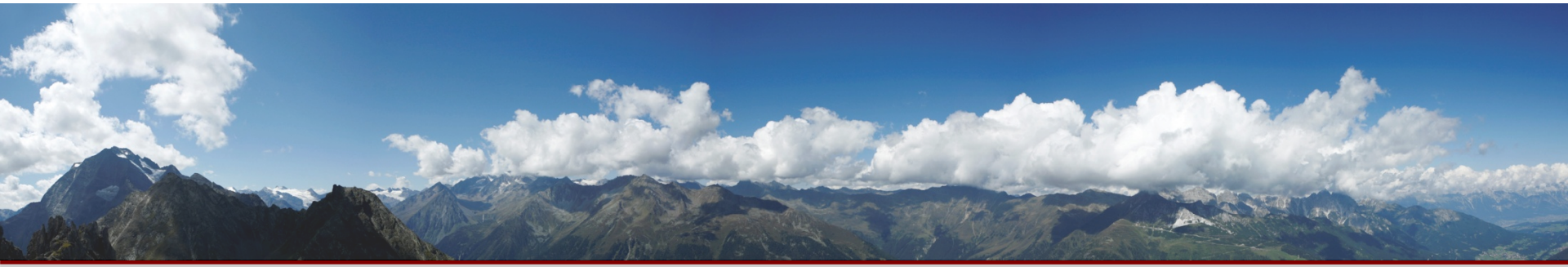
Introduction to modeling
WS 2015/16

General Concepts

Anna Fensel



Where are we?

STI · INNSBRUCK

#	Title	Date
1	Introduction	07.10.2015
2	**General concepts**	**21.10.2015**
3	ORM modeling	04.11.2015
4	Relational modeling	18.11.2015
5	ER modeling	02.12.2015
6	OO modeling	16.12.2015
7	Services and process modeling	13.01.2016
10	Exam	27.01.2016

Models

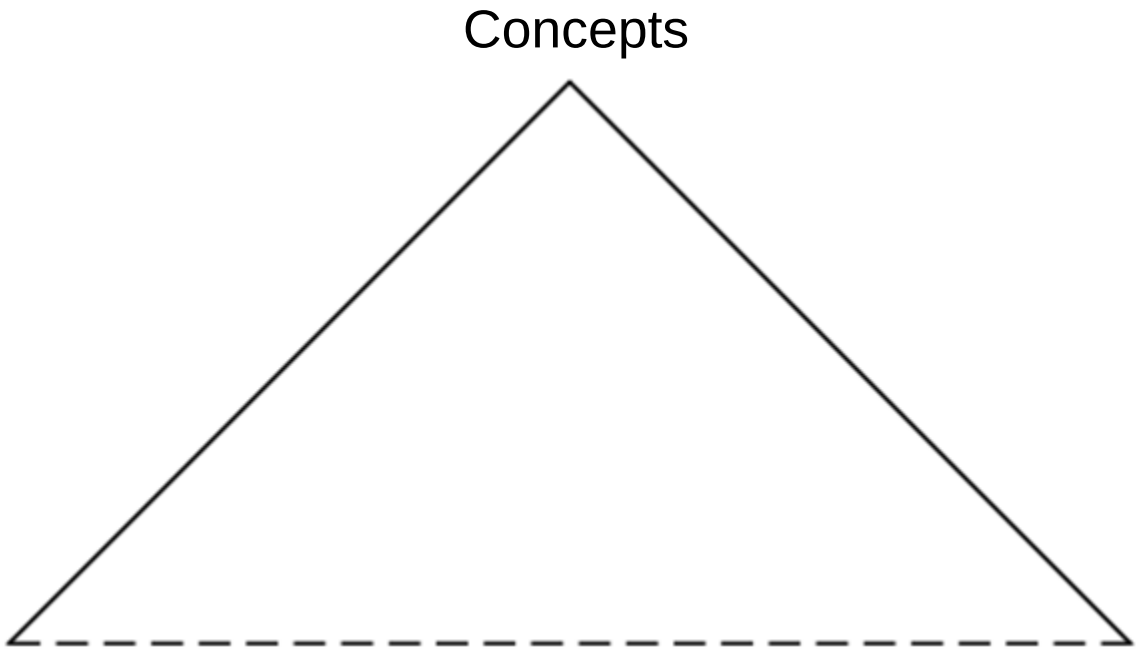
- A **model** (*plural* **models**)
  - A person who serves as a subject for artwork or fashion, usually in the medium of photography but also for painting or drawing.
    - *The beautiful **model** had her face on the cover of almost every fashion magazine imaginable.*
  - A miniature representation 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’.

“Triangle of reference”
(aka “triangle of meaning”)

STI · INNSBRUCK

- How linguistic symbols are related to the objects they represent



General principles

STI · INNSBRUCK

- 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

STI · INNSBRUCK

- 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 model

- 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

STI · INNSBRUCK

	What (Data)	How (Function)	Where (Locations)	Who (People)	When (Time)	Why (Motivation)
Scope {contextual} Planner	List of things important to the business	List of processes that the business performs	List of locations in which the business operatses	List of organizations important to the business	List of events/ cycles important to the business	List of business goals/strategies
Enterprise Model {conceptual} Business Owner	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
System Model {logical} Designer	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Process Structure	e.g. Business Rule Model
Technology Model {physical} Implementer	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
Detailed Representation {out-of-context} Subcontractor	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Definition
Functioning System	e.g. Data	e.g. Function	e.g. Network	e.g. Organization	e.g. Schedule	e.g. Strategy

http://www.zifa.com/

Conceptual/semantic models

STI · INNSBRUCK

- 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

STI · INNSBRUCK

- 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**

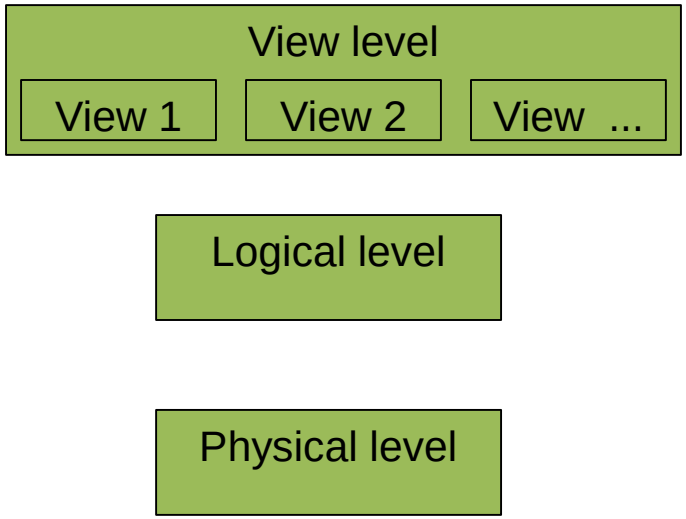
General principles

STI · INNSBRUCK

- A **model** describes some domain of interest in a simplified, abstract way
- Abstraction, classification, aggregation
- A model is built according to a **modeling theory**
  - Object-role modeling (ORM), ER modeling , OO modeling, relational modeling, ontologies, semantic networks, 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

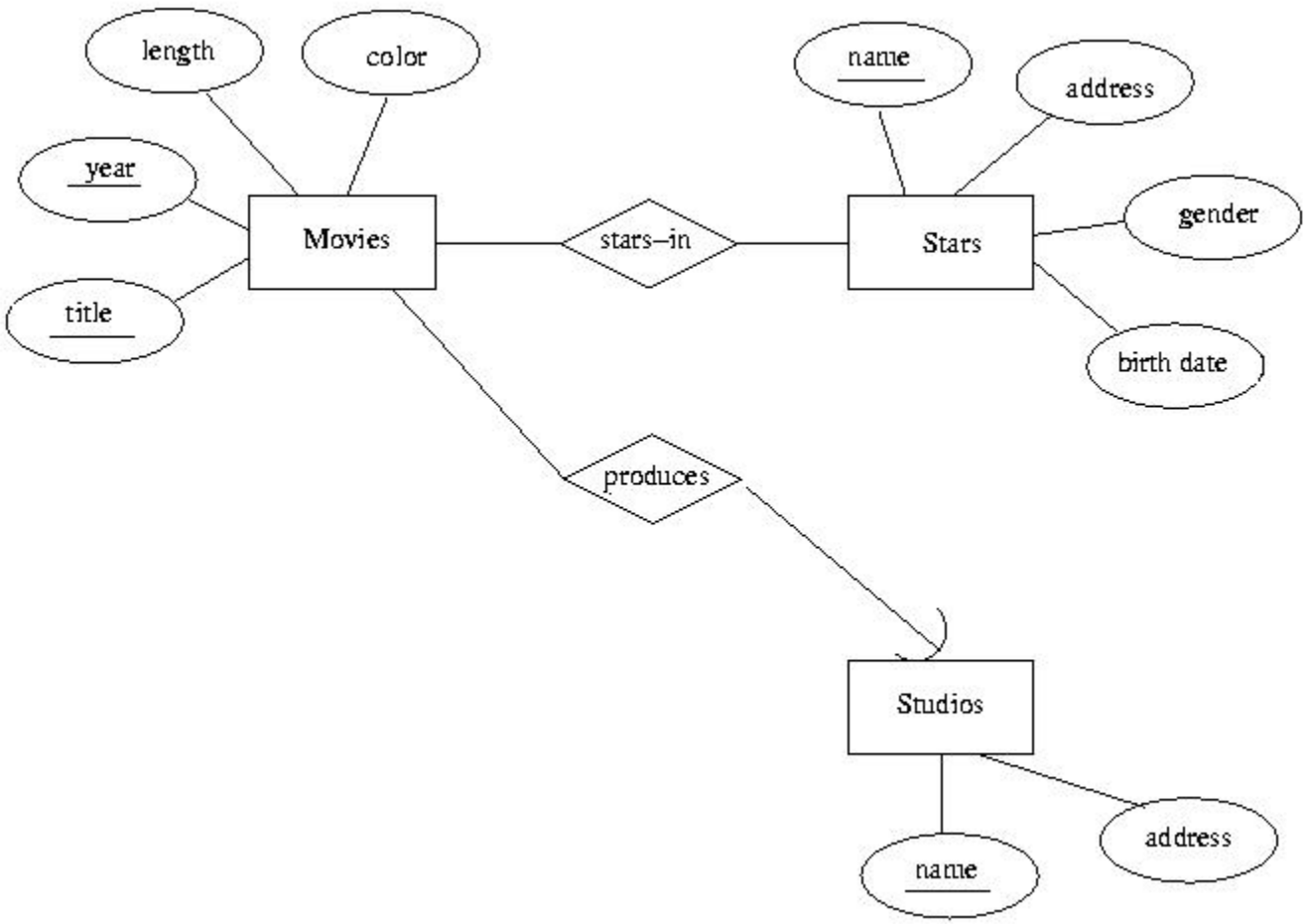


Types of data models

STI · INNSBRUCK

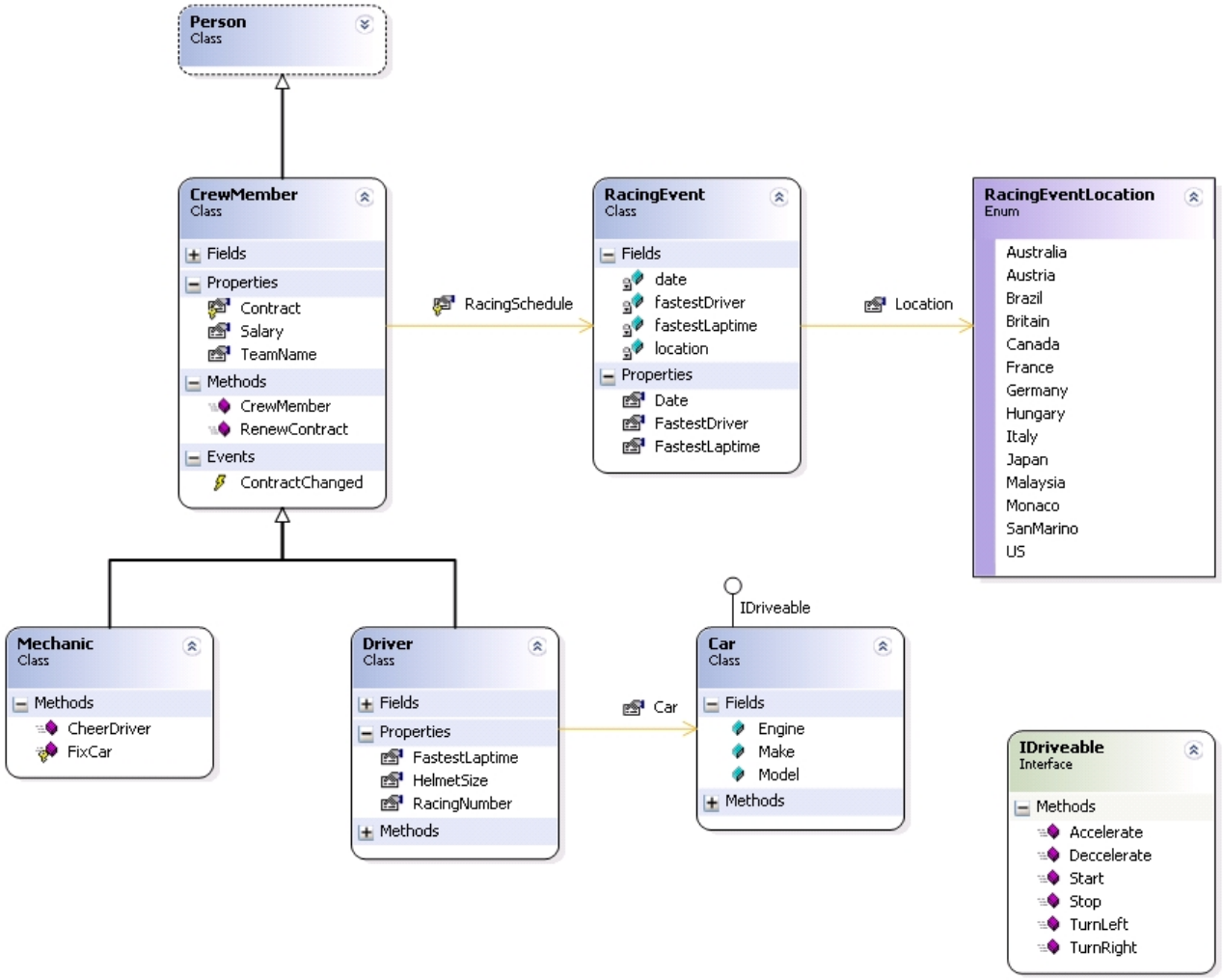
- Object-based Logical Models
  - Describe data at the conceptual level
  - Provide fairly flexible structuring capabilities, semantics further specified through constraints
  - Examples: Object-role model, Entity-relationship model, Object-oriented model, Semantic data model
- Record-based Logical Models
  - 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


length
color
year
title
Movies
stars–in
name
address
Stars
gender
birth date
produces
Studios
address
name

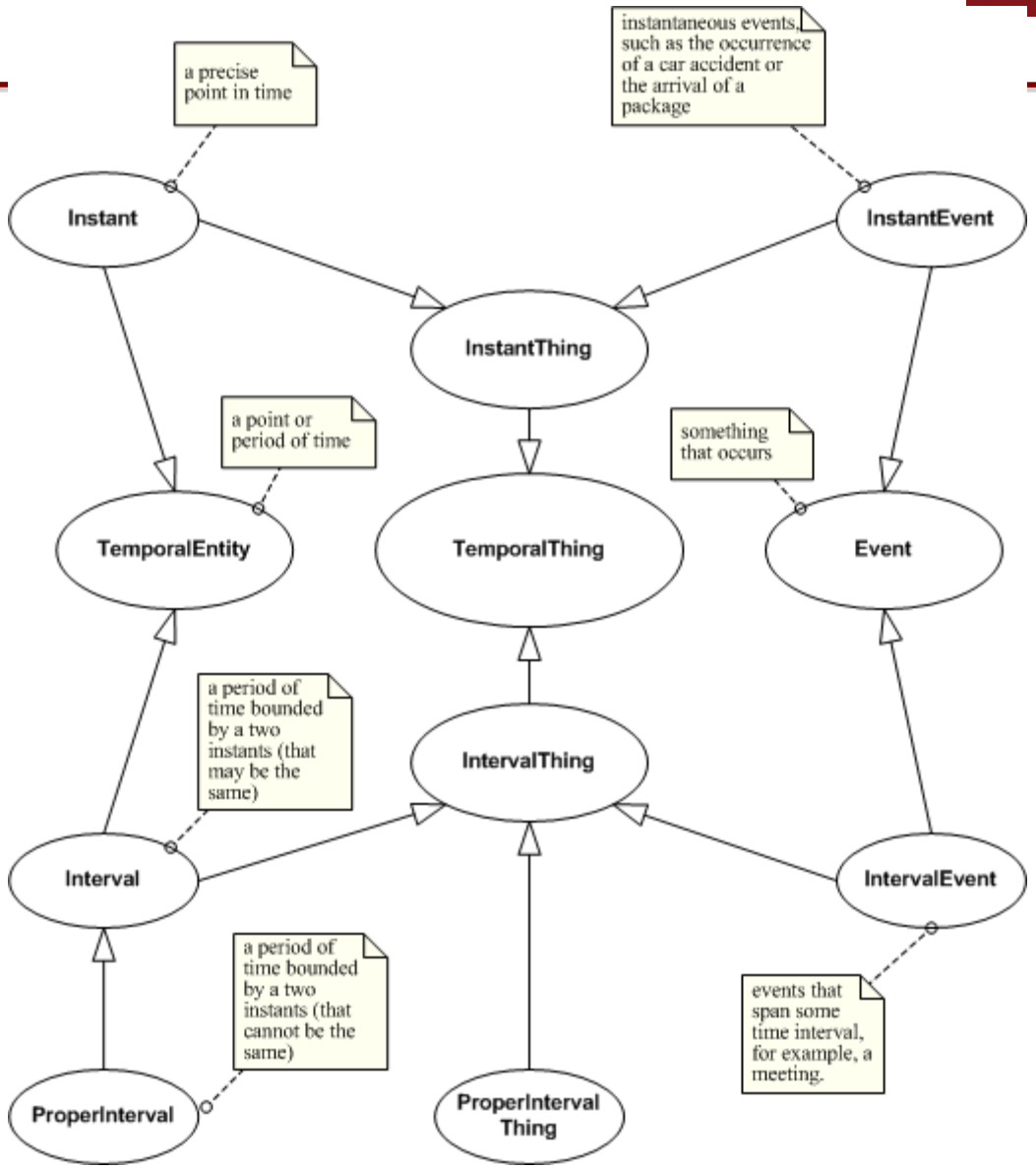
Example: UML

STI · INNSBRUCK



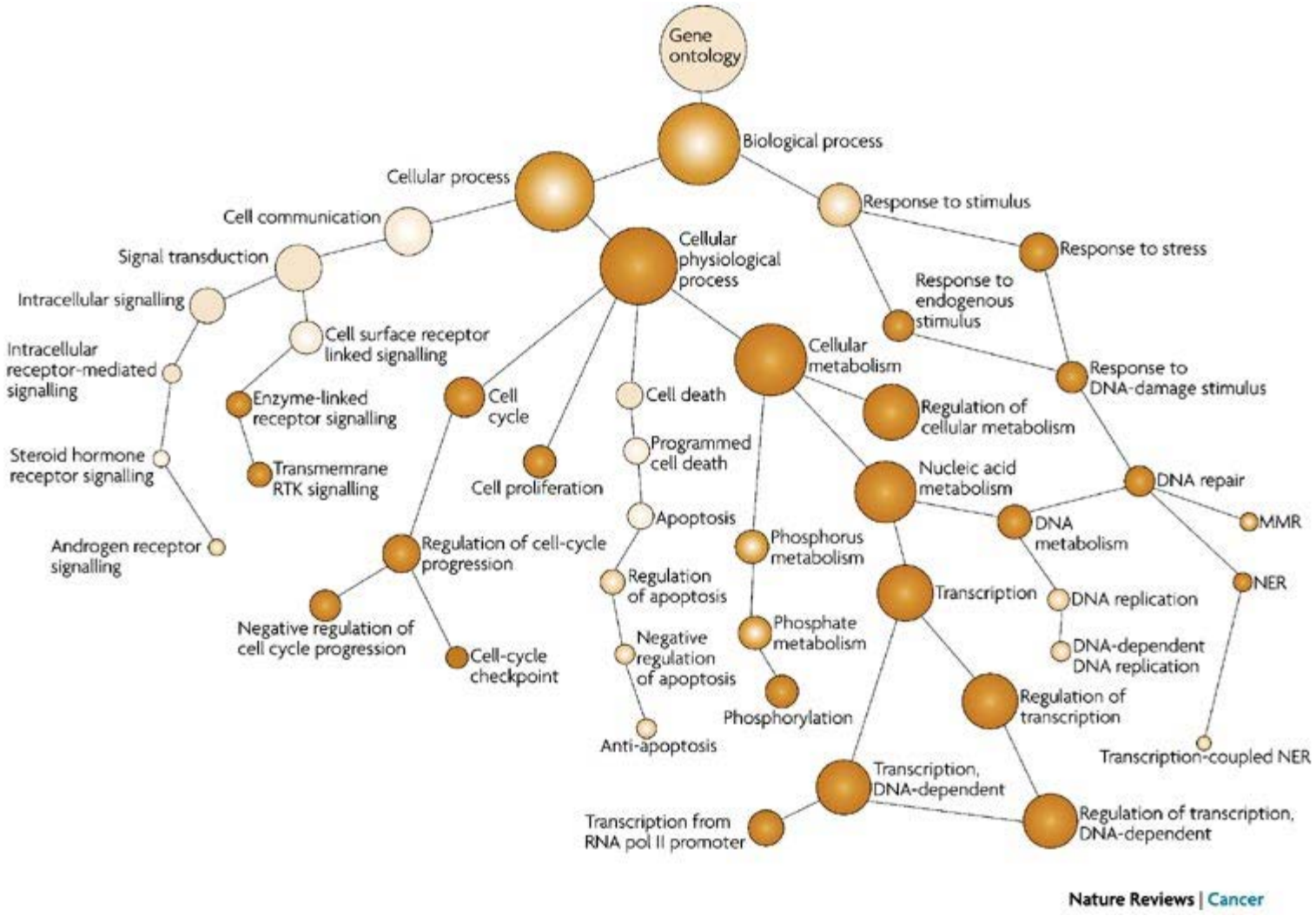
http://www.softwarefactories.com

Example: OWL Time



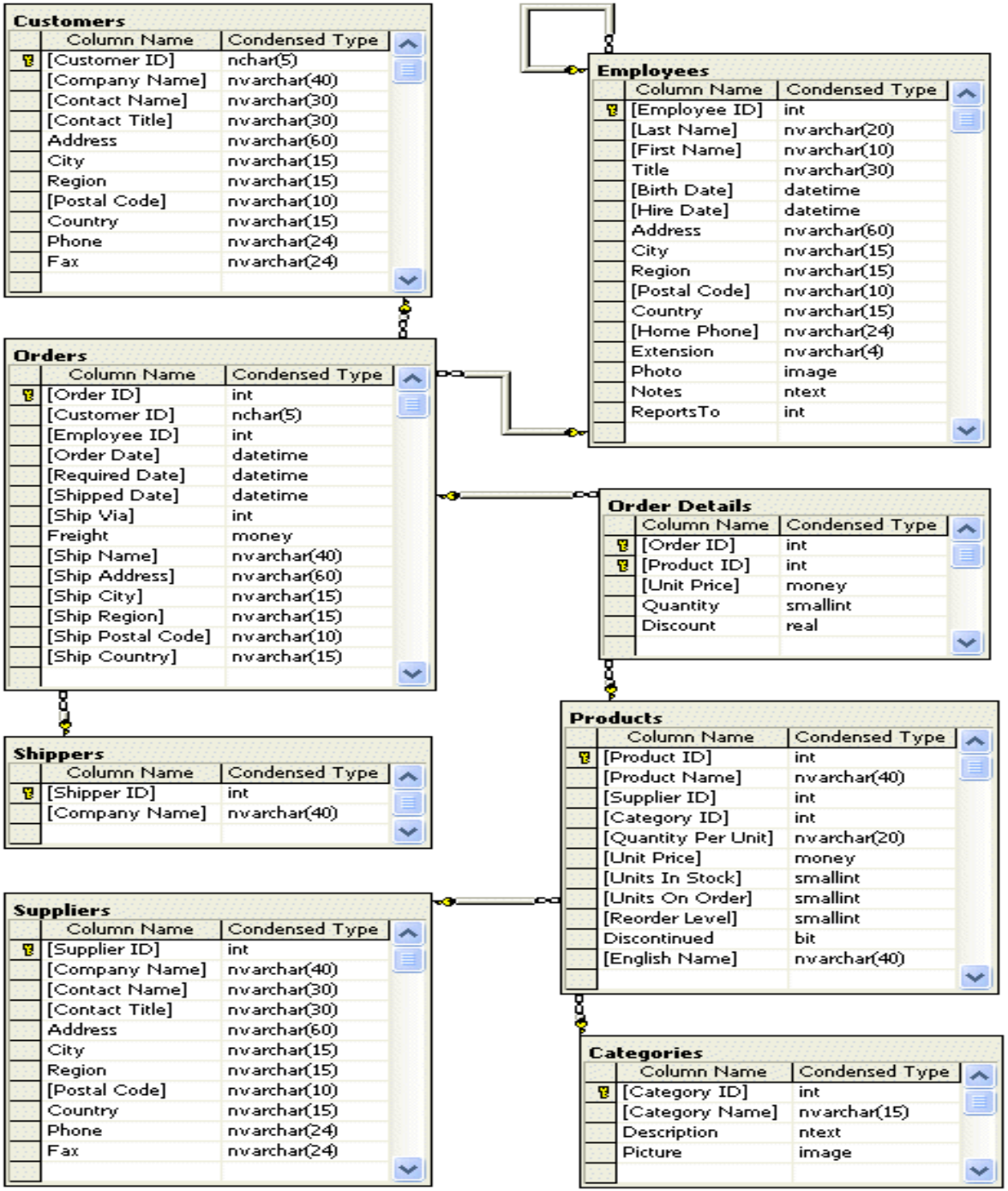
http://iandavis.com/2006/owl-time/owl-time-ontology-20050404.png

Example



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

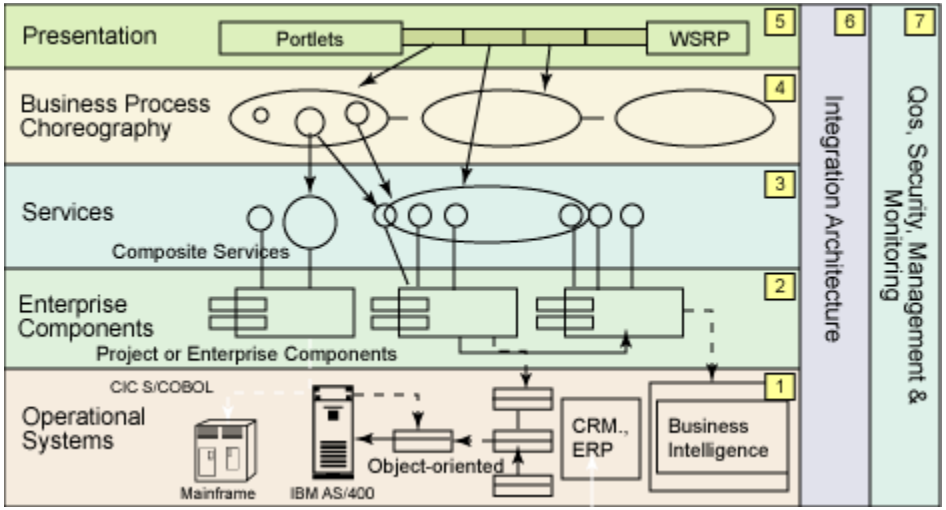
Example



http://www.microsoft.com/library/media/1033/technet/images/prodtechnol/sql/2000/maintain/sscpop07_big.gif

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.*



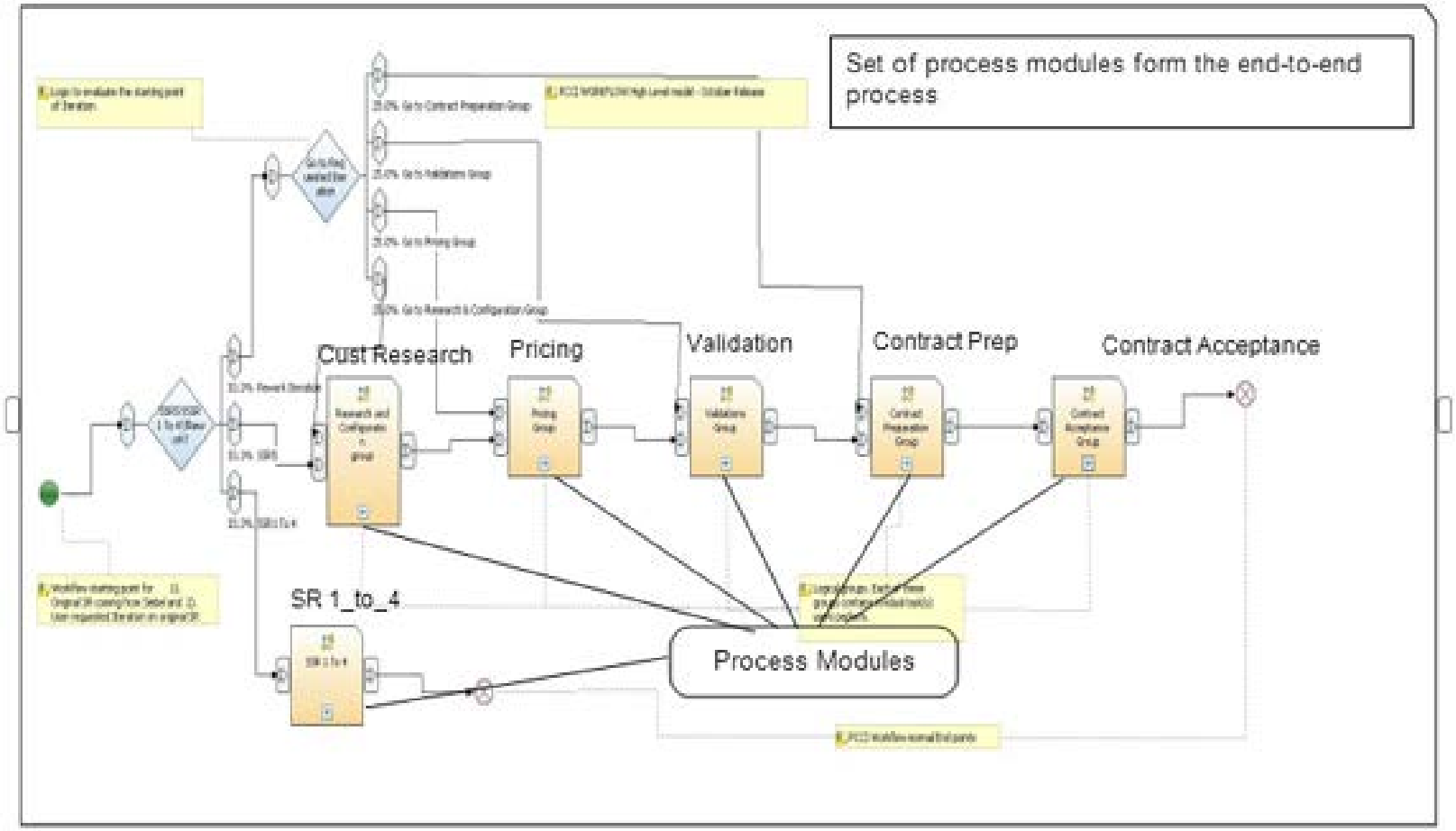
The layers of SOA (http://www.ibm.com/developerworks/library/ws-soa-design1/)

General principles

STI · INNSBRUCK

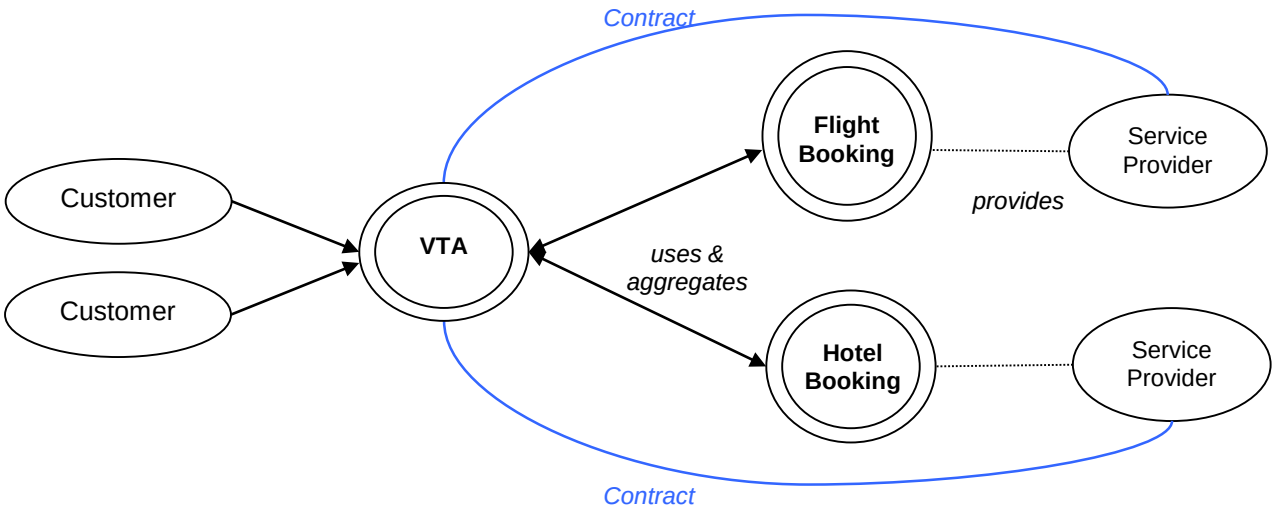
- A process model describes a process in a simplified, abstract way
- Contains dynamic information
- A model is built according to a modeling theory
- A model uses modeling primitives
- A model is represented using a particular notation

Example



http://www.ibm.com/developerworks/library/ar-workflow2/WorkflowProcessModel_Full.jpg

Example


Contract
Flight Booking
Service Provider
provides
Customer
Customer
VTA
uses & aggregates
Hotel Booking
Service Provider
Contract

Modeling and model engineering

STI · INNSBRUCK

- 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*

Modeling languages

STI · INNSBRUCK

- 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

STI · INNSBRUCK

- ORM: fact-based modeling language; genral purpose
- 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
- XML Schema/DTD: general-purpose schema languages which can be used for data modeling

Use cases

STI · INNSBRUCK

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

Benefits

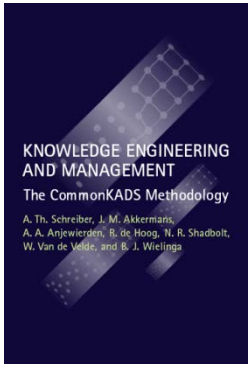
- Productivity
  - Faster decision making
  - Increased productivity
  - Enhanced problem-solving
  - Solve complex problems
  - Reliability
  - Equipment operation
- Knowledge preservation
  - Capture scarce expertise
  - Use in remote locations
- Quality improvement
  - Increased quality of decisions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

Schreiber et al. Knowledge Engineering and Management.

Example: CommonKADS

STI · INNSBRUCK

- Knowledge Acquisition and Documentation Structuring
- 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.



Knowledge engineering

STI · INNSBRUCK

- Process of
  - eliciting
  - structuring
  - formalizing
  - operationalizing
- information and knowledge involved in a knowledge-intensive problem domain,
- in order to construct a program that can perform a difficult task adequately

Knowledge engineering problems

STI · INNSBRUCK

- Complex information and knowledge is difficult to observe
- Experts and other sources differ
- Multiple representations:
  - textbooks
  - graphical representations
  - skills

Importance of proper knowledge engineering

STI · INNSBRUCK

- Knowledge is valuable and often outlives a particular implementation
  - knowledge management
- Errors in a knowledge-base can cause serious problems
- Heavy demands on extendibility and maintenance
  - changes over time

The CommonKADS Methodology

STI · INNSBRUCK

- 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

CommonKADS principles

STI · INNSBRUCK

- CommonKADS: a comprehensive methodology for KBS development
- Knowledge engineering is ***not*** some kind of `mining from the expert's head', but consists of constructing different aspect models of human knowledge
- The knowledge-level principle: in knowledge modeling, first concentrate on the conceptual structure of knowledge, and leave the programming details for later
- Knowledge has a stable internal structure that is analyzable by distinguishing specific knowledge types and roles.

CommonKADS Terminology

STI · INNSBRUCK

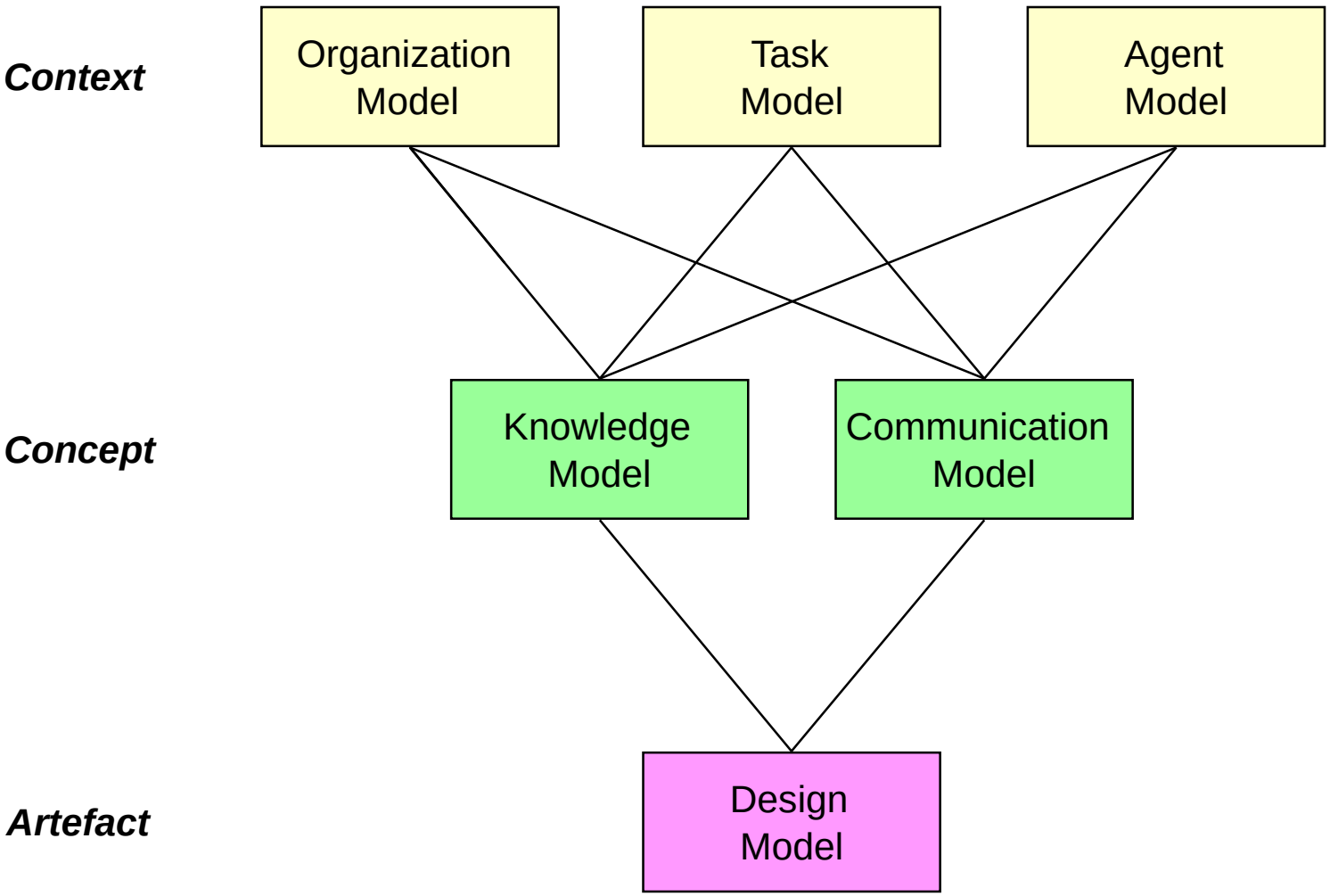
- Domain
  - some area of interest
    banking, food industry, photocopiers, car manufacturing
- Task
  - something that needs to be done by an agent
    monitor a process; create a plan; analyze deviant behavior
- Agent
  - the executor of a task in a domain
    typically either a human or some software system
- Application
  - The context provided by the combination of a task and a domain in which this task is carried out by agents
- Application domain
  - The particular area of interest involved in an application
- Application task
  - The (top-level) task that needs to be performed in a certain application

CommonKADS Terminology

STI · INNSBRUCK

- knowledge system (KS)
  - system that solves a real-life problem using knowledge about the application domain and the application task
- expert system
  - knowledge system that solves a problem which requires a considerable amount of expertise, when solved by humans.

CommonKADS Model Set



Model Set Overview (1)

STI · INNSBRUCK

- Organization model
  - supports analysis of an organization,
  - Goal: discover problems, opportunities and possible impacts of KBS (knowledge-based system) development.
- Task model
  - describes tasks that are performed or will be performed in the organizational environment
- Agent model
  - describes capabilities, norms, preferences and permissions of agents (agent = executor of task).

Model Set Overview (2)

STI · INNSBRUCK

- Knowledge model
  - gives an implementation-independent description of knowledge involved in a task.
- Communication model
  - models the communicative transactions between agents.
- Design model
  - describes the structure of the system that needs to be constructed.

Roles in knowledge-system development

STI · INNSBRUCK

- knowledge provider
- knowledge engineer/analyst
- knowledge system developer
- knowledge user
- project manager
- knowledge manager

many-to-many relations between roles and people

Knowledge provider/specialist

STI · INNSBRUCK

- “traditional” expert
- person with extensive experience in an application domain
- can provide also plan for domain familiarization
  - “where would you advise a beginner to start?”
- inter-provider differences are common
- need to assure cooperation

Knowledge engineer

STI · INNSBRUCK

- specific kind of system analyst
- should avoid becoming an "expert"
- plays a liaison function between application domain and system

Knowledge-system developer

STI · INNSBRUCK

- person that implements a knowledge system on a particular target platform
- needs to have general design/implementation expertise
- needs to understand knowledge analysis
  - but only on the “use”-level

Knowledge user

STI · INNSBRUCK

- interact with the prospective system or are affected indirectly by the system
- Level of skill/knowledge is important factor
- May need extensive interacting facilities
  - explanation
- His/her work is often affected by the system
  - consider attitude / active role

Project manager

STI · INNSBRUCK

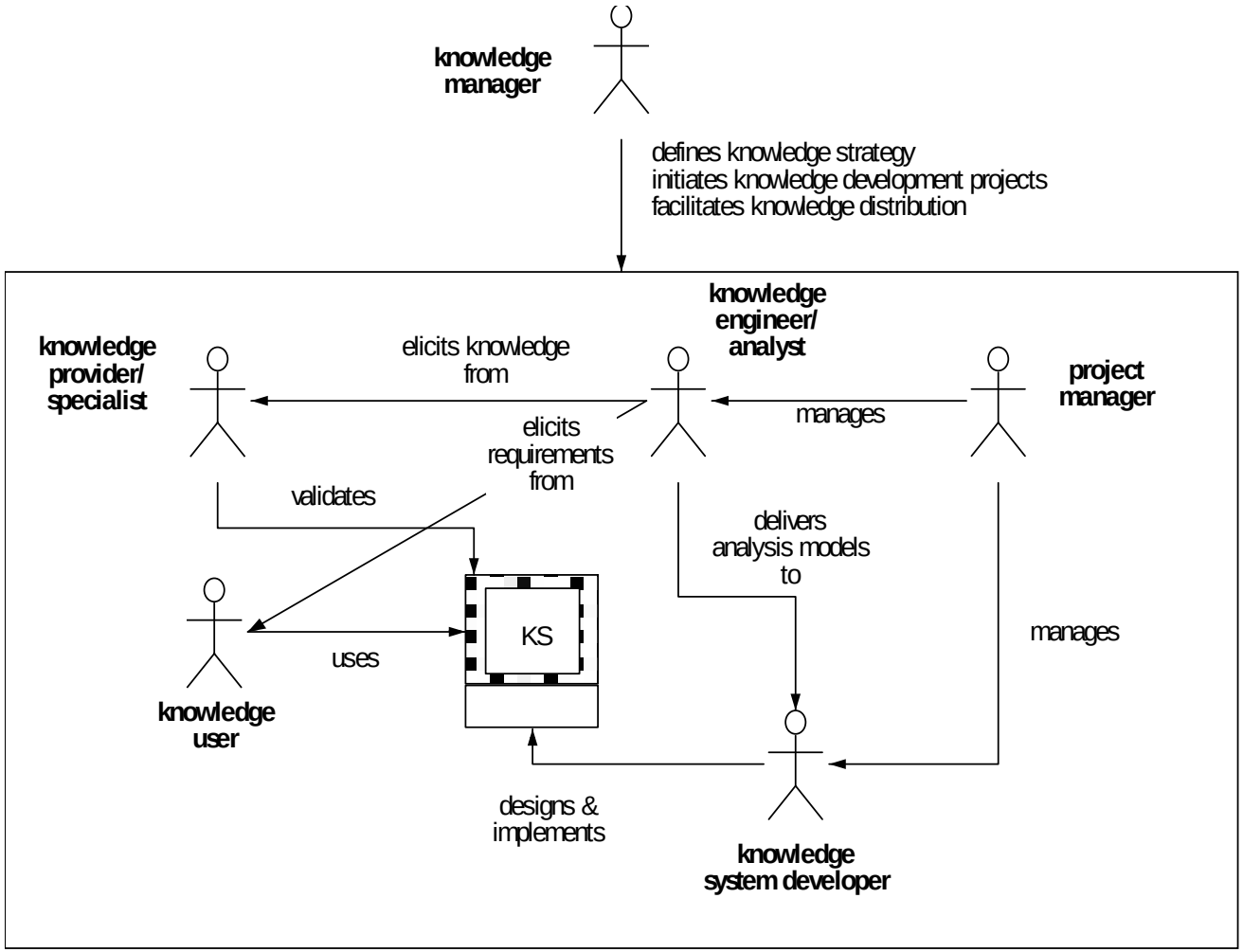
- responsible for planning, scheduling and monitoring development work
- liaises with client
- typically medium-size projects (4-6 people)
- profits from structured approach

Knowledge manager

STI · INNSBRUCK

- background role
- monitors organizational purpose of
  - system(s) developed in a project
  - knowledge assets developed/refined
- initiates (follow-up) projects
- should play key role in reuse
- may help in setting up the right project team

Roles in knowledge-system development



Knowledge model construction

Process & Product

STI · INNSBRUCK

- so far: focus on knowledge model as product
- bottleneck for inexperienced knowledge modelers
  - how to undertake the process of model construction.
- solution: process model
  - as prescriptive as possible
  - process elements: stage, activity, guideline, technique
- but: modeling is constructive activity
  - no single correct solution nor an optimal path
- support through a number of guidelines that have proven to work well in practice.
- knowledge modeling is specialized form of requirements specification
  - general software engineering principles apply

Stages in Knowledge-Model Construction

STI · INNSBRUCK

STAGES	TYPICAL ACTIVITIES
knowledge identification	- domain familiarization (information sources, glossary, scenarios) - list potential model components for reuse (task- and domain-related components)
knowledge specification	- choose task template (provides initial task decomposition) - construct initial domain conceptualization (main domain information types) - complete knowledge-model specification (knowledge model with partial knowledge bases)
knowledge refinement	- validate knowledge model (paper simulation, prototype of reasoning system) - knowledge-base refinement (complete the knowledge bases)

Stage 1:
Knowledge identification

STI · INNSBRUCK

- goal
  - survey the knowledge items
  - prepare them for specification
- input
  - knowledge-intensive task selected
  - main knowledge items identified
  - application task classified
    - assessment, configuration, combination of task types
- activities
  - explore and structure the information sources
  - study the nature of the task in more detail

Exploring information sources

STI · INNSBRUCK

- Factors
  - Nature of the sources
    - well-understood?, theoretical basis?
  - Diversity of the sources
    - no single information source (e.g. textbook or manual)
    - diverse sources may be conflicting
    - multiple experts is a risk factor.
- Techniques
  - text marking in key information sources
  - some structured interviews to clarify perceived holes in domain
- main problem:
  - find balance between learning about the domain without becoming a full expert

Guidelines

STI · INNSBRUCK

- Talk to people in the organization who have to talk to experts but are not experts themselves
- Avoid diving into detailed, complicated theories unless the usefulness is proven
- Construct a few typical scenarios which you understand at a global level
- Never spend too much time on this activity. Two person weeks should be maximum.

Results exploration

STI · INNSBRUCK

- Tangible
  - Listing of domain knowledge sources, including a short characterization.
  - Summaries of selected key texts.
  - Glossary/lexicon
  - Description of scenarios developed.
- Intangible
  - your own understanding of the domain
    - most important result

Stage 2:
Knowledge specification

STI · INNSBRUCK

- goal: complete specification of knowledge except for contents of domain models
  - domain models need only to contain example instances
- activities
  - Choose a task template.
  - Construct an initial domain conceptualization.
  - Specify the three knowledge categories

Guidelines

STI · INNSBRUCK

- use as much as possible existing data models:
  - useful to use at least the same terminology basic constructs
  - makes future cooperation/exchange easier
- limit use of the knowledge-modeling language to concepts, sub-types and relations
  - concentrate on "data"
  - similar to building initial class model
- If no existing data models can be found, use standard SE techniques for finding concepts and relations
  - use “pruning” method
- Constructing the initial domain conceptualization should be done in parallel with the choice of the task template

Guidelines for specifying domain knowledge

STI · INNSBRUCK

- domain-knowledge type used as static role not required to have exactly the “right’” representation
  - design issue;
  - key point: knowledge is available.
- scope of domain knowledge is typically broader than what is covered by inferences
  - requirements of communication, explanation

Stage 3: Knowledge Refinement

STI · INNSBRUCK

- Validate knowledge model
- Fill contents of knowledge bases

Validate knowledge model

STI · INNSBRUCK

- internally and externally
- verification = internal validation
  - “is the model right?”
- validation = validation against user requirements
  - "is it the right model?"

Validation techniques

STI · INNSBRUCK

- Internal
  - structured walk-troughs
  - software tools for checking the syntax and find missing parts
- External
  - usually more difficult and/or more comprehensive.
  - main technique: simulation
    - paper-based simulation
    - prototype system

Maintenance

- model development is a cyclic process
- models act as information repositories
  - continuously updated
- but: makes requirements for support tools stronger
  - transformation tools

Domain Documentation Document

STI · INNSBRUCK

- Knowledge model specification
- list of all information sources used.
- list of model components that we considered for reuse.
- scenarios for solving the application problem.
- results of the simulations undertaken during validation
- Elicitation material (appendices)

Summary process

- Knowledge identification
  - familiarization with the application domain
- Knowledge specification
  - detailed knowledge analysis
- Knowledge refinement
  - completing the knowledge model
  - validating the knowledge model
- Feedback loops may be required
  - simulation in third stage may lead to changes in specification
  - Knowledge bases may require looking for additional knowledge sources.
  - general rule: feedback loops occur less frequently, if the application problem is well-understood and similar problems have been tackled

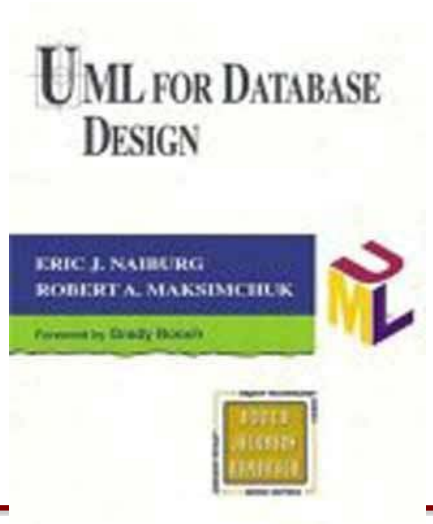
Summary

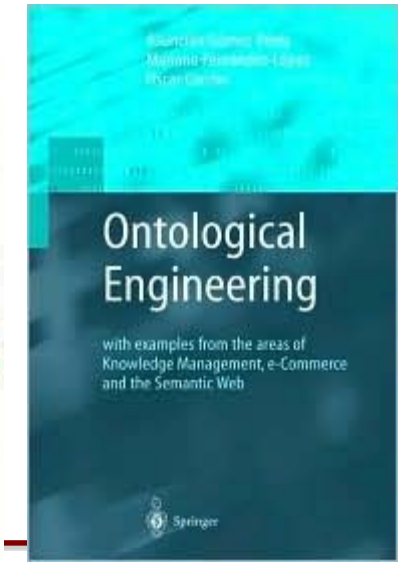
- Modeling and model engineering.
- Types of models and their properties.
- Modeling languages.
- Use cases.

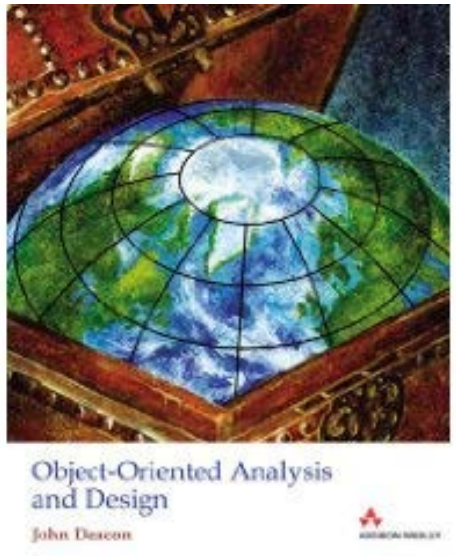
References

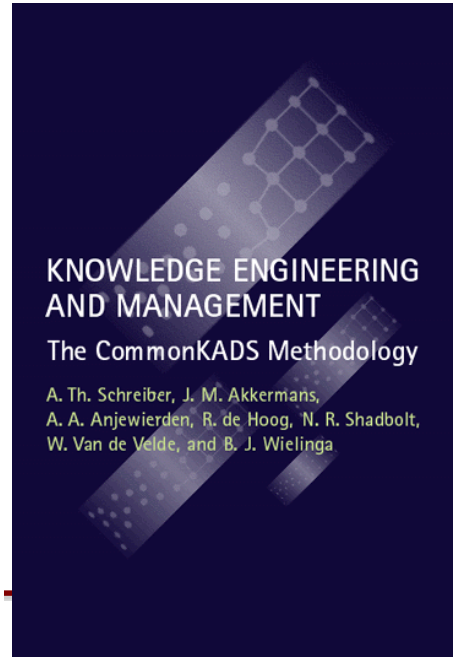
- A. Newell. The Knowledge Level. Artificial Intelligence, 18(1):87-127, 1982.
- M. Uschold and M. Gruninger. Ontologies: principles, methods and applications. KER, 11(2), 1996.

http://www.uml.org/









Next lecture

STI · INNSBRUCK

#	Title	Date
1	Introduction	07.10.2015
2	General concepts	21.10.2015
3	**ORM modeling**	**04.11.2015**
4	Relational modeling	18.11.2015
5	ER modeling	02.12.2015
6	OO modeling	16.12.2015
7	Services and process modeling	13.01.2016
10	Exam	27.01.2016

Questions?