Reusability of Requirements Ontologies

By Rania Alghamdi
Outline

- Introduction
- Requirements Reuse
- Requirements ontologies
- Criteria of reusable requirements
- Examples of reusable ontologies
- Discussion and conclusion
Introduction

- Requirements Engineering (RE) consists of different activities [8].

  - Elicitation: Understanding the application domain.
  - Analysis: Dealing with requirements classification, modeling, and resolving conflicts.
  - Documentation: Aiming to produce requirements document (specification).
  - Validation: Checking the final draft of a requirements document.

- The main goal of RE is to translate the needs of the stakeholders accurately to produce concise requirements specification.
- That takes a considerable time and effort.
Requirements Reuse

As a result...

- Requirements Reuse became necessary [2-4]
  - To decrease cost and time-to-market.
  - To benefit from the other existing high-quality requirements.

- Goldin and Berry [3] found:
  - Reusing is difficult since it depends on choosing the correct components to reuse.
  - Those components should be initially “written to be reusable”.
  - Working on requirements reuse is worth it and pays off.
Requirements Reuse

According to Chernak’s global online IT survey [4]

- Did you reuse requirements on your latest projects?
  - Yes: 59%
  - No: 41%

- What are the obstacles for adopting requirements reuse?
  - No other: 5%
  - Reuse is not supported: 12%
  - Reuse is not important: 18%
  - Outdated: 19%
  - Unstructured: 21%
  - Incomplete: 25%
Requirements Reuse

Approaches applied requirements reuse successfully

Software Product Line (SPL)
- It depends on exploiting the commonalities and the variations of all products in the line [5].
- It focuses on developing a family of products [3].
- It is conformed to large-sized software projects [2].

Software Requirements Catalog (SRC)
- A set of related and sorted requirements.
- Requirements are classified based on their functions and priorities.
- More suitable to be applied in small-sized projects [2].

Requirements ontologies
- It is used to make the knowledge more sharable and reusable.
- It is used for large to small scale software project.
## Requirements ontologies

The concept of ontology is used to reduce the negative effects of RE challenges

<table>
<thead>
<tr>
<th>Some RE challenges</th>
<th>Benefits of requirements ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stakeholders may give incomplete requirements.</td>
<td>Using requirements ontology for specific domain help to capture the missing requirements such as the non-functional requirements [14].</td>
</tr>
<tr>
<td>Stakeholders may provide redundant requirements with different vocabularies (ambiguity).</td>
<td>Ontology is used to restrict vocabulary interpretations and the semantic relations between different entities [8].</td>
</tr>
<tr>
<td>Requirements change continuously.</td>
<td>The explicit relations between entities in the ontology help to trace any changes [8].</td>
</tr>
</tbody>
</table>
Requirements ontologies

- According to the systematic review of Dermeval et. al [1]
  - ontologies have been used to accomplish different RE activities.
  - they found empirical evidences of their advantages.
  - 34% of the studies reused existing ontologies in their contributions to achieve various purposes.
  - They did not study the reusability criteria of those reused ontologies and they mentioned this part briefly.
Different studies reused ontologies for different purposes [1] :

- Some studies reused:
  - Generic requirements engineering ontologies
  - Domain knowledge ontologies
  - Security ontologies
  - Goal-oriented requirements engineering (GORE) ontology
  - Business ontology
  - Scenario-extended problem ontology

Requirements ontologies are generally not widely reused.

Blomqvist et. al related that to the poorly designed and documented ontologies [6].
Criteria of reusable requirements

1. Using the technical contents of worldwide accepted standards of a particular domain provides the guidelines to guarantee the quality of ontology.

2. Being consistent

3. Being Traceable

4. Being unambiguous

5. Using clear implementation and documentation

6. Being complete

7. Using priority ranking

1. Using well-known standards
Criteria of reusable requirements

1. Using well-known standards
2. Being consistent
3. Being Traceable
4. Being unambiguous
5. Using clear implementation and documentation
6. Being complete
7. Using priority ranking

2. It means there is no conflicts between different requirements specifications [7].
Criteria of reusable requirements

1. Using well-known standards
2. Being consistent
3. Being traceable
4. Being unambiguous
5. Using clear implementation and documentation
6. Being complete
7. Using priority ranking

3. The ontology is traceable if there is explicit links between requirements that define the dependency relationships between the requirements in bidirectional manner (back and forth) [7][8]
Criteria of reusable requirements

1. Using well-known standards
2. Being consistent
3. Being Traceable
4. Being unambiguous
5. Using clear implementation and documentation
6. Being complete
7. Using priority ranking

4. It means all the parties (analysts and stakeholders) agree on the same meaning.
Criteria of reusable requirements

1. Using well-known standards
2. Being consistent
3. Being Traceable
4. Being unambiguous
5. Using clear implementation and documentation
6. Being complete
7. Using priority ranking

5. The ontology should be written in a clear and widely known language. Example: (OWL) AND the documentation should be easy to understand by analysts and stakeholders.
Criteria of reusable requirements

1. using well-known standards
2. Being consistent
3. Being Traceable
4. Being unambiguous
5. Using clear implementation and documentation
6. Being complete
7. Using priority ranking
6. All the necessary requirements are included.
1. Using well-known standards
2. Being consistent
3. Being Traceable
4. Being unambiguous
5. Using clear implementation and documentation
6. Being complete
7. Using priority ranking

7. Each requirement has a priority and an expected frequency of changes.
Criteria of reusable ontologies

- Which requirements ontologies have been reused in other studies?
- What are the criteria of the reused requirements ontologies?
Examples of reusable requirements ontologies (1)

- Dzung and Ohnishi [11]

<table>
<thead>
<tr>
<th>Contribution</th>
<th>an ontology-based tool which extracts the initial requirements from the written stakeholders’ interviews.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements type</td>
<td>functional requirements</td>
</tr>
<tr>
<td>Scope</td>
<td>Small</td>
</tr>
<tr>
<td>RE activity</td>
<td>Elicitation</td>
</tr>
</tbody>
</table>
| Reusability Criteria | - consistent  
|                     | - traceable  
|                     | - unambiguous  
|                     | - well documented  
|                     | - implemented in well known language (OWL)                                                              |
| Reused in          | Study in [12] reused the same ontology-based tool for education management system.                        |
Examples of reusable requirements ontologies (1)

- Predefined reasoning rules of [11]

1. Complementary rule
\[ \forall x \exists y. (\text{Complementary}(x,y) \land (y \in \text{Req})) \rightarrow (x \in \text{Req}) \]

2. Supplementary rule
\[ \forall x \exists y. (\text{Supplementary}(x,y) \land (x \in \text{Req})) \rightarrow (y \in \text{Req}) \]
Examples of reusable requirements ontologies (1)

- Predefined reasoning rules of [11]

3. Aggregation Rule
\[ \forall x \exists y. (\text{Aggregation}(x,y) \land (y \in \text{Req})) \rightarrow (x \in \text{Req}) \]

4. Inheritance rule
\[ \forall y \exists x. (\text{Inheritance}(x,y) \land (x \in \text{Req})) \rightarrow (y \in \text{Req}) \]
Examples of reusable requirements ontologies (1)

- Predefined reasoning rules of [11]

5. Inconsistency rule

\[ \forall x \exists y \, (\text{Inconsistency}(x,y) \land (x \in \text{Req}) \land (y \in \text{Req})) \rightarrow ((x \notin \text{Req}) \land (y \in \text{Req})) \lor ((x \in \text{Req}) \land (y \notin \text{Req})) \]

6. Redundancy rule

\[ \forall x \exists y \, (\text{Redundancy}(x,y) \land (x \in \text{Req}) \land (y \in \text{Req})) \rightarrow ((x \notin \text{Req}) \land (y \in \text{Req})) \lor ((x \in \text{Req}) \land (y \notin \text{Req})) \]
Examples of reusable requirements ontologies (1)

An example of requirements ontology for an online store system [11]

An example of checking requirements with ontology [11]
Examples of reusable requirements ontologies (2)

- Wang et al. [9]

<table>
<thead>
<tr>
<th>Contribution</th>
<th>A QoS Ontology Cooperated with Feature Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements type</td>
<td>Non-functional requirements</td>
</tr>
<tr>
<td>Scope</td>
<td>Large/industrial</td>
</tr>
<tr>
<td>RE activity</td>
<td>Elicitation</td>
</tr>
</tbody>
</table>
| Reusability Criteria | -using ISO/IEC 9126 quality model  
- using different priorities  
- well documented  
- implemented in well known language |
| Reused in | The model was reused totally in [10] for requirements elicitation |
Discussion and Conclusion

- Most of the reused ontologies are functional requirements ontologies.
- Requirements ontologies are mostly reused for requirements elicitation purpose.
- Some authors reused their own requirements ontologies in their other works.
- Reusing requirements ontologies is not popular.
- Requirements ontologies could be reused partially or totally.
- Most of the authors stated that comparing to the gained advantages, the effort and time spent on reusing the ontologies were small.
References


References


Thank you,,,
Any questions?