26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

"Functional Analysis for Virtual IP Multimedia Subsystem"

Author: Arevik Gevorgyan
PhD candidate at
Ecole Polytechnique (Palaiseau, FR),
Nokia (Paris area, FR)

Summary

1. Network Functions Virtualization (NFV):

Context, Problems

2. Motivation of Systems Engineering (SE) Approach:

Why an extended Architectural Framework?

3. Functional Analysis Procedure for Virtual IP Multimedia Subsystem:

3.1 Operational Analysis Outcomes
3.2 Functional Analysis Outcomes
3.2.1 Trade-Off for Optimal Architectural Choices

4. Discussion:

Conclusion, Perspectives, Questions

1. Network Functions Virtualization (NFV): *Context Analysis (1)*

NFV is the greatest Telecom world transformation:
"Enabling network access to a scalable and elastic pool of sharable physical or virtual resources with self-service provisioning and administration on-demand" [ISO-IEC_17788].

July

Contextual Complextiy Axes:

- System *(Network Elements Specificities, Interoperability issues, etc)*,
- Organizational *(Traditional Silo decompositions, Distributed geography, etc)*,
- Direct Market *(Heterogeneous Technologies, Different Maturity Levels, etc)*,
- Overall evolving Environment (Diverse stakeholders and Eco-systems)

1. Network Functions Virtualization (NFV):
Problem Definition (2)

The Problem:

Modeling the Transformation from Traditional (monolithic) to Virtual Network Architecture

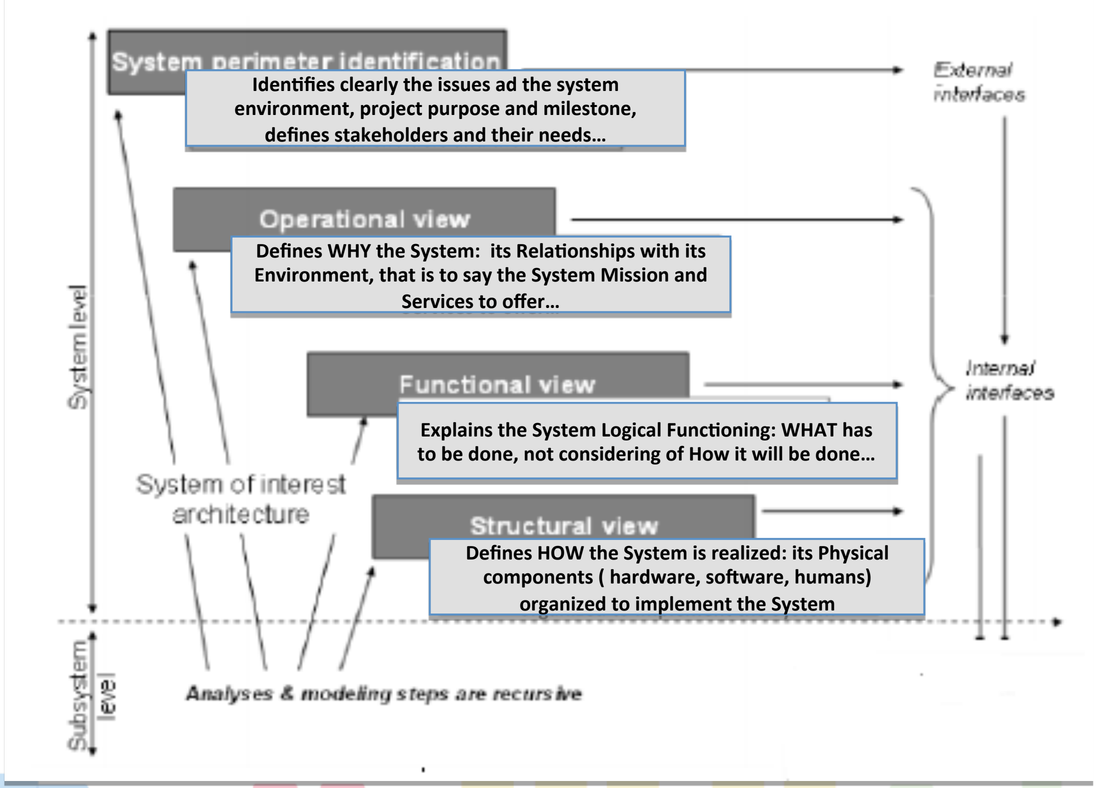
The Problem Question:

July

Defining the **Dynamics of Functional Organization**, while satisfying the standards driven functional and performance constraints and diverse stakeholders diverse conflicting objectives.

2. Motivation of Systems Engineering Approach:

Why an extended Architectural Framework?



Ref.: D. Krob "Eléments d'architecture des systèmes complexes, in "Gestion de la complexité et de l'information dans les
grands systèmes critiques", 179-207, CNRS Editions, 2009
Ref.: D. Krob, Enterprise Architecture, Modules 1-10, Ecole Polytechnique, 2009-2010 (personal communication)

3. Why **Functional Analysis Procedure for Virtual IP Multimedia Subsystem**

Objective: Defining the (Optimal) Virtual Architecture adapted to the Context and Equilibrium from Infrustructural Standpoint

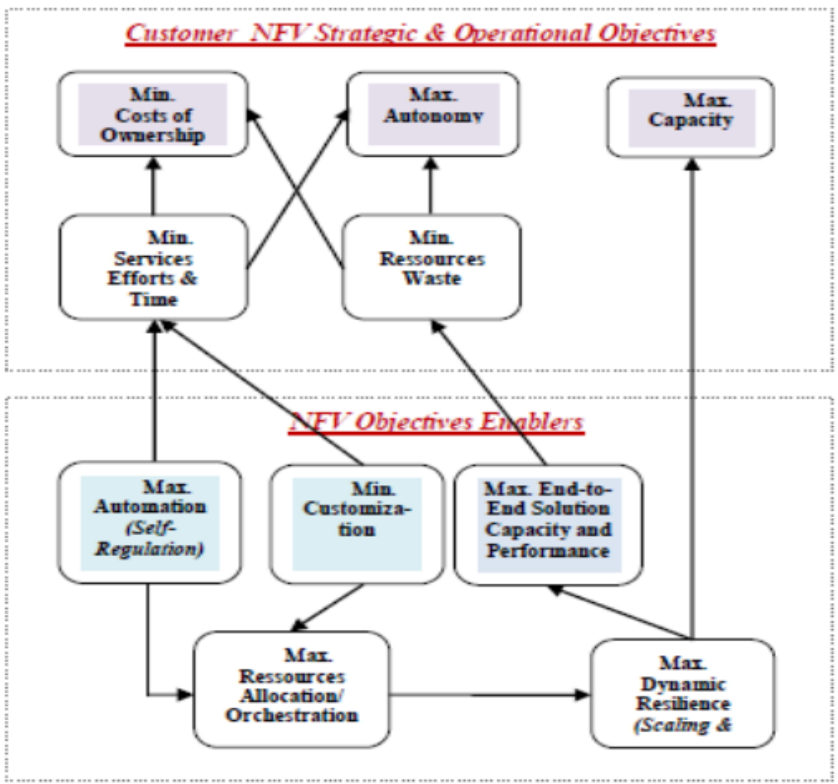
July

Analysis Outcomes:

- Operational Invariants Defined
- Functional Invariants Defined
- Optimal Functional Options Defined

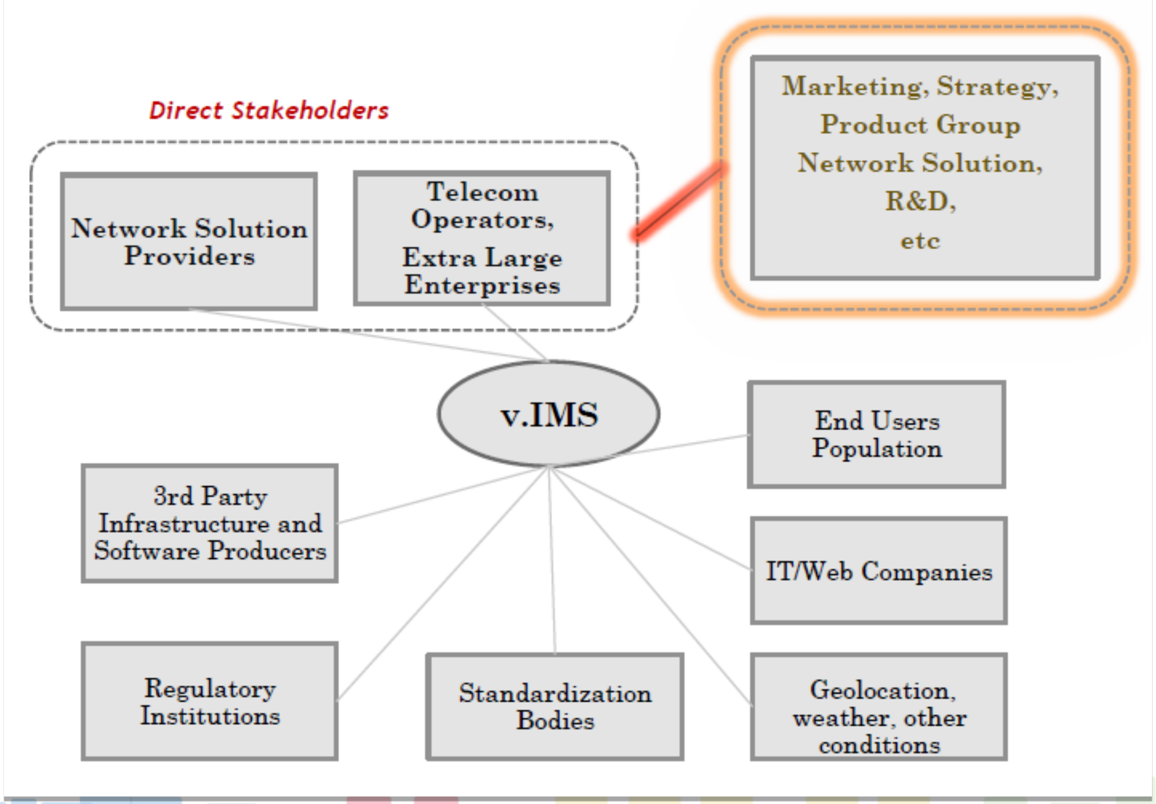
3. Trade-Off Baseline For Functional Architecture Definition

Aligning the Stakeholders Strategic, Operational Objectives with NFV Technical Enablers



3. Virtual IMS Environment Analysis: Defining the Stakeholders

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016



Incorporating with **PESTEL** Analysis Frame

Ref. A. Gevorgyan, P. Spencer "Operational Analysis of Virtual IP Multimedia Subsystem (IMS) Through a Model-Based Architectural Framework" *CSDM 2015 Proceedings*

Ref.: A. Gevorgyan, D. Krob, P. Spencer "Holistic Integrated Decisions Trade Off Baseline for an Optimal Virtual IMS Architecture." *CSDM Asia 2016 Proceedings*

3. Definition & Refinement of Stakeholders Needs

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

Table1: Macro Needs

N1	Operators want a system that will support significantly higher traffic loads
N2	Operators want a robust system
N3	Operators want assurance for the maintenance and support
N4	Operators want capabilities to easily deploy/support new applications/services
N6	Operators want significant savings of CAPEX and/or OPEX
N7	Operators want operational easiness: i.e. to drastically reduce time to market
...	....

Table2: Macro Needs Refinement

N2.1	Operators want an automatic adjustment of resources allocation for traffic growth and de-growth
N2.2	Operators want maximal availability and speed for huge traffic rates
N2.3	Operators do not want to feel the physical limitations of the system
N2.4	Operators want predictable behavior of network functions
...	

Incorporating with **FURPSE (Software Characteristics)** Analysis Frame

Ref. A. Gevorgyan, P. Spencer "Operational Analysis of Virtual IP Multimedia Subsystem (IMS) Through a Model-Based Architectural Framework" *CSDM 2015 Proceedings*

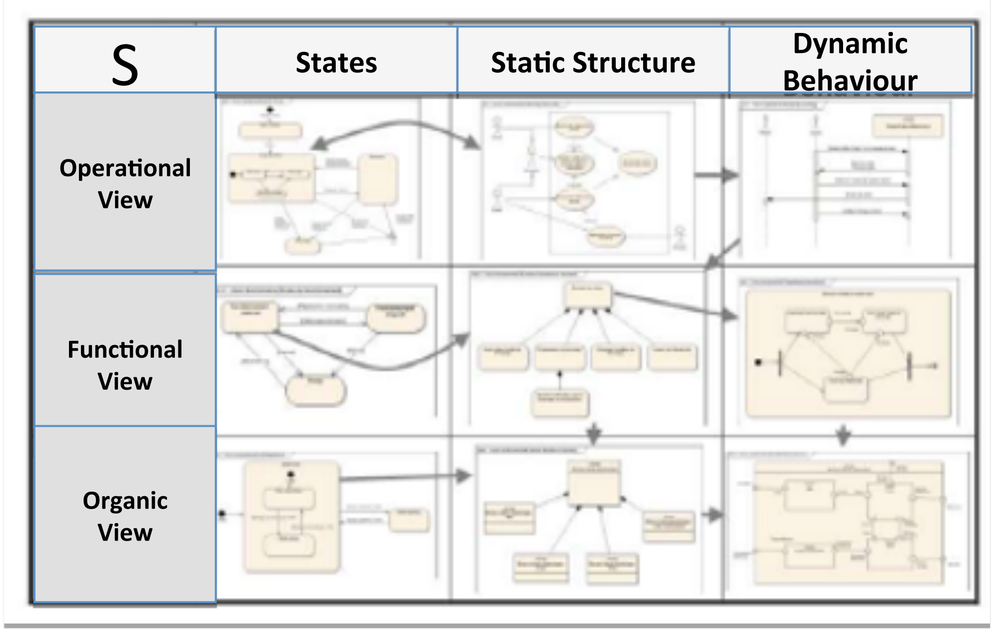
Ref.: A. Gevorgyan, D. Krob, P. Spencer "Holistic Integrated Decisions Trade Off Baseline for an Optimal Virtual IMS Architecture." *CSDM Asia 2016 Proceedings*

3. Derived Operational Analysis Invariants

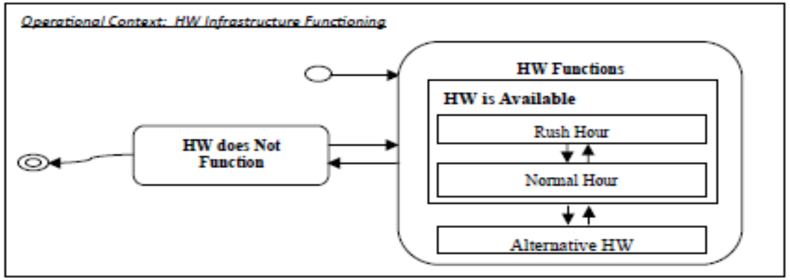
26th annual INCOSE international symposium
Edinburgh, UK
18 - 21, 2016

Operational Analysis Outcomes:
The Derived Invariants

- System Mission
- Contexts,
- Uses Cases,
- Scenarios,
- States,
- External Interfaces
- Internal Interfaces



3. Brief Example of Operational Invariants



Operational Scenarios

- The system lacks capacity (memory or CPU)
- Hence new virtual machines are initiated (automatically)
- The system then transmits the needed amount of traffic with the new VMs

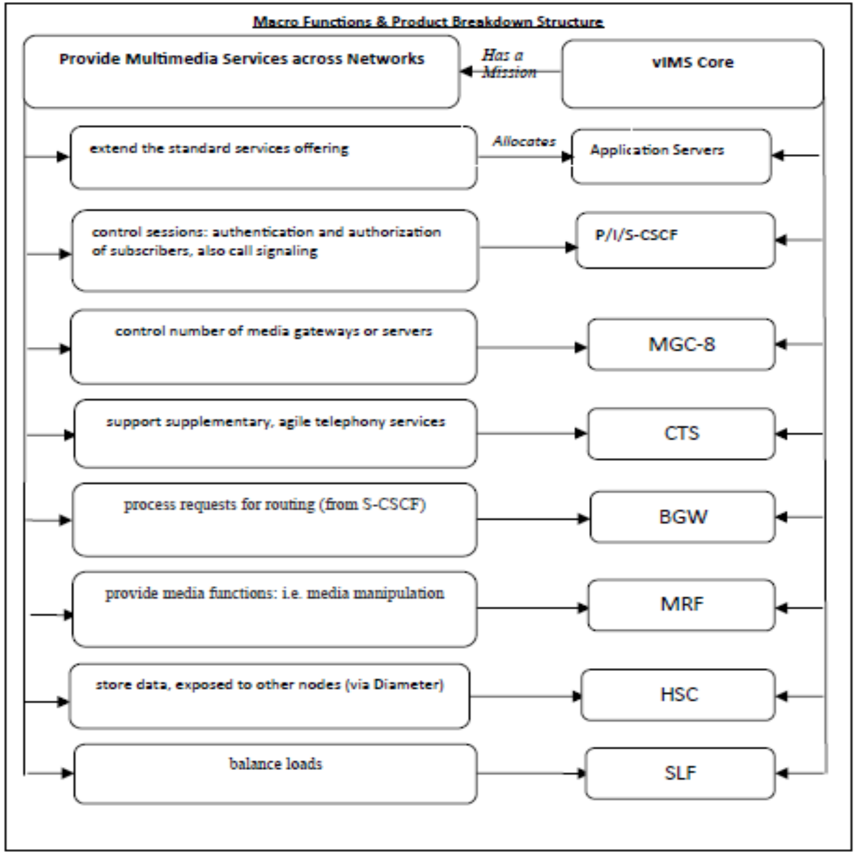
<table>
<tr><th colspan="3">States of vIMS</th><th colspan="3">States of Hardware Infrastructure</th></tr>
<tr><td rowspan="5">OPERATE
Phase</td><td colspan="2">vIMS does Not Function</td><td colspan="3"></td></tr>
<tr><td colspan="2" rowspan="3">vIMS is Deployed</td><td rowspan="2">HW Functions</td><td>HW is Available</td><td>Normal Hour</td></tr>
<tr><td>Alternative HW</td><td>Rush Hour</td></tr>
<tr><td>HW does Not Function</td><td colspan="2"></td></tr>
<tr><td>vIMS Functions</td><td>New Virtual Machines Instantiated</td><td></td><td colspan="2"></td></tr>
</table>

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

3. Functional Analysis

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

Functions definition and allocatation to Physical Components/ Products



3. Defining Inputs for Optimization Models

26th annual INCOSE international symposium
Edinburgh, UK
July 18 - 21, 2016

Parameters and Values Definition

N	General Traffic Model Parameters	Measure Units	Values
TM1	Min. number of CPU Cores	Unit	
TM2	Min. number of Core Network Elements	Unit	
TM3	Min. Dynamic Memory	GB	
TM4	Min. Direct Attached Storage	GB	
TM5	Min. Block Storage (Cinder)	GB	
TM6	Min. Object Storage	GB	
TM7	Min. Estimated Performance	Msgs/Sec.	
TM8	Max. Response/Ressources Allocation Latency	Sec.	< 2
TM9	Max. Failure Latency	Sec.	
TM10	Max. Failures/Incidents per Year	Percentage	
	Min. Availability	Percentage	

3. Choice of Optimization Method/s to extend the Framework

We opt for:

I. **Multi-Objective Optimization** method to constitute the optimal functional options per market, technilogical and any other criteria

II. **Game Theory**: to define architectural equilibrium for most efficient infrastructural and other ressources usage

4. **Discussion:** Conclusion, Perspectives

NFV Paradigm Redefines:

- The ways of thinking
- The engineering procedures required to build and manage networks.

3. Optimization Procedure

Multi-Objectives Optimization problems request clear definition and distinction between the measures of Objectives:

- *Measures of Effectiveness (MOE)* estimate achievement of the system's mission or operational objectives (under a specified set of conditions) within the corresponding operational environment. They are derived during the Operational Analysis phase [INCOSE 2010].

- *Measures of Performance (MOPs)* should be derived from or provide insight upon MOEs or User needs [INCOSE 2010]. MOPs are the key performance characteristics the system must have. They assess whether the system satisfies design or performance (i.e. technical, economic, etc) requirements to meet MOEs.

Applying INCOSE Rules for writing high-quality requirements in Industry

Anabel Fraga
anabel.fraga@uc3m.es

Wednesday, July 20, 2016

KNOWLEDGE REUSE

Short Bio

- PhD in Computer Science on Universal Knowledge Reuse in 2010
- E-Commerce and Networking Master 2004
- Computer Science Engineer 1999
- Professor at UC3M (Carlos III of Madrid University in Madrid)
- Knowledge Manager expert in The Reuse Company

Researchgate and LinkedIn profiles for more information

Abstract

- Reviews activities are **time consuming** and their performances are not totally satisfying.
- Rely on the **expertise** of the Systems Engineers completely.
- More **advanced practices should combine Requirements Engineering and Knowledge Management** (NLP – Patterns – Ontologies)
- **Identify in a large set of requirements gaps** such as **inconsistencies** or superfluous **similarities**.
- Typical examples are:
  - inconsistent use of measurement units in requirements,
  - lack of domain vocabulary,
  - multiple requirements that mean the same requirement at the end.
- We point out during the lecture **additional instances and focus on the several quality assessment functions** that can be applied.
- In particular, **Guide for Writing Requirements (International Council on Systems Engineering)** is a light in the process.

Agenda

- Problems found in the Requirement Engineering practice
  - Impact of low quality requirements
- Previous and current efforts (CRYSTAL project)
- Tools supporting the ideas and the research developed so far
  - The Requirements Quality Suite
  - Metrics
  - Rules implemented
- Some results

Problems found in the Requirement Engineering practice

- Chaos report and some other surveys (e.g. PMI: Pulse of the profession study):
  - [“40%-70% of defects in the projects are related to requirements.”]

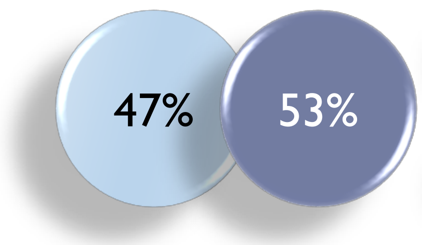
Source:
Chaos Report 2004

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

- Capers Jones:
  - [“The average rework is over 40%-50%.”]

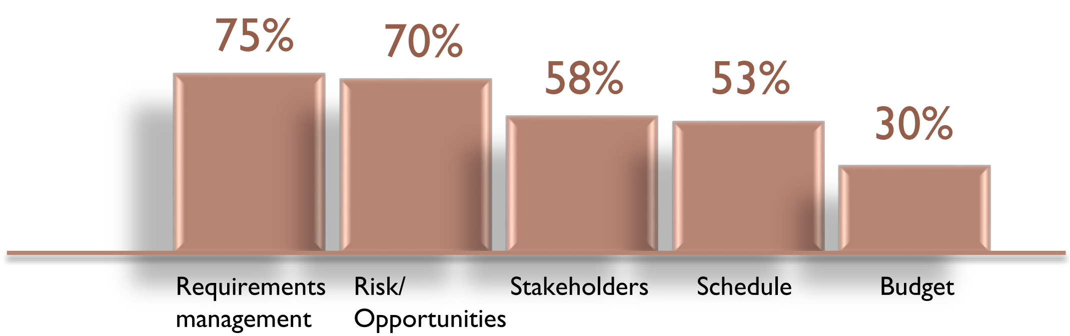
Poor requirements = Poor performance

- "Poor requirements management is a major cause of project failure, second only to changing organization priorities"



Unsuccessful projects where poor requirements management is the primary cause of not meeting goals

Other reasons



Poor communication is primary cause of failure in these areas; and requirements are the reason for this poor communication

Source: PMI 2014. Pulse of the Profession study

Poor Requirements = Poor Performance

The impact of low quality requirements

- In some cases:

[“A good requirements specification may lead to a bad project outcome”]

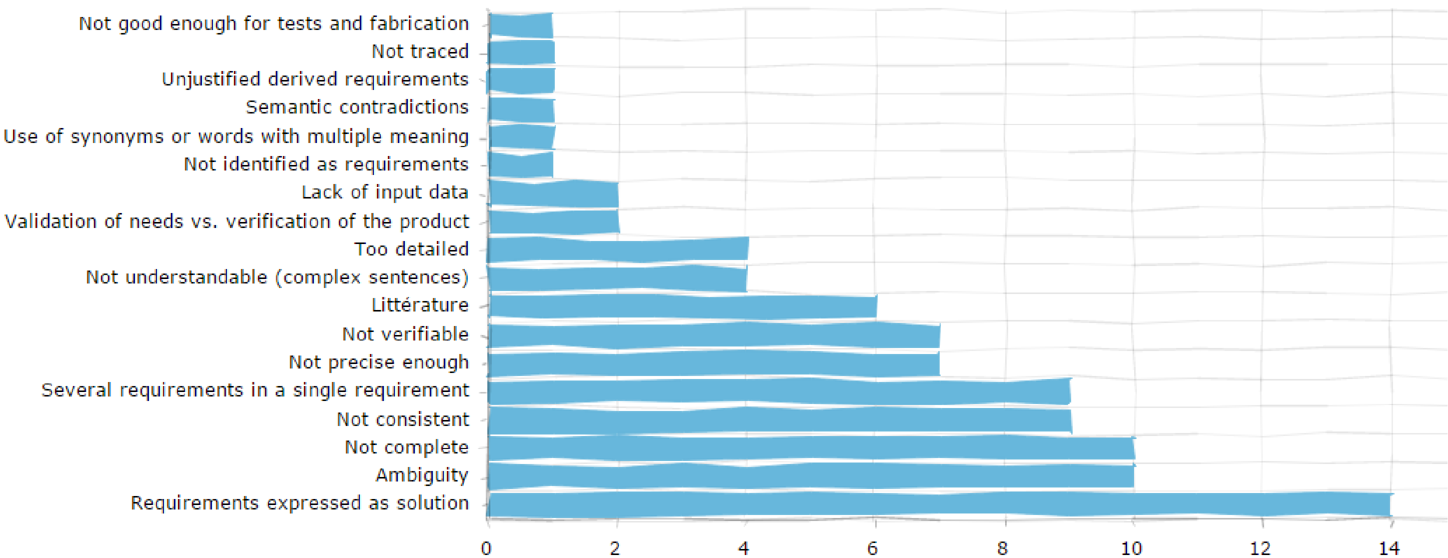
- But clearly:

[“A bad requirements specification will always lead to a bad project outcome”]

The impact of low quality requirements

Previous and current effort

- 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%
  - …



Source:
AFIS & Gauthier Fanmuy
RAMP Project

Previous and current effort (CRYSTAL project)

- **CRYSTAL** - CRITICAL SYSTEM ENGINEERING ACCELERATION (ARTEMIS JU project – European)
- This standard will allow loosely coupled tools to share and interlink their data based on standardized and open Web technologies that enables common **interoperability** among various life cycle domains. This reduces the complexity of the entire integration process significantly. Compared to many other research projects.
- CRYSTAL is strongly industry-oriented and will provide ready-to-use integrated tool chains having a mature technology-readiness-level (up to TRL 7).
- In order to reach this goal, CRYSTAL is driven by **real-world industrial use cases** from the automotive, aerospace, rail and health sector and builds on the results of successful predecessor projects like CEASAR, SAFE, iFEST, MBAT on European and national level.
- Creating and establishing a **new standard** on a large scale in an already consolidated market cannot be achieved by individual organizations. With a budget of more than 82 million Euro and 71 partners from 10 different European countries, CRYSTAL has the critical mass to accomplish this endeavor. The project consortium is made up of participants from all relevant stakeholders, including OEMs, suppliers, tool vendors and academia.

Requirements quality characteristics vs quality metrics

- Can we translate between the set of quality characteristics and a set of quality metrics more *easy to measure*?
- Some approaches to define a set of *easy-to-measure* requirements quality metrics:
  - ARM (Automated Requirement Measurement Tool) by NASA

INDICATORS OF QUALITY ATTRIBUTES

Categories of Quality Indicators	Quality Attributes										
	1. Complete	2. Consistent	3. Correct	4. Modifiable	5. Ranked	6. Testable	7. Traceable	8. Unambiguous	9. Understandable	10. Validatable	11. Verifiable
1. Imperatives	X			X			X	X	X	X	X
2. Continuances	X			X	X	X	X	X	X	X	X
3. Directives	X		X			X		X	X	X	X
4. Options	X					X		X	X	X	
5. Weak Phrases	X		X			X		X	X	X	X
6. Size	X					X		X	X	X	X
7. Text Structure	X	X		X	X		X		X		X
8. Spec. Depth	X	X		X			X		X		X
9. Readability				X		X	X	X	X	X	X

Requirements quality characteristics vs quality metrics

- Well-known 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
  - Achievable
  - Relevant
  - Traceable

"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 vs quality metrics

- More approaches:
  - **INCOSE Guide for Writing Requirements**
  - Describes a set of quality characteristics (based on ISO/IEC 29148):
    - Necessary
    - Implementation independent
    - Unambiguous
    - Complete
    - Singular
    - Conforming
    - Feasible
    - Verifiable
    - Correct
  - But also describes a number of more precise rules
  - And the matching among characteristics and easy-to-measure rules

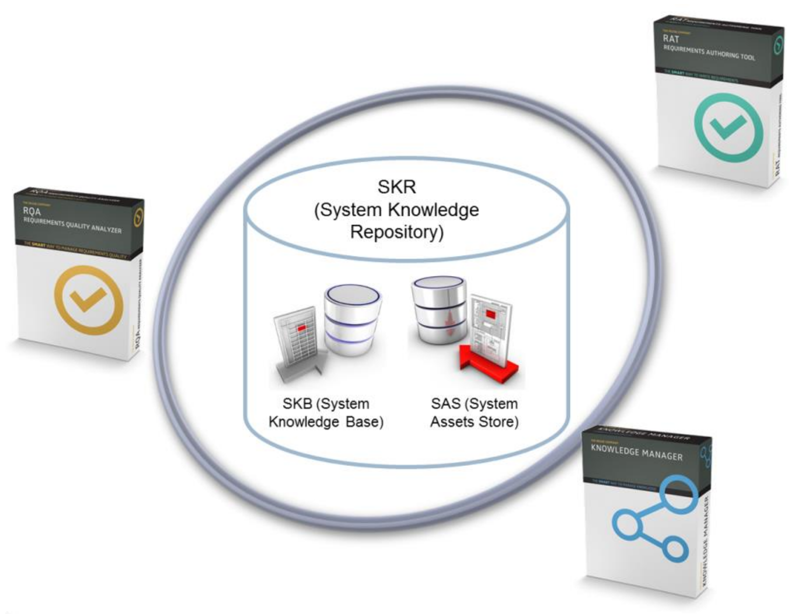
Characteristic Cxx – Characteristic name
Rationale: xxxx
Strategy: xxxx
Rules that help establish this characteristic:
Rxx - /Section/Rule name Avoid xxxx Ryy - /Section/Rule name Avoid yyy

Requirements quality characteristics vs quality metrics (Summary)

- Quality Characteristics vs Quality Metrics:
  - Quality Metrics are **easy to measure**
  - can be **used in formal requirements verification and validation**
  - **can be automated in a requirements management tool**
- Quality Characteristics and Quality Metrics, both at two levels:
  - Characteristics and Metrics for **individual** requirements
    - Correctness
  - Characteristics and Metrics for **sets** of requirements (specifications)
    - Consistency
    - Completeness

The Requirements Quality Suite

- The Requirements Quality Suite (RQS) intends to tackle requirements quality management by offering a set of tools and processes
- Automatic measurement of requirements quality metric
- Support to Requirements Authoring
- RQS models requirements quality metrics using the CCC approach (Correctness, Consistency and Completeness)



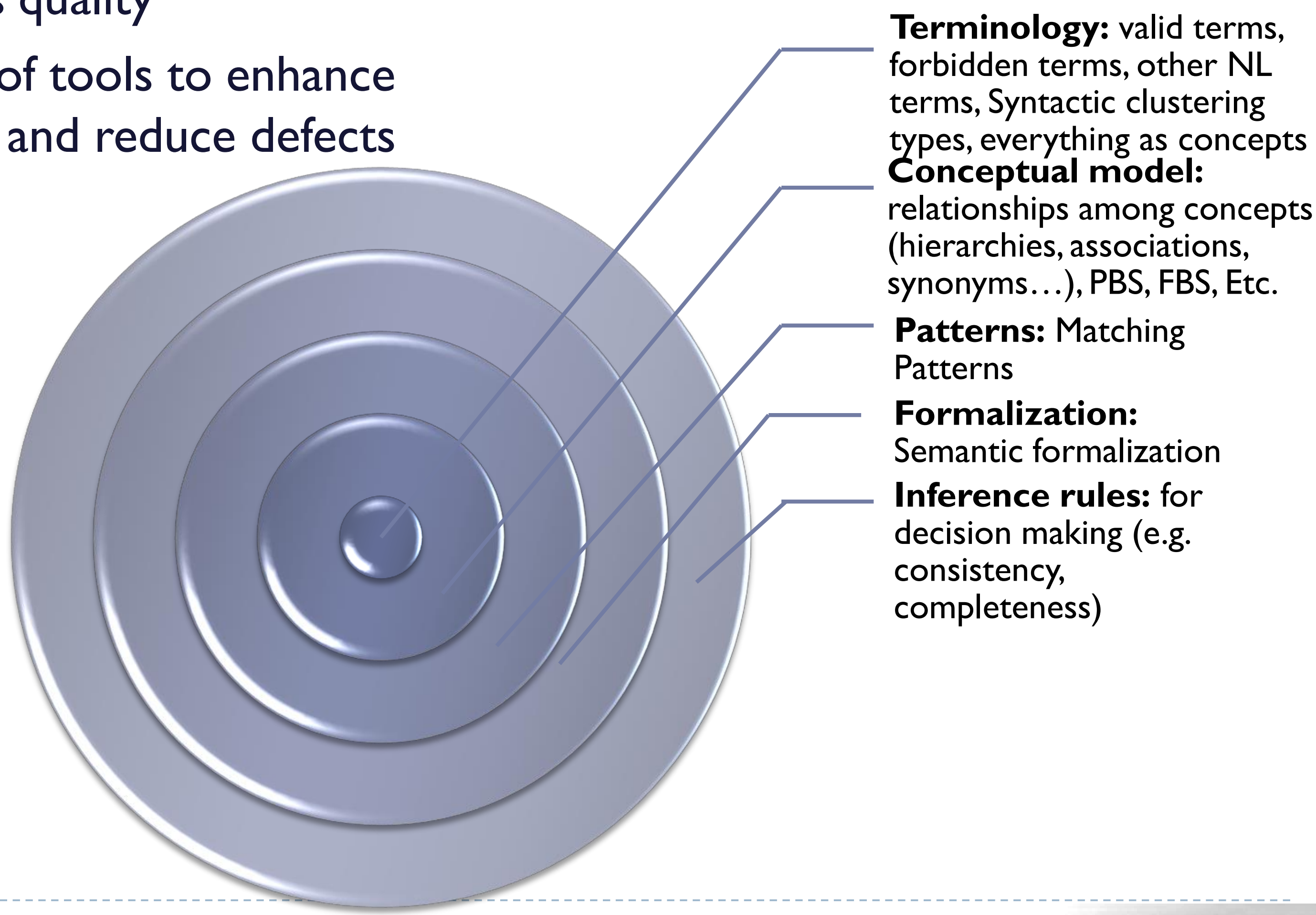
Requirements Quality Analyzer (RQA):
to setup, check and manage the quality of a requirements specification.

Requirement Authoring Tool (RAT):
to assist authors while they are creating or editing requirements.

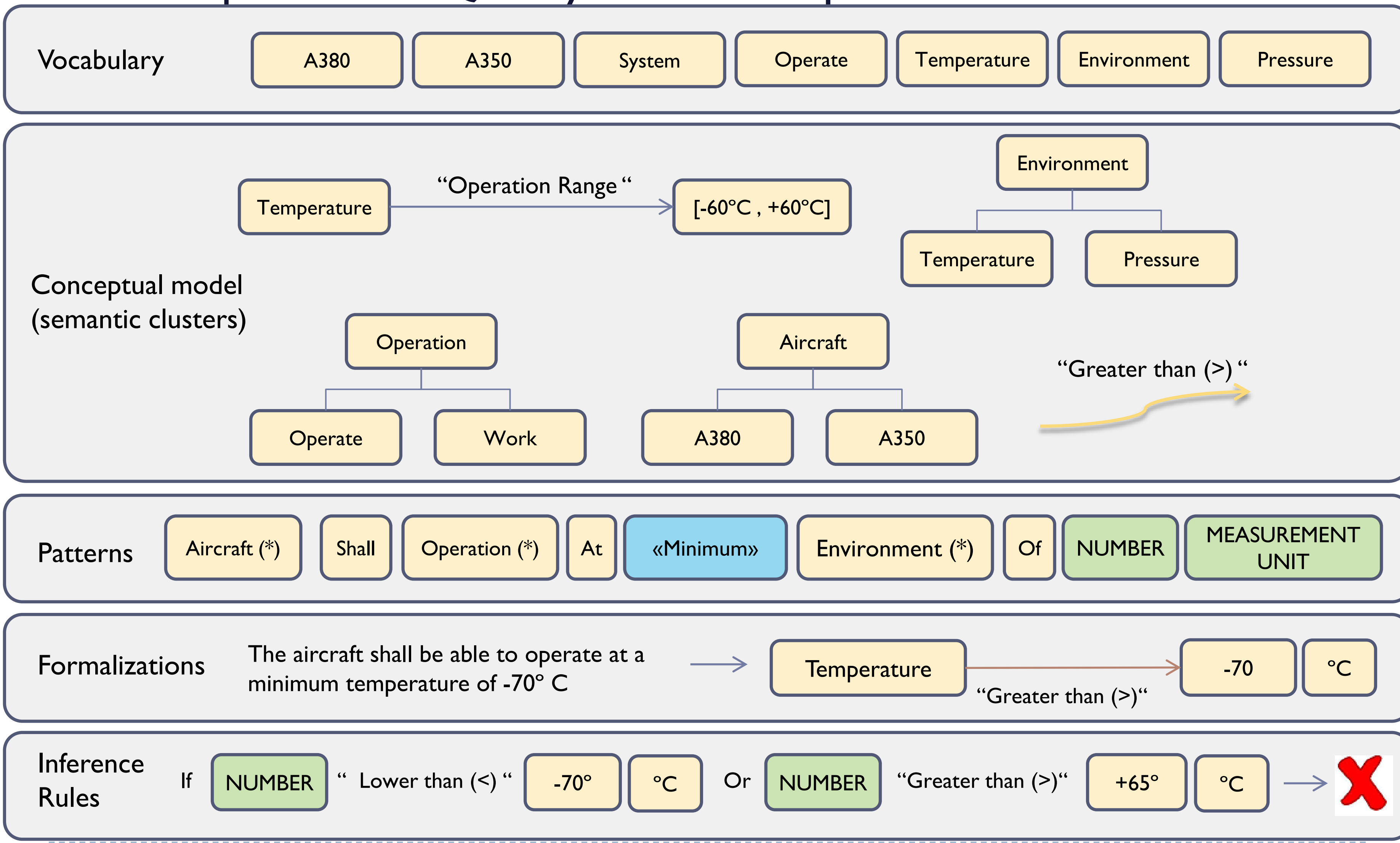
Knowledge Manager (KM):
to manage knowledge around a requirements specification: the ontology it is based on, the structure of the requirements to be used in the project, the communication between authors and domain architects.

The Requirements Quality Suite: Ontology view

- Ontologies as the driving element for requirements quality
- Allows a set of tools to enhance performance and reduce defects



The Requirements Quality Suite: Example



Controlled Vocabulary

- 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:
  - Business related Terms : those terms central to the business area to be treated
  - General Language Terms
  - Syntactically relevant phrases: Adverbs, Adjectives, etc.
  - Invalid terms: those terms that could be of no relevance.
- Idiomatic information

Doppler radar

Radar

Sonar

identify

Detect

A

OF

Syntactic Information

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

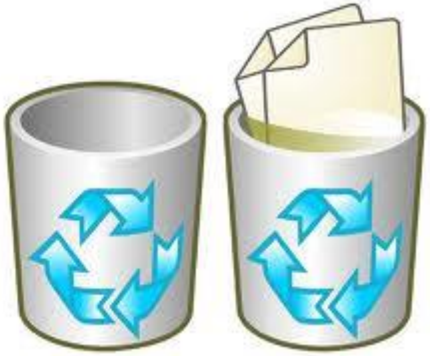
NOUNS	Doppler radar	Radar	Sonar
VERBS	Identify	Detect	
PREPOSITIONS	A	Of	

Controlled Vocabulary

UR044 : The Rad8 shall be able to identify hits at a minimum rate of 10 units per second

- Shall
- Identify
- Hit
- Minimum
- Unit
- …..
- Second

- The
- To
- At



- Rad8

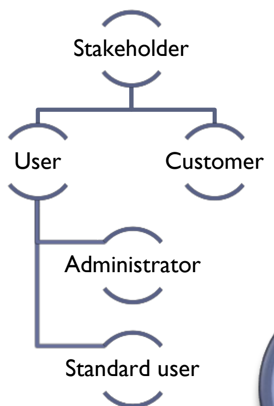
Conceptual Model: Semantic Search Engine

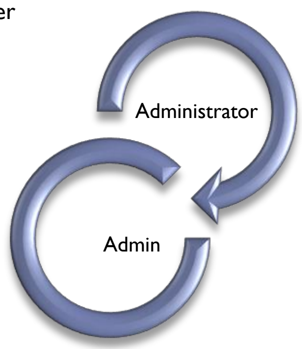
- A Thesaurus stores relational information regarding the terms in the vocabulary.

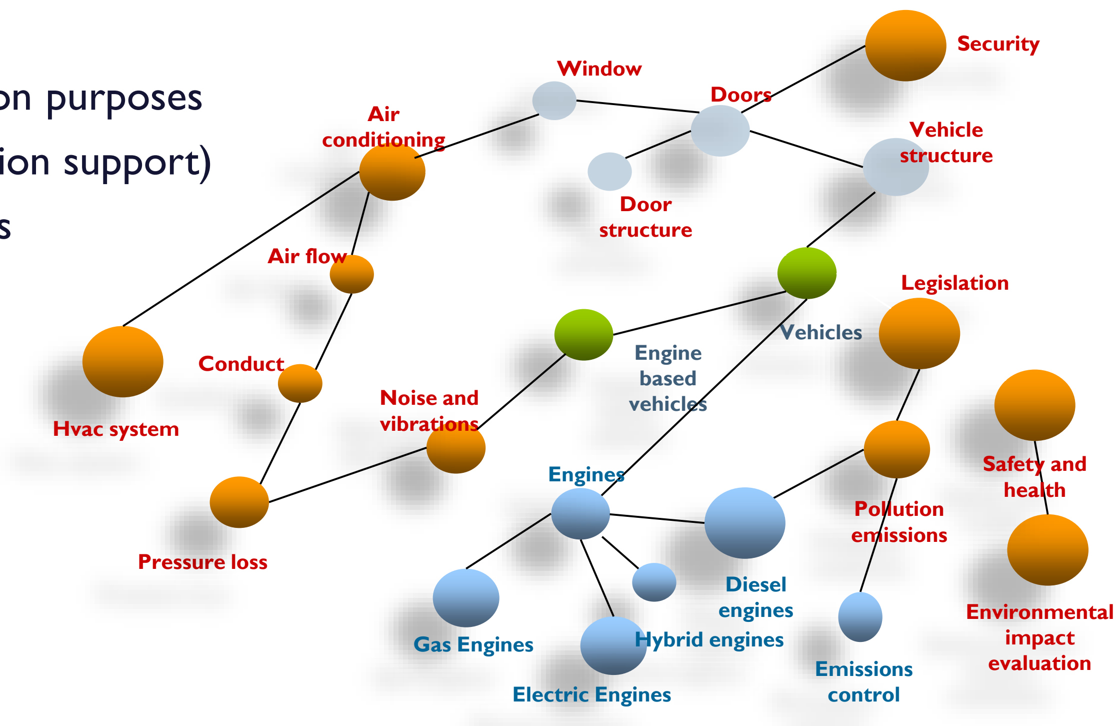
"thesaurus" is derived from (*thēsauros*), literally "treasure store", generally meaning a collection of things which are of big importance or value (Wiki)

Used For:

- Retrieval purposes
- Representation normalization purposes
- Suggestion purposes (Decision support)
- “solution specific” purposes





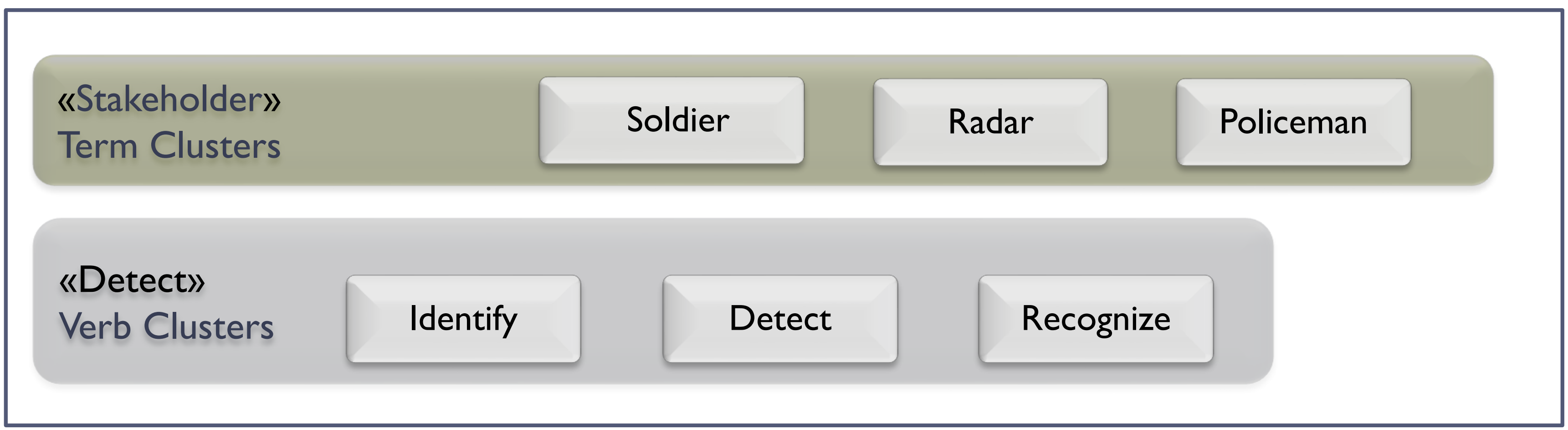


Semantic Clusters: grouping terminology in a smart way

- Semantic Clusters
  - For specific Pattern Restrictions

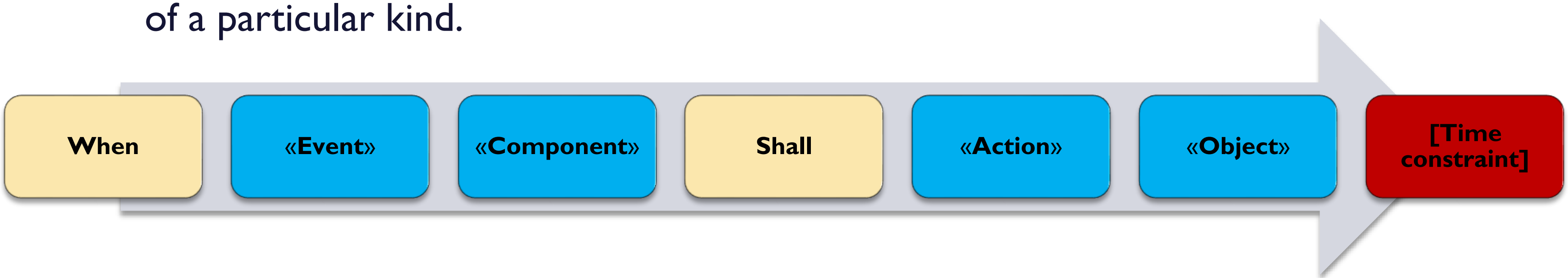
UR044 :The Radar shall be able to detect hits at a minimum rate of 10 units per second

UR563 :The policeman shall be able to Identify thieves after 8 pm



Requirement Pattern

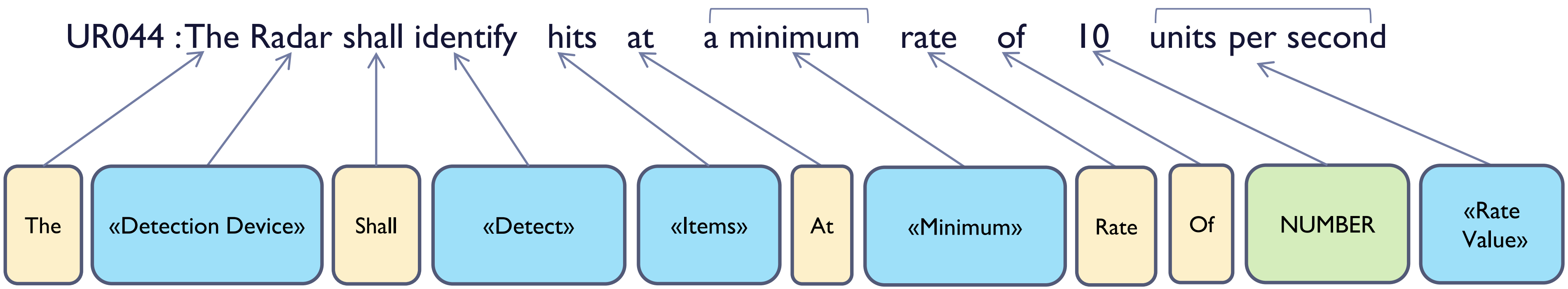
- Sequential restrictions structure with place-holders for the specific terms and values that constitute a particular knowledge statement, where the restrictions can be grammatical, semantic, or even both, as well as other patterns.
- A pattern encapsulates the rules for writing and validating a knowledge statement of a particular kind.



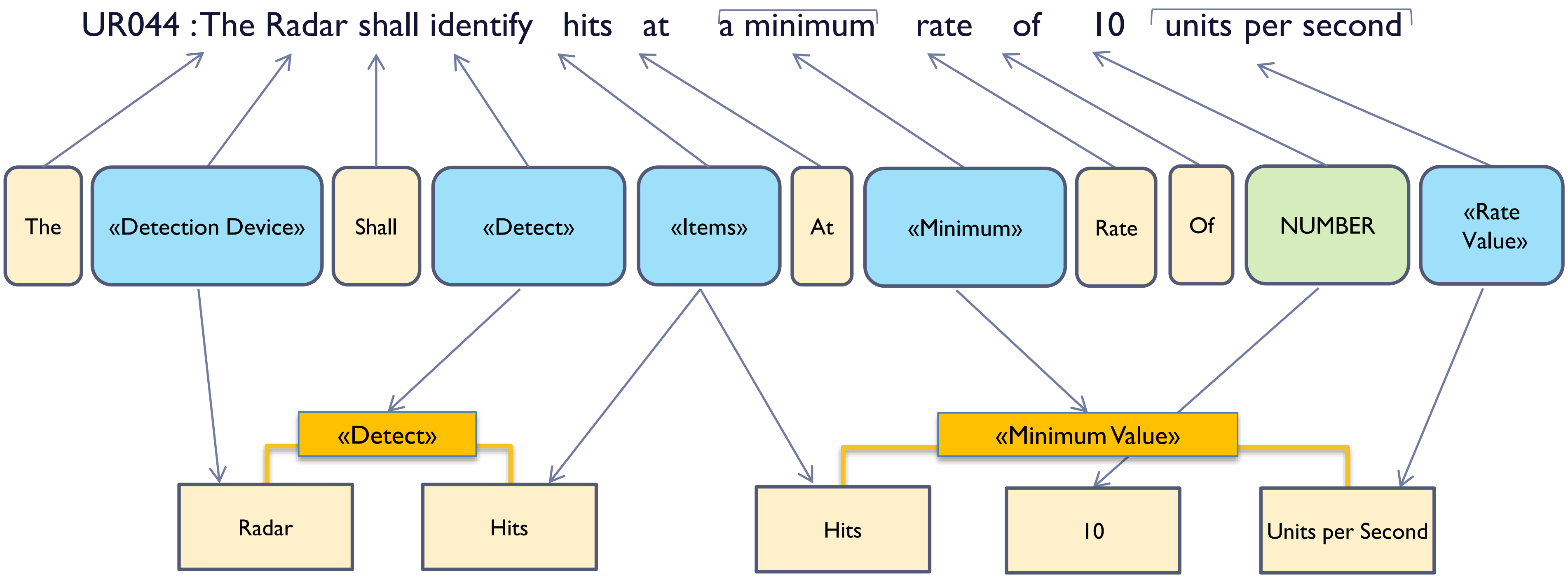
- Different types of restrictions: terms, syntactic tags, Clusters, sub-patterns
- Other options:
  - Combined restrictions: syntactic + semantic
  - Optional slots
  - Repetitive groups
  - Propagation according to the ontology

Patterns: Example

Detection Pattern 1



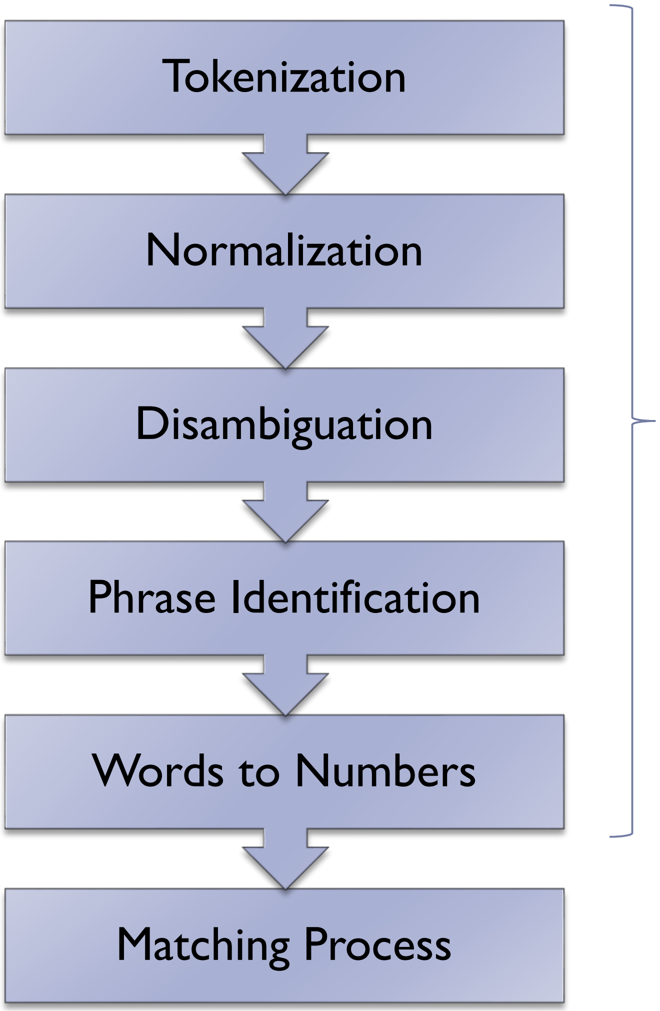
Requirement + Pattern + formalization



The Indexing process

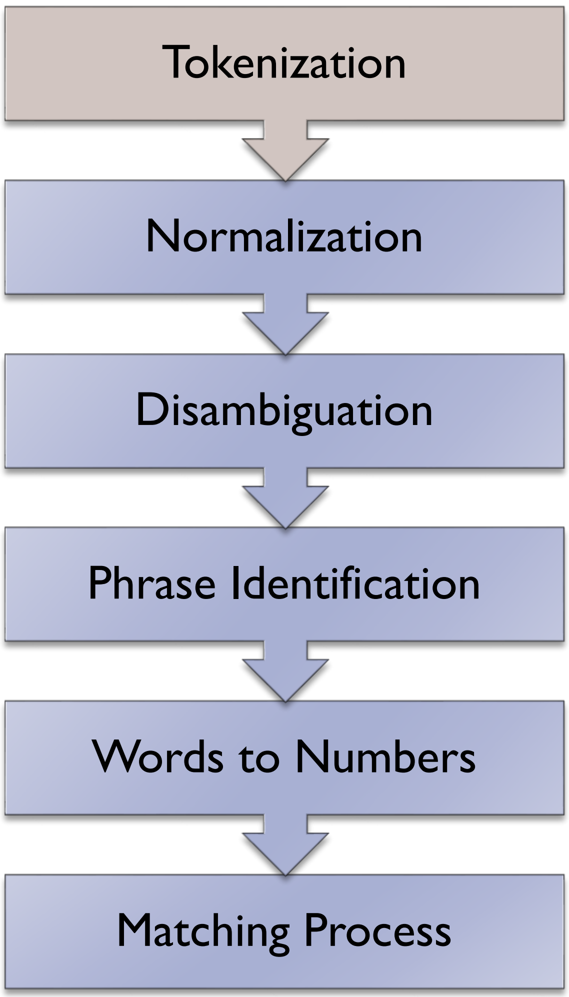
- In order to match a pattern from a natural language text, it is necessary to identify and extract the **TERMINOLOGY** accepted by the organization:
  - Identify **valid** terminology
    - Singulars, ranges, Measurement Units
  - Locate terminology in the **Conceptual Model**
  - Check for **restriction** matching
- Without terminology It is not possible to match patterns.
- **Terminology identification implies Natural language processing**

The Pattern Matching process: Stages



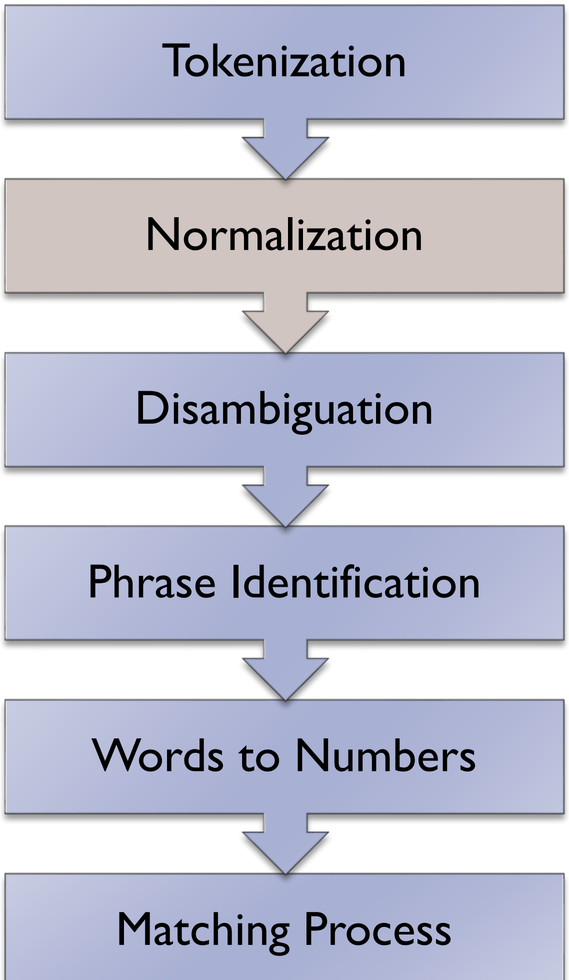
- Natural Language Processing (NLP)

Tokenization step



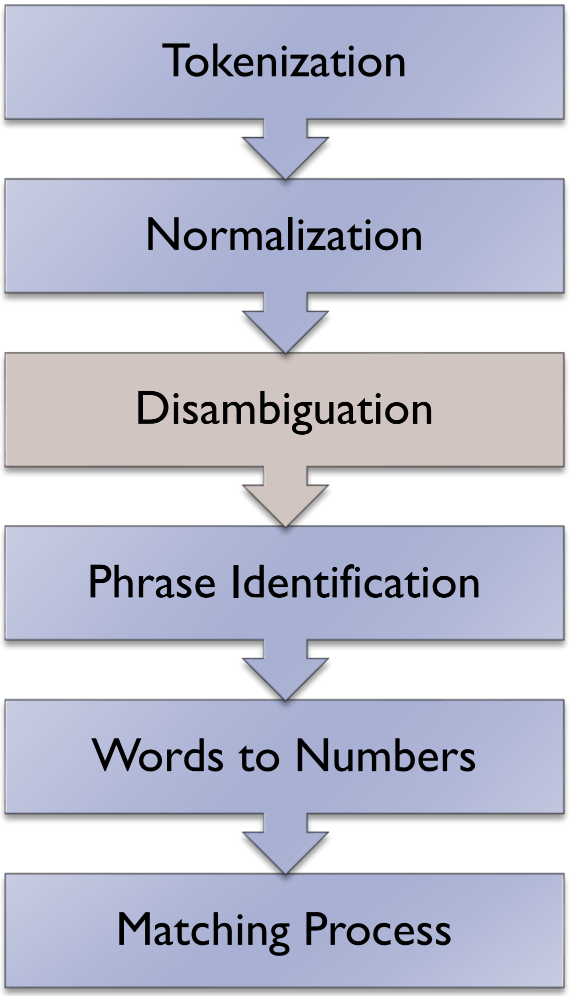
- Input: natural language text
- Output: list of simple terms
- Method: use of tokenization rules
- Example:
  - Input: The flight system should have three engines
  - Output:
    - The
    - Flight
    - System
    - Should
    - Have
    - Three
    - Engines

Normalization step



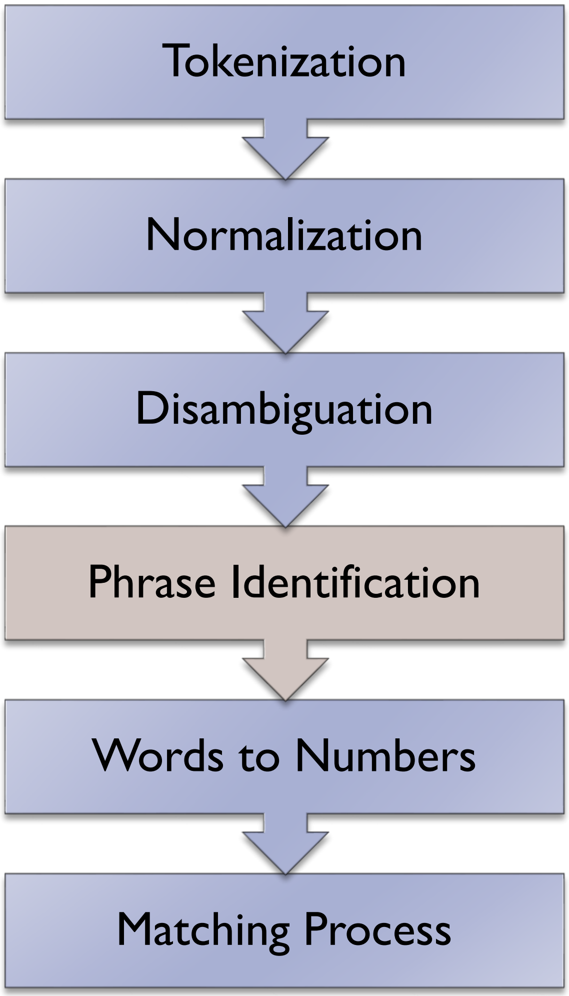
- Input: list of simple terms
- Output: list of simple terms with candidates term tags
- Method: use of normalization rules
- Example:
  - Input:{The, Flight, System, should, have, three, engines}
  - Output:
    - [The| DETERMINER],
    - [Flight| NOUN],
    - [System | NOUN, PROPER NOUN],
    - [should| VERB, MODAL VERB],
    - [have| VERB],
    - [Three| NUMBER],
    - [Engine| NOUN]

Disambiguation step



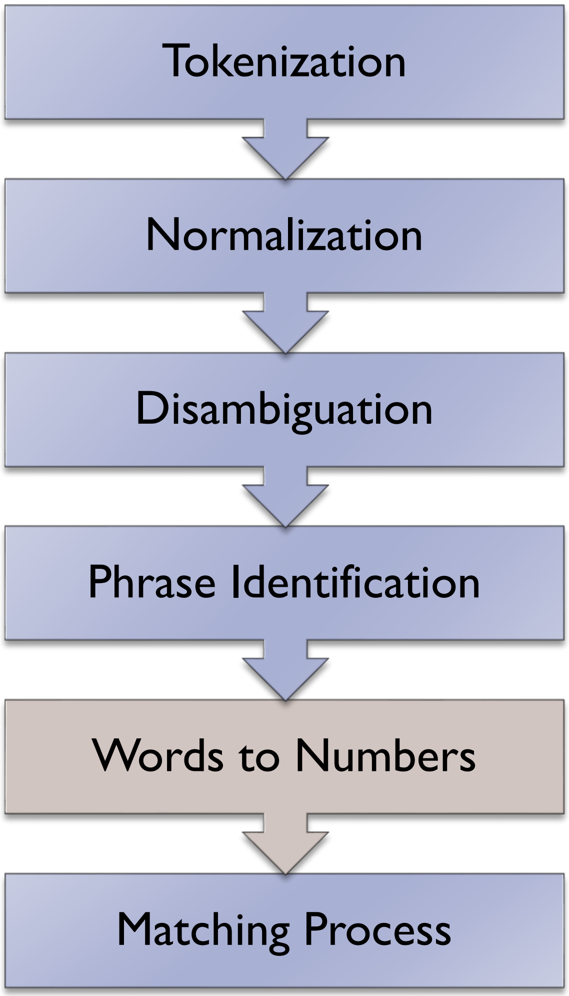
- Input: list of simple terms with candidates term tags
- Output: list of simple terms with only term tag chosen for each element
- Method: use of disambiguation rules, bigrams and tags probabilities
- Example:
  - Input:
    - [The| DETERMINER],
    - [Flight| NOUN],
    - [System | NOUN, PROPER NOUN],
    - [should| VERB, MODAL VERB],
    - [have| VERB],
    - [Three| NUMBER],
    - [Engine| NOUN].
  - Output:
    - [The| DETERMINER],
    - [Flight| NOUN],
    - [System | NOUN, **PROPER NOUN**],
    - [should| VERB, **MODAL VERB**],
    - [have| VERB],
    - [Three| NUMBER],
    - [Engine| NOUN]

Phrase Identification step



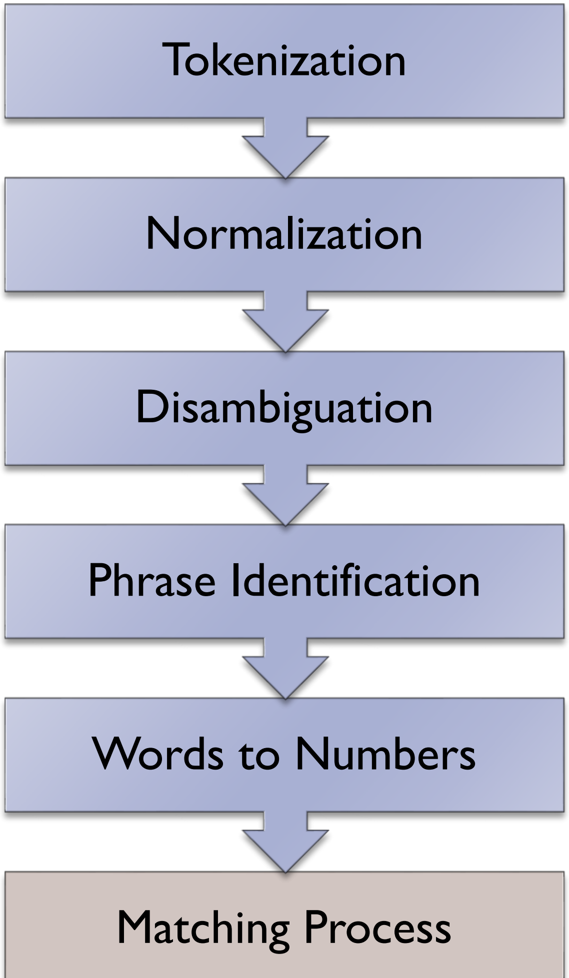
- Input: list of simple terms with only term tag chosen for each element
- Output: list of simple terms or compound terms with only term tag chosen for each element
- Example:
  - Input:
    - [The| DETERMINER],
    - [Flight| NOUN],
    - [System | NOUN, **PROPER NOUN**],
    - [should| VERB, **MODAL VERB**],
    - [have| VERB],
    - [Three| NUMBER],
    - [Engine| NOUN].
  - Output:
    - [The| DETERMINER],
    - [Flight System | NOUN]
      - [Flight| NOUN],
      - [System | NOUN, **PROPER NOUN**],
    - [should| VERB, **MODAL VERB**],
    - [have| VERB],
    - [Three| NUMBER],
    - [Engine| NOUN].

Words to Numbers step



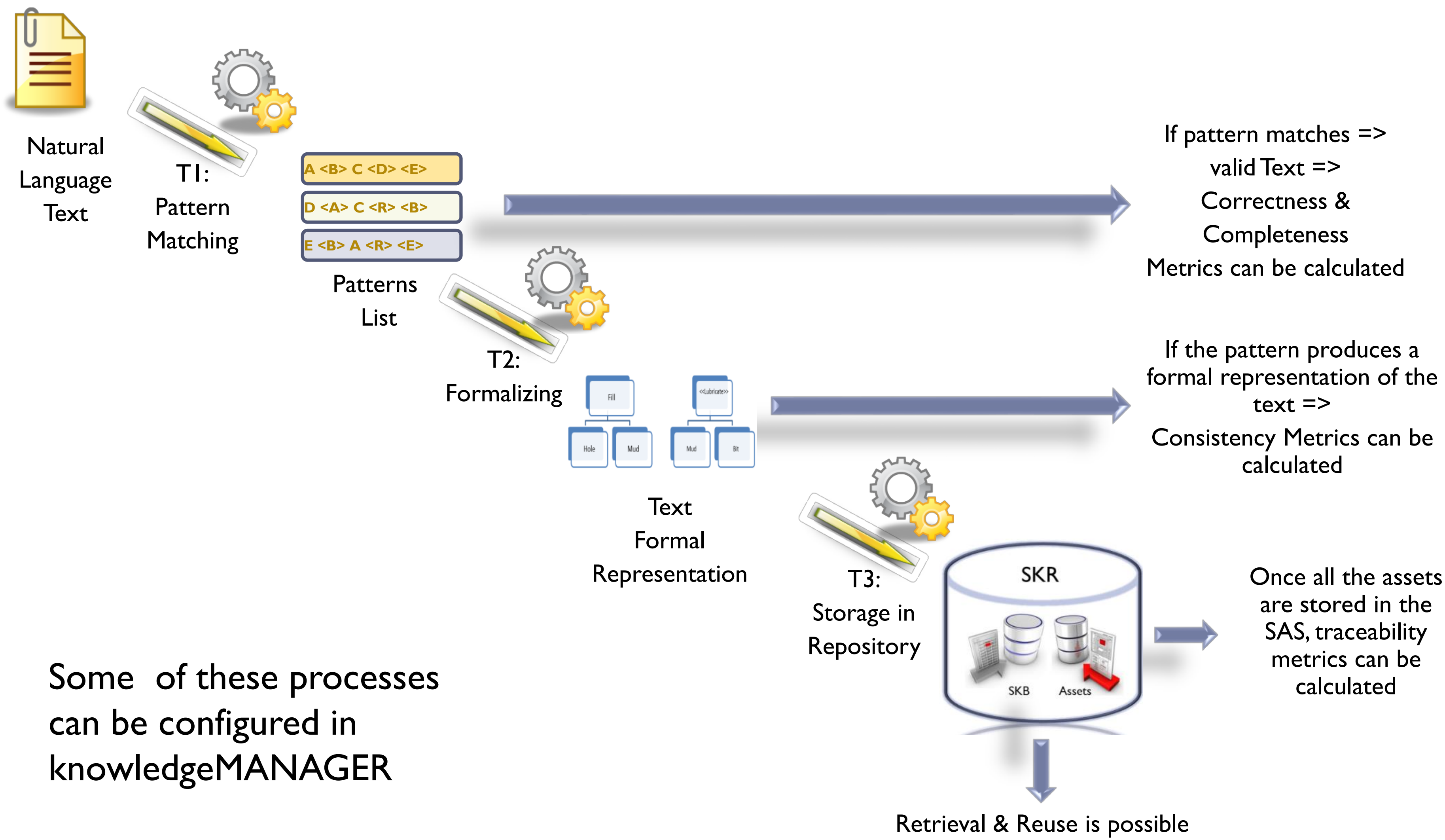
- Input: list of simple terms or compound terms with only term tag chosen for each element
- Output: list of simple terms or compound terms with only term tag chosen for each element changing the numbers written in text to digits
- Example:
  - Input:
    - [The| DETERMINER],
    - [Flight System | NOUN]
      - [Flight| NOUN],
      - [System | NOUN, **PROPER NOUN**],
    - [should| VERB, **MODAL VERB**],
    - [have| VERB],
    - [Three| NUMBER],
    - [Engine| NOUN].
  - Output:
    - [The| DETERMINER],
    - [Flight System | NOUN]
      - [Flight| NOUN],
      - [System | NOUN, **PROPER NOUN**],
    - [should| VERB, **MODAL VERB**],
    - [have| VERB],
    - [3| NUMBER],
    - [Engine| NOUN].

Pattern Matching step: Example



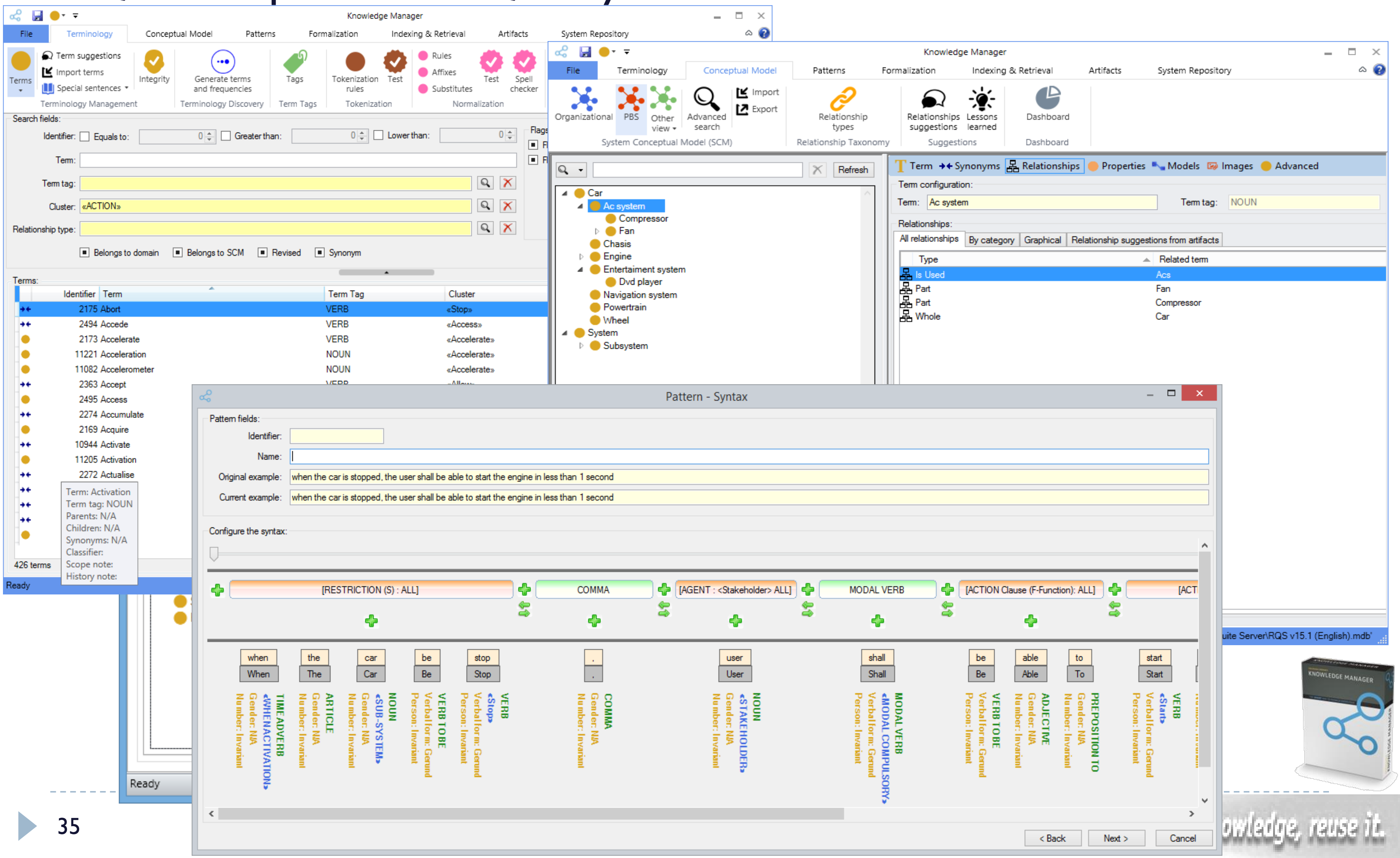
- **Input**
  - [The| DETERMINER],
  - [Flight System | NOUN]
    - [Flight| NOUN],
    - [System | NOUN, **PROPER NOUN**],
  - [Should| VERB, **MODAL VERB**],
  - [Have| VERB],
  - [3| NUMBER],
  - [Engine| NOUN].
- **Output:**
  - [SUBJECT]
    - [The | DETERMINER]
    - [Flight System | PROPER NOUN]
      - [Flight| PROPER NOUN, NOUN],
      - [System| NOUN]
  - [VERB]
    - [Should | VERB, MODAL VERB]
    - [Have | VERB]
  - [PREDICATE]
    - [3 | NUMBER]
    - [Engine | NOUN]

Indexing process in RQS (RQA and RAT)

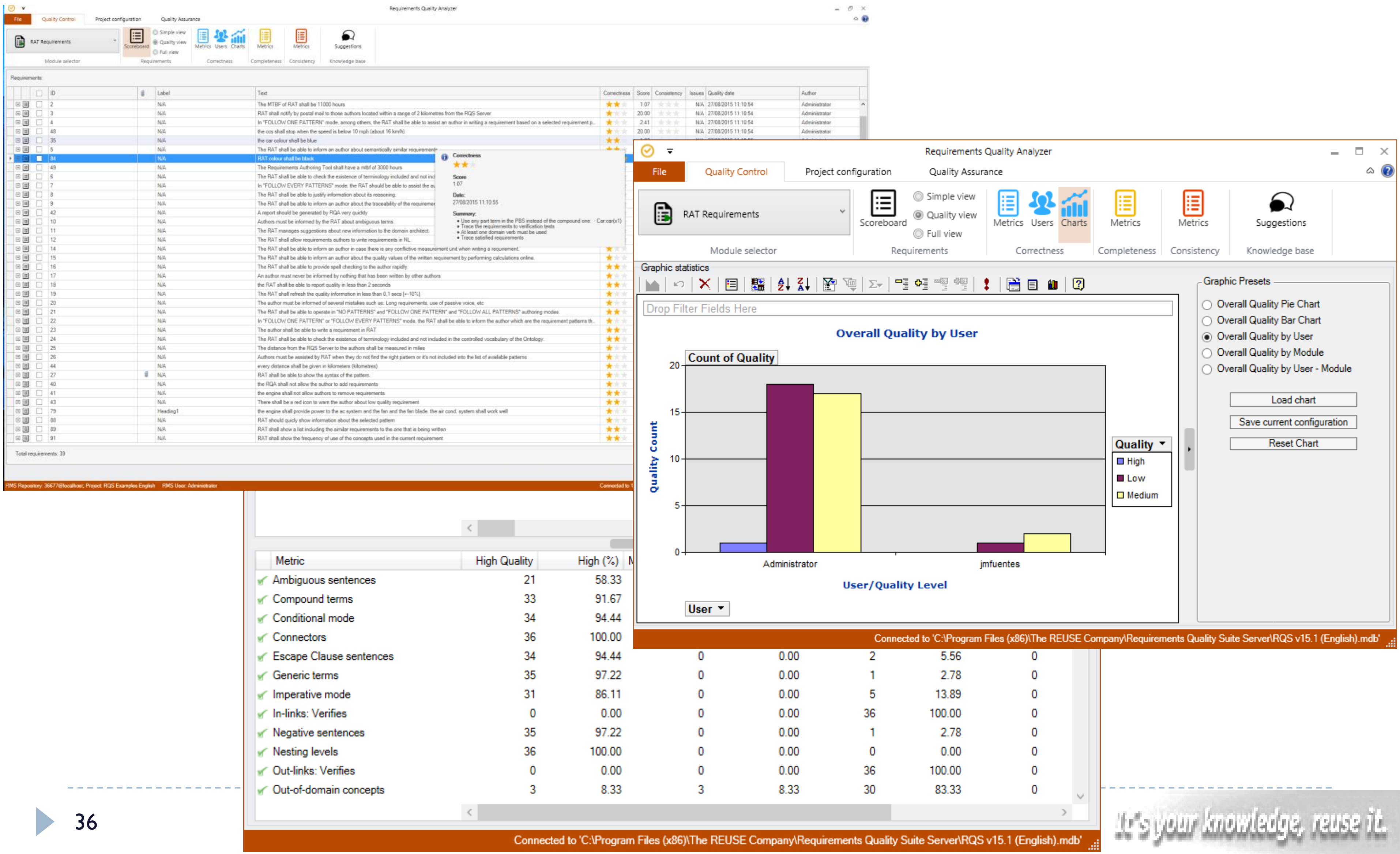


Some of these processes can be configured in knowledgeMANAGER

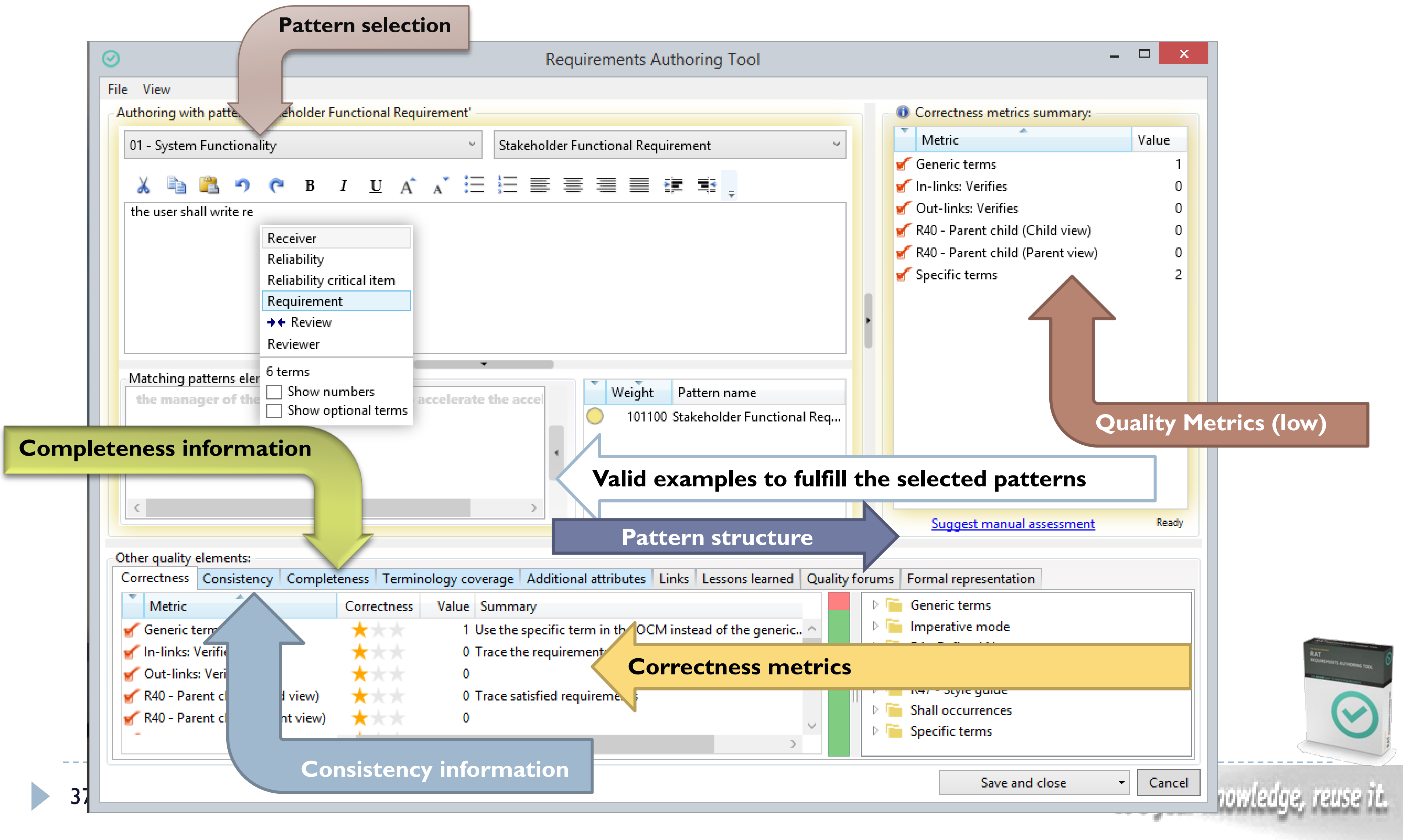
RQS – Requirements Quality Suite: KM



RQS – Requirements Quality Suite: RQA



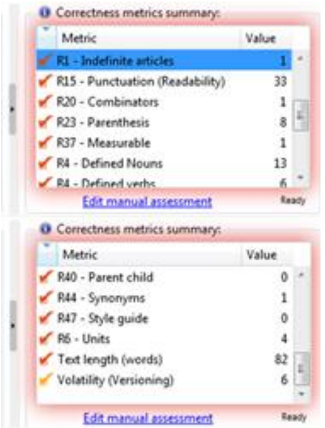
RQS – Requirements Quality Suite: RAT

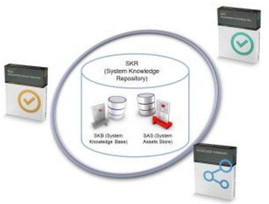


RQS – Requirements Quality Suite: characteristics and implemented rules

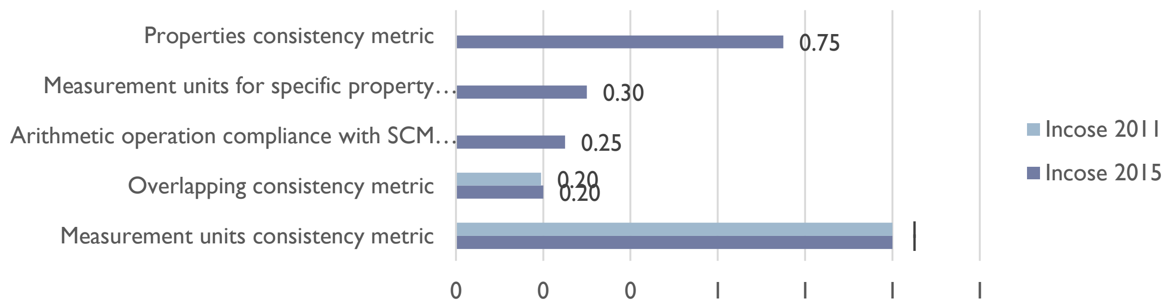
- Fully customizable
- **+60 out-of-the-box quality metrics for correctness**
- **+20 out-of-the-box quality metrics for completeness and consistency**
- Parameterized metrics
- Custom-coded metrics
- Support for requirements verification
- Support for Requirements authoring
- Reporting system
- Knowledge management and semantic approach
- Semantic search engine and requirements reuse

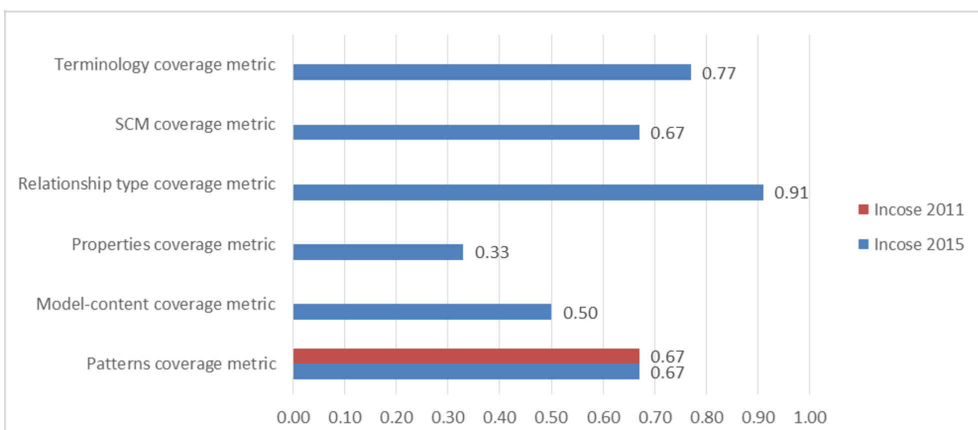
Rule description - INCOSE R1	Metric in RQA	Solution
This rule defines the definite articles to be used. The use of indefinite articles as A leads to ambiguity, it is better to use "THE" instead of "A".	Avoid Articles	Create a list of words: a, an Available in the out of the box version of RQA

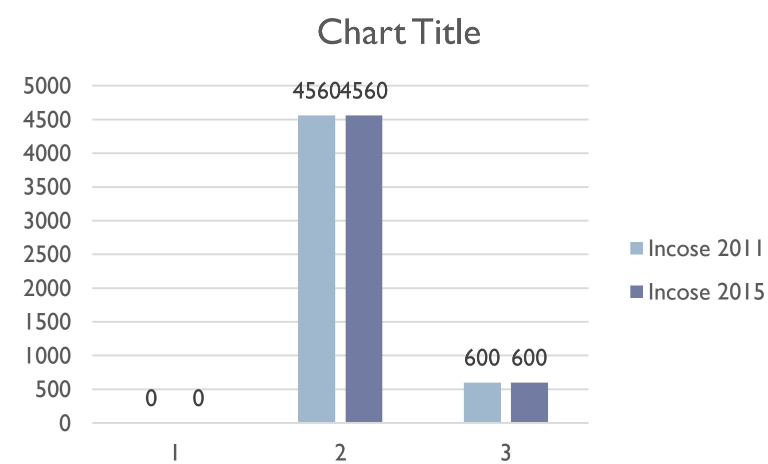




RQS – Some results with INCOSE rules







Applying INCOSE Rules for writing high-quality requirements in Industry

Anabel Fraga
anabel.fraga@uc3m.es

Wednesday, July 20, 2016

KNOWLEDGE REUSE