Ontology Assisted Systems Engineering Process
Application to Requirements Engineering

Jose Fuentes – The Reuse Company
Luis Alonso – The Reuse Company
Anabel Fraga – Universidad Carlos III de Madrid
Juan Llorens – Universidad Carlos III de Madrid

CSD&M’14
Paris, 12-14 November 2014
Ontology Assisted Systems Engineering Process

Agenda

- Systems Engineering and Requirements Engineering: current situation
- What can we do and when should we do it?
- KCSE – Knowledge Centric Systems Engineering
- Applications of KCSE to the Requirements Engineering process:
  - Requirements quality improvement
  - Requirements reuse
  - Requirements authoring
Systems Engineering and Requirements Engineering: Current situation
Successful projects

Chaos Report, 2009

- Succeeded: 32%
- Failed: 24%
- Challenged: 44%
Ontology Assisted Systems Engineering Process

Requirements quality: Successful projects
Requirements Quality: source of defects

<table>
<thead>
<tr>
<th>Project Success Factors</th>
<th>% of Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. User Involvement</td>
<td>15.9%</td>
</tr>
<tr>
<td>2. Executive Management Support</td>
<td>13.9%</td>
</tr>
<tr>
<td>3. Clear Statement of Requirements</td>
<td>13.0%</td>
</tr>
<tr>
<td>4. Proper Planning</td>
<td>9.6%</td>
</tr>
<tr>
<td>5. Realistic Expectations</td>
<td>8.2%</td>
</tr>
<tr>
<td>6. Smaller Project Milestones</td>
<td>7.7%</td>
</tr>
<tr>
<td>7. Competent Staff</td>
<td>7.2%</td>
</tr>
<tr>
<td>8. Ownership</td>
<td>5.3%</td>
</tr>
<tr>
<td>9. Clear Vision &amp; Objectives</td>
<td>2.9%</td>
</tr>
<tr>
<td>10. Hard-Working, Focused Staff</td>
<td>2.4%</td>
</tr>
<tr>
<td>Other</td>
<td>13.9%</td>
</tr>
</tbody>
</table>

(Source: CHAOS Report, 2004)

+40% directly related with requirements definition and management
Requirements Quality: source of defects

Study in the scope of **RAMP project** (Requirements Analysis and Modeling Process) in partnership with Airbus Group, RENAULT, EDF, ADN, CORTIM, ENSTA, IRIT, PARIS 1 UNIVERSITY

(end 2010 over 22 industrials in several domains worldwide: interviews and questionnaires)

**Most common requirements defects**

- Requirements not good enough for tests and fabrication: 1
- Requirements not traced: 1
- Derived requirements not justified: 1
- Semantic contradictions: 1
- Use of synonyms or words with multiple meanings: 1
- Not identified as requirements: 1
- Lack of input data (vague needs, non mastered scenarios...): 2
- Confusion between validation of needs and verification of the product: 2
- Too much details: 4
- Not understandable (complex sentences): 4
- Littérature: 6
- Not verifiable: 7
- Not precise enough: 7
- Several requirements in a single requirement: 9
- Not consistency: 9
- Not complete: 10
- Ambiguity: 10
- Requirements expressed as solutions: 14
Requirements Quality characteristics

› IEEE Std. 830:
  › Correct
  › Unambiguous
  › Complete
  › Consistent
  › Ranked
  › Verifiable
  › Modifiable
  › Traceable

› ESA PSS-05,
  ISO/IEC 29148, others:
  › Pretty much the same characteristics

› SMART:
  › Specific
  › Measurable
  › Aligned
  › Realistic
  › Time-limited

"I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the Moon and returning him safely to Earth"
Requirements Quality characteristics

- Good characteristics to check but…
- Can we measure how correct, how complete, how consistent, how measurable… a specification is??
- Are those characteristics SMART?
  - Are they specific?
  - Easy to measure? From a objective point of view?
  - Is it realistic to ask for those characteristics?
Ontology Assisted Systems Engineering Process

What can we do and when should we do it?
Need of knowledge for better Systems Engineering

- The “smarter” we need systems engineering to be, the more dependent on “semantic” knowledge must it be.

- Knowledge must be represented within a knowledge structure (KOS)
  - from informal representations to glossaries, to ..., to ontologies

- The proper selection of the knowledge structure allows different quality enhancement opportunities to the organization
Why managing requirements quality?

Impact of not well defined requirements: “The sooner the better”

INCOSE Handbook V 3.2: Figure 2-4 Committed Life Cycle Cost against Time
Requirements Quality: from things we know → Requirements Authoring

- Experiences showed that about **25% of system Requirements are critical and can grammatically be improved**
  - No Shall: 8 to 10%
  - Forbidden words: 10 to 15%
  - Subject, multiple objects, design: 15%
  - Incorrect grammar: 50%, ...

- Requirements **error costs are high**
  - Fixing requirements after delivery may cost up to **100 times more** than fixing in the requirements definition stage

- Training, best practices and **verifying requirements by reviewers** can help to get SMART requirements:
  - But the process is still **costly and time consuming**

- Introducing quality analysis during the **authoring** activity:
  - **Reduce the number of iterations between System Engineers and sub-contractors and improve the verification activities**
Requirements Engineering Process

- It’s good to automate the verification process but…
- … it’s even better to provide such a help to requirements authors
- “Right the first time”
KCSE – Knowledge Centric Systems Engineering
Key concepts of KCSE

- Semantic indexing
- SKR
- Patterns
- Ontologies
Knowledge Organization in Systems Engineering

“Managing” knowledge for “managing” requirements quality

- System Knowledge Repository (SKR)
  - Allows representing, storing, managing and retrieving
    - Relevant knowledge around the System and its domain (including the SE Process)
    - Digital content (Assets) regarding a particular System

- The SKR is formed by
  - SKB – System Knowledge Base
  - SAS – System Assets Store
System Knowledge Base (SKB): Ontology for Requirements Engineering

Terminology layer: valid terms, forbidden terms, other NL terms, syntactic tagging

Thesaurus layer: relationships among concepts (hierarchies, associations, synonyms…) as well as semantic clusters and relationship types

Patterns layer: definition of requirements grammar

Formalization layer: Semantic formalization: from textual requirements to semantic graphs

Inference layer: for decision making (e.g. consistency, completeness)
**Requirement Pattern**

- Sequential restrictions structure with place-holders for the specific terms and values that constitute a particular knowledge statement,
- A pattern encapsulates the rules for writing and validating a knowledge statement of a particular kind

- Different types of restrictions: terms, syntactic tags, Clusters, sub-patterns
- Other options:
  - Combined restrictions: syntactic + semantic
  - Optional slots
  - Repetitive groups
  - Propagation according to the ontology
Ontology Assisted Systems Engineering Process

Ontology: full example

**Controlled Vocabulary**
- A380
- A350
- System
- Operate
- Temperature
- Environment
- Pressure
- Shall
- At a minimum
- Of

**Thesaurus**
- Temperature
  - “Operation Range“
  - [-60°C, +60°C]
  - “Greater than (>)”
  - “Lower than (<)”
- <<Aircraft>>
- <<Minimum>>
- At a minimum
- Operate
- Work
- Environment
- Pressure
- A380
- A350

**Patterns**
- <<Aircraft>> Shall <<Operation>> At <<Minimum>> Environment Of NUMBER [MEASUREMENT UNIT]

**Formalizations**
- The aircraft shall be able to operate at a minimum temperature of -70°C
- temperature “Greater than (>)” -70°C

**Inference Rules**
- If NUMBER “Lower than (<)” -60°C °C || NUMBER “Greater than (>)” +60°C °C
Semantic indexing

- Tokenization
- Normalization
- Disambiguation
- Pattern matching
- Formalization
Semantic indexing

- Inputs and outputs:

SR0254: “When ice is detected, the car shall show an ice icon in less than 0.5 s”

Pattern matching:
Semantic indexing

- Inputs and outputs:

SR0254: “When ice is detected, the car shall show an ice icon in less than 0.5 s”

Pattern matching:

- When
- <Event> ice is detected
- <System> the car
- Shall
- <Action> show an icon ice
- [Time constraint] in less than 0.5 s
Semantic indexing

- Inputs and outputs:

SR0254: “When ice is detected, the car shall show an ice icon in less than 0.5 s”

Semantic formalization:
Applications
Requirements Quality Metrics

Different initiatives to use a set of easy-to-measure metrics/rules instead of the former characteristics:

- INCOSE Guide for Writing Requirements
- Describes a set of quality characteristics (based on ISO/IEC 29148):
  - Necessary
  - Implementation independent
  - Unambiguous
  - Complete
  - Singular
  - Feasible
  - Verifiable
  - Correct
  - Conforming

But also describes a number of more precise rules
And the matching among characteristics and easy-to-measure rules
How “Smart” must we be in SE: “Requirements quality challenges”

### Correctness
- Text content metrics
- RMS metrics

### Consistency
- Redundant requirements
- Inconsistent units

### Completeness
- Missing or redundant requirements
- Missing links
Requirements semantic retrieval

Based on patterns, every requirement can be formalized by means of semantic graphs.

**UR044**: The **Radar** shall **identify** hits at a minimum **rate of 10 units per second**.

**UR852**: Targets shall be **detected** by the **Electromagnetic sensor** at a frequency not lower than 10 units per second.

**Semantic equivalences**:
- Identifies
- Find
- Distinguish
- Discover
- ...

**System Knowledge Repository**

```
<<Detect>>

Radar

<<Minimum Value>>

Hits

10

Units per Second
```

**Taxonomy**
- System
- Electromagnetic device
- Electromagnetic sensor
- Radar

**Synonyms**
- Target
- Hit
- ...

Requirements semantic retrieval

- Semantic retrieval: for consistency
  - Do we have duplicated requirements in the specification?
  - Do we have conflicting requirements?

- Traceability:
  - Help discovering related requirements that must be traced

- Requirements reuse:
  - Looking for similar requirements in previous projects
  - Traceability and quality are other key factors for requirements reuse
Requirements authoring

- From a requirements quality control process, to empower the author in earlier stages: “Right the first time”

  - Tools for the authors:
    - Correctness quality assessment on the fly
    - Some kinds of requirements consistency on the fly:
      - Vocabulary and grammar consistency
      - Others: e.g. consistency in terms of measurement units
    - Completeness for individual requirements
    - Requirements traceability
    - Requirements reuse
Ontology Assisted Systems Engineering Process

Requirements Authoring Tool

- Author assistance on the fly (typing requirement)
Ontology Assisted Systems Engineering Process

Requirements Authoring Tool

- Semantically similar requirements on the fly

There's a requirement in the DB very similar to the writing requirement.
Ontology Assisted Systems Engineering Process

Requirements Authoring Tool

- Inconsistent measurement units on the fly

There’s a requirement in the DB conflicting with the writing requirement.

There’s a requirement in the DB conflicting with the writing requirement.