Ontology-based Enterprise Modeling for Human and Machine Interpretation

Prof. Knut Hinkelmann

knut.hinkelmann@fhnw.ch
Why Modeling

- If the object you want to create or change is simple, then you can do it directly.
- For complex systems that are likely to change over time, you need a model.
- Without explicit modeling there is a high risk that the implementation is not what is intended.

(John Zachmann, 2012)
Business Process Management

- Process Design
- Process Optimization
- Process Digitalization
- ...
Enterprise Architecture
Graphical Models are appropriate for Humans

Communication/
Analysis/
Decision Making

Models

human-interpretable models

Reality
Models

- Models are not mere pictures; rather, they
  - provide a precise, meaningful description that can be visualized in different ways for different stakeholders;
  - can also be used to analyze the impact of changes, cost, risk, security, compliance and other relevant KPIs.
Models should allow automated analysis, decision making and digitalization
Graphical Models are Represented in a Database

Models

human interpretable

machine processable

application

Knowledge

Data

Scripts

Reality

Example: Adoxx

Prof. Dr. Knut Hinkelmann
Modelling Environment

Definition of syntax and (type) semantics

- Meta^2-model layer
- Metamodel layer
- Model layer

Notations and classes
The Semantics of a model language is defined by:
- Classes of elements and relations
- Class hierarchy
- Attributes of the elements

The Syntax is defined by notation:
- Adoxx: attribute GraphRep

Metamodel: Semantics and Syntax of a Modeling Language
Class Hierarchies

- ADOxx distinguishes
  - Classes
  - Relation classes
Attributes

- Kinds of Attributes
  - Properties of Models
  - Graphical Representation
  - References
Notation

GraphRep: A script language for the graphical representation
Appearance of Classes in the Modelling Toolkit
Change of Metamodel

- Example: new task type Cloud Task

Metamodel Engineer

Metamodel

Cloud Task

Notation Design

Metamodeling

Feedback
Amendments
Improvements

Modeling

Modelling Environment

Modeler
The AMME LifeCycle
Agile Modeling Method Engineering

(Karagiannis 2015)
Knowledge in Models
Interpretation of Models
Dimensions of a Knowledge Space

**Dimensions of the Knowledge Space**

**Use:** Stakeholders and their concerns determine the relevant subset of the knowledge

- process optimization requires knowledge about time and costs
- selection of a cloud service require knowledge about data and functionality

**Form:** Syntax and semantic of modeling language elements.

**Content:** Domain in which knowledge engineering is applied, is represented in the labels

**Interpretation:** Giving meaning to a model:
- Graphical models are cognitively adequate for human
- Machines need more formal representation
Making the Knowledge in Models explicit

■ Humans «know» the meaning of the modeling objects.
  ♦ Elements of the model language
  ♦ Labels represent domain knowledge

■ Examples:

  ♦ Model element: Application Component
  ♦ Domain: «ERP System» is business software

  ♦ Model element: Task
  ♦ Domain: «Cook pasta» is about preparing food

■ The objective is to represent the knowledge so that it can be interpreted by a system for decision making and problem solving.
Semantic Lifting
Semantic Lifting: Map Models into an Ontology

Models

Reality

Knowledge

reasoning

application

graphical models
(human interpretable)

data models
(machine processable)

Knowledge base
(machine interpretable)

Knowledge

Data Scripts

Rules Ontology
Semantic Lifting: Map Models into an Ontology

Modelling Environment

- Meta²-model layer
- Metamodel layer
- Model layer

Definition of syntax and (type) semantics

Ontology

- Semantics definition: commonly accepted ontology

ontological metamodelling (lifting): explication of type semantics
Semantic Lifting

- Map models into an ontology
  - Semantics: Classes of the metamodel are aligned with classes in the ontology
  - Interpretation: For each element in a model an instance of the ontology is created
  - Content: Model elements are annotated with domain knowledge from ontology
  - Inference of the ontology can be applied to the knowledge base
Example: Business Process as a Service

BPaaS Modelling Environment

- Meta model
- Models

Workflow Process

BPaaS Ontology

- Classes/Rules
- Inference

Instances

human interpretation
informal and semi-formal

machine interpretation
formal

semantic alignment

semantic annotations

transformation and mapping

From: CloudSocket Project
Example: Business Process as a Service

The semantics of the meta-model elements is defined in the BPaaS Ontology.

BPaaS Modelling Environment

- Meta model
- Models
  - Workflow Process

BPaaS Ontology

- Classes/Rules
  - Semantic annotations
- Instances
  - Transformation and mapping

Inference

- Smart Business and IT in the Cloud Alignment

From: CoudSocket Project
Semantic Alignment

The ontology contains classes for all modelling elements

BPMN Modelling Language in ADOxx

BPMN Ontology
Transformation and Mapping

The model elements are exported as instances ontology classes
**Semantic Annotations**

Annotate modeling elements with classes from the domain ontology

**Example: Functionality of a Service**

![Diagram showing the process of annotating function with external functional ontology and a list of process classification framework categories.]

**Domain Ontology:**
**APQC Process Classification Framework**

- owl:Thing
  - American Productivity and Quality Center
    - Acquire, Construct, and Manage Assets
    - Deliver Physical Products
    - Deliver Services
    - Develop and Manage Business Capabilities
    - Develop and Manage Human Capital
    - Develop and Manage Products and Services
    - Develop Vision and Strategy
    - Manage Customer Service
    - Manage Enterprise Risk, Compliance, Remediation, and Resiliency
    - Manage External Relationships
    - Manage Financial Resources
    - Manage Fixed-Asset Project Accounting
    - Manage Internal Controls
    - Manage International Funds/Consolidation
    - Manage Taxes
    - Manage Treasury Operations
    - Perform General Accounting and Reporting
    - Perform Global Trade Services
    - Perform Planning and Management Accounting
    - Perform Revenue Accounting
    - Invoice Received
  - Generate Customer Billing Data
    - Maintain Customer/Product Order Files
    - Post Receivable Entries
    - Resolve Customer Billing Inquiries
    - Transmit Billing Data to Customers
    - Manage and Process Adjustments/Deductions
    - Manage and Process Collections
    - Process Accounts Receivable (AR)
    - Process Customer Credit
    - Process Accounts Payable and Expense Reimbursements
    - Process Payroll
    - Manage Information Technology (IT)
    - Market and Sell Products and Services
Application Example for Semantic Lifting

Cloud Service Selection

Functionality

- APQC category that reflect the functional requirement:
  type to search *

- Action that reflect the functional requirement:
  type to search *

- Object that reflect the functional requirement:
  type to search *

Non-functional requirements

- Select your preferred payment plan:
  - Prepaid Annual Plan
  - Try Free First
  - Customizable Plan
  - Monthly Fee
  - None

Performance

- Monthly Availability in %:
  Insert your value here *
Drawbacks of Semantic Lifting

- Separate Environments for
  - Modelling
  - Knowledge Base (Inferencing)

- Inconsistency: Both metamodel and ontology must be aligned but are maintained independently:
  - Metamodel and ontology must represent the same semantics
  - Each change in metamodel must be reproduced in the ontology and vice versa

- Effort: After each change the models must be translated again into the ontology instances
Example: New Model Element

- New task type: Cloud Task

Change in the meta model:

Change in the ontology:
Ontology-based Metamodelling
Ontology-based Metamodelling

Models

graphical models (human interpretable)

Knowledge base (machine interpretable)

Reality

Knowledge
Ontology-based Metamodelling (1):

Metamodel is represented as an Ontology
Modelling Language Ontologies

BPMN

Archimate

Invoice

ERP

Send invoice
Ontology-based Metamodelling (2): Ontologies for Metamodel and Content

- meta²-model layer
  - meta²-model (e.g. GraphRep)
- metamodel layer
  - Notation
  - Language ontology (Abstract Syntax)
  - Domain ontology (Semantics)
- model layer
  - Abstract Syntax

(Laurenzi et al. 2018)
Domain Ontologies

Enterprise Ontology (excerpt)

Domain Ontology: APQC Process Classification Framework

Action type
Object
APQC Class

Send invoice

Prof. Dr. Knut Hinkelmann
Ontology-Based Modeling

- Single environment for modelling and ontology
- Model elements are directly created as instances in the ontology
Agile Meta-Modeling
Objective

Adapt modeling languages and ensure a precise shared interpretation of new modeling constructs to both **humans and machines**
Integration Modeling and Metamodelling in a Single Environment

- Modeling and metamodelling in a single environment
- Tight collaboration between metamodel developer and modeler
- Modeler can also take the role of metamodel developer

(Laurenzi et al. 2018)
Integration of Meta-modeling and Modeling: On-the-fly Modeling Language Adaptation

- Name of the parent class
- Name for the new modeling construct
- Graphical notation to be shown in palette and canvas
- Create New Modeling construct and store in the ontology

(Laurenzi et al. 2018)
Agile and Ontology-Aided Modeling Environment (AOAME)

Ontology-based Models (human- and machine-interpretable)

Models + Knowledge

Reality

(Laurenzi et al. 2018)