Introduction to modeling

12.10.2009

Lecture 1

Course overview

• Introduces modeling as a discipline within Computer Science and Engineering, and some of the the most important modeling paradigms of the last decades.
  – Core primitives and principles.
  – ER modeling.
    • Entity relationship modeling
    • Conceptual schema
    • Entities, relationships, and attributes
  – OO modeling.
    • Object oriented modeling
    • Computer programming
Course overview (cont’d)

• Ontologies.
  – Formal representation of a shared conceptualization of a domain.
  – Concepts and relationships, axioms.
  – Reasoning.
  – Semantic Web.

• Services and process modeling.
  – (Semantic) descriptions of Web services.
  – Representation of processes within an enterprise.

• Modeling best practices.

Overall learning goal

• Get an overview on the most important principles of (domain) modeling.

• Learn how to model correctly and usefully.

• Learn some well-known modeling paradigms, their commonalities and differences.
Administrative issues

- The course is structured as VO1 + PS1

- Both VO and PS will be held biweekly (VO 13.15 – 14.45, PS 15.00 – 16.30)

- Attendance policy
  - **VO**
    - This is a graduate course, assignments require material discussed in class in addition to slides, textbooks and additional reading.
    - Attendance not mandatory.
  - **PS**
    - Students are allowed to miss maximally 2 hours of the total number of PS hours.
    - Attendance mandatory.

About the lecturer

Dr. Katharina Siorpaes  
University of Innsbruck  
Technikerstr. 21a  
6020 Innsbruck, Austria  
e: katharina.siorpaes@sti2.at

*Office hours: only by appointment. Please contact me per e-mail at katharina.siorpaes@sti2.at*
Examination and grading

• VO
  – **Examination**: Last session of the term.
  – If needed, dates for a second and a third examination will be proposed.

• PS
  – **Attendance**: Students are allowed to miss maximally 2 hours of the total PS time.
  – **Assignments**: 6 assignments in total, at least 4 assignments should be successfully solved by every student; the solutions developed by each student will be evaluated during the PS.
  – **Presentation**: every student should give at least two assignment presentations in front of the other students during the PS.

Resources

• Course home page
  [http://www.sti-innsbruck.at/teaching](http://www.sti-innsbruck.at/teaching)

• Textbooks (selected chapters) and additional reading available for each lecture.

• Please subscribe to the mailinglist! (see website)
Course syllabus

- Mo 12.10. 13.15 - 14.45
- Mo 09.11. 13.15 - 14.45
  - Modeling elements. Core relationship types and their properties.
- Mo 23.11. 13.15 - 14.45
  - ER modeling. Basic modeling elements, foundational theory, examples. Differences to other modeling paradigms.
  - OO modeling. Basic modeling elements, UML. Differences to other modeling paradigms.
- Mo 07.12. 13.15 - 14.45
  - Ontologies. Definitions, types, modeling elements, RDFS, OWL, ontology reasoning. Differences to other modeling paradigms

Course syllabus (cont)

- Mo 11.01. 13.15 - 14.45
  - Process modeling. Modeling elements, BPMN, BPEL, UML. Differences to other modeling paradigms.
- Mo 25.01. 13.15 - 14.45
  - Examination
- Mo 25.01. 15.00 – 1730
  - Proseminar
INTRODUCTION AND MOTIVATION

Models

• **A model (plural models)**
  - A person who serves as a subject for artwork or fashion, usually in the medium of photography but also for painting or drawing.
  - The beautiful model had her face on the cover of almost every fashion magazine imaginable.
  - A miniature representation of.
  - The boy played with a model of a World War II fighter plane.
  - A simplified representation (usually mathematical) used to explain the workings of a real world system or event.
  - The computer weather *model* did not correctly predict the path of the hurricane.
  - A style, type, or design.
  - He decided to buy the turbo engine *model* of the sports car.
  - The structural design of a complex system.
  - The team developed a sound business *model*.
  - A praiseworthy example to be copied, with or without modifications.
    - British parliamentary democracy was seen as a *model* for other countries to follow.

• From Latin *modellus*, diminutive form of *modulus* 'small measure'.

[www.sti-innsbruck.at](http://www.sti-innsbruck.at)
General principles

- Abstraction.
  - A concept or idea not associated with any specific instance.
  - Example: the concept of a human represents many individuals.

- Classification.
  - The act of distributing things into classes or categories of the same type.

- Aggregation.
  - Several things grouped together or considered as a whole.

- Type of information captured: structural or dynamic information.

- Reusability: application-independent vs. application-dependent models; modularity.

Data models

- A **data model** is an abstract model that describes how data is represented and accessed.

- A **data model theory** is a formal description of how data may be structured and accessed.

- A **data model instance** is obtained from applying a data model theory to create a practical data model for some particular application.
Types of data models

- We differentiate among three types of data model instances (ANSI, 1975):
  - A **conceptual model** describes the semantics of a domain of interest in terms of modeling primitives.
  - A **logical model** describes the semantics, as represented by a particular data manipulation technology.
  - A **physical model** describes the physical means by which data is stored.

The Zachman Framework

- Enterprise architecture
- Formal and highly structured way of viewing and defining an enterprise
- Taxonomy
- Artifacts
  - Target of the artifact (e.g., owner or builder)
  - Particular issue (e.g., data and functionality)
- Matrix, 1980
- A framework to organize and analyze data,
- A framework for enterprise architecture.
- A classification system, or classification scheme
- A matrix, often in a 6x6 matrix format
- A two-dimensional model or an analytic model.
The Zachman Framework

http://www.zifa.com/

http://www.va.gov/OIRM/conference/Orlando2002/Monday/EnterpriseArchitectureFeb152002.ppt

The Zachman Framework
The Zachman Framework

**Contextual**
- *(Why) Goal List* – primary high level organization goals
- *(How) Process List* – list of all known processes
- *(What) Material List* – list of all known organizational entities
- *(Who) Organizational Unit & Role List* – list of all organization units, sub-units, and identified roles
- *(Where) Geographical Locations List* – locations important to organization; can be large and small
- *(When) Event List* – list of triggers and cycles important to organization

**Conceptual**
- *(Why) Goal Relationship Model* – identifies hierarchy of goals that support primary goals
- *(How) Process Model* – provides process descriptions, input processes, output processes
- *(What) Entity Relationship Model* – identifies and describes the organizational materials and their relationships
- *(Who) Organizational Unit & Role Relationship Model* – identifies enterprise roles and units and the relationships between them
- *(Where) Locations Model* – identifies enterprise locations and the relationships between them
- *(When) Event Model* – identifies and describes events and cycles related by time
The Zachman Framework

Logical

• (Why) Rules Diagram – identifies and describes rules that apply constraints to processes and entities without regard to physical or technical implementation
• (How) Process Diagram – identifies and describes process transitions expressed as verb-noun phrases without regard to physical or technical implementation
• (What) Data Model Diagram – identifies and describes entities and their relationships without regard to physical or technical implementation
• (Who) Role Relationship Diagram – identifies and describes roles and their relations to other roles by types of deliverables without regard to physical or technical implementation
• (Where) Locations Diagram – identifies and describes locations used to access, manipulate, and transfer entities and processes without regard to physical or technical implementation
• (When) Event Diagram – identifies and describes events related to each other in sequence, cycles occur within and between events, without regard to physical or technical implementation

Physical

• (Why) Rules Specification – expressed in a formal language; consists of rule name and structured logic to specify and test rule state
• (How) Process Function Specification – expressed in a technology specific language, hierarchical process elements are related by process calls
• (What) Data Entity Specification – expressed in a technology specific format; each entity is defined by name, description, and attributes; shows relationships
• (Who) Role Specification – expresses roles performing work and workflow components at the work product detailed specification level
• (Where) Location Specification – expresses the physical infrastructure components and their connections
• (When) Event Specification – expresses transformations of event states of interest to the enterprise
The Zachman Framework

- **Rule 1:** Columns have no order
- **Rule 2:** Each column has a simple, basic model
- **Rule 3:** Basic model of each column is unique
- **Rule 4:** Each row represents a distinct view
- **Rule 5:** Each cell is unique
- **Rule 6:** Combining the cells in one row forms a complete description from that view
- **Rule 7:** The logic is recursive

Conceptual/semantic models

- A conceptual/semantic model is a mental model which captures ideas in a domain of interest in terms of modeling primitives.
- The aim of conceptual model is to express the meaning of terms and concepts used by domain experts to discuss the problem, and to find the correct relationships between different concepts.
- The conceptual model attempts to clarify the meaning of various usually ambiguous terms, and ensure that problems with different interpretations of the terms and concepts cannot occur.
- Once the domain of interest has been modeled, the model becomes a stable basis for subsequent development of applications in the domain.
- A conceptual model can be described using various notations.
Domain/use case models

- A domain model is a conceptual model of a system which describes the various entities involved in the system and the relationships among them.
- The domain model is created to capture the key concepts and the vocabulary of the system.
- It identifies the relationships among all major entities within the system, as well as their main methods and attributes.
- In this way the model provides a structural view of the system which is normally complemented by the dynamic views in use case models.
- The aim of a domain model is to verify and validate the understanding of a domain of interest among various stakeholders of the project group. It is especially helpful as a communication tool and a focusing point between technical and business teams.

General principles

- A model describes some domain of interest in a simplified, abstract way.
- Abstraction, classification, aggregation.
- A model is built according to a modeling theory.
  - ER modeling, OO modeling, ontologies, semantic networks, object-role modeling etc.
- A model uses modeling primitives.
  - Concepts, classes, entities, objects, elements.
  - Attributes, properties, methods.
  - Relationships.
  - Axioms, constraints, restrictions, rules.
- A model is represented using a particular notation.
  - Tables and columns, XML, UML, OWL etc.
View of data

- Physical level/internal level/internal schema
  - how data are actually stored.
- Logical level/conceptual level/conceptual schema
  - what data are stored in the database, and what relationships exist among those data.
- View level/user view/external schema
  - describes only part of the entire database.

Types of data models

- Object-based Logical Models.
  - Describe data at the conceptual level.
  - Provide fairly flexible structuring capabilities, semantics further specified through constraints.

- Record-based Logical Models.
  - Describe data at the conceptual and view levels.
  - Are used to provide a higher-level description of a (database) system or implementation.
  - Examples: relational, network, and hierarchical models.

- Physical Data Models.
  - Are used to describe data at the physical level.
Example

Example: UML

http://www.softwarefactories.com
Example: OWL Time

Example

http://www.w3.org/2005/Incubator/urw3/group/images/section4_figure3.jpg
Example

http://www.sequenceontology.org/images/image001.jpg

Example

http://www.nature.com/nrc/journal/v7/n1/images/nrc2036-f1.jpg
Process models

- A process model is a description of a process. Process models are often associated with business processes.

- A business process is a collection of related, structured activities that produce a service or product that meet the needs of a client.

- Related: Service-oriented modeling, SOA, Business process management, Workflows.

The layers of SOA (http://www.ibm.com/developerworks/library/ws-soa-design1/)
General principles

- A process model describes a process in a simplified, abstract way.
- Contains dynamic information.
- A model is built according to a modeling theory.
- A model uses modeling primitives.
- A model is represented using a particular notation.

Example

Example

![Diagram](http://faculty.maxwell.syr.edu/gmbonham/CognitiveProcessModel-2.gif)

Fig. 1. Flowchart version of group cognition model.
Modeling and model engineering

- Modeling is the process of creating or building a **model**.

- **Model engineering** describes the main principles, phases and activities of the model life cycle.
  - How to build a domain model.
  - How to integrate/align different models.
  - How to evaluate a model.
  - How to modify a model.
  - How to build a shared model.
  - How to reuse a model.
  - …

- Related: Business process management, knowledge engineering, domain model engineering, service-oriented modeling.

---

Example: CommonKADS

- Describes the process of constructing knowledge models in terms of stages, ordered activities, support techniques.

- Stages:
  - Knowledge identification.
    - Domain familiarization, list of potential model components for reuse.
  - Knowledge specification.
    - Choose task template, construct initial domain conceptualization, complete knowledge-model specification.
  - Knowledge refinement.
    - Validate knowledge model, refine knowledge base.

- Roles: knowledge engineer, domain expert.
The CommonKADS Methodology

• Originally, CommonKADS aimed at providing a methodology for developing knowledge (-based) systems
  – Thus, CommonKADS is a knowledge engineering methodology

• However, there exist strong ties between knowledge engineering and knowledge management
  – Knowledge management exploits the methods and tools of advanced information and knowledge systems
  – Knowledge (-based) systems have to be embedded into the organizational and human resources context

The CommonKADS Methodology

• Subsequently we consider
  – The overall knowledge management framework as defined by CommonKADS
  – Those models of the CommonKADS model suite that are relevant for knowledge management
The Knowledge Management Framework of CommonKADS

• Knowledge management (KM) strategy is defined in an outside-in direction
  – What are the value-creation goals of the organization?
  – How is this value delivered by the organization's business processes?
  – What knowledge is used in these processes to deliver value?

• KM question
  – What actions are useful for increasing the leverage of knowledge underlying these business processes?

Knowledge management in relation to the business processes and value creation by the organization.
CommonKADS guidelines

• Knowledge identification
  – Challenges: knowledge engineer needs to find a balance between learning about the domain without becoming a full domain expert.
  – Guideline: talk to people in the organization who have to talk to experts, but are not experts themselves.
  – Guideline: avoid diving into detailed, complicated theories unless their usefulness is proven.
  – Guideline: construct a few typical scenarios which you understand at a global level.

CommonKADS guidelines (cont)

• Knowledge specification
  – Guideline: Prefer task templates which have been used more than once.
  – Guideline: Base domain-specific conceptualizations on existing data models as much as possible.
  – Guideline: Use few modeling primitives (concepts, subtypes, relations).
  – …
CommonKADS guidelines (cont)

- Knowledge refinement.
  - Guideline: if it turns out to be difficult to find instances of certain knowledge types, reconsider this type of the schema.
  - Guideline: look also for existing knowledge bases in the same domain.

Metamodeling

- **Metamodeling** is the analysis, construction and development of the frames, rules, constraints, models and theories applicable and useful for modeling a predefined class of problems.
- It applies the notions of meta- and modeling.

- Related: MDAs, Model Transformation Language, Meta-process modeling, Meta-data modeling

  (Brinkkemper, 1990)
Modeling languages

• Languages have a syntax and a semantics.

• The same applies for modeling languages, though their syntax and, more often, semantics is not always precisely defined.

• Modeling languages can be divided in two categories:
  – Graphical modeling languages.
  – Textual modeling languages.

Examples

• UML: general-purpose modeling language typically used to specify software systems.
• IDEF: family of modeling languages IDEF0-IDEF14
  – IDEF1: information modeling.
  – IDEF1X: database design.
  – IDEF4: object-oriented design.
  – IDEF5: ontology description capture.
  – ...
• ER: database design modeling language.
• RDFS/OWL: ontology modeling languages.
• PetriNets: modeling language for the description of discrete distributed systems.
• BPEL: language for specifying business process behavior based on Web services.
• WSML: language for specifying semantic descriptions of Web services.
• SysML: domain modeling language used in systems engineering.
• XML Schema/DTD: general-purpose schema languages which can be used for data modeling.
Use cases

- Software and system engineering.
- Communication among people and machines.
- Knowledge representation.
- Information management: search, indexing, classification, navigation.

Benefits

- **Productivity**
  - Faster decision making
  - Increased productivity
  - Enhanced problem-solving
  - Solve complex problems
  - Reliability
  - Equipment operation
- **Knowledge preservation**
  - Capture scarce expertise
  - Use in remote locations
- **Quality improvement**
  - Increased quality of decisions
  - Dealing with uncertainty
- **Training**
  - Educational benefits
- **Job enrichment**
  - Flexibility
  - Integrating knowledge of several experts
Summary

- Modeling and model engineering.
- Types of models and their properties.
- Modeling languages.
- Use cases.

Further reading


http://www.uml.org/
Assignment 1

a. Explain what is the difference between a model and a miniature? What is the difference between a model and a prototype?

b. Consider one or two domains of interest that you are familiar with. You might or might not have developed computer systems dealing with your chosen subject matters. Note down 10 of the key entities in this domain.

c. Explain the process of knowledge model construction in CommonKADS (look up on the Web).

Thank You!

Questions?