Towards Integration of Modeling and Reusing Software Cases

SVM Workshop (Helsinki), 19.04.2007

Thorsten Krebs, Katharina Wolter, Lothar Hotz

HITeC

Hamburger Informatik Technologie-Center e.V.

ReDSeeDS: Project Partners

Industry:

- Infovide S.A. (Poland) *(Project Leader)*
- Fraunhofer Gesellschaft (Germany)
- Algorithmus Sistemos, UAB (Lithuania)
- C/S Enformasyon Teknolojileri Limited Sirketi (Turkey)
- PRO DV Software AG (Germany)

Research Institutes:

- Wasaw University of Technology (Poland)
- HITeC e.V. (Germany)
- University of of Koblenz-Landau (Germany)
- University of Latvia (Latvia)
- Vienna University of Technology (Austria)
- Heriot-Watt University (UK)

ReDSeeDS: The Goal

Problem:

- Software projects tend to ignore past experience
- No standard way to capture knowledge about complete cases…
  - …leading from the problem to its solution
- Customer needs are often ambiguous and imprecise
  - Requirements models abstract from complex requirements
  - Architecture models abstract from complex systems

Solution:

- Framework for systematic creation and reuse of *all* software development artifacts

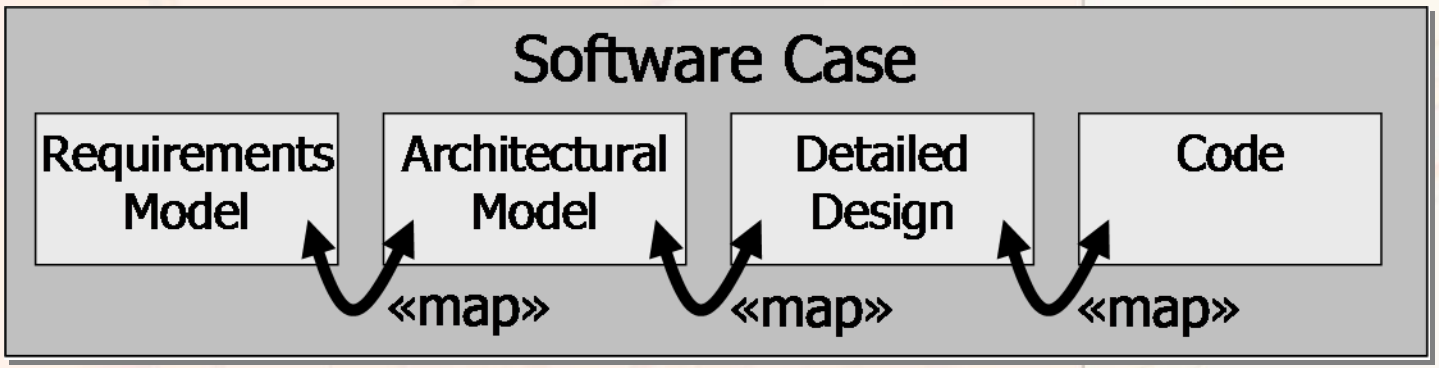
Software Case

Problem specification

- Requirements

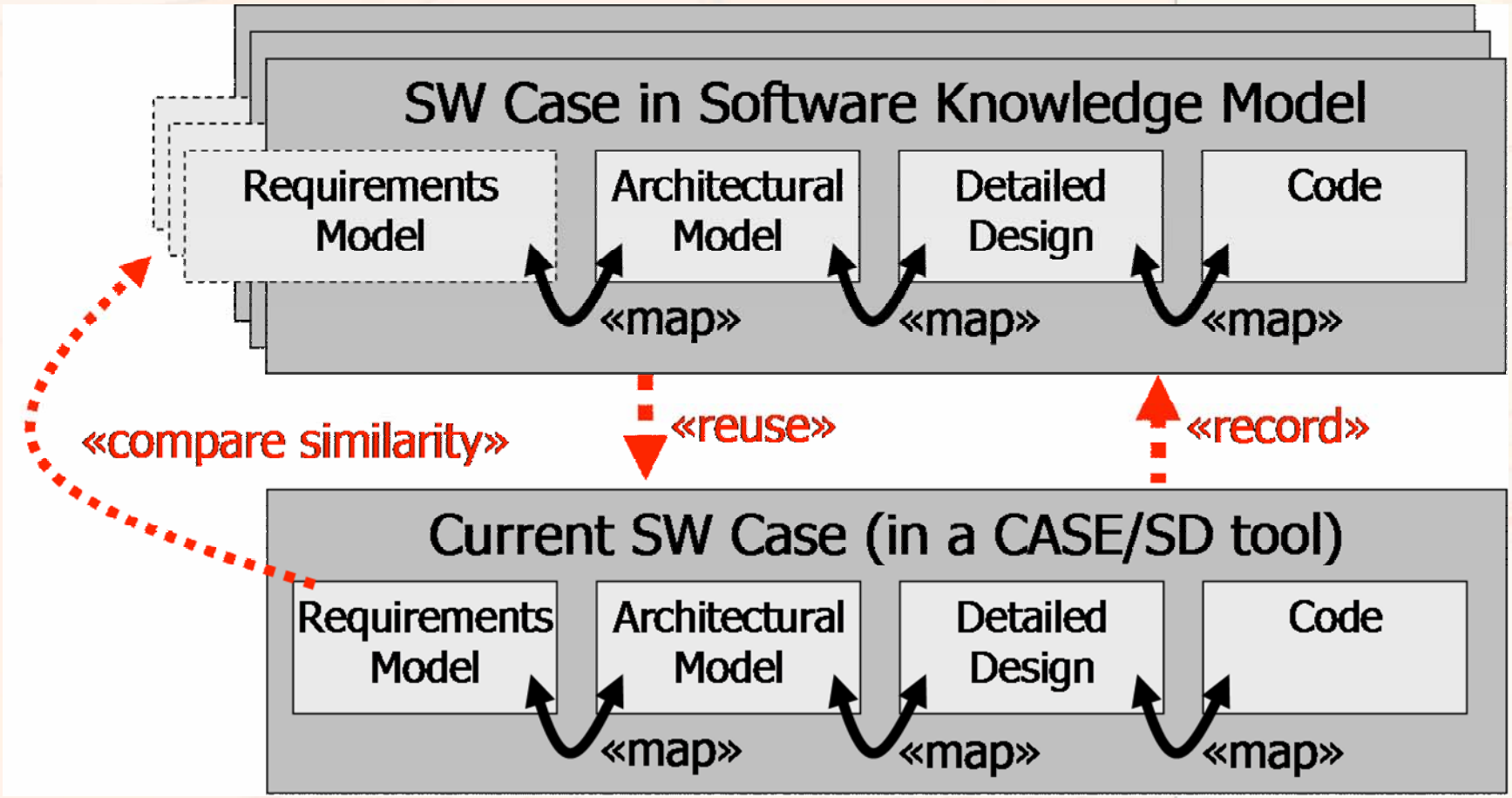
Solution specification

- Architecture
- Design
- Code



Case Retrieval

Identification based on requirements specification



Application

Prerequisites

- Uniform requirements specification throughout various software development projects
- Standard modeling notation (UML & MOF 2.0) for denoting all models transformed from the requirements model

Requirements of concern:

- Restricted English (e.g. SVO sentences)
- Use Case & Sequence diagrams

Variability Modeling

No variability in one case

- One case incl. all assets from requirements to code

Variability through new cases

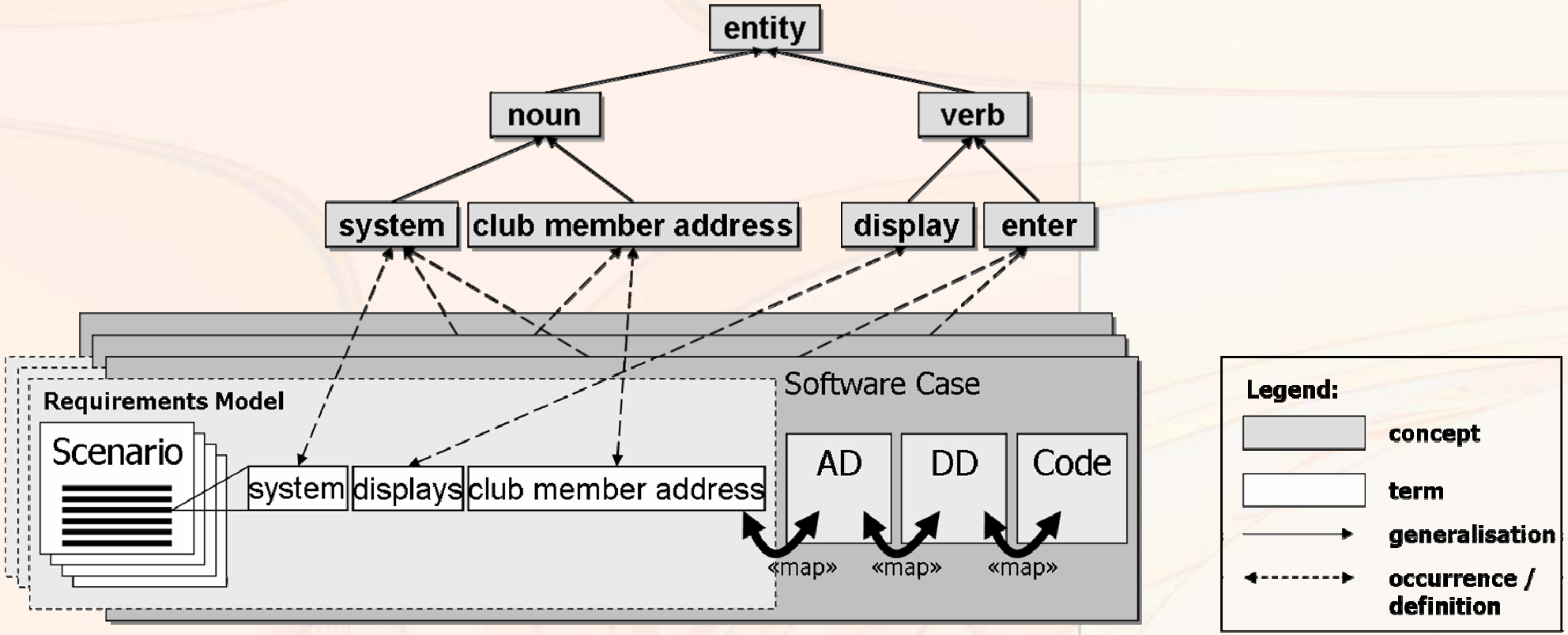
- Every new case introduces variability

Collecting cases: Software Knowledge Model

- ALL previously stored cases
  - All requirements
  - All solutions (architecture, design & code)
- Consolidated vocabulary (no ambiguous terms)

Software Knowledge Model

All cases comprehending vocabulary



Similarity Measure

Lexical comparison

- Names / IDs

Structural comparison

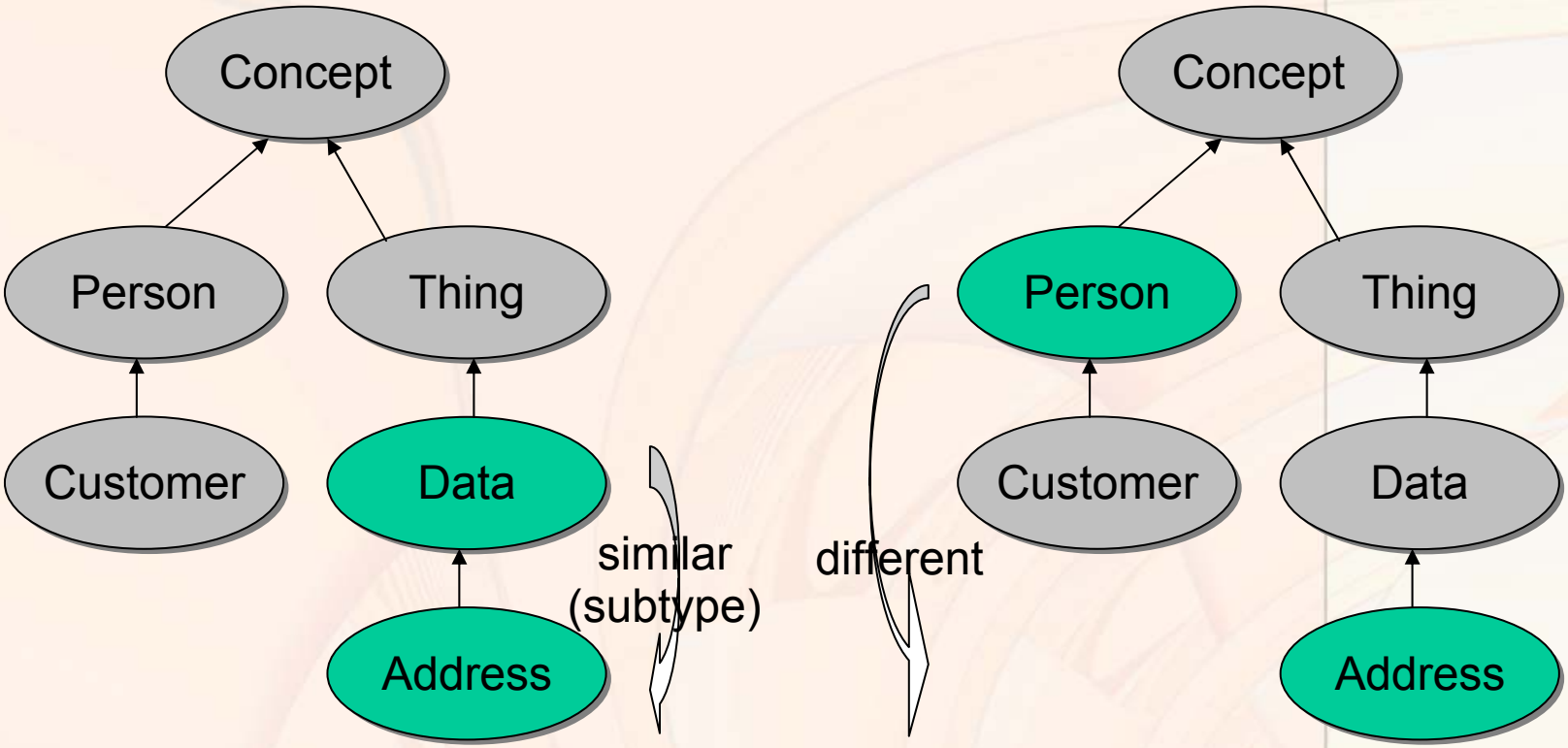
- Synonyms (different names, similar meaning)
- Homonyms (same name, different meaning)
- Taxonomy
- Partonomy

…it's all heuristic measure!

Similarity Measure

Taxonomy

- Term subsumption (*is-a*)
- Inheritance
- E.g. **address** *is-a* **data**

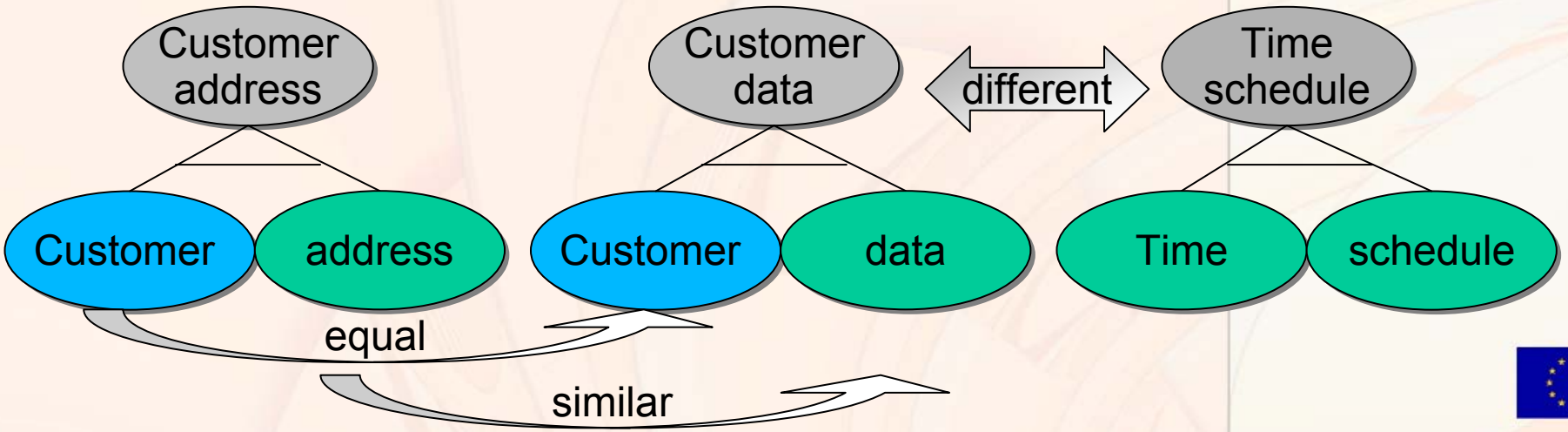


Similarity Measure

Partonomy

- Term composition (*has-parts*)
- E.g. **customer address** composed of **customer** and **address**

- Plus: using taxonomy for the parts
- E.g. **customer data** and **customer address**
  - (**address** *is-a* **data**)



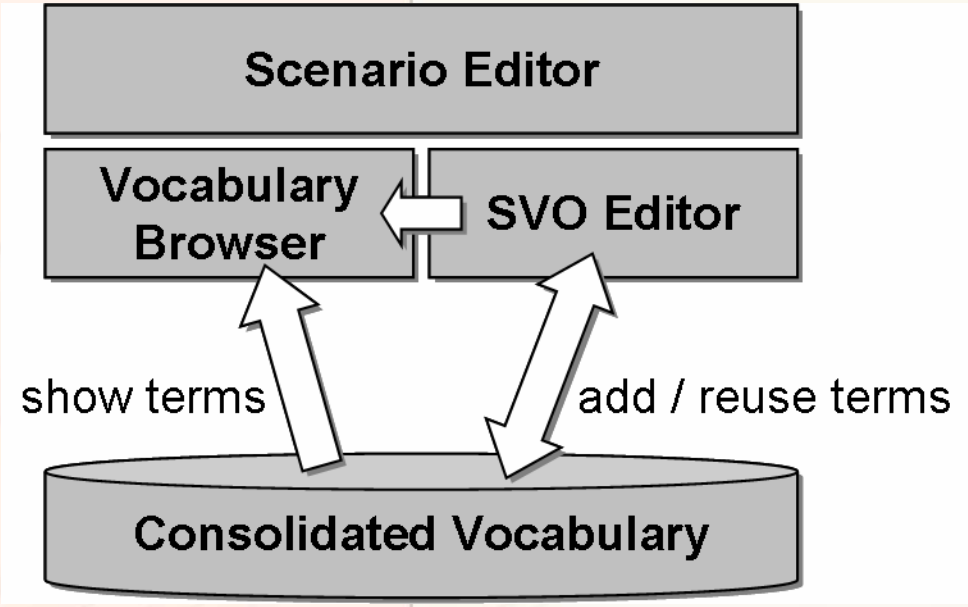
Modeling & Reusing Cases

SVO Editor

- Entering requirements statements
- Statements belong to scenarios

Vocabulary Browser

- Shows related terms
- Reusing terms
- Entering new terms



Modeling & Reusing Cases

Explicit:

- Modeling
- Reuse of requirements specifications

Implicit:

- Modeling of solution specification
  - …through collecting ALL software development artifacts
- Reuse of (partial) case solutions

Variability is managed by merging new cases into one Software Knowledge Model