Are Standards an Ambiguity-free Reference for Product Validation?

Alessio Ferrari, Mario Fusani, Stefania Gnesi

ISTI – CNR, Pisa, Italy

alessio.ferrari@isti.cnr.it, mario.fusani@isti.cnr.it, stefania.gnesi@isti.cnr.it

Foreword

- The question is a particular case of a more general one:

Could the responsibility for any problems of “standard-compliant” systems also depend on the standards themselves?
What if any of such problems includes safety?

Grounds and motivations - 1

- Efficacy of Standards (software Standards for safety critical systems) investigated in 1990's
- Renewed interest on the efficacy of [Safety] Standards
  - ECCT 2014: "Are Standard unplanned experiments?"

    A common belief was under discussion:
    "the ability of Safety Related Standards to provide safe systems is generally taken for granted and safety problems with the systems are caused by non-compliance"
- Experience gained at ISTI-CNR on Requirement Engineering, typically on "requirement disambiguation" (also an output of ECCT2014)

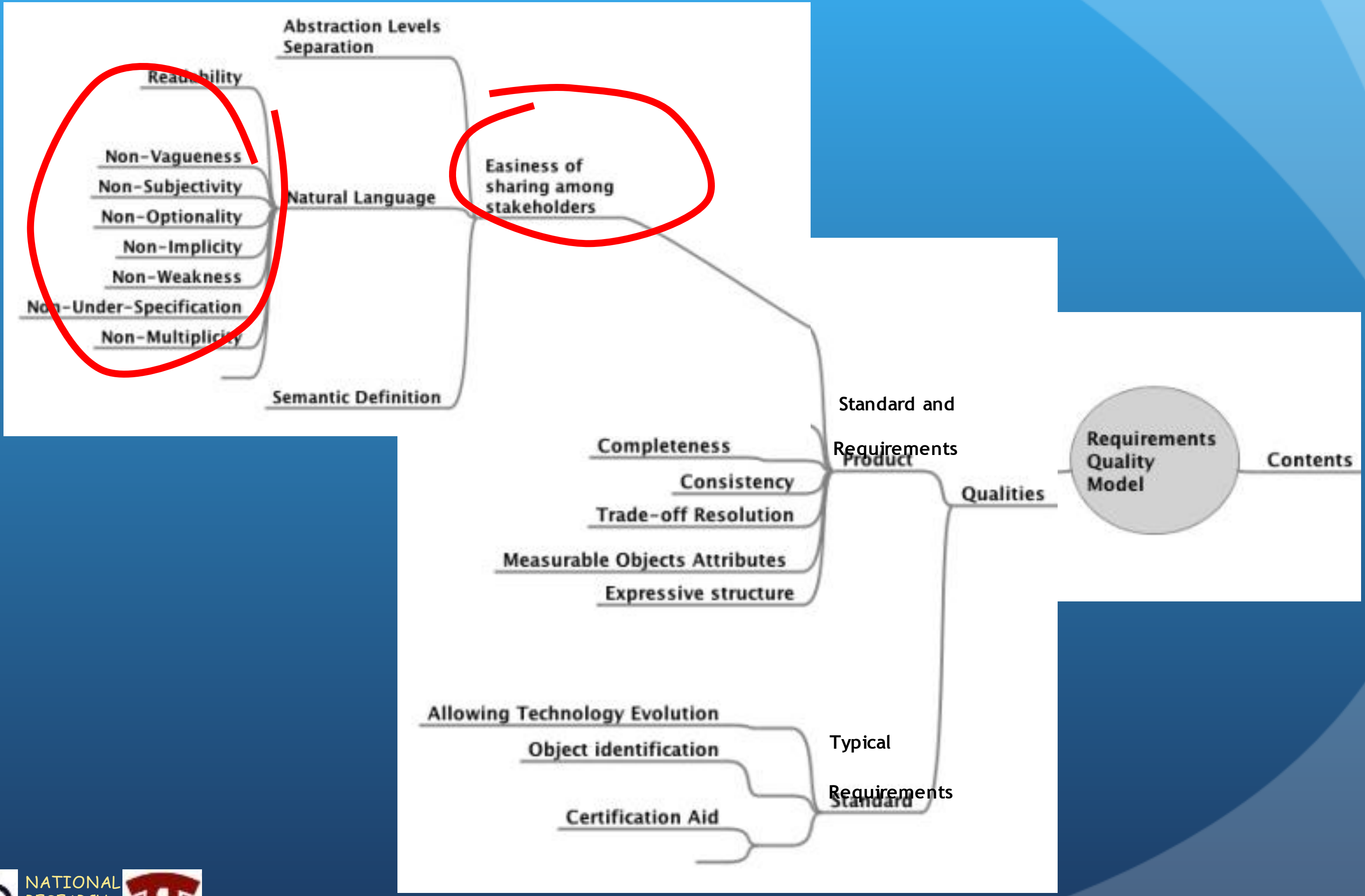
Grounds and motivations - 2

- Outcomes from the 1990's (works by Fenton et al.):
  - overemphasis on process rather than product
  - Imprecisely testable requirements
  - Non-consensus technical recommendations
  - Unclear about risks and benefits
  - too big and poorly organized (and abstraction levels mixing)
- Outcomes from an 2014-2015 investigation (NASA et al.):
  - What *exactly* developers must do for compliance?
  - One-developer assumption
  - Ambiguous assurance requirements
  - Unsolved problem of complex software behaviour at system level
  - Confidentiality of data

Grounds and motivations - 3

- Detecting and removing ambiguity from requirements expressed in natural language has been one of the objectives of RE: methods and tools have been proposed worldwide.
- The investigation has been extended in other areas (manuals, contracts, legal documents, laws)
- Why not Standards?
  - Can methods and tools used in RE be used also in Standards? With what differences?

Reqs. non-ambiguity as an aspect of a general quality model for requirements

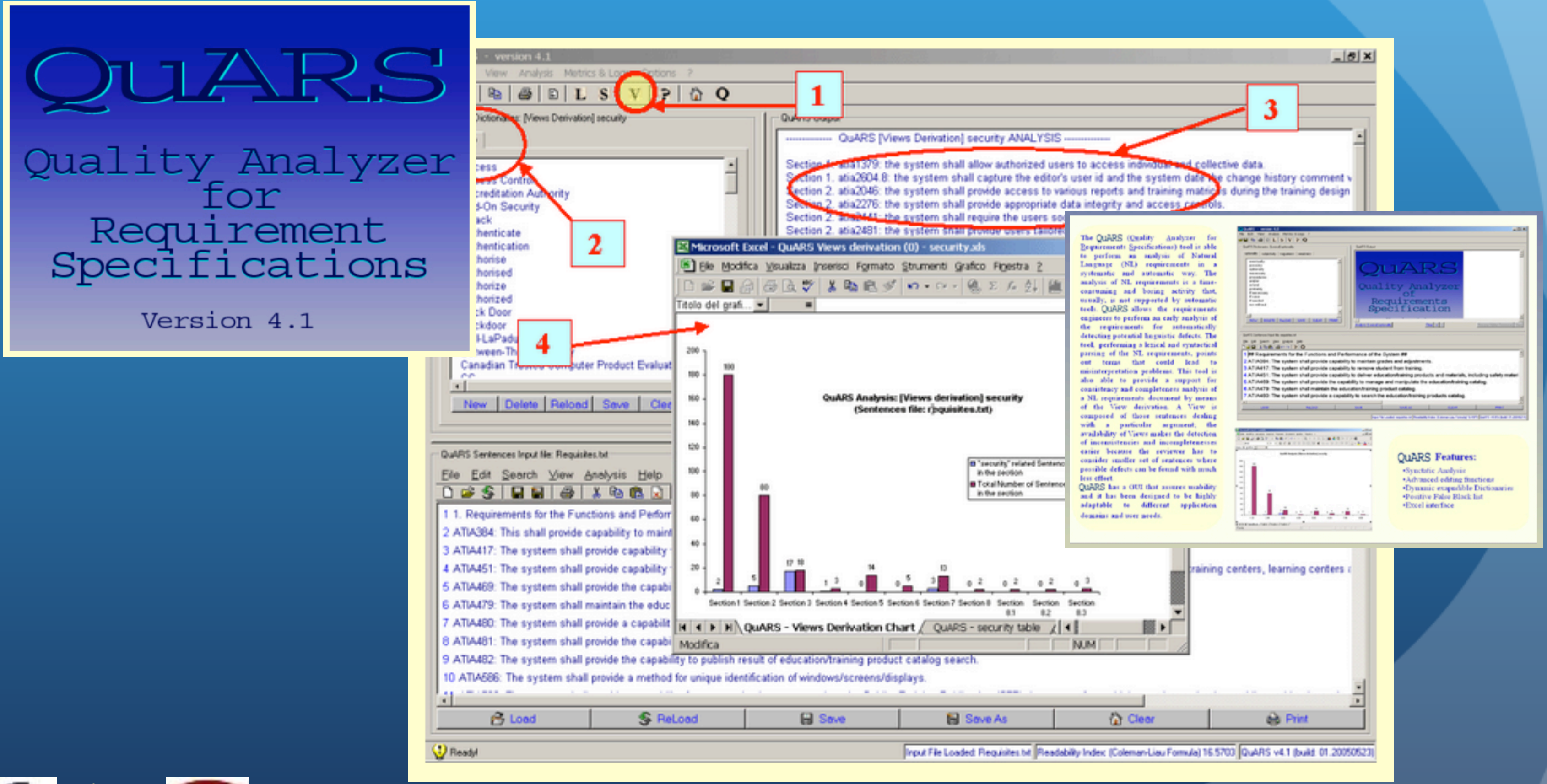


QuARS tool – ambiguity related detectable characteristics

Quality Characteristics and Sub-characteristics	
Lexical	**Vagueness**: items having a non-uniquely quantifiable meaning (*“adequate”, “easy”, “bad”, “clear”, “far”, “close”, …)*
	Subjectivity: personal opinions or feelings (*“simple”, “known”, “similar”, “taking into account”, …*)
	Optionality: optional parts (may or may not be considered) (*“possibly”, “if needed”, “if appropriate”, …*)
Syntactical	**Implicity**: subjects or objects are not expressed by means of their specific name (*“The previous task”, “it”, …*)
	Under-specification: Generic terms are used without adjectives or specification (such as “of …”) (*“ The manual ”, “access to”, “interface”, “function”, “document” …*)
	Multiplicity: more than one main verb or subject occur in requirement (*“< sentence> and / or <sentence>”, …*)

A tool developed at ISTI-CNR

- QuARS (**Quality Analyzer for Requirement Specification**), late 1990's – 2012


QuARS
Quality Analyzer for Requirement Specifications
Version 4.1
version 4.1
View Analysis Metrics & Logs Options ?
L S V ? Q
1
2
3
4
Dictionaries: [Views Derivation] security
QuARS Output
New Delete Reload Save Clear
Microsoft Excel - QuARS Views derivation (0) - security.xls
File Modifica Visualizza Inserisci Formato Strumenti Grafico Finestra ?
Titolo del grafi...
QuARS Analysis: [Views derivation] security
(Sentences file: requisites.txt)
"security" related Sentences in the section
Total Number of Sentences in the section
200
180
160
140
120
100
80
60
40
20
0
2 180
5 80
17 18
1 3
0 14
0 5
3 13
0 2
0 2
0 2
0 3
Section 1
Section 2
Section 3
Section 4
Section 5
Section 6
Section 7
Section 8
Section 8.1
Section 8.2
Section 8.3
QuARS - Views Derivation Chart
QuARS - security table
Modifica
NUM
QuARS Sentences Input file: Requisites.txt
File Edit Search View Analysis Help
Load
ReLoad
Save
Save As
Clear
Print
Ready!

---------- QuARS [Views Derivation] security ANALYSIS ----------

Section 1. atia1379: the system shall allow authorized users to access individual and collective data.
Section 1. atia2604.8: the system shall capture the editor's user id and the system date the change history comment w
Section 2. atia2046: the system shall provide access to various reports and training matrices during the training design
Section 2. atia2276: the system shall provide appropriate data integrity and access controls.
Section 2. atia2441: the system shall require the users so
Section 2. atia2481: the system shall provide users tailored

1 1. Requirements for the Functions and Perform
2 ATIA384: This shall provide capability to maint
3 ATIA417: The system shall provide capability
4 ATIA451: The system shall provide capability
5 ATIA469: The system shall provide the capabi
6 ATIA479: The system shall maintain the educ
7 ATIA480: The system shall provide a capabilit
8 ATIA481: The system shall provide the capabi
9 ATIA482: The system shall provide the capability to publish result of education/training product catalog search.
10 ATIA586: The system shall provide a method for unique identification of windows/screens/displays.

Input File Loaded: Requisites.txt | Readability Index (Coleman-Liau Formula) 16.5703 | QuARS v4.1 (build 01.20050523)

The QuARS (Quality Analyzer for Requirements Specifications) tool is able to perform an analysis of Natural Language (NL) requirements in a systematic and automatic way. The analysis of NL requirements is a time-consuming and boring activity that, usually, is not supported by automatic tools. QuARS allows the requirements engineers to perform an early analysis of the requirements for automatically detecting potential linguistic defects. The tool, performing a lexical and syntactical parsing of the NL requirements, points out terms that could lead to misinterpretation problems. This tool is also able to provide a support for consistency and completeness analysis of a NL requirements document by means of the View derivation. A View is composed of those sentences dealing with a particular argument, the availability of Views makes the detection of inconsistencies and incompleteness easier because the reviewer has to consider smaller set of sentences where possible defects can be found with much less effort.
QuARS has a GUI that assures usability and it has been designed to be highly adaptable to different application domains and user needs.

QuARS Features:
- Syntactic Analysis
- Advanced editing functions
- Dynamic expandible Dictionaries
- Positive False Black list
- Excel interface

NATIONAL RESEARCH COUNCIL

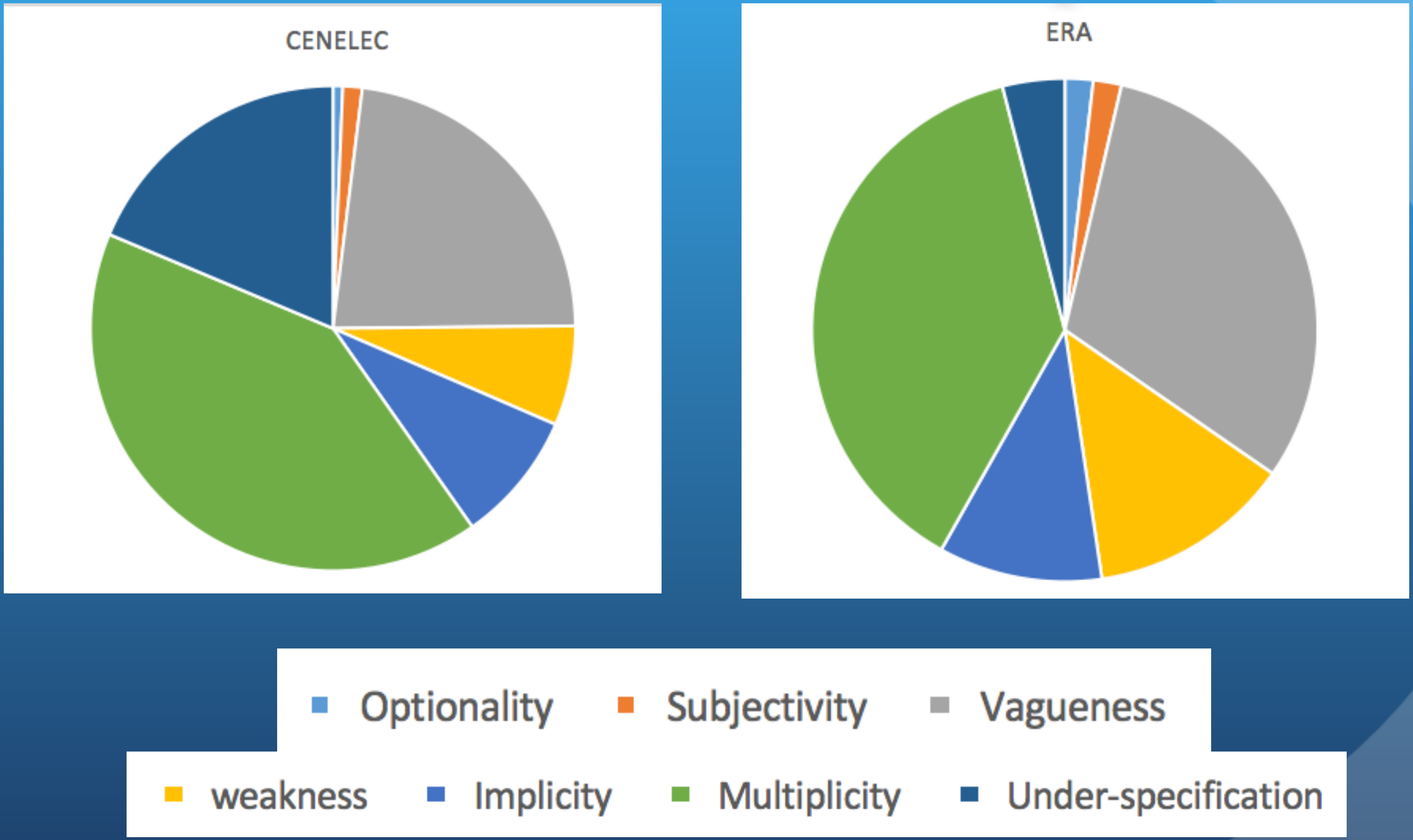
Project Reqs. vs. Standard Clauses

	Project requirements	Standard clauses, annexes, …
Traditional review effort	Moderate and limited	Very high
Impact (how many users, ..)	Limited	Full domain (developers, assessors)
Generality	Limited (product oriented)	Higher (process oriented)
Stability	Moderate (Continuous requirement process)	Higher (reference for a community)

Findings using QuARS

QuARS ANALYSIS	EN 50128: 2011 - Clauses			ERA ERTMS		
	sentences	flagged	%	sentences	flagged	%
Lexical						
Optionality	624	4	0,6	645	5	0,8
Subjectivity	624	8	1,3	645	5	0,8
Vagueness	624	141	22,6	645	86	13,3
weakness	624	41	6,6	645	36	5,6
Syntactic						
Implicity	624	54	8,7	645	29	4,5
Multiplicity	624	253	40,5	645	105	16,3
Under-specification	624	115	18,4	645	11	1,7

Findings using QuARS



Meaning and impact of the warnings

- The warnings are just potential defects. A manual inspection is needed to decide if warnings denote real ambiguities (false positive removal)
- False positives run to about 40-50% of warnings, higher for Standards
- Much actual research work is devoted to reduce false positive removing effort
- Once false positives are removed, an impact or risk analysis is expected to justify text correction

QuARS results on EN50128:2011

- Clause 7.3.4.19 (interface description)

The line number:
131. g) existence of synchronization mechanisms between functions (see e)).
contains a unspecified sentence because the term: function

- Which function (safety, non-safety, ..) ? At what abstraction level?

QuARS results on EN50128:2011

- Clause 8.4.4.8 (application data/algorithms production)

The line number:
554. a) that the application test specification meets the general requirements for readability and traceability (5.3.2.7 to 5.3.2.10 and 6.5.4.14 to 6.5.4.17) as well as the specific requirements expressed in the subclause (8.4.4.6),
is defective because it contains the wording: general

- Clearly a false positive, since there is a definition of "general requirements for …"

QuARS results on EN50128:2011

- Clause 8.4.4.6 (application data/algorithms production)

The line number:

581. *8.4.8.6 care must be taken in the verification process and validation test phase of the generic software in order to assure that all relevant combinations of data and algorithms are considered.*

is defective because it contains the wording: relevant

QuARS results on EN50128:2011

- Clause 6.7.4.9 (support tools)

The line number:
500. 6.7.4.9 where automatic code generation or similar automatic translation takes place, the suitability of the automatic translator for safety-related software development shall be evaluated at the point in the development lifecycle where development support tools are selected.
is defective because it contains the wording: similar

- Which other translators? From what to what?

NATIONAL
RESEARCH
COUNCIL

QuARS results on EN50128:2011

- Clause **6.6.4.1 (change management)**

The line number:
392. a) the documentation needed for problem reporting and/or corrective actions, with the aim of giving feedback to the responsible management;
is defective because it contains the wording: and/or

QuARS results on EN50128:2011

- Clause 7.3.4.31 **(software integration test specifiction)**

The line number:
169. b) it shall be shown that the software behaves in an appropriate manner when the interface is subjected to inputs which are out of specification;
contains a unspecified sentence because the term: interface

- Here, possible uncertainty about “interface” is resolved in the preceding sub-clause. In all the other clauses, the term “interface” is specified by an adjective or a complement

Evolution of related research work (at ISTI-CNR and in general) after QuARS

- Broader field of Requirement Engineering, also including railway domain
- Extension of Ambiguity dimensions beyond traditional quality characteristics
- To resolve false positive issues and perform impact analysis, recent research provides methods and tools that address domain specific expressions

Domain-dependent Ambiguities - 1

- Compute the ambiguity potential of typical Computer Science words when they are used in different domains.
- Domain-specific ambiguity can be:
  - **Lexical:** e.g., *windows* – operating system or glass openings of a vehicle?
  - **Pragmatic:** e.g., *machine*: a software system or a specific medical system for diagnostic support?
  - **Generality:** e.g., *interface*: software or hardware interface?
- 5 Domains: Electronic Engineering, Mechanical Engineering, Medicine, Literature, and Sports.

Domain-dependent Ambiguities - 2

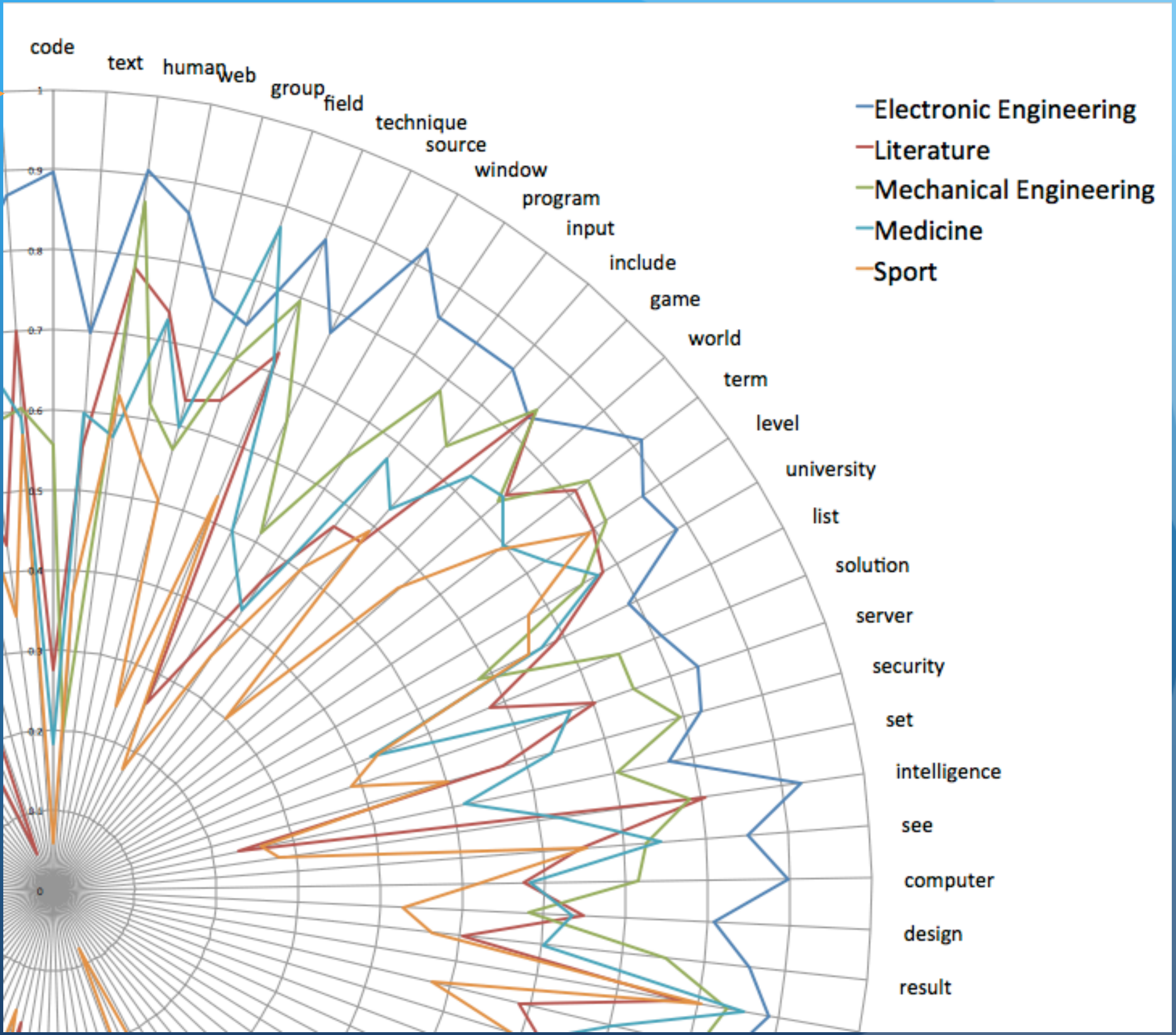
- Approach based on Wikipedia Crawling and Word Embeddings
- Results:
  - Some terms are ambiguous in all the domains (*code*, *database*, *support*)
  - Some terms are ambiguous in some domains (*part*, *interface*, *machine*)
  - Some terms are never ambiguous (*user, level*, *change*, *information*)
- Reference: Alessio Ferrari, Beatrice Donati, Stefania Gnesi: Detecting Domain-Specific Ambiguities: An NLP Approach Based on Wikipedia Crawling and Word Embeddings. RE

Comparing the meaning of terms across domains

Meaning in Computer Science domain (outer border)

vs.

Meaning in other domains (colors)



Application to Railway Requirements -1

- 1866 Industrial Requirements (from ATP, CTC, Axle Counter)
- Rule-based approach implemented by railway domain expert, to adapt the approach to the language of the company, based on the GATE tool for NLP
- Check for ambiguity, vagueness, incompleteness, missing references, passive voice, etc.
- Results: 85.39% recall, 83.16% precision

Application to Railway Requirements -2

- Domain-specific adaptation is crucial to improve the performance of rule-based tools:
  - Example: terms such as *light* and *sound* are not vague, when they are used as nouns instead of adjectives
  - Example: the company may systematically use expressions such as *It shall be possible*
- If a tool is developed internally, these systematic false positive cases can be discovered and discarded
- Reference: Benedetta Rosadini, Alessio Ferrari, Gloria Gori, Alessandro Fantechi, Stefania Gnesi, Iacopo Trotta, Stefano Bacherini: Using NLP to Detect Requirements Defects: An Industrial Experience in the Railway Domain. REFSQ 2017: 344-360

Application to Standards

- No further research work happened for Standards, but new technology may be adopted, to cope with Standards peculiarity
- Traditional (QuARS) and new tools might be adopted in Standard developing WGs

Annex – ambiguity: examples, anecdotes

- Ambiguous requirements
    - ISO26262: “Requirements verification” double meaning (probably contextually solvable, glossary definition vs. usage):
        - To verify requirements against some rules (testability, clarity, traceability, …)
        - To verify software behaviour against requirements (test, code inspection, ...)
- SW development for <US City> tram : meaning of “parameters” for different developing teams
    - Compiled function parameters vs. PTU provided configuration data

Conclusions

- Problems related to “if, why and how” a Software Safety-related Standard “works” were faced in technical literature and discussed in a first thematic Workshop and Report.
- None of the position seems to prevail, but the questions analysed showed the opportunity for specific, planned studies and surveys.
- Here we focus on ambiguity detection and removal
  - Impact analysis research should also be performed: can we exclude that ambiguity in Standard text has any impact in safety?

Other References

- **Fenton, N. E.; and Neil, M.: A Strategy for Improving Safety Related Software Engineering Standards. Transactions on Software Engineering, vol. 24, no. 11, 1998, pp. 1002-1013.**

- **Proceedings of "Planning the Unplanned Experiment: Assessing the Efficacy of Standards for Safety Critical Software (AESSCS), @** EDCC '14, Newcastle upon Tyne, May 2014.

- **Patrick J. Graydon and C. Michael Holloway "Planning the Unplanned Experiment: Assessing the Efficacy of Standards for Safety-Critical Software", NASA/TM-2015 218804.**