Knowledge Management - Overview

Table of Contents

1 Terminology.......................................................................................................... 2
2 Knowledge Representation (Requirements)...................................................... 5
3 Structured Knowledge Representation .............................................................. 7
4 Knowledge Profiling........................................................................................... 11
5 Situation-Oriented Knowledge Representation............................................... 14
1 Terminology

Data

- Information in a computer-readable form.
- Information in a computer's memory.
- ...

Example:
- Paper Book is an Information
- Electronic Book is data

Data Structure

- Components of a Data Object;
- Properties of the components;
- Relationships between components

Example:

Data management principles are essentially different depending on nature of information to be processed.

Thus, we can speak about well-structured data and ill-structured data at least.
Separating Structure and Content

We speak about well-structured data if structure of data objects may be separated from its actual content.

- There are a number of data objects having identical structure (meta-structure).
- Any data object is just an instance of a previously defined data type.

**Knowledge**

Knowledge is a combination of data and rules for applying the data to solve a particular problem.

Often, ... rules to infer new data of particular interest.

Example 1:

- **Data**: Collection of Learning resources
Knowledge Management - Overview

- **Problem**: Need to learn topic "X"
- **Rules**: How to build a sequence of Units sufficient to learn the topic "X"

Example 2:
- **Data**: Collection of materials on previous projects
- **Problem**: Need to co-ordinate project "X"
- **Rules**: How to co-ordinate a new project reusing info on similar ones.

Example 3:
- **Data**: Collection of software architectural templates
- **Problem**: Need to develop an architecture for a new software system
- **Rules**: How to build a new software system architecture out of existing templates.

**Terminology: Knowledge Management**

Thus, architecture of a knowledge management system may be seen as follows:
2 Knowledge Representation (Requirements)

Knowledge Representation paradigm is a set of significant decisions about the organization/definition of data, rules and problems to find a solution.

Example:
- Semantic Nets;
- Frame-Based Definition;
- Deductive Conclusions

Knowledge Representation Requirements:
- rules are always incomplete - Modifiability;
- rules are defined by different experts - Integrability;
- rules are (may be) contradictory and ambiguous - Verifiability;
- rules are defined by a human being and interpreted by a computer: -
  - Learnability;
  - Efficiency;
  - Memorability;
  - Visualability;
Knowledge Management - Overview

Knowledge Management Tasks:

- **Knowledge Definition** including Integration and Verification;
- Situation Recognition;
- **Reasoning** - resolving the situation (user problem)
3 Structured Knowledge Representation

Structured Knowledge Representation paradigm attempts to split a human knowledge into a number of somewhat simpler knowledge elements, and to define relationships between such elements.

**Alpha Calculus:**

- **Knowledge Definition** - basic facts + rules of inference;
- **Situation Recognition** - Goals;
- **Reasoning** - backward and forward chaining procedures

Database: explains("Sorting Technique","sort1.htm"); interested("Herman", "Algorithms");
isaRelation("Sorting Technique","Algorithms")

\[
\text{visit}(User, Doc) \iff \exists \text{Topic} \left[ \text{interested}(User, \text{Topic}) \& \text{explains}(\text{Topic}, \text{Doc}) \right] \\
\text{visit}(User, Doc) \iff \exists \text{Topic1} \exists \text{Topic2} \left[ \text{interested}(User, \text{Topic1}) \& \text{isaRelation}(\text{Topic1}, \text{Topic2}) \& \text{explains}(\text{Topic2}, \text{Doc}) \right]
\]

? visit("Hermann",x);
? [visit("Hermann","sorting.htm") \& new("Hermann","sorting.htm")]

**Lambda Calculus:**

- **Knowledge Definition** - Lambda Abstractions;
- **Situation Recognition** - Application of an abstraction;
- **Reasoning** - Beta and Alpha reduction mechanisms

\[
\text{visit}(user) \\
\text{explains}(true) \Rightarrow \text{explains}(\text{interested}(user)) \\
\text{explains}(topic) \\
\text{(topic = "Sorting Techniques") \Rightarrow \{"sort.htm","mySort.htm",...\} \\
\text{(topic = "Algorithms") \Rightarrow \{explains("Sorting Techniques"), \\
\quad \text{explains("Search")}\}} \\
\text{(true) \Rightarrow \{\}} \\
\text{visited}(user) \\
\text{(user = "Hermann") \Rightarrow \{"Algorithms","Hypermedia",...\} \\
\text{(user = "Nick") \Rightarrow \{"Databases",\text{interested("Hermann")}} \\
\text{(true) \Rightarrow \{\}}
\]

Normal form:
visit("Hermann") \Rightarrow \{"sort.htm","mySort.htm",...\}
Knowledge Management - Overview

Semantic Nets (General):
- **Knowledge Definition** - Concepts & Relationships;
- **Situation Recognition** - Selection a concept;
- **Reasoning** - Not defined as a part of the concept

![Knowledge Map (Example)]

Knowledge Maps = Semantic Nets (Document Processing):
- **Knowledge Definition** - Concepts, Relationships & Associated Documents;
- **Situation Recognition** - Selection a concept;
- **Reasoning** - Backward Chaining to infer all relevant documents

![Knowledge Map (Example)]
Frames:

- Knowledge Definition - Frames, Slots & Facets (Filters, Procedures, ...);
- Situation Recognition - Selection a Frame;
- Reasoning - Backward Chaining to calculate slot values

Frames (Spin Off)

The Frame concept was further elaborated as different spin off techniques:
- Structured Calculations (mainly CAD & FMS)
- Active Frames = Blackboards (mainly DSS)
- Frame-Channel Model

Structured Calculations:
A special procedure controls a frame appearance (different for different slots)
Active Frames:

- A special event "onSlotValue" that triggers an appearance procedure.
- Calculations are carried out in a backward chaining manner.

Frame-Channel Model:

- Frames are connected by means of channels
- Frames are activated asynchronously depending on special conditions.
4 Knowledge Profiling

The previously discussed Knowledge Representation methods has two-levels of abstraction:

- Data Management Level
- Knowledge (Rule) Management Level

If an informal method of knowledge representation is used, Integrability and Verifiability barely can be achieved.

Obvious solution is based on an additional level of abstraction called "Meta-Knowledge" or "Meta-Rules".

The meta-knowledge level controls knowledge representation provided by different experts.
Semantic Nets combined with a special meta-knowledge about a particular application domain are called Knowledge Domains.

A Knowledge Domain is:
- A set of so-called semantic categories
- A number of meta-links between categories

A particular semantic net is built as an instance of the knowledge domain:
- Users create instances of categories (nodes)
- Relationships are created as instances predefined meta-links.
Current State (Knowledge)

Contributing with new knowledge element

New "Knowledge"

Description:

Needed:

- Project Manager
- Multimedia Developer

Explains:

- Quality Management
- Project Management
- Multi Media

Save  Cancel
5 Situation-Oriented Knowledge Representation

Situation-Oriented *Knowledge Representation paradigm* attempts to gather as many as possible typical solutions, and offers a "best-match" existing solution to a user.

We can speak about following methods of the Situation-Oriented Knowledge Representation:

- Metrics-Based Methods
- Clustering-Based Methods
- Template Recognition Methods

### Metrics-Based Knowledge Representation

In a simplest case, we can speak about so-called "Best-Match solution":

- Offer a number of typical solutions gathered from different experts
- Identify a special metrics allowing to calculate a distance between a concrete situation and a situation where we have a solution
- Offer a "Best-Match" existing solution for a particular situation

![Diagram of Metrics-Based Knowledge Representation]

There may be two different algorithms for processing "Best Match" queries:

- select just one solution closest to an existing one
- identify whether the user is satisfied
- if "Yes", save the solution as corresponding to the user definition of the problem
- if "No", ask the user to redefine his situation, if this is not possible, ask an expert to elaborate a new typical situation

Best results were achieved with a natural language problem definition and with Levenshtein Metrics to calculate distance between sentences. Levenshtein Metrics:

- List of non-informative words
- Semantic net of synonyms: "Is a" relationships
Weight for finding common words (one word found, two words in the same order, etc.)

**Metrics-Based Knowledge Representation**

There may be two different algorithms for processing "Best Match" queries:
- select a number of existing solutions close to a problem
- use distance to calculate a parameter "relevance"
- combine the solutions using the relevance parameter

Example:

Existing solution 1: Go to "WBT-Master", select a knowledge card "Algorithms" (relevance 0.7)
Existing solution 2: Go to "WBT-Master", search for "Sorting Technique" (relevance 0.3)

Solution offered to a user: Go to "WBT-Master", and most probably select a knowledge card "Algorithms". You can also try to search for "Sorting Technique".

Note, Neural Networks may be seen as an application of this technique to Boolean description of a problem and existing solutions (Sample data).

**Clustering-Based Knowledge Representation**

Clustering technique may be seen as follows:
- Identify an application space (hyper cube - a dimension for each criteria)
- Split the space into a number of hypercubes using a special procedure
- Apply the procedure recursively to each sub-cubes until a single solution may be defined for each sub-cube.
- Ask experts to define solutions for each non-divisible hypercube.

![Cluster Diagram]

Now, a similar procedure should be applied to a user's situation to identify a corresponding sub-cube.

offer a solution defined for this hypercube to the user
Knowledge Management - Overview

- identify whether the user is satisfied
- if "No", apply the splitting procedure, and ask experts to define solutions for new hypercubes.

### Template Recognition Methods

This method identifies a number of typical situations - templates, and identifies a solution for each template.

It should especially note that templates/solutions may be defined using special variables.

Sample templates:

- request for an explanation (topic "X")
- request for a prototype document (topic "X")
- request for an assistance in performing some job (job description "X")

The most challenging part is a recognition of a template in user problem definition.