Industrial Impact of Ontology Engineering
-- A success story --

Riichiro Mizoguchi

The Institute of Scientific and Industrial Research
Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka, Japan
http://www.ei.sanken.osaka-u.ac.jp/
Agenda

- A brief summary of
  - Ontological engineering
  - Hozo tool
- A success story of ontological engineering
  - Engineering knowledge management
  - Needs in industries
  - Functional ontology
  - Deployment
  - The current status
- Concluding remarks
Dichotomy of ontology

- **Light-weight** Ontology
  - One like Yahoo ontology
  - **Vocabulary** rather than concepts

- **Heavy-weight** Ontology
  - DOLCE by Nicola, SUOs in IEEE WG
  - One I will discuss
  - **Concepts** rather than vocabulary
  - Philosophical foundation
Typology of Ontology applications

- **Type 1**: Ontology as a common vocabulary
  - Many of the existing ones

- **Type 2**: Ontology for the help of information access
  - Metadata, Ontology-based information search, Knowledge management (KM)

- **Type 3**: Ontology as the medium for mutual understanding
  - Between human and computer = SW; between agents = FIPA, etc.

- **Type 4**: Ontology as specification
  - Ontology-aware (authoring) tools

- **Type 5**: Ontology as foundation of knowledge systematization
  - EPISTLE ontology for Process/Product modeling in industries
Ontology as a MODEL of Knowledge

- Ontology as a means of knowledge systematization
  - Currently, knowledge is accumulated and systematized in the literature for human consumption
  - Ontological engineering contributes to knowledge systematization for computer consumption
Systematization of knowledge

How does Ontological Engineering make a contribution? It provides us with:

– A guideline of articulation of the world
  • How to organize concepts
  • How to discriminate between basic and role concepts
  • etc.

– A common platform on which different models/theories can be interoperable
Ontology as a system of concepts

- Used as conceptual building blocks of knowledge-intensive systems
  - Something deeper than metadata
  - It provides foundation on which a KB or an application system is built

An explicit specification of a hidden conceptualization of the target domain
Ontology Engineering as Content Technology

- It enables you
  - to do Consistent knowledge representation
    - by giving you an explicit viewpoint/guideline
    - Without consistency, computers cannot deal with knowledge properly
  - to share/reuse Knowledge across various domains
  - to make pieces of knowledge interoperable
  - to transform and compare pieces of knowledge used in different domains
What is a Role?

Role is something played by another

The world is full of roles

Unfortunately, there is few tool which can deal with Role properly
Examples of “Roles”

Classification by Context

- **Task role**
  - Symptom role (Fault Diagnosis)
  - Conclusion role (Reasoning)

- **Functional role**
  - Steering Wheel role (Steering Function)
  - Level control valve: played by a flow control valve (Function)

- **Action-related role**
  - Actor role (Any action)
  - Teaching Agent role (Teaching Action)
  - Target object role (Action object)

- **Process-related role**
  - Product role (Final output)
  - Residue role (How it is processed)

- **Organizational(Social) role**
  - Staff role
  - Student role, Nurse role, etc.

- **Relational role**
  - Friend role (Friendship)
  - Parent role (Parent-Child Relation)

- **Compound role**
  - **Japanese Prime Minister** role (Head role + Japanese Minister role + Japanese Citizen role)
  - **Teacher** role (School Staff + Teaching Agent)

- **Attribute role**
  - **Height** role: played by length

- **State-related role**
  - **The Sick** role (Sickness)
  - **Beginner** role (history)
  - **UFO** (Unidentified Flying Object)

- **Intransitive action-related role**
  - **Walker, Runner**
  - **Rotating object**

- **Abstract role**
  - **School staff** role (Super class role of **School teacher** and School clerk roles)

Scope of Our Tool

Scope of Our Framework
The individual corresponding to *Teacher* is the composite of these two instances and totally dependent on them.
Role models

In OWL

```
Person teacherOf Person

School hasTeacher Person

ObjectProperty(hasTeacher domain(School) range(Person))
```

OWL translation of Hozo Role

```
School

hozo:component
hozo:structuralComponent
hozo:dependOn
hozo:roleOf
rdfs:subClassOf

Teacher Role

Teacher

hozo:Role Concept
hozo:Role Holder
rdfs:subClassOf

hozo:Basic Concept

Person

hozo:playedBy
hozo:dependOn
hozo:inheritFrom
rdfs:subClassOf

Teacher

rdfs:subClassOf

Role Holder

rdfs:subClassOf

Teacher

rdfs:subClassOf

Role Concept

rdfs:subClassOf

Basic Concept
```
Characteristics of Hozo

- It properly deals with Role
- It has a version management function
- It has a powerful function to support distributed development of an ontology
- It has friendly GUI
- It translates its ontology into OWL
- It is available at http://www.hozo.jp/
- etc.
Agenda

- A brief summary of ontological engineering
- A success story of ontological engineering
  - Engineering knowledge management
  - Needs in industries
  - Functional ontology
  - Deployment
  - The current status
- Concluding remarks
Engineering Knowledge Management based on Functional Ontology

- It is definite that KM is important
  - It helps people share knowledge and experiences across divisions within an organization and
  - It boosts the daily design/production/maintenance activity

- However, it is currently not very successful, especially for management of engineering knowledge.

- Needs of the industries?
Engineers’ difficulties and needs

- Most of the technical reports are sleeping in DBs
- Most of the technical reports are written in jargons of the respective domains
- Most of the know-how is hidden in engineers head. They don’t know how to share it with others
- Most of the technical documents such as FTA are not reusable because they are specific to the target machine
- Writing a patent is time consuming. Communication with patent writers is very influent
- The conventional documents are not powerful enough for design review
- Documenting and reusing design rationale are hard
Problem analysis

■ Difficulty
  – Ad hoc modeling and authoring incur non-sharable knowledge
  – It is not easy to describe “good” knowledge in a “good” way
    • Knowledge should be consistent and applicable to other domains
    • How to generalize the instance models into generic knowledge?
    • How to describe proper functional models?

■ Issues
  – What a framework for functional knowledge representation
  – How to realize product- and task-independence
  – How to define a common vocabulary
  – How to realize consistent organization and classification of functional knowledge
Ontological engineering

A functional knowledge representation framework based on layers of ontology

- A common language meaningful to multiple (hopefully all) engineers
- Easy to write and to understand
- Applicable to other machines across domains

**A machine**

**Explication of functional structure**

**Way library (KB)**

**Describe**

Cut ingot

Press wire against ingot

Move wire

Hold wire against roller

Rotate roller

**Extract/store**

Wire-saw way

Rotating way

Inner-blade way

**Retrieved**

Other domains
Why function?

- Functional structure is a major part of design rationale of an artifact
  - Explanation of how a machine works is done in terms of function
  - Functional integrity is the identity of an artifact

- Function is an interface between users and products
  - Requirements are written in terms of function
  - Function is a goal to achieve
The current state of the art of functional modeling

- No technology-support
  - Implicit, subjective
  - Not used across domains
- Computer-supported organization/use
  - Circulation and sharing is hard
  - No effective help for various tasks

- Issues
  - Principle
  - Consistency
  - Interoperability

- Causes of difficulties (Lack of generality): Dependent on:
  - Object
  - Task
  - Viewpoints in classification
  - How to recognize the object (Role)
Layers of ontology and knowledge

Specific to an object.
Dependent on designers viewpoint
General knowledge
General concepts
Fundamental

Functional model of the target artifact
Function decomposition tree
Generic function decomposition tree
Attribute trees
Viewpoint-specific structuring
Ways of functional achievement

Description of way of achievement
reference

Conceptualization of function
Specialization from device-centered view

Functional concept ontology
Part library
Physical law Principle

Extended device ontology

Top level ontology(entity, process, time, etc.)
Functional concept ontology

- Function = teleological interpretation of input-output behaviors under a goal

(a) Base functions

(c) Function types

(b) Meta-functions

- Mandatory contribution
- Optional contribution
- Allow
- Prevent
- Provide
- Control
- Drive
- Enable
- Improve
- Enhance

- Is-a
- Make
- Maintain
- Contribute
- Compensate
- Detach
- Separate
- Increase
- Decrease
- Make energy existent
- Consume
- Store
- Convert
- Pass tr.
- Give
- Take
- Transmit
- Remove
- Change
- Make
- Consume
- Drive
- Enabling
- Improve
- Enhance
- Control
- Prevent
Functional ontology

- Functional concepts
- Meta-function, Function type
- Function achievement way
- Functional decomposition
Function achievement way

Functional Decomposition

Subfunction1
Subfunction2
Subfunction3

Way

Method

Theory
Phenomena
Structure
Component
Secret of way description

- **Detach “how to achieve” from the function**
  - To weld = to join by fusion way
- **A way consists of one principle**

**Way knowledge**

- **Sup.Func** Connect together
- **Sub.Func**
  - melt
  - cool

**Theory**
- reorganization through heat
- property: non-decomposable
Example: Organizing step

Ways for “splitting”
- Removing way
  - Lose combination force of a part (the kerf loss)
  - Move the part away

Force
- Ways for Exerting force
  - Exert force
  - Collision way
  - Frictional way
  - Screw-type washing machine
  - Whirlpool friction way
  - Drum-type washing machine
  - Liner fiction way
  - Wire-saw

Ways for losing combination force
- Changing way
  - Physical force way
  - Chemical way
  - Melting way
  - Electrolysis way

Electrolysis cutting

Tensile stress way
- Breaking way
- Make stress
- Breaking by stress

Shearing stress way

Lateral pressure cutting
Functional knowledge sharing tool: SOFAST®
品質要素

要求品質

プロセス条件

風味がある

打つ

こしがある

圧力、時間

工程機能

粘度

水分比率

QFD

Library of function

Achievement ways etc.

Support of many tasks

FTA

QFD

Support of maintenance

方式サーバ

Knowledge/Knowhow

LAN

SOFAST

Clients

Client
Layered Functional vocabulary

Company/domain-dependent layer

Friendly vocabulary layer
SOFAST standard-1

Concept layer
SOFAST standard-2
(Simplified version of the original)

Original functional ontology

To XXX
To land
To put together
To weld
To contact
To join
To put on

Domain-X
Aviation

Way KB
Fusion way
Deployment in a Manufacturing Company

- Production Systems Engineering Division of Sumitomo Electric Industries Ltd., Japan

- Target
  - Production equipment of semiconductor ingots, optical fiber etc.
    - Ingot slicing machines (wire-saw, inter-blade), a tension control system, an optical fiber connector adjusting machine, an inspection machine, etc.

- Situation
  - Number of functional models: more than 100 machines
  - Number of people using: 50
  - Used in daily work in 2 factories
  - SOFAST users’ group: Currently, 13 companies are using
Ontology-based modeling framework

(e) Functional concept ontology

(f) Extended device ontology

Instances of generic functions

“is-achieved-by” relations among functions with ways

(d) Attribute trees of ways (specific to each viewpoint)

“is-a” relations between generic functions

Abstractions of “what to achieve”

Organizing step

Modeling step

Guidelines

(a) Function decomposition tree

(b) General function decomposition tree

(c) The generic ways of function achievement

Ways for

Ways for

OR

AND

OR

AND

AND

Abstractions of “how to achieve”
### Examples of the success stories

<table>
<thead>
<tr>
<th>Problems solved</th>
<th>Effects in daily activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Extention of the claim items in patents</td>
<td>*3 items =&gt; 7 items in average</td>
</tr>
<tr>
<td>2. Speeding up of patent writing</td>
<td>*3 – 4 weeks =&gt; 1 week</td>
</tr>
<tr>
<td>3. Resolution of a trouble of low quality</td>
<td>*Resolved by the same engineer in 2 weeks of the problem left unsolved for 4 months</td>
</tr>
<tr>
<td>4. Fault diagnosis</td>
<td>*Resolved by the same engineer in 3 weeks of the problem left unsolved for 5 months</td>
</tr>
<tr>
<td>5. Fast development of a new machine by a novice</td>
<td>*Resolved in 5 days of the problem left unsolved for 3 months</td>
</tr>
<tr>
<td>6. Fast and efficient design review</td>
<td>*System development within 3 days by a novice reusing the way KB. The job which usually needs 2 weeks by an expert</td>
</tr>
<tr>
<td>7. Improvement of existing machines</td>
<td>*Many examples</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Application to Creative Design Support
Statistical property of patents
adapted from the book written by Prof. Hatamura, 1999

Hierarchy of patents in the order of difficulty

- Simple combination of ordinary techniques: easy
- Complex combination of ordinary techniques: possible
- Simple combination of innovative techniques: difficult
- Complex combination of innovative techniques: very difficult
- Discovery of innovative findings: impossible

Support by system

Level
Display Functions to select one

Search ways and show the result

Select a way

Make functional decomposition tree based on the search result

Continue functional decomposition of sub-functions

Send out light

Direct way

Send out light way

Conversion WL way

Synthesis way

Principle: Three primary colors

Send out light

Send out red light

Send out green light

Send out blue light

Synthesize colors

Send out light

Excite electron

Lower energy

Emit light

Resistor heating way

Principle: Radiation

Prepare Ag with thermion

Bombard Ag with thermion

Excite electron

Luminescence way

Principle: Luminescence

Apply the current to filum

Register heat up

Resist light

Prepare Ag

Luminescence way

Principle: Radiation

Excite electron

Lower energy

Emit light
Interoperability

- Mizlab functional ontology is NOT the unique one in the world
- We are currently doing alignment between ours and Functional basis built by NIST(USA)
In terms of the Semantic Web...

- The semantic web
  - Raw data(URI) <= a document, a device
  - Metadata <= Function decomposition tree
  - Higher order data <= Functional ontology

![Diagram showing alignment between Company-A and Company-B](attachment:image.png)
Typology of function decomposition trees

- **Agent=human**
  - **To function**
  - **To design**
    - **FDT of design process**
  - **To produce**
    - **FDT of a product**
  - **To inspect**
    - **FDT of device inspection**
      - **Target of inspection = Production machine**
  - **To use**
    - **FDT of use case**
The Semantic Web

Ontology/Schema level (OWL classes)

Funnotation Schema

Reference ontology for functions

FMEAS Ontology

Functional basis

Ontology mapping

Meta-data level (RDF statements)

Funnotation

Metadata

FMEAS Ontology

Functional basis

Ontology mapping

Data level (HTML documents)

Document  
d_filtering

Document  
d_distill

Example of an FMEA sheet

Figure 2. Overview of the Funnotation Framework and Systems
Concluding remarks

- **Ontology Engineering as** Content-technology
- **Ontology helps to** systematize the knowledge

- Functional ontology has already been deployed
  
  Two facts which show the value of our work
  
  - **Academic**: We were the only invitee from engineering domain at a WS held by philosophers who work on Function
  
  - **Industrial**: NASA sent us an agent to give us a research fund intended to use our framework saying “Your functional ontology and modeling framework is the best in the world”

- **Ontology engineering is matured enough to give an impact to industries**
The latest activity

- **OntoGear** is to be commercialized soon from Just System Corporation.
  - Extended version of SOFAST
  - Implemented in XML technology as an integrated module of a knowledge management platform
  - It has a friendly user-level vocabulary and a powerful function of Way knowledge extraction from documents