2.1 Meta-Level Representation

Definition: Meta-X

Meta-X means X about X.

Make the inherent relationships between pieces of knowledge visible by switching to a higher-level perspective.

A. Meta-representation for formalized knowledge

Queries:

"Which regulations deal with the surrender of 3b contracts?"

"Which office tasks are affected by a regulation and how?"

Independent representation of regulation:

$$\forall c : \neg\exists t : (contract(c) \wedge type(c, 3b) \wedge surrender\text{-}task(t) \\ \wedge\, object\text{-}of\text{-}surrender(t, c) \wedge elapsed\text{-}time(c) < 3 \text{ years})$$

Another independent representation of regulation:

$$\forall c : (contract(c) \wedge type(c, 3b) \\ \wedge\, (elapsed\text{-}time(c) \geq 3 \text{ years} \Rightarrow possible\text{-}to\text{-}surrender(c))$$

Facts that cannot be derived *within* a theory but can be asserted *about* the theory are meta-knowledge.

Axioms:

$$\forall x : (human(x) \Rightarrow mortal(x)) \tag{1}$$

$$\forall x : (mortal(x) \Rightarrow fallible(x)) \tag{2}$$

Inference Rule:

$$\frac{\forall x : (A \Rightarrow B) \quad \forall x : (B \Rightarrow C)}{\forall x : (A \Rightarrow C)}$$

Within the theory the following theorem can be derived:

$$\forall x : (human(x) \Rightarrow fallible(x)) \tag{3}$$

About the theory something like the following can be stated:

$$\forall f : \exists i, p_1, p_2 : f = mkforall(i, mkimp(appl(p_1, i), appl(p_2, i)))$$

How to link the object level with the meta level?

Definition: Naming Relation

A naming relation is a mapping from syntactic constructs in the object language $\mathcal{L}_{\mathcal{O}}$ to variable-free terms in the meta-language $\mathcal{L}_{\mathcal{M}}$.

Two traditional types of naming relations:

Quotation mark names:

An object expression $\phi \in \mathcal{L}_{\mathcal{O}}$ is written as the constant $\lceil \phi \rceil \in \mathcal{L}_{\mathcal{M}}$

Structural descriptions:

Object expressions become complex terms of $\mathcal{L}_{\mathcal{M}}$ which reflect the syntactic structure of the object level expression.

A meta-theory can only refer to properties of object expressions that are made visible on the meta-level through the naming relation.

Example:

$$object\ sentence: \quad p(f(x)) \wedge q(g(x,y)) \tag{1}$$

$$quotation\ name: \quad \lceil p(f(x)) \wedge q(g(x,y)) \rceil \tag{2}$$

$$structural\ name: \quad p'(f'(var1)) \wedge' q'(g'(var1, var2)) \tag{3}$$

To express that an object-level predicate is commutative:

$$\forall x : \forall y : (prove(p'(x,y)) \Rightarrow prove(p'(y,x)))$$

This, however, expresses something different:

$$prove(\lceil \forall x : \forall y : (p(x,y) \Rightarrow p(y,x)) \rceil)$$

More naming relations:

$$\begin{aligned}
\textit{object sentence}: &\quad p(f(x)) \wedge q(g(x,y)) && (1)\\
\textit{another name}: &\quad p'(func(f,[var1])) \wedge' q'(func(g,[var1,var2])) && (4)\\
\textit{and another}: &\quad pred(p,[func(f,[var1])]) \wedge' \\
&\qquad pred(q,[func(g,[var1,var2])]) && (5)\\
\textit{yet another}: &\quad logical\text{-}const(and, pred(p,[func(f,[var1])]), \\
&\qquad pred(q,[func(g,[var1,var2])])) && (6)
\end{aligned}$$

What is the additional expressiveness allowed by (4) to (6)?

Classically, names are similar if they denote similar syntactic expressions.

But: we can also encode semantic aspects of expressions in $\mathcal{L}_{\mathcal{O}}$ into their names in $\mathcal{L}_{\mathcal{M}}$

Example:

Regulation for dealing with the surrender of 3b contracts:

$$\forall c : \neg\exists t : (contract(c) \wedge type(c, 3b) \wedge surrender\text{-}task(t) \\ \wedge object\text{-}of\text{-}surrender(t, c) \wedge elapsed\text{-}time(c) < \text{3years})$$

A possible name:

forall(var1, neg(exists(var2, contract'(var1) $\wedge'$ type'(var1, 3b')$\wedge'$
surrender-task'(var2) $\wedge'$ object-of-surrender'(var2, var1) $\wedge'$
leq(elapsed-time'(var1), 3years'))))

Encode range of variables into their name:

forall(instance(var1, contract), neg(exists(instance(var2, surrender),
type'(var1, 3b') $\wedge'$ object-of-surrender'(var2, var1) $\wedge'$
leq(elapsed-time'(var1), 3years))))

Let $rule8$ be a short name for the above term.

Then we can state on the meta level e.g.

$regulation(rule8)$

$affects(rule8, surrender)$

... or we can encode that into the name, too:

regulation(surrender, forall(concept(var1, contract),
neg(exists(concept(var2, surrender), type'(var1, 3b') $\wedge'$
object-of-surrender'(var2, var1) $\wedge'$
leq(elapsed-time'(var1), 3years'))))))

B. Meta-representation for semi-structured data

HTML:

⟨HEAD⟩

⟨META NAME="Author" CONTENT="Ulrich"⟩

⟨/HEAD⟩

XML:

⟨BODY⟩

This page is written by

⟨AUTHOR⟩Ulrich Reimer⟨/AUTHOR⟩

⟨/BODY⟩

RDF:

Author(http://research.swisslife.ch/~reimer)=Ulrich

Meta-representation for semi-structured data is needed for

- searching information
- extracting information
- maintaining text sources
- automatic document generation/presentation

Dublin Core: A meta-data standard for *resource discovery*

- resources are web-based, document-like objects
- defines 15 properties ("elements") a resource can have
- each property is defined using 10 attributes
  (like name, definition, comment, version, obligation, etc.)
- widespread consensus
- enables
  - interoperability
  - cost-effective development of tools

(see http://purl.oclc.org/dc)

Element: Title
Name: Title
Definition: A name given to the resource
Comment: A name by which the resource is formally known

Element: Creator
Name: Creator
Definition: An entity primarily responsible for making the content of the resource
Comment: Examples are a person, an organisation, a service

Element: Subject

Name: Subject and Keywords

Definition: The topic of the content of the resource

Comment: A Subject will be given as keywords, key phrases or classification codes (preferably from a controlled vocabulary or formal classification scheme)

Element: Description
Name: Description
Definition: An account of the content of the resource
Comment: May be an abstract, table of contents

Element: Publisher
Name: Publisher
Definition: An entity responsible for making the resource available
Comment: Examples are a person, an organisation, a service

Element: Date
Name: Date
Definition: A date associated with an event in the life cycle of the resource
Comment: Date will e.g. be associated with the creation, modification, availability, validity of the resource

etc.