RESI

A Natural Language Specification Improver

Dipl. Inform. Sven J. Körner
Torben Brumm
Prof. Dr. Walter F. Tichy

Institute for Programming and Data Structures (IPD) Tichy
University of Karlsruhe, Germany
“The hardest part of the software task is arriving at a complete and consistent specification, and much of the essence of building a program is in fact the debugging of the specification.”

[Brooks 1987]
specification of a ferrari

- shift-levers on steering wheel
- 450 hp
- bright leather
- aluminum rims
- 335/45 tires
- cup holder
what the customer wanted
that's a lamborghini
Motivation (2): An Example

„After signing-in, the user is redirected to his personal page.“

„When signing in, the username and password are entered.“
Related Work w/o Ontology Support

• Inspections

• Controlled Languages
  [ACE 1995, Propel 2002]

• Quality of Sentences

• Tools for Quality Checks
  [QuARS 2001]
Related Work w/ Ontology Support

- Improving Correctness, Completeness, Consistency and Unambiguousness [Kaiya und Saeki 2005/2006]
- Finding Ambiguities [Kiyavitskaya 2008]
Why Ontologies?

- Ontologies offer world knowledge to a computer system
- They provide semantics and therefore the meaning of a sentence
- How to get that information?
Ontologies, cont’d

Tom saw the plane flying.
Tom saw the mountains flying.
RESI – Requirements Engineering Specification Improver

Specification

Import

Export

RESI

Specification (base)
Pre-Annotation

Specification (pre-annotated)
Rule Application

Specification (improved)
Example From a Specification

„Every pallet is returned after transport.“
Rule: “Check for Nominalization”

- Get Nouns
  - pallet, transport
- Rule
  - Check pallet
  - No Nominalization
  - Check transport
  - Main Verb: transport
- Mark transport
- Ontology
„Every pallet is returned after transport.“

Sentence

Every pallet is returned after transport

Nominalization: Verb *transport*
Rule: Check for Nominalizations

Every pallet is returned after transport.

Nominalization. Use verb transport instead.
Example From a Specification

„Every pallet is returned after transport.“

Sentence

Every, pallet, is, returned, after, transport
Rule: “Complete Process Words”

1. Get all Verbs
2. Specfication ➔ Rule ➔ Ontology

- "return"
- "Check return"
- "2 Argument Lists"
- "Link Argument List"
"Every pallet is returned after transport."

Process Word:
Missing Args: giver and givee
Rule: Complete Process Words

Every pallet is **returned** after transport.

This predicate relates the recipient of a gift to the event in which s/he/it is given that gift. (#$givee GIVING AGENT) means that the #$Agent-PartiallyTangible AGENT is the 'givee' in the #$GivingSomething event GIVING. For the gift-giver, see #$giver.
Example From a Specification

„Every pallet is returned after transport.“
Rule: “Check Articles & Quantifiers”

- Get Quantifiers/Articles
- Check Every
- Note Intended Meaning

- Every
- All (w/o exception)
Example From a Specification

„Every pallet is returned after transport.“

Sentence

Every pallet is returned after transport

Quantor: All (w/o exception)
Rule: Check Articles and Quantifiers

Every pallet is returned after transport.

Quantifier (specify further)

- Exactly: 0
- Minimum: 0
- Maximum: 0
- From: 0 to 0
- None (without exception) [0]
- All (without exception)
- Arbitrary amount [0..*]

Add Mark  OK  Cancel
Evaluation

• 2 Specifications
  • „Video Rental“; manually checked for errors by Kiyavitskaya, Zeni, Mich, and Berry
  • Example from Berry’s ambiguity handbook
• “Video Rental” used for direct comparison
• Problems, that occur most:
  • Passive constructs
  • Definite articles without reference
  • Use of synonyms
Evaluation – Results I

- Similar Meanings
- Indefinite Articles
- Definite Articles
- Incomplete Process words
- Nominalizations
### Evaluation – Results II

#### Statistics

<table>
<thead>
<tr>
<th></th>
<th>Spec1</th>
<th>Spec2</th>
<th>Spec3</th>
</tr>
</thead>
<tbody>
<tr>
<td># Sentences</td>
<td>17</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td># Words</td>
<td>222</td>
<td>99</td>
<td>7</td>
</tr>
<tr>
<td># Characters (no spaces)</td>
<td>1036</td>
<td>486</td>
<td>38</td>
</tr>
<tr>
<td># Nouns</td>
<td>76</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td># Verbs</td>
<td>36</td>
<td>14</td>
<td>1</td>
</tr>
</tbody>
</table>

#### KPI

**Ambiguity Statistics:**

<table>
<thead>
<tr>
<th></th>
<th>Spec1</th>
<th>Spec2</th>
<th>Spec3</th>
</tr>
</thead>
<tbody>
<tr>
<td># Ambiguous Words</td>
<td>44</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td># Additional Meanings</td>
<td>198</td>
<td>42</td>
<td>5</td>
</tr>
<tr>
<td># Exacter Meaning</td>
<td>12</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td># Nominalizations</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td># Incomplete Process Words (Verbs)</td>
<td>14</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td># Missing Arguments</td>
<td>18</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td># Detected Cluttering</td>
<td>11</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td># Quantifiers</td>
<td>8</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td># Definite Articles</td>
<td>24</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td># Indefinite Articles</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>
Summary

• Good software is founded on good specifications
• RESI
  • supports the expert
  • uses ontology knowledge
  • easy application of new rules, various specification formats and new ontologies
if it’s not working

it better be the customer’s fault
Thanks for Your Attention.

Any Questions?