Dynamic Typing and Non-monetonic Reasoning

Principles for a Semantic Interpreter

Hans Rudolf Straub – Semfinder AG
Content List

1. Background and History
2. Basic principles - Interpretation paradigm
3. Concept architecture and semantic space
4. Transformation process
Background and History

Semfinder  Program for automatic ICD-10 / OPS coding of medical diagnoses and procedures in daily use in more than 300 hospitals

1989   First Program for automatic ICD-coding (Clinic St. Leonhard, St. Gallen)

1996   Starting point: concept molecules

1999   Foundation of the Semfinder company

2009   300 hospitals in Germany
       25 hospitals in Switzerland
Basic Principles ➔ Interpretation Paradigm

- Not a copy ➔ Models are an interpretation of reality
- Information reduction ➔ Models contain less information than reality
- No model is complete ➔ Different models are possible
Basic Principles → Interpretation Paradigm

Not a copy → Models are an interpretation of reality
Information reduction → Models contain less information than reality
No model is complete → Different models are possible

Consequences

A) Look for the perfection of your model
   → static view → monotonic reasoning (DL, FOL)

B) Look for a way to transform models
   → dynamic view → non-monotonic reasoning
Basic Principles → Semiotic Triangle

Space-Time Reality

Mind

Subject

Object 1

Object 2

Model

World

- Reality
- Individual minds
- Models

→ Elements

- Space-time objects
- Concepts
- Symbols

Different worlds

Different behaviour
Basic Principles → Semiotic Triangle

Mind follows reality = realism
Model follows mind = conceptualism

World → Elements
- Reality → Space-time objects
- Individual minds → Concepts
- Models → Symbols
<table>
<thead>
<tr>
<th></th>
<th>Objects</th>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>World</td>
<td>space/time reality</td>
<td>mental space (mind)</td>
</tr>
<tr>
<td>Visibility</td>
<td>visible</td>
<td>invisible</td>
</tr>
<tr>
<td>Boundaries</td>
<td>disjunct, concrete</td>
<td>overlapping, diffuse (fields)</td>
</tr>
<tr>
<td>Existence</td>
<td>autonomous</td>
<td>context dependent</td>
</tr>
<tr>
<td>Composition</td>
<td>addition</td>
<td>complex mix</td>
</tr>
<tr>
<td>Composed of</td>
<td>similar, but smaller objects</td>
<td>different (and bigger) concepts</td>
</tr>
</tbody>
</table>
Basic Principles  $\rightarrow$ concept $\neq$ object

Example (context dependency of concepts)

foot (as a body part)

lower limb (as a body part)

foot (as a location)

lower limb (as a location)
Basic Principles → concept ≠ object

Example (context dependency of concepts)

<table>
<thead>
<tr>
<th>foot (as a body part)</th>
<th>is_part_of</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower limb (as a body part)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>foot (as a location)</th>
<th>is_a</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower limb (as a location)</td>
<td></td>
</tr>
</tbody>
</table>
Basic Principles  \[\rightarrow\]  concept \(\neq\) object

Example (context dependency of concepts)

<table>
<thead>
<tr>
<th>foot (as a body part)</th>
<th>is_part_of</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower limb (as a body part)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>foot (as a location)</th>
<th>is_a</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower limb (as a location)</td>
<td></td>
</tr>
</tbody>
</table>

If concepts are not typed, false conclusion are inherent.
The same concept (foot) can have different types.
Different Relationships follow.
\[\rightarrow\]  allow dynamic typing!  \[\leftarrow\]  use typed (composite) concepts
(not untyped, unlinked objects)
Basic Principles  \( \rightarrow \) concept \( \neq \) object

\[ \text{use composition (postcoordination) whenever possible} \]
\[ \text{keep the atomic concepts inside the clusters} \]
\[ \text{give the cluster an adequate structure} \]

composite concept cluster ("molecule")
Basic Principles  →  concept ≠ object

- use composition (postcoordination) whenever possible!
- keep the atomic concepts inside the clusters!
- give the cluster an adequate structure!
Basic Principles ⇒ Semiotic Triangle

Space-Time Reality

Mind

Subject

Object 1

Object 2

Model

World

- Reality
- Individual minds
- Models

⇒ Elements

→ Space-time objects
→ Concepts
→ Symbols

Different worlds

Different behaviour
Content List

1. Background and History
2. Basic principles - Interpretation paradigm
3. Concept architecture and semantic space
4. Transformation process
How to structure concepts? \(\rightarrow\) 3 formal elements

\[
\text{atomic concept}
\]

\[
\text{injury}
\]
How to structure concepts? → 3 formal elements

atomic concept

An atom is equally type and expression

→ Bifaciality
How to structure concepts? \(\rightarrow\) 3 formal elements

- **atomic concept**
  - An atom is equally type and expression
  - \(\rightarrow\) Bifaciality

- **Hierarchic chain**
How to structure concepts? → 3 formal elements

1. **atomic concept**
   
   An atom is equally type and expression

2. **hierarchic relationship**
   
   \[ \text{Is a} \]

   diagnosis → injury → fracture
How to structure concepts? → 3 formal elements

1. **atomic concept**
   - An atom is equally type and expression
   - Bifaciality

2. **hierarchic relationship**

3. **attributive relationship**

---

Dynamic Typing  
H.R. Straub / OBML-Workshop / Nov. 2009
How to structure concepts? → 3 formal elements

The relationships are invisible on screen, but their positions indicate and distinguish them → Economy
The two relationships span the semantic space

Set theory conditions work inside hierarchies (hierarchic relationships link concepts inside one semantic dimension)

Attributive relationships knit the different semantic dimension
The two relationships span the semantic space

Compositionality with different dimensions

- Diagnose
- Verletzung
- Fraktur
  - offen
  - extraartikulär
  - Mehrfragment
  - nicht disloziert

- Verkehrsunfall
- Knochen
- Radius
- Unterarm
  - distal
  - frisch

- nachgewiesen
- radiologisch nachgewiesen
The two relationships span the semantic space

Focus with encapsulated dimensions

- Diagnose
- Verletzung
  - Fraktur
    - offen
    - extraartikulär
    - Mehrfragment
    - nicht disloziert
- Verkehrsunfall
- Knochen
- Radius
- Unterarm
  - distal
  - frisch
- nachgewiesen
- radiologisch nachgewiesen
The two relationships span the semantic space

Pseudotree structure helps computability

- Diagnose
- Verletzung
- Fraktur
  - offen
  - extraartikulär
  - Mehrfragment
  - nicht disloziert

- Verkehrsunfall
  - Knochen
  - Radius
  - Unterarm
    - distal
  - frisch
  - nachgewiesen
  - radiologisch nachgewiesen
The two relationships span the semantic space.
Top Level Ontology

We distinguish between **formal** and **content** TLO:

**Formal TLO**

→ 3+1 Elements
1 atomic concept + 2 relationships + 1 multiplication element
These 4 Elements are not to change,
but can be filled (named) with any content

**Content TLO**

Content TLO is not proprietary in our system
In principle any content can be given to the four formal elements
Thus we can model other systems and transform them
(semantic interoperability)
Transformation process (inference process)
Transformation process

Intelligence
The rule pool represents the intelligence of the system (includes thus the human engineers brain)

Form of rules
Rules have the same multidimensional, strict composite and pseudohierachical form like all other concepts

No definitions
The non-monotonic system comprises only rules (which can be overruled) and no definitions
Transformation process

Input

Pool of rules

Match
The composite and multidimensional form of both input and rules allow a fast match of the most appropriate rule.
Transformation process

Input

Pool of rules

Execution
The rule contains if's for the match and then's for the modification of the input
Transformation process

Input

Pool of rules

Next rules
The modified input matches with a different rule which is then executed

Procedure continues until no more rules do match → output

Output
Transformation process

Input

Pool of rules

Output

Semantic way of chronology control

Which rule is executed?

Chronology is controlled by the information of the actual input and by dynamic triggers installed into the status during the execution of the process.

The control lies therefore in the concepts of input and rules (in the semantics) and not in algorithm steps of the program.
Transformation process

Input: includes **goals**

Match

Execution

Output

Semantic way of chronology control

Which rule is executed?

The Input contains information about the **observations** to be processed (objective input) as well as information about the **goal** of the interpretation process (subjective input).

Both are given to match in the pool of rules and to select the rule to be executed.
Transformation process

Input

Pool of rules

match

execution

Output

Semantics way of chronology control

Chronology of inference is controlled by the semantic concepts of input and rules.

Non-monotonism

The actual state of the transformation process determines which rule matches. Contradicting rules can thus coexist. Context decides which one is active.

Visibility

All rules and steps are visible to the human knowledge engineer.
Transformation example: from words to meaning
Transformation example: from words to meaning

First step
"Fracture" becomes a diagnosis = conceptualisation
Transformation example: from words to meaning

First step
"Fracture" becomes a diagnosis = conceptualisation

Second step
Attributes from the input are added
Transformation example: from words to meaning

Implicit information
"Bone" was not mentioned in the input, but is implicitly deduced by the system
Transformation example: from words to meaning

2 Molecules are joined
Transformation example: from words to meaning

More implicit information + Coding
Conclusions

- It is possible to have a non-monotonic system working
- Concepts are not objects
- Use well-structured composite concept clusters
Thank you for your attention!