Applying Ontologies and Intelligent Text Processing in Requirements Reuse

Mónica Marrero, Sonia Sánchez-Cuadrado, Anabel Fraga, Juan Llorens

University Carlos III of Madrid

25th May 2008
Reuse in Software Engineering...why?

Cost savings in Software Engineering

- Work more quickly: 28%
- Work better: 17%
- Reuse: 8%
- Other: 47%

[Boehm, B, 1993]

+ Abstraction level of artifacts

+ Reuse Benefits

[Sommerville, 2001]
Reusing Requirements Specifications

- **Quantitative benefits:**
  - Reuse of requirements previously specified
  - Reuse of artifacts linked to them
  - Improvement of development times (*Time to Market*)

- **Qualitative benefits:**
  - Proved quality
  - Related information about the development (problems, solutions)
  - Reduce potential deviations on later developments

- **Problems**
  - Requirements written in natural language
  - Enterprise environments
  - Integrated method, techniques and professional tools
Requirement Reuse Approaches

- Requirements patterns
- Parameterization
- Domain Models

…but no integrated proposals that can deal with:

- **Domain** Modeling
- **Requirements** Modeling
- **Flexibility** in the modeling updates
- **NLP** systems for the analysis and retrieval
- Requirement management **tools**
Reuse System Architecture

1. Requirement description
2. Indexing
3. Query
4. Similarity Analysis
5. Retrieved requirements ordered by relevance

Requirements modeled using a meta-model
REOntology

Incremental Reuse Method
Requirements Engineering Ontology

- Controlled vocabulary of the Requirements Engineering Domain
- Types of relationships
- Explicit relationships between the controlled vocabulary: static relationships
- Lexical-syntactic patterns: dynamic relationships

The users can remove the identifiers in the salary table.

The diagram shows the relationship between the users, remove, identifiers, in, and salary table, with labels [SN1], remove, [preposition], and [SP].
### Example

<table>
<thead>
<tr>
<th>Requirement</th>
<th>The manager could remove the users</th>
</tr>
</thead>
</table>

- **Query done without the Requirements Ontology**

<table>
<thead>
<tr>
<th>Req3</th>
<th>The manager may manage the users from the tables</th>
</tr>
</thead>
</table>

- **Query done with the Requirements Ontology support**

<table>
<thead>
<tr>
<th>Req1</th>
<th>The administrator could unsuscribe the users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req2</td>
<td>User’s table will be modifiable. The administrator could do the elimination of the users</td>
</tr>
<tr>
<td>Req3</td>
<td>The manager may manage the users from the tables</td>
</tr>
</tbody>
</table>
Results

- **Recall and precision rates**

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>With Req. Ontology</td>
<td>0.78</td>
<td>1.00</td>
</tr>
<tr>
<td>Without Req. Ontology</td>
<td>0.72</td>
<td>0.78</td>
</tr>
</tbody>
</table>

- **Ranking of the most relevant requirement in each question:**
  - With the Req. Ontology: 92% are ranked first
  - Without the Req. Ontology: 51% are ranked first
Conclusions and Future lines

Conclusions

- **Applied research:** availability of professional tools to support the techniques and methods described.
- Useful in *any organization* dedicated to software development.
- **Interoperability** between diverse specific domains
- **Incremental** Reuse process

Future lines

- Measure parameters to support the real cost savings in companies