Transforming Information into Value

empolis
BERTELSMANN MOHN MEDIA GROUP

The Role of Ontologies and Taxonomies in Knowledge Technologies

Carsten Tautz

Knowledge Technologies
Seattle, USA, March 11, 2002

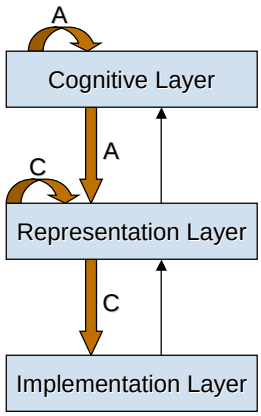
Paradigm of Declarative Programming

empolis
BERTELSMANN MOHN MEDIA GROUP

- All types of programs (can) contain knowledge.
- Imperative programs (C, C++, Java, …):
  implicit representation of knowledge (hidden in algorithms)
- Declarative programming:
  - Program operations are **not** defined step by step
  - Instead: Facts and laws are represented
  - Consequences are drawn by the system itself
- Knowledge-based systems are written in declarative programming languages

3 Layer Model of Knowledge-Based Systems (KBS)

- Cognitive Layer
  - rational
  - informal
- Representation Layer
  - logical
  - formal
- Implementation Layer
  - data structures
  - programs



empolis knowledge management 7

Formal and Informal Knowledge Representation

The concepts *formal* and *informal* are not absolute:

- Knowledge is formal with respect to some system if the system is able to process this knowledge in an adequate way (to react, to initiate some action etc.)
- Otherwise the knowledge is informal with respect to the system, i.e., the knowledge is merely a set of symbols with no meaning to the system

In particular, knowledge may be formal with respect to a human (e.g., a lawyer) but informal to the interpreter of the language Prolog

For a human, to be informal does not mean useless

Computer systems need the knowledge represented in a way which is formal with respect to this system

empolis knowledge management 8

Semiformal Systems

***Semiformal* systems have formal and informal parts. They are concerned with**

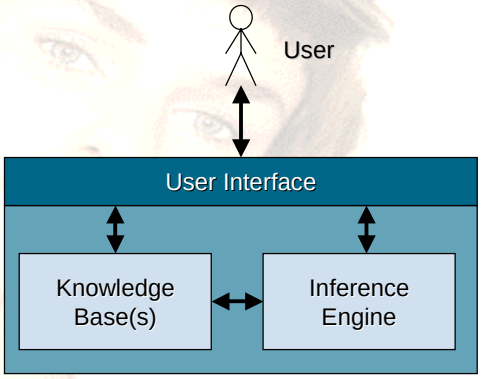
- The structure in which knowledge units are organized: This is usually formal (e.g. a list, a labelled graph, etc).
- The knowledge units themselves:
  - Formal
    (e.g., executable programs,
    elements of a formal representation language)
  - Informal
    (e.g., comments for the user in natural language)

Semiformal systems are useful for:

- The development of formal systems as intermediate products
- Interactive systems for the needs of the human

empolis ■ knowledge management 9

Structure of a KBS in Its Simplest Form



Optional Components:

- Explanation component
- Learning component
- Component for special computations
- Interfaces for special users

Inference Engine:

- May be complex
- May be made up of sub components

Main characteristic is the separation of the knowledge base(s) and the inference engine

empolis ■ knowledge management 10

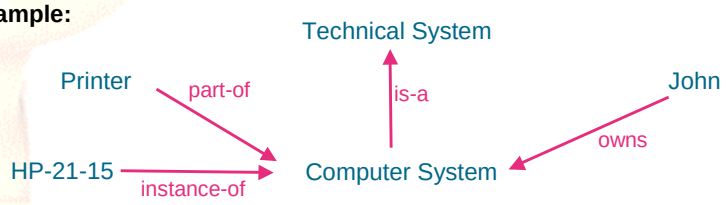
Semantic Nets

… are typical examples of semiformal knowledge representations

… are intended to represent (binary) relations

Taxonomies (hierarchy of concepts) are special cases of semantic nets

Example:



The intention of this semantic net is clear to the human due to the vocabulary used

Definition of Ontology

***ontology* := explicit specification of a shared conceptualization**

- Conceptualization:
  - Abstract model of some area of interest (domain) by identifying *relevant* concepts of that domain
  - Consists of:
    - Relevant concepts of a domain
    - Relations between concepts
    - Axioms about these concepts and relations
- Explicit:
  - Types of concepts used and constraints are explicitly defined
- Shared:
  - Ontology captures consensual knowledge
  - Ontology is not private to some individual
  - Ontology is accepted by a group

Types of Ontologies [Studer]

empolis
BERTELSMANN MOHN MEDIA GROUP

Light-Weight Ontologies (e.g., semantic nets/Topic Maps)

- Concepts, atomic types
- Is-a hierarchy among concepts
- Relationships between concepts

Heavy-Weight Ontologies (e.g., OIL/DAML)

- Cardinality of constraints
- Taxonomies of relations
- Axioms/semantic entailments of various tastes
  - Expressiveness
    (description logics, horn, first order logic, higher order)
  - Inference systems

Case-Based Reasoning

empolis
BERTELSMANN MOHN MEDIA GROUP

A doctor remembers past patient records

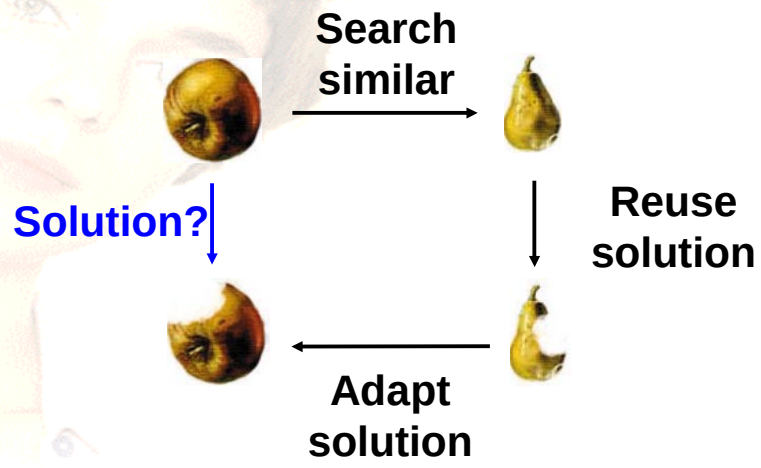
An advocat argues by precedence

An architect reuses designs of existing buildings

A sales agent explains a new product by referring to satisfied customers

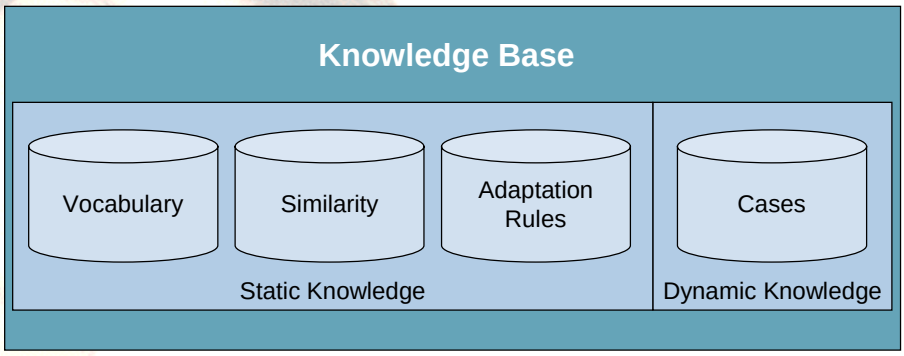
A service technician remembers a similar defect from another machinery

Case-Based Problem Solving



empolis knowledge management 16

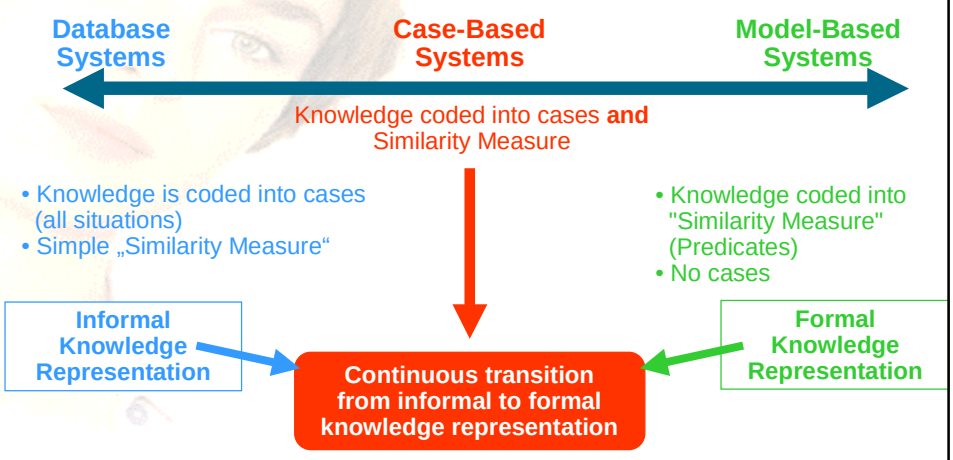
Knowledge Containers of CBR Systems [Richter]



Knowledge can be transferred between containers!

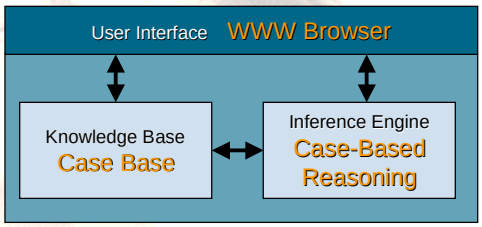
empolis knowledge management 17

Case-Based Reasoning Systems



empolis ■ knowledge management 18

empolis orenge – A Case-Based Reasoning System



orenge

Inferences:

- Similarity-based retrieval
- Completion of queries and cases (Inference using rules)

Optional components

- orenge:explain
  - Explanation component
  - Explains why a hit is relevant
- orenge:caseBuilder
  - Learning component
  - Insertion/Indexing of new cases
- orenge:dialog
  - Component for special computations
  - Intelligent dialog based on principle of "Information Gain"
- orenge:script, orenge:rules
  - Interface for special users
  - Freetext / form / mixed input
  - Automatic recognition of user preferences
    - Example: Low price means price-consciousness → Price becomes important criterion for relevance rating

empolis ■ knowledge management 19