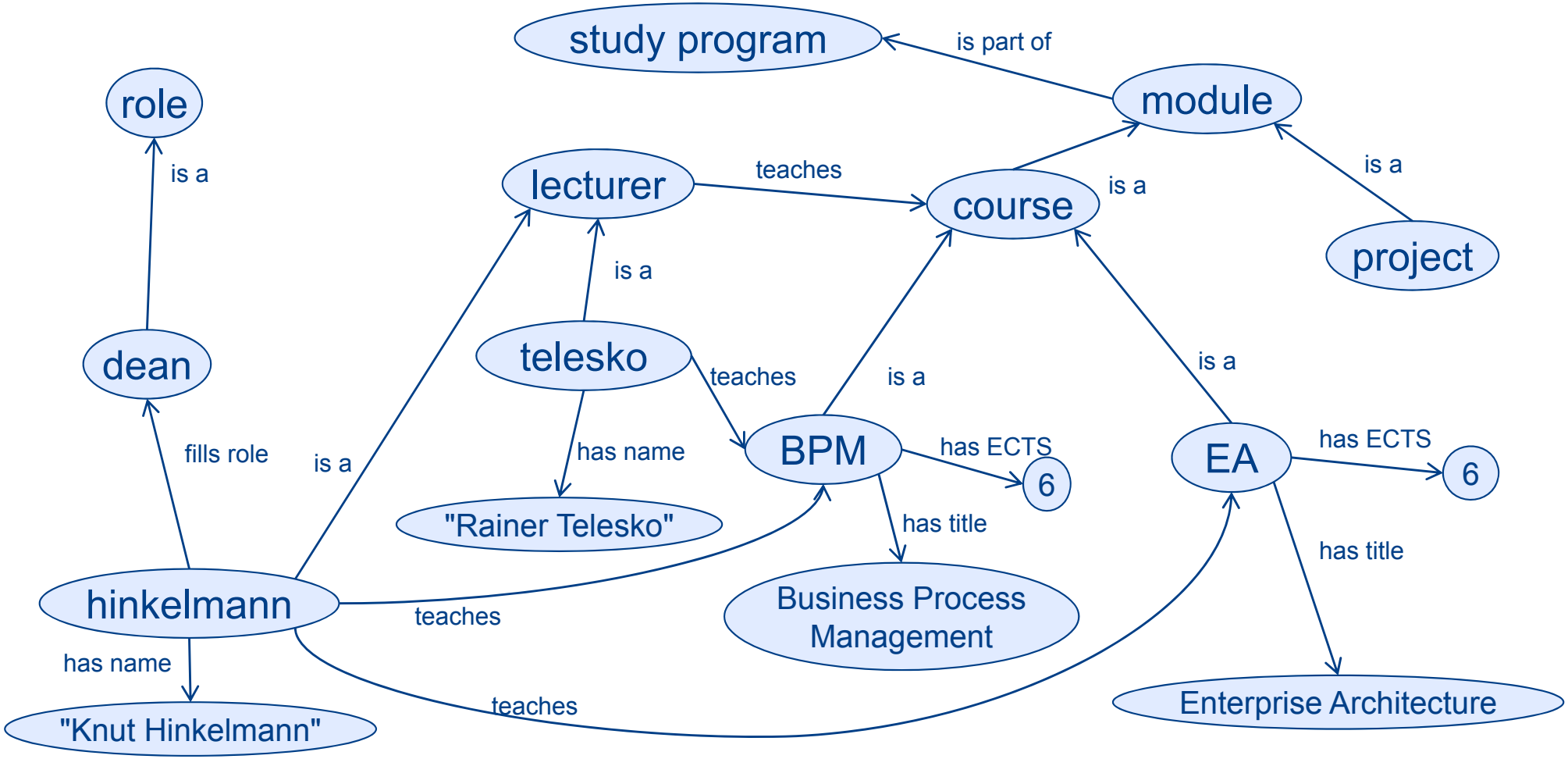
- Generic modeling languages can be used to represent any kind of knowledge
- There are a wide range of generic modeling languages
  - Natural language allows to express any knowledge
  - Formal languages: Typically a subset of Logic
  - Graphical Notations

Graphical Diagrams for Generic Modeling

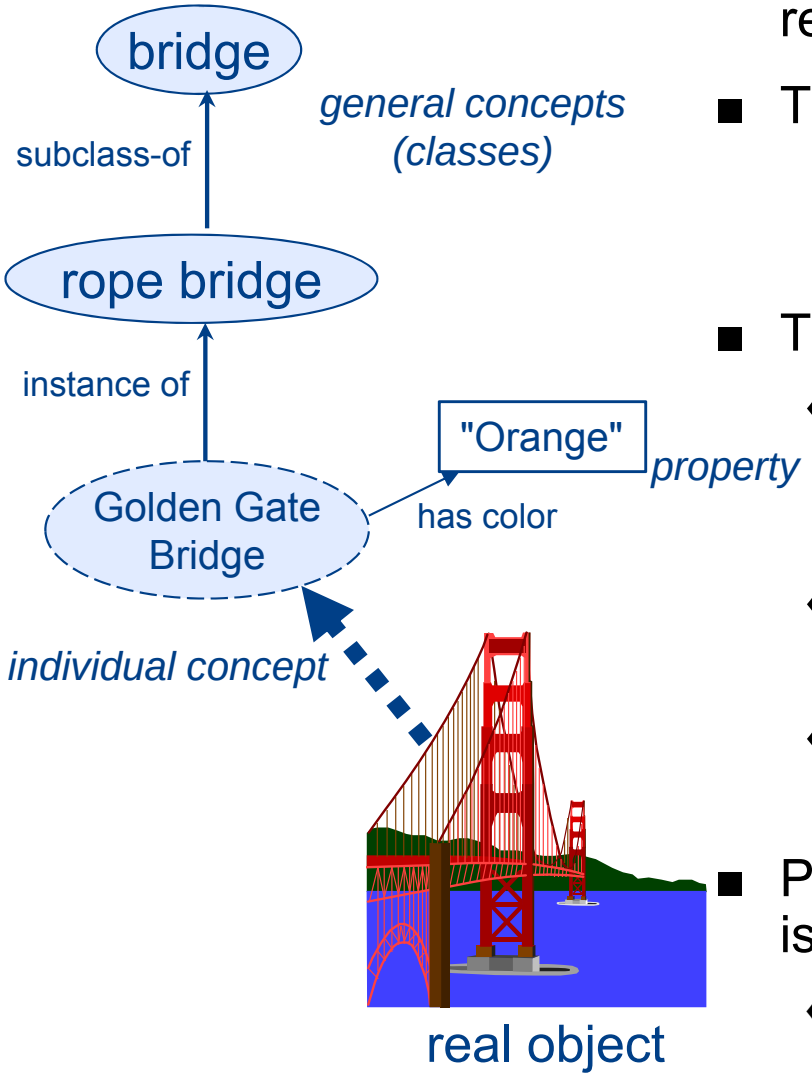
- Generic graphical modeling languages have been developed in a many difference fields:
  - Artificial Intelligence: Semantic networks, Description Logics
  - Data Modeling: Entity Relationship Diagrams
  - Object-Oriented Programming: UML Class Diagrams
- Although having different notations these generic modeling languages typically allow to represent
  - **objects**
  - **properties** of objects
  - **relations** between objects
  - **constraints** (e.g. cardinalities, type restrictions)

A Semantic Network

A Semantic Network consists of nodes and relations.
Example: This is a semantic network about university courses.

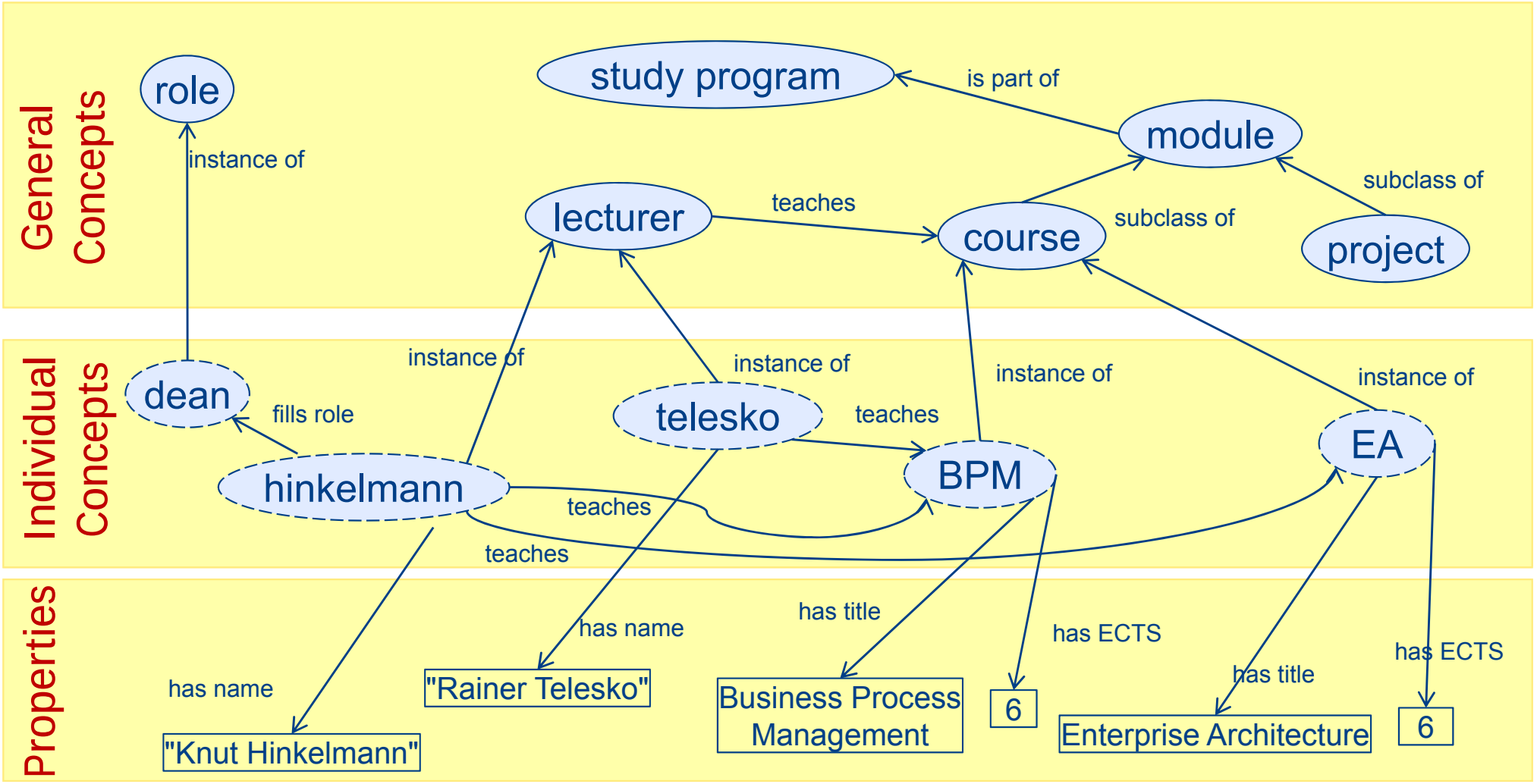


Nodes and Relations in a Semantic Networks



- It is now common to distinguish different kinds or relations and nodes in a semantic network
- There are two kinds of nodes:
  - general concepts (also called classes)
  - individual concepts (also called instances)
- There are different kinds of relationships
  - generalisation ("is a")
    - between classes (subclass of)
    - between individual and class (instance of)
  - aggregation
    - "part-of" relationship
  - associations
    - any other kind of relationship
- Properties can be regarded as associations whose value is not node but is of a primitive type (number, string).
  - That value of a property does not have relations or properties

A Semantic Network distinguishing Kinds of Nodes



UML Class Diagrams and Object Diagrams

- Subsets of UML Class Diagrams and UML Object Diagrams are often used as generic modeling language
  - classes
  - attributes
  - associations (incl. generalisation, aggregation, composition)
  - objects

- UML Class diagrams where originally designed for modeling in object-oriented programming. This is why they contain operations, which are not relevant for most modeling languages

The Semantic Network modeled in UML