Ontology-Assisted Systems Engineering Process with Focus in the Requirements Engineering Process

Anabel Fraga, Juan Llorens, Luis Alonso, and José M. Fuentes

Abstract. Problems found in the current Systems Engineering with focus in the Requirements Engineering Process shown that it could be improved using ontologies for aiding in the process. Requirements engineering in the Systems Engineering process is enhanced and quality of requirements enriched as well, improving Systems Engineering capabilities clearly can result in better Project Performance. One of that is the Requirement improvement and of course the benefit goes to the whole process of development. The more correct, complete and consistent it is, the best performance it will have and ontologies enable a more exhaustive and fast quality process.

1 Introduction

The Systems Engineering Division (SED) of the National Defense Industrial Association (NDIA) established the Systems Engineering Effectiveness Committee (SEEC) to obtain quantitative evidence of the effect of Systems Engineering (SE) best practices on Project Performance. The SEEC developed and executed a survey of contractors for the defense industry (i.e., government suppliers) to identify the SE best practices utilized on defense projects, collect performance data on these projects, and search for relationships between the application of these SE best practices and Project Performance. As shown in the

Anabel Fraga · Juan Llorens · Luis Alonso
Universitad Carlos III de Madrid, Madrid, Spain
e-mail: {afraga,llorens,luis.alonso}@inf.uc3m.es

José M. Fuentes
The Reuse Company, Madrid, Spain
e-mail: jose.fuentes@reusecompany.com

© Springer International Publishing Switzerland 2015

DOI: 10.1007/978-3-319-11617-4_11

report [16], improving Systems Engineering capabilities clearly can result in better Project Performance. One of that is the Requirement improvement and of course the benefit goes to the whole process of development. The more correct, complete and consistent it is the best performance it will have.

The most common defects encountered within requirements were the ambiguity and expressing needs in the form of solution [14]. The adequate requirements management is the most important factor in the success of any Project, even more tan tests, design and programming. If you don't know what you want, you don't know where you go.

The application of ontology engineering in systems engineering seems to be a very promising trend [20], [7]. We call it system verification based on Knowledge Management, and it deals with assisting System Engineers to get a complete and consistent set of requirements (e.g. compliance to regulation, business rules, non-redundancy of requirements…) by using Ontologies, which represent the domains of knowledge of an organization. The combination of Requirements Engineering with Knowledge Management, throughout Information Retrieval from existing sources, allows the verification process to measure quality of a set of requirements by traceability, consistency/redundancy, completeness and noise. Information retrieval enables also to verify the completeness of the ontology using a PDCA (Plan-Do-Check-Act) cycle of improvement. Requirements engineering is the first step and by traceability and Ontology based systems, similar assets of any phase of the development process used in analogous projects could be reused and adapted to a new challenge.

For instance, by using a semantic approach, a requirement can be translated into a graph and by means of NLP (Natural Language Processing) techniques. It could be compared with another requirement or test or document by similarity, as for instance:

UR044: The Radar shall be able to detect hits at a minimum rate of 10 units per second

This example will be used to illustrate the rest of the sections.

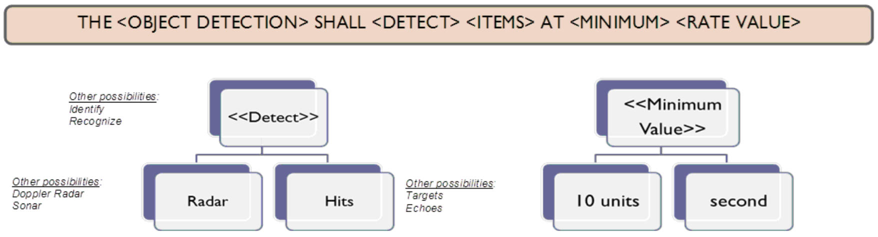


Fig. 1 A requirement similarity comparison

The use of a semantic tool for representing the knowledge of the requirement (for instance) allows us to compare diverse requirements written in a different way because what matters is in this case the minimum value of detection and it must be 10 seconds, if another requirement says it must be 15 seconds, then a conflict or contradictory requirements for the radar are available in the Systems requirements which means a huge error in calculus if it is an important measure for the radar. Extending NLP with inference rules capabilities, taxonomies, thesaurus and semantic groups enable computer tools to enhance systems engineering requirements, models, architectures, tests or documentations consistency. Even though, 10 seconds could be compared in another range of metrics in order to guarantee that this value is unique within the requirements of the system.

One of the problems found is the similarity of requirements, incompleteness, different use of measurement units, and so on. It is hard to compare sequentially a group of thousands of requirements. Most common requirement defects are the following: not verifiable, not precise enough, several requirements gathered in a single one, lack of consistency, not completeness, ambiguous, requirements expressed as solutions, and so on. An ontology-assisted System Engineering Process can sort out these problems in a reasonable time.

The reminder of this paper is structured as follows: Section 2 explains how is nowadays the requirements engineering process, Section 3 explains how ontologies are considered and applied in the ontology-assisted requirements engineering process within the systems engineering process, Section 4 explains an overall application of the ontology-assisted vision in the systems engineering process, and finally Section 5 includes some conclusions.

2 How Is Nowadays the Requirements Engineering Process

A requirement is an identifiable element of a function specification that can be validated, and against which an implementation can be verified. [3] [10] [9] [19] [8] [6] [5] The main requirements attributes are mentioned below [8]:

- Each requirement must be uniquely identifiable.
- Verifiable: it must be very simple to determine the success/failure property for every requirement.
- Clear: it must describe what the software does, without involving other parts of the system.
- Practical: each requirement must be derived from a user need.
- Complete: describe the whole customers’ situation case.
- Consistent: without internal conflicts between requirements.
- Correct: describes accurately and exactly the customer’s situation and need.
- Modifiable: documented in a structured and accessible manner.
- Traceable: against whatever number of artifacts used in the life cycle.
- Accurate: the requirement should only be understood in only one way.

Software requirements engineering [3] [10] [9] [19] [8] is a disciplined, process-oriented to the definition, documentation, and maintenance of software requirements throughout the software development life cycle. Software Requirements Engineering is made up of two major processes: requirements development and requirements management. [7]

- Requirements Development encompasses all of the activities involved in eliciting, analyzing, specifying and validating the requirements.
- Requirements Management encompasses the activities involved in requesting changes to the requirements, performing impact analysis for the requested changes, approving or disapproving those changes, and implementing the approved changes. Furthermore, includes the activities used for ensuring that work products and project plans are kept consistent and tracking the status of the requirements as one progresses through the software development process.

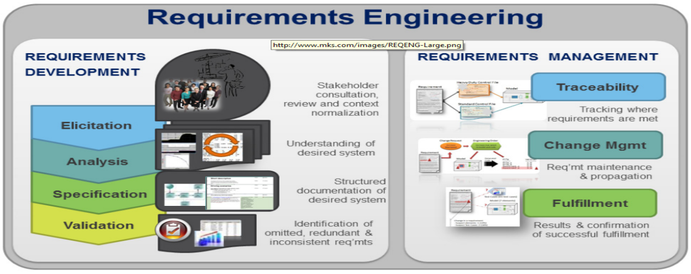

Requirements Engineering
http://www.mks.com/images/REQENG-Large.png
REQUIREMENTS DEVELOPMENT
Elicitation
Analysis
Specification
Validation
Stakeholder consultation, review and context normalization
Understanding of desired system
Structured documentation of desired system
Identification of omitted, redundant & inconsistent req'mts
REQUIREMENTS MANAGEMENT
Traceability
Tracking where requirements are met
Change Mgmt
Req'mt maintenance & propagation
Fulfillment
Results & confirmation of successful fulfillment

Fig. 2 Requirements Engineering Process by PTC Company. [17]

Requirements specification produces a formal software requirements document from the set of requirements. Its purpose is to give visibility of the requirements to the system/test engineers and to allow the formal verification of the requirements specification. [3] After that, requirements verification is the final process that ensures that the requirements are correct. It represents the formal agreement and acceptance of the requirements that the software must implement. It ensures that the specification is complete and that the requirements are feasible in terms of its technical implementation and verification, taking into account time/budget constraints. Once this process is finished, the software requirements document is formally issued and constitutes the technical baseline for further developments.

The requirements management process handles the modification of the software requirements after the document has been formally reviewed and agreed. Thus, the activity is done in order to ensure that the baseline of the software is known and to analyze affordability of change in terms of budget and design. The change is proposed by the system engineers and must be agreed by the software development team. Once agreed, the change is included in the software requirements document and a new baseline is established.

There are some rules that establish how the requirements must be written and which mistakes must be avoided. The INCOSE (International Council on Systems Engineering) rule states that the requirements must be clear and concise, complete, consistent, verifiable, affordable, achievable, necessary, bounded, traceable with independent implementation. [11]

The SMART (mnemonic criteria to guide in the setting of objectives) criteria define that a requirement must be Specific, Measureable, Achievable, Relevant and Traceable. It is important to keep the requirement as simple as possible, avoiding unnecessary information. [11]

A suggested list of requirement attributes is [8]:

- Write using the same words with exact meaning established.
- Utilize clear, unambiguous phraseology and punctuation.
- Do not misplace comas.
- Use correct verb tense: o Shall - a demand. o Will - a future happening. o Must - a strong desire. o To be, is to be, are to be, should, should be - nice to have, desired capabilities.

The use of certain words should be avoided, because they can convey uncertainty:

- Superlatives such as “best” and “most”.
- Subjective language: “user friendly”, “cost effective”, “Easy to use”…
- Vague pronouns: he, she, this …
- Ambiguous adverbs and adjectives: “minimal”, “almost always”, “significant”, “quick/rapid/safe”, “sufficient”,…
- Open ended, non-verifiable terms: “provide support”, “but not limited to”,…
- Comparative phrases: “better than“, “higher quality”
- Loopholes: “as applicable“, “if possible”
- Other indefinites: and so on, TBD, TBC, etc.

Other important attributes of good requirements are:

- Identify all stakeholders across all products lifecycle phases and involve them during requirements development and validation in order to build what is really needed.
- Take ownership of requirements.
- Always use the imperative shall and identify subject, result and success criteria in measurable terms.
- Requirements shall be uniquely and identified.
- Write requirements clearly and explicit.
- Requirements must be verifiable in order to allow proving that they have been met.
- Justify each requirement by a rationale and/or trace to its source.
- Allocate and flow down requirements.

Therefore it is necessary to comply with these rules avoiding possible risk of failure in the implementation and in the final product that results in an unsatisfied customer or corporative damage.

Any mistake in the requirements definition phase is distributed downwards until low level requirements being almost impossible to fix. Thereby, those mistakes must be caught in the early development process.

When defining any system we will have a set of requirements that attends to their relation and dependences among them, which have to comply with the CCC philosophy: Consistency, Completeness and Correctness. [18] Completeness means that the set of requirements does not need further improvement because it contains everything concerning the system definition. Consistency states that the requirements do not have contradictions within them, not duplicated or even that a term is used for the same semantic in all the requirements.

3 Ontology-Assisted Requirements Engineering Process in the Systems Engineering Process

The solution developed to improve the problems when writing requirements is establishing ontological structures.

It can be defined as a common vocabulary in which shared knowledge is represented. A specification of a representational vocabulary for a shared domain of discourse (definition of classes, relations, functions and other objects) is called ontology. It is an explicit and shared specification of conceptualization. Ontology is a knowledge-base within the system development process, which has information about the structure of the System, in the subject of application of the written requirements. It consists of controlled vocabulary, thesaurus, light ontology and full-ontology with patterns and representation schemas. [14] In a nut shell, this is the vision of the ontology:

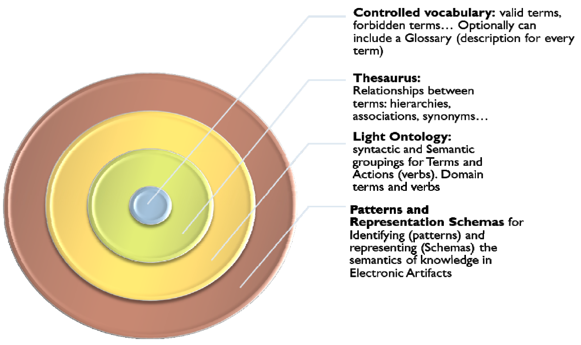

Controlled vocabulary: valid terms, forbidden terms… Optionally can include a Glossary (description for every term)
Thesaurus: Relationships between terms: hierarchies, associations, synonyms…
Light Ontology: syntactic and Semantic groupings for Terms and Actions (verbs). Domain terms and verbs
Patterns and Representation Schemas for Identifying (patterns) and representing (Schemas) the semantics of knowledge in Electronic Artifacts

Fig. 3 Ontology Layers [9]

3.1 Controlled Vocabulary

It is needed for standardizing and normalizing the terminology used in the custom application. The input information must/should match the controlled vocabulary. Using a glossary with different categories of terms, the ontology may store:

- Client business related Terms: those terms focused into the customer area to be considered.
- General Language Terms: those terms related to the idiomatic field.
- Syntactically relevant phrases: Adverbs, Adjectives, and so on.
- Invalid terms: those terms that could be of no relevance.

Following the example introduced in the first section, the terms extracted from that requirements are shown in Fig. 4.

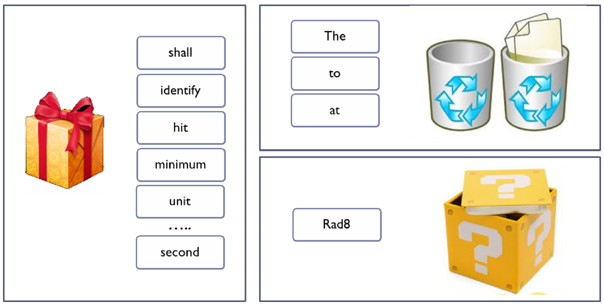


Fig. 4 Stop words, general language terms and business terms

3.2 Thesaurus

A Thesaurus stores relational information regarding the terms in the glossary. It is used for:

- Retrieval purposes
- Representation and normalization purposes
- Suggestion purposes (Decision support)
- "solution specific" purposes

It enriches the controlled vocabulary for including specific relationships of the represented domain: synonyms, hierarchies, and general associations. Following the example, a new requirement is introduced:

UR03442 : The Radar shall be able to distinguish hits at a minimum rate of 10 elements per s

Fig. 5 shows a relationship between Rad8 PTT and Rad8 of equivalence, as well between distinguish and identify, and between s and second. On the other hand, Radar is a super class of Rad8, and due to the synonymy of Rad8 PTT. Also a generic relationship between Radar and Sensor is shown.

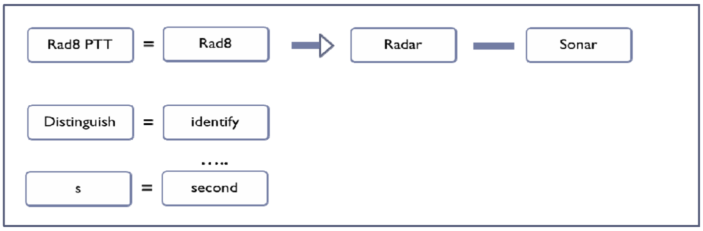


Fig. 5 Relationships in the requirements examples UR044 and UR03442

3.3 *Light Ontology*

The Light ontology contains:

- Syntactic Information: For NLP purposes and for specific Pattern Restrictions

An example of the kind of syntactic information to be represented following the example shown before:

UR044 :The Radar shall be able to detect hits at a minimum rate of 10 units per second

UR563 :The Doppler Radar shall be able to Identify hits at a minimum rate of 10 units per second

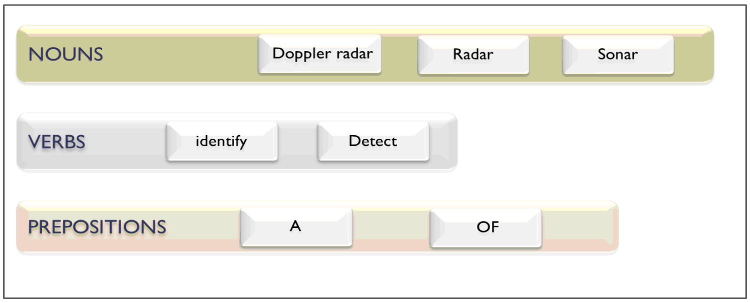
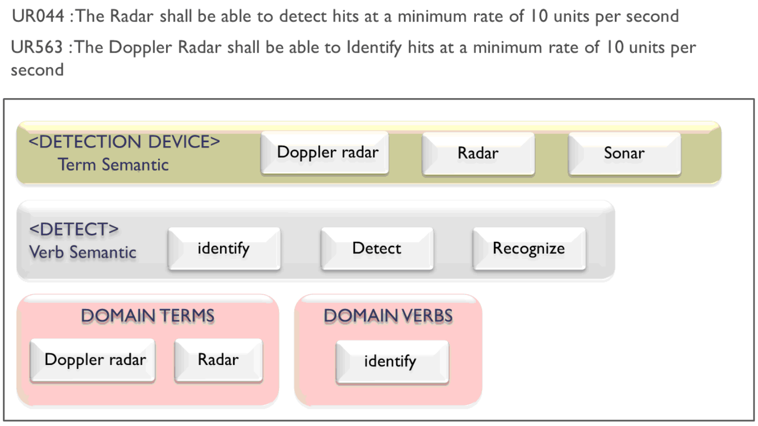


Fig. 6 Nouns, Verbs and prepositions recognized in the requirements

- Semantic Information: For Retrieval purposes and for specific Pattern Restrictions. The semantic information extracted from the requirements is shown as follows (Fig. 7).
- Idiomatic Information: For NLP purposes.
- Artifact Type Information: For Retrieval Filtering purposes.


UR044 :The Radar shall be able to detect hits at a minimum rate of 10 units per second

UR563 :The Doppler Radar shall be able to Identify hits at a minimum rate of 10 units per second

Fig. 7 Semantic grouping of verbs and terms, domain terms and domain verbs

3.4 *Patterns and Representation Schemas*

The patterns [12], also called boilerplates, are sequential restrictions based on a place-holders structure for the specific terms and values that constitute a particular knowledge statement, where the restrictions can be grammatical, semantic, or even both, as well as other patterns.

A pattern encapsulates the rules for writing and validating a knowledge statement of a particular kind. It may be possible to abstract away from the textual representation by using conceptual graphs in the style of [13]. It needs to be completed following these phases:

- Creating the detection pattern.
- Creating the formal representation of the knowledge statements based on the pattern information.

An example of a pattern created for the requirement example UR044 is shown as follows in Fig. 8.

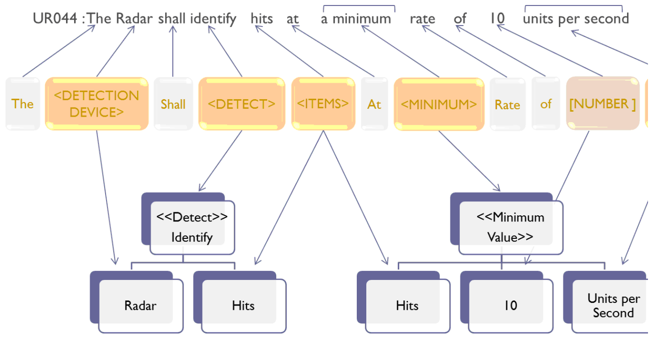

UR044 :The Radar shall identify hits at a minimum rate of 10 units per second

Fig. 8 A pattern and its knowledge representation

3.5 The Ontology in the Center of the Process

The appropriate selection of the knowledge structure allows different possibilities to the organization.

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

The SKR is formed by:

- SKB – System Knowledge Base
- SAS – System Assets Store

The System Knowledge Base (SKB) supports the complete system knowledge-base for the application of semantic services around the system life cycle (Including SE). The System Assets Store (SAS) manages a formal representation of the System Assets: Requirements, Models, and so on. It is the base for offering services around these assets: Reuse, Traceability, MDE, TDD, etc.

3.6 Ontology Tools Available

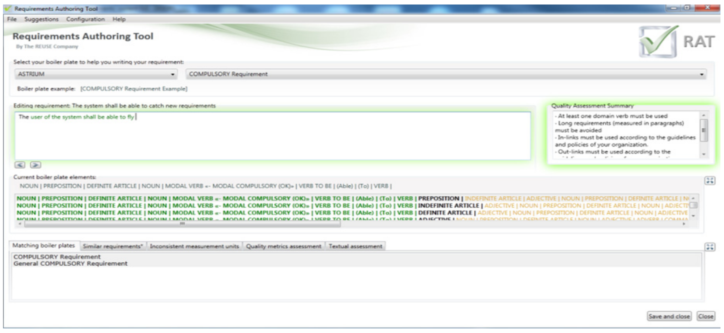
Diverse tools are available for building ontologies [1], the most known among practitioners is Protégé [15], but it is difficult to build an ontology in any of the available tools and then use it for analyzing the meaning of the information available in the company. There is a suite of tools called Requirements Quality Suite (RQS) that contains a knowledge management system called knowledgeMANAGER [20], it is connected to the whole suite and a semantic use of the text is done when dealing with requirements within the system engineering process. The process and knowledge-assisted process is supported by this suite of tools [20].

4 Applied In

The application of ontologies in the systems engineering process and mainly in the systems requirements can be applied in:

4.1 Authoring

The image below shows the tool Requirement Authoring Tool (RAT). RAT is part of the Requirements Quality Suite (RQS) and leads authors during the process or requirements writing. The list box in the center of the screen represents the proper grammar of the requirements, while the editing box (in the top of the screen) provides dropdown lists with the set of valid concepts for every part of the requirement.



Select your boiler plate to help you writing your requirement:
ASTRIUM
COMPULSORY Requirement
Boiler plate example: [COMPULSORY Requirement Example]

Editing requirement: The system shall be able to catch new requirements
The user of the system shall be able to fly

Quality Assessment Summary
- At least one domain verb must be used
- Long requirements (measured in paragraphs) must be avoided
- In-links must be used according to the guidelines and policies of your organization.
- Out-links must be used according to the

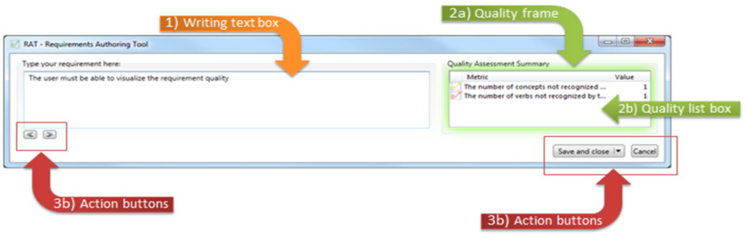
Current boiler plate elements:
NOUN | PREPOSITION | DEFINITE ARTICLE | NOUN | MODAL VERB «- MODAL COMPULSORY (OK)» | VERB TO BE | (Able) | (To) | VERB |

NOUN | PREPOSITION | DEFINITE ARTICLE | NOUN | MODAL VERB «- MODAL COMPULSORY (OK)» | VERB TO BE | (Able) | (To) | VERB | PREPOSITION | INDEFINITE ARTICLE | ADJECTIVE | NOUN | PREPOSITION | DEFINITE ARTICLE | N
NOUN | PREPOSITION | DEFINITE ARTICLE | NOUN | MODAL VERB «- MODAL COMPULSORY (OK)» | VERB TO BE | (Able) | (To) | VERB | INDEFINITE ARTICLE | ADJECTIVE | NOUN | PREPOSITION | DEFINITE ARTICLE | NOUN | ADJECTI
NOUN | PREPOSITION | DEFINITE ARTICLE | NOUN | MODAL VERB «- MODAL COMPULSORY (OK)» | VERB TO BE | (Able) | (To) | VERB | DEFINITE ARTICLE | ADJECTIVE | NOUN | PREPOSITION | DEFINITE ARTICLE | NOUN | ADJECTIVE

Matching boiler plates | Similar requirements* | Inconsistent measurement units | Quality metrics assessment | Textual assessment
COMPULSORY Requirement
General COMPULSORY Requirement

Fig. 9 Requirement Authoring Tool based on Patterns (Ontology).

It is important to note that, aside of providing intellisense support for writing the requirements, RAT also provide quality information on the fly according to the agreed set of metrics. The image below represents this quality analysis on the fly (Fig. 10).



Type your requirement here:
The user must be able to visualize the requirement quality

Quality Assessment Summary

Metric	Value
The number of concepts not recognized ...	1
The number of verbs not recognized by t...	1

Fig. 10 Intellisense information detailed

4.2 Quality

The ontology will aid in the process of checking the CCC criteria, as shown in Fig. 11, two requirements that contains the same information but using different words are detected as duplicated requirements even they are not typed in the exact way, but the domain relationships in the ontology and also the semantic grouping aids in the detection of this kind of requirement, we can call it a quality detection system.

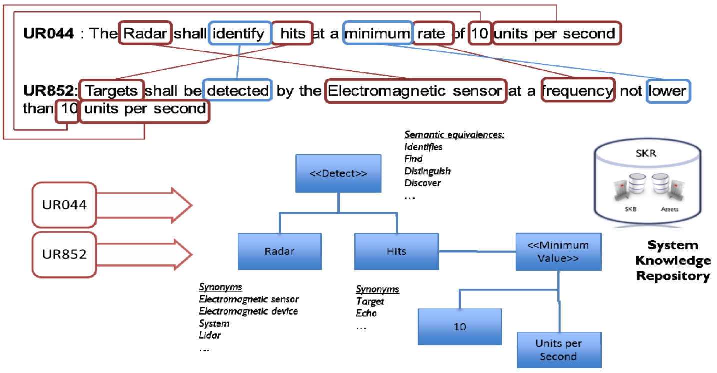


Fig. 11 Example of an ontology used for detecting duplicated requirement

There is one (similarity) of the diverse metrics of quality that could be measured by using ontologies.

4.3 *Reuse*

Once a project has been created and the ontology build properly, when the next project arrives to the system engineering process, the requirements related to the project in a specific area could be reused as well as the ontology involved for the small set of requirements to be reused.

5 Conclusions

The requirements in any project are one of the most important assets, if not the most. A bad group of requirements might have terrible implications in a developed system. For instance a requirement detailed in various parts of the requirement list using different measurement units might cause an accident or a failure during operation of any system.

Classical sequential review process of requirements is costly in terms of time consuming. Then support of tools for lexical, syntactic analysis enables to correct bad requirements writing before business of project reviews.

One of the next challenges in the Industry is to reach an ontology-assisted system engineering process to write SMART requirements at a first shot.

The use of ontologies and patterns is a promise of doing better requirements engineering and knowledge reuse in any system engineering project, for instance CRYSTAL [2] project that willl apply ontologies and the patterns to facilitate the human job, avoid mechanical checks and increase quality.

References

1. Fraga, A.: Universal Knowledge Reuse. PhD Thesis. Carlos III of Madrid University (2010)
2. CRitical sYSTem engineering AcceLeration (CRYSTAL EU Project), http://www.crystal-artemis.eu/ (last visited November 13, 2013)
3. Braude, E.: Software Engineering. An Object-Oriented Perspective. John Wiley & Sons (2001)
4. Fanmuy, G., Fraga, A., Lloréns, J.: Requirements Verification in the Industry
5. Genova, G., Fuentes, J., Llorens, J., Hurtado, O., Moreno, V.: A framework to measure and improve the quality of textual requirements. Requirements Engineering, http://dx.doi.org/10.1007/s00766-011-0134-z, doi:10.1007/s00766-011-0134-z
6. Guide for Writing Requirements, INCOSE Product INCOSE-TP-2010-006-01, V. 1 (April 2012)
7. Chale, H., et al.: Reducing the Gap between Formal and Informal Worlds in Automotive Safety-Critical Systems. In: INCOSE Symposium 2011, Denver (2011)
8. Alexander, I.F., Stevens, R.: Writing better requirements. Addison-Wesley (2002)
9. Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. John Wiley & Sons (1997)
10. Sommerville, I.: Software Engineering. Pearson-Addison Wesley (2005)
11. INCOSE, Systems Engineering Handbook, http://www.incose.org/ProductsPubs/products/sehandbook.aspx (last visited November 13, 2013)
12. Dick, J., Llorens, J.: Using statement-level templates to improve the quality of requirements. In: ICSSEA 2012 (2012)
13. Sowa, J.: Conceptual structures: Information processing in mind and machine, Addison Wesley (1983)
14. OBSE Fundamentals, The Reuse Company, Juan Llorens (UC3M) - José Fuentes (TRC), http://www.reusecompany.com (last visited November 13, 2013)
15. Protégé. Stanford University Development, http://protege.stanford.edu (last visited November 15, 2013)
16. Report CMU/SEI-2008-SR-034
17. Requirements Engineering, PTC Product and Service Advantage, http://www.mks.com/solutions/discipline/rm/requirements-engineering (last visited November 13, 2013)
18. Requirements Patterns for Authoring, MASI, Panagiotis Mourtis & Susana Beltrán. Internal Report UC3M (2012)
19. Pressman, R.: Software Engineering: a practical guide, 6th edn. McGraw-Hill
20. The Reuse Company (TRC), http://www.reusecompany.com (last visited November 13, 2013)