TRC

WEBINARS 2020

Ensuring Completeness, Consistency, and Correctness with the MASTER Patterns by Sophist and RAT – Authoring Tools

February, 2020

Presenters' profile

- Ilyes Yousfi
  - Sales and Consulting Engineer, The REUSE Company

Ilyes Yousfi
Ilyes.yousfi@reusecompany.com

- José Fuentes
  - Chief Operating Officer, The REUSE Company

José Fuentes
jose.fuentes@reusecompany.com

Table of Contents

About The REUSE Company (TRC)

01 The company was created in **1999**

As a spin-off of a local university in Madrid (Spain)

02 **System + Software Engineers**

Smart combination between Company staff and R&D from Academia

03 **Head Quarters:** Madrid (Spain)

International offices: London (UK) Stockholm (Sweden)

04 Offering a **knowledge centric** approach to leverage system engineering activities in our customers

Research and innovation in our DNA. Public projects

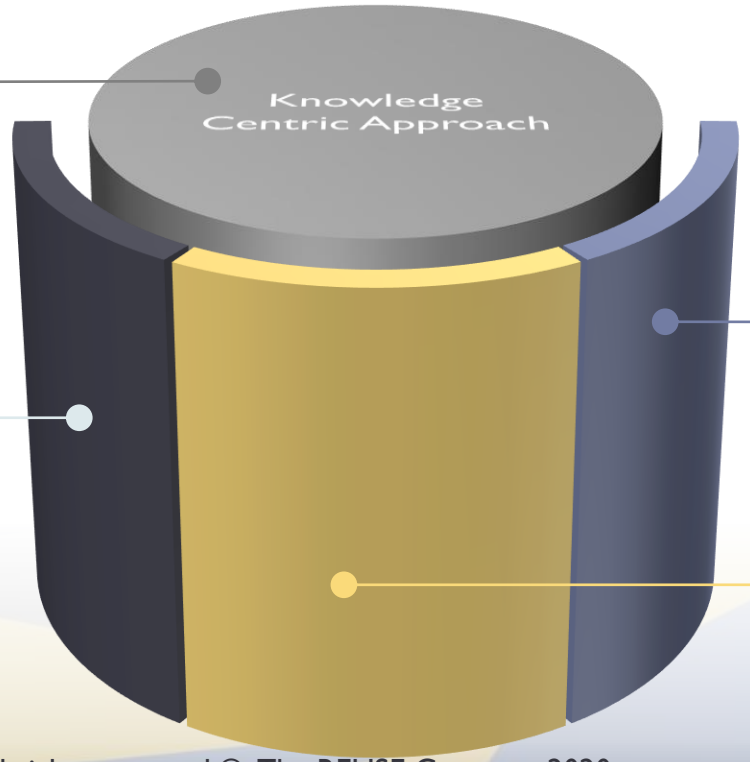
Research and Innovation in our DNA

- Spin-off of Carlos III University of Madrid
- TRC's headquarter is in the Legatec Technology Park of the University
- ≈10% of revenues are devoted to R&D
- TRC is actively involved in several large EU research projects

Introduction to The REUSE Company

T(he) R(euse) C(ompany)

Leveraging Systems Engineering activities

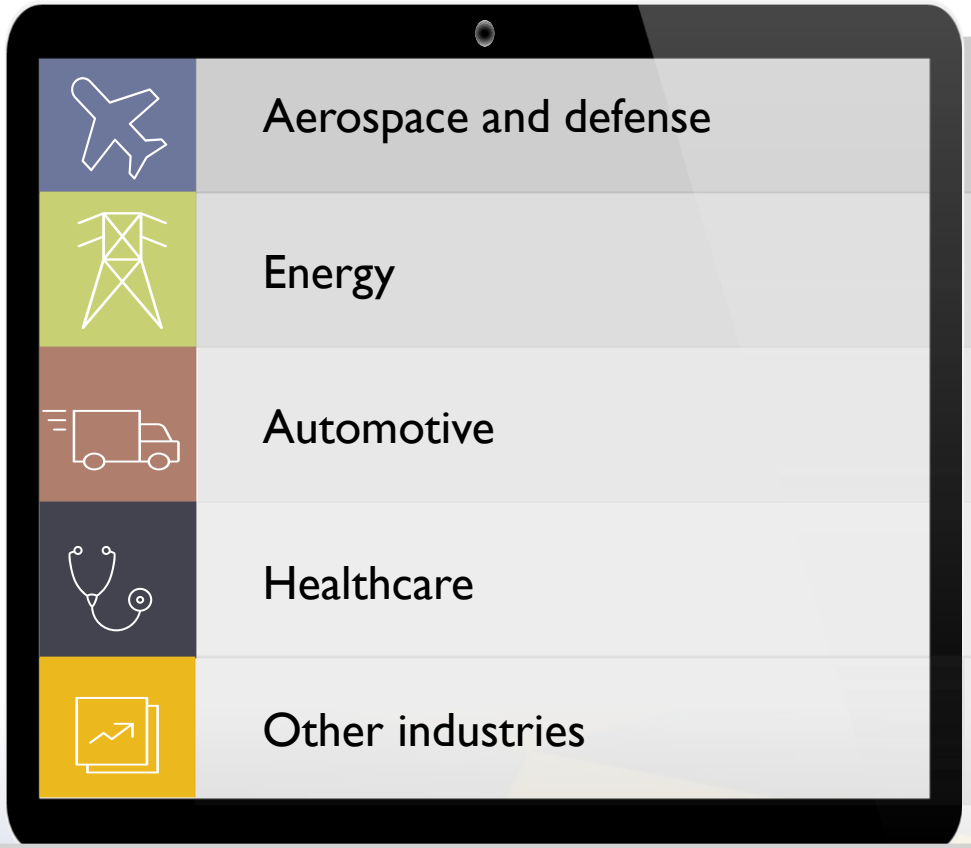


TRACEABILITY

QUALITY

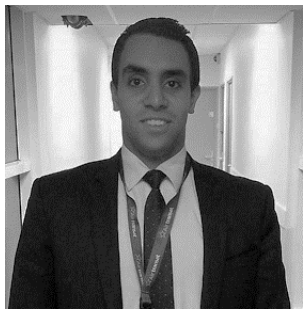
REUSABILITY

Who is using our technology?



AIRBUS
AIRBUS DEFENCE & SPACE
SAFRAN AIRCRAFT ENGINES
HENSOLDT
AIRBUS GROUP INNOVATIONS
THALES
THALES Communications & Security S.A.S.
ariane GROUP
FUSION FOR ENERGY
iter the way to new energy
EDF
REPSOL
RENAULT
THINK SURGICAL
Health Net
SIEMENS
acciona Agua
orange
rtve
ELRA European Land Registry Association
tirant lo blanch

Ilyes YOUSFI



- **Sales & Consulting Engineer** at The REUSE Company.
- Ilyes has experience in **Energy Engineering** and Life Cycle Assessment methods in different industrial sectors such as energy and aeronautics.
- Ilyes' main missions are: **international sales** of our systems engineering solutions, **consulting** of our customers **and account management**.
- His main interests include **knowledge** management, **industrial** engineering, **requirements** engineering, and **sustainable engineering processes**.

TRC WEBINARS 2020

the REUSE company

Ensuring Completeness, Consistency, and Correctness with the MASTER Patterns by Sophist and RAT – Authoring Tools

SOPHIST

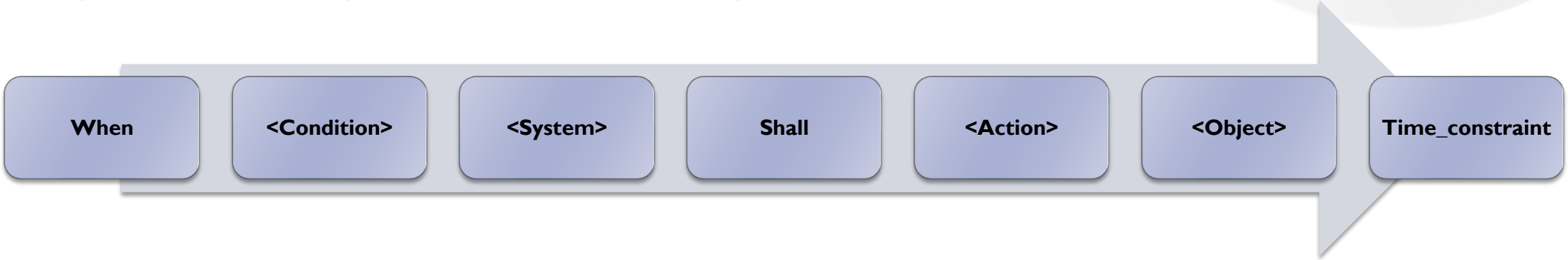
Why using patterns ?

History of requirement patterns

- First introduced in 1998 : Future Surface Combatant (FSC) Defence Project – UK
- Development of Nature Language Processing (NLP) tools
- Also known as "boilerplates" or "templates"
- Developed first in software engineering and extended to systems engineering (INCOSE patterns working group)

Definition and benefits of requirement patterns

- Represents the structures every correct requirement should meet
- Different types of requirements → different patterns (templates)
- Customizable for every domain, customer and content of each requirements document
- Libraries with sets of patterns
- Represented as a sequential set of restrictions: placeholders



Example : When ice is detected, the car shall show an ice icon in less than 0,5 s from its detection

The Ontology for KCSE

01 Vocabulary

Controlled Organizational and Project Vocabulary for a common understanding among stakeholders

02 SCM/Architectures

Recreate and capture the system architectures represented in views and models. Stablish relationships among system and system elements

03 Patterns

Represent requirements similarities and enable formal representation, automatic recognition and aid authors

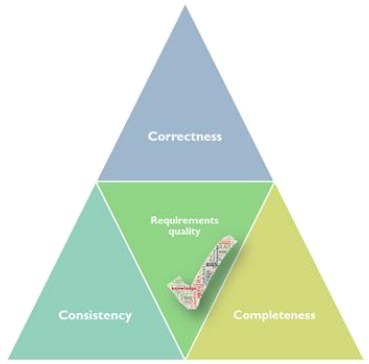
04 Formalization

Representation of assets semantic through SRL – System Representation Language

05 Reasoning

A combination of rules, tasks and groups to infer information from valuable assets

Requirements quality metrics: knowledge needs



SKB – System Knowledge Base

Consistency metrics (most)
Completeness metrics (some)
Correctness metrics (few)
Semantic retrieval capabilities

05. Reasoning

04. Formalization

Writing assistance (RAT)
Correctness metrics (some)
Improve accuracy of other quality metrics
Completeness/Consistency metrics (some)
Semantic retrieval capabilities
Requirements and knowledge elicitation

03. Patterns

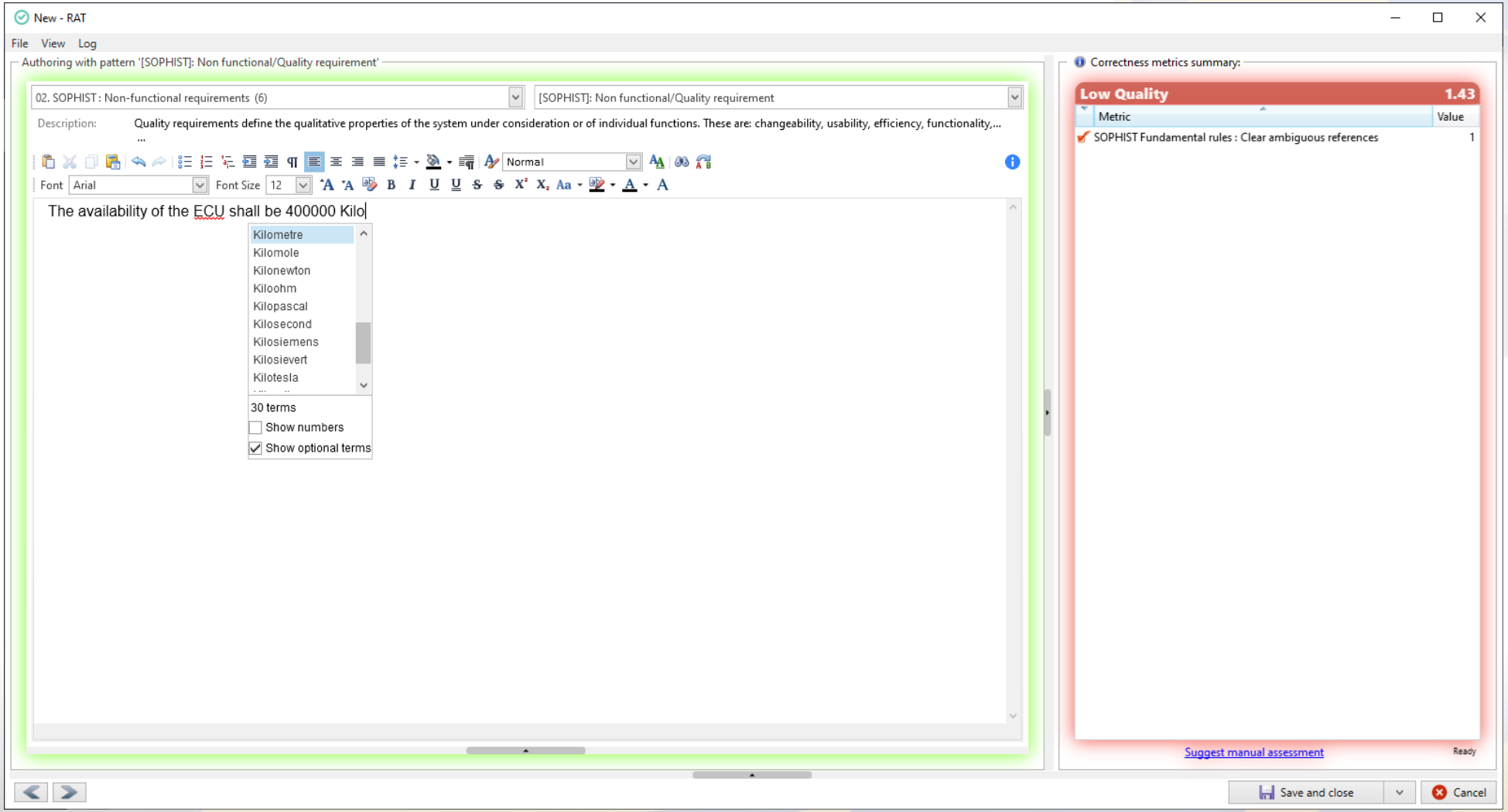
Completeness metrics (some)
Correctness metrics (few)
Semantic retrieval capabilities
Writing assistance

02. SCM / architectures

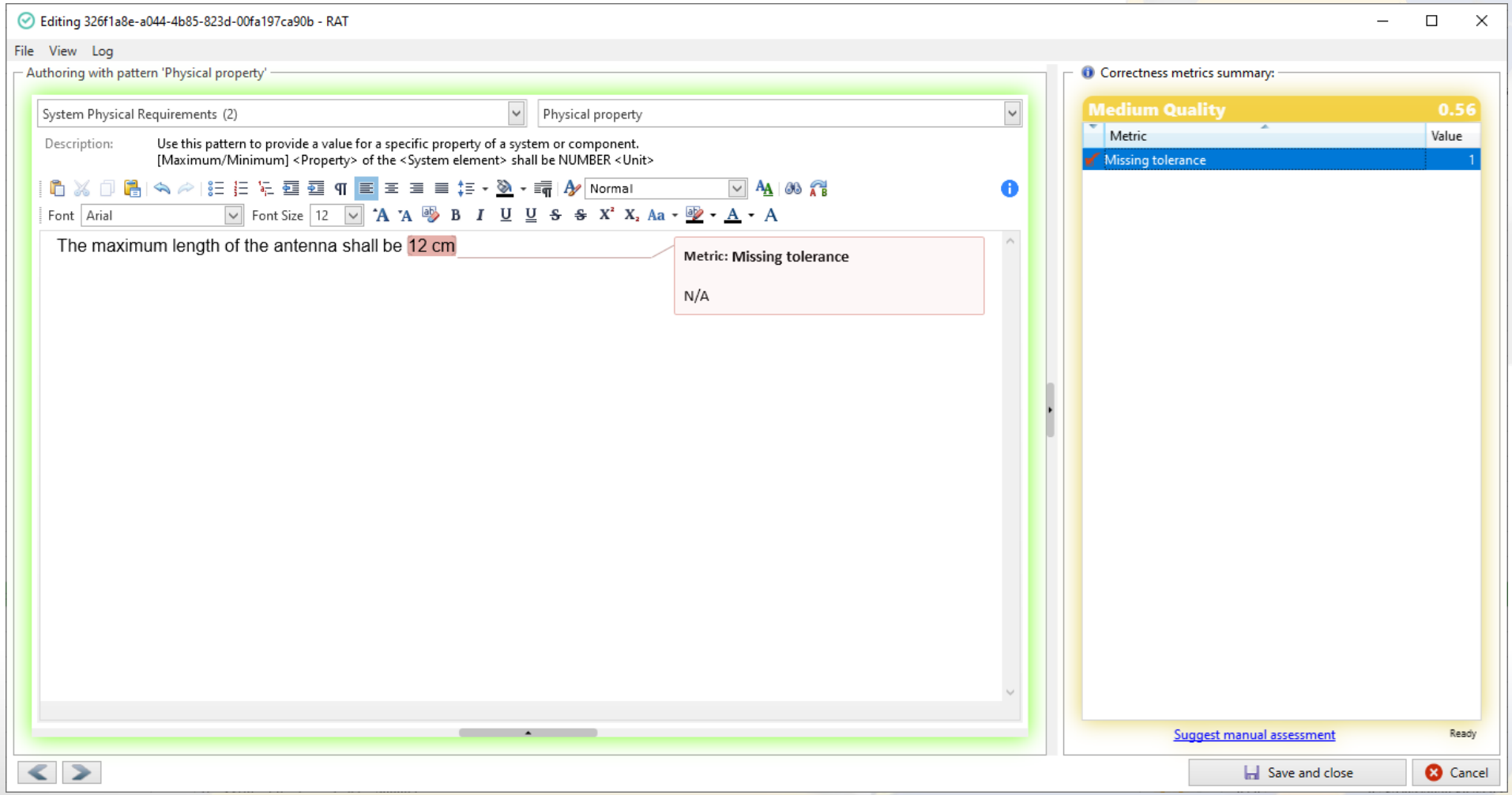
01. Vocabulary

CCC Metrics (most)

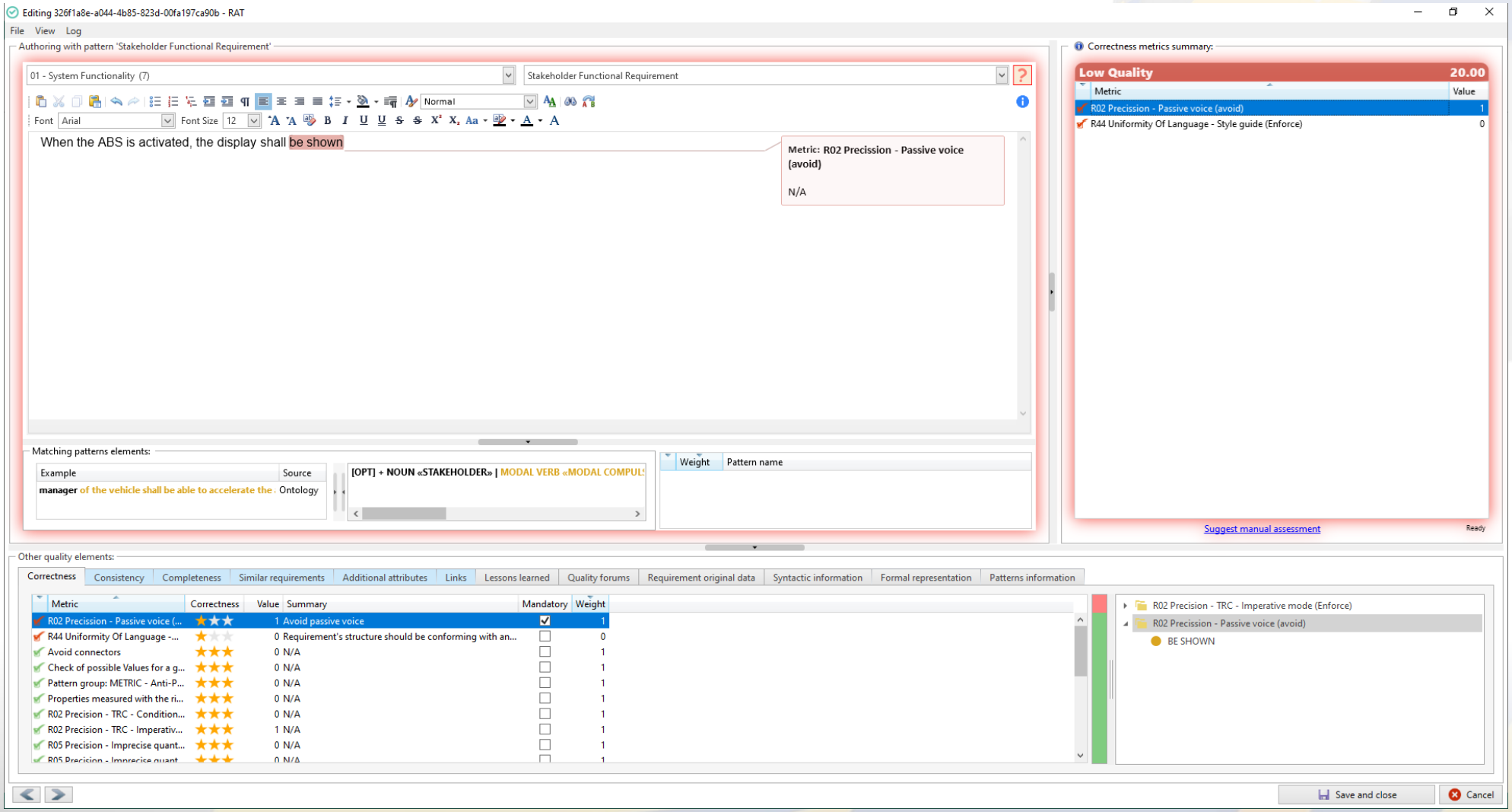
Some examples : Dictionary references (acronyms)



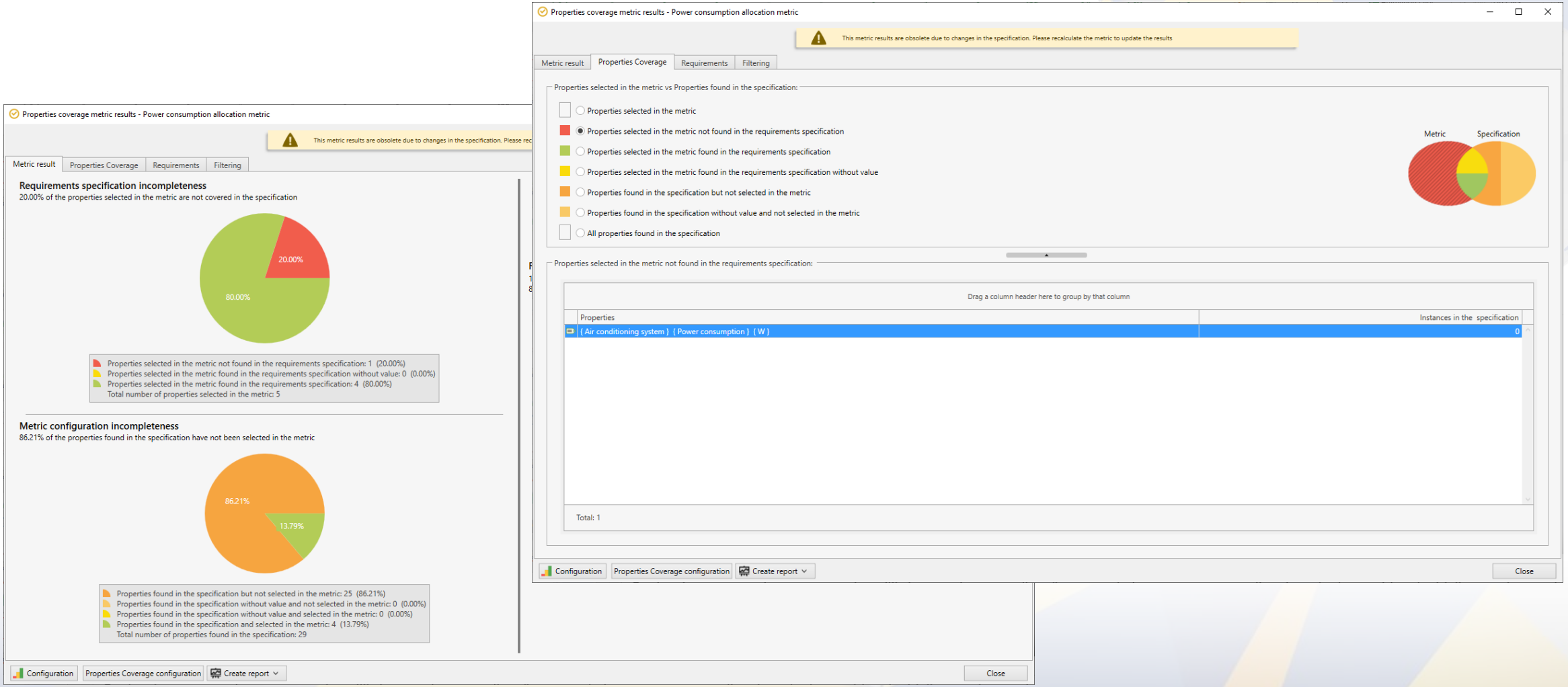
Some examples : enforcing the tolerance when using measurement units



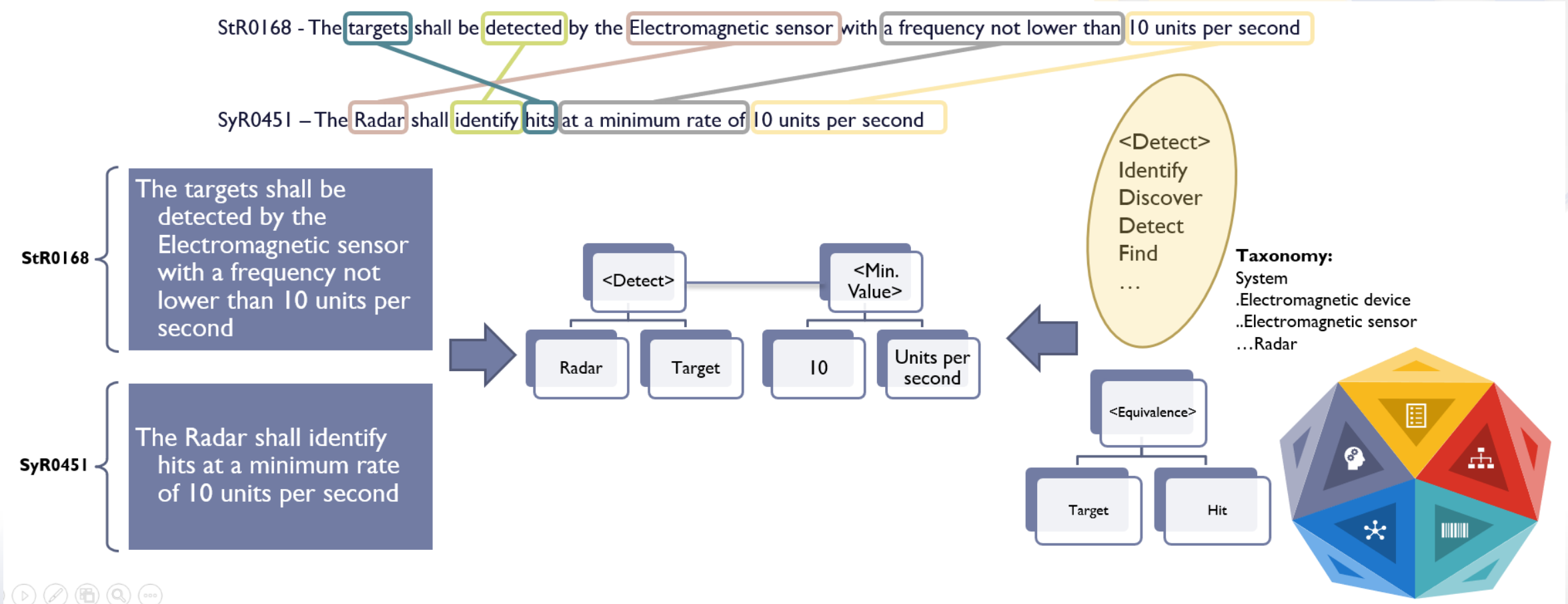
Some examples : filter the passive voice detection only on selected groups of patterns



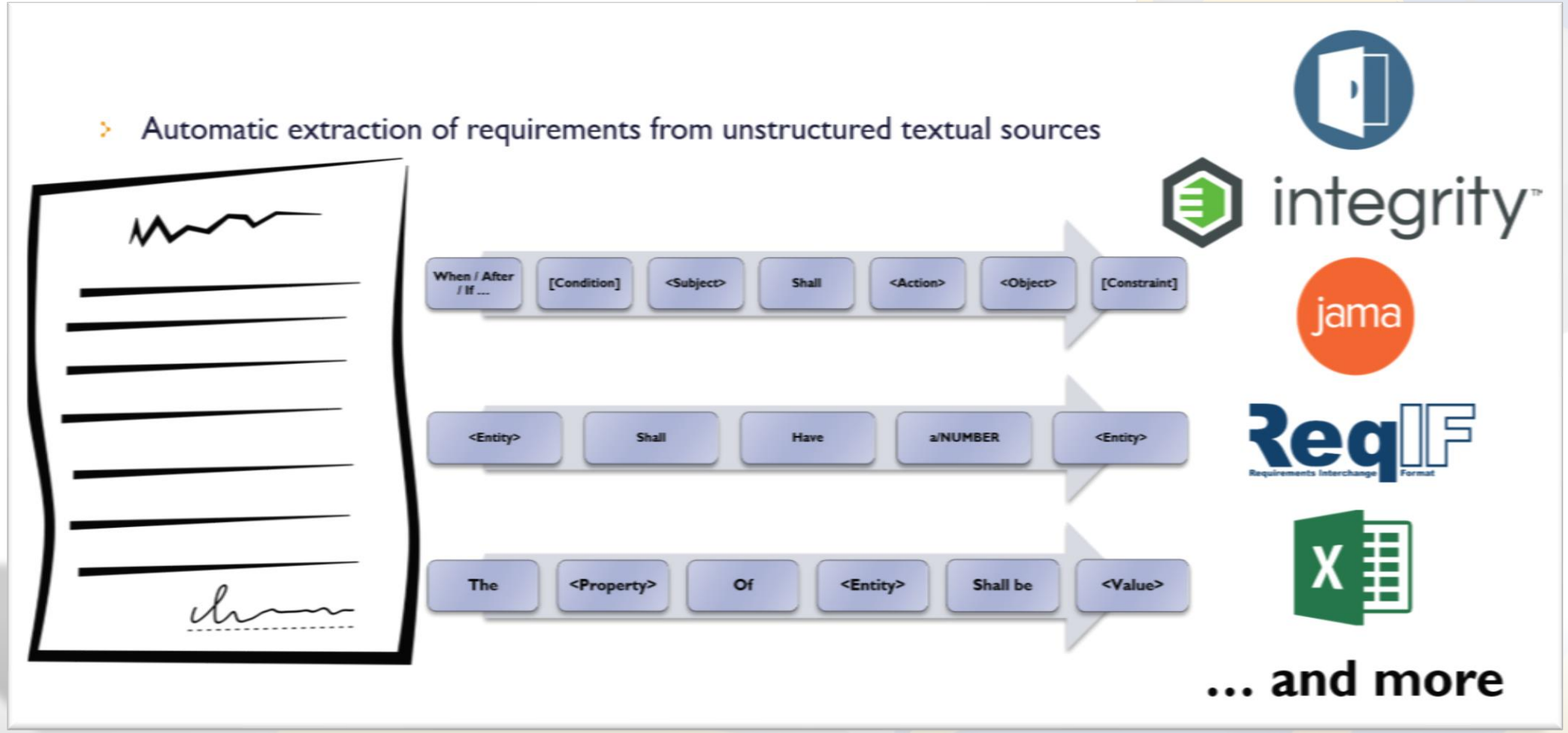
Some examples : completeness metrics (property described for all the elements of a system/sub-system)



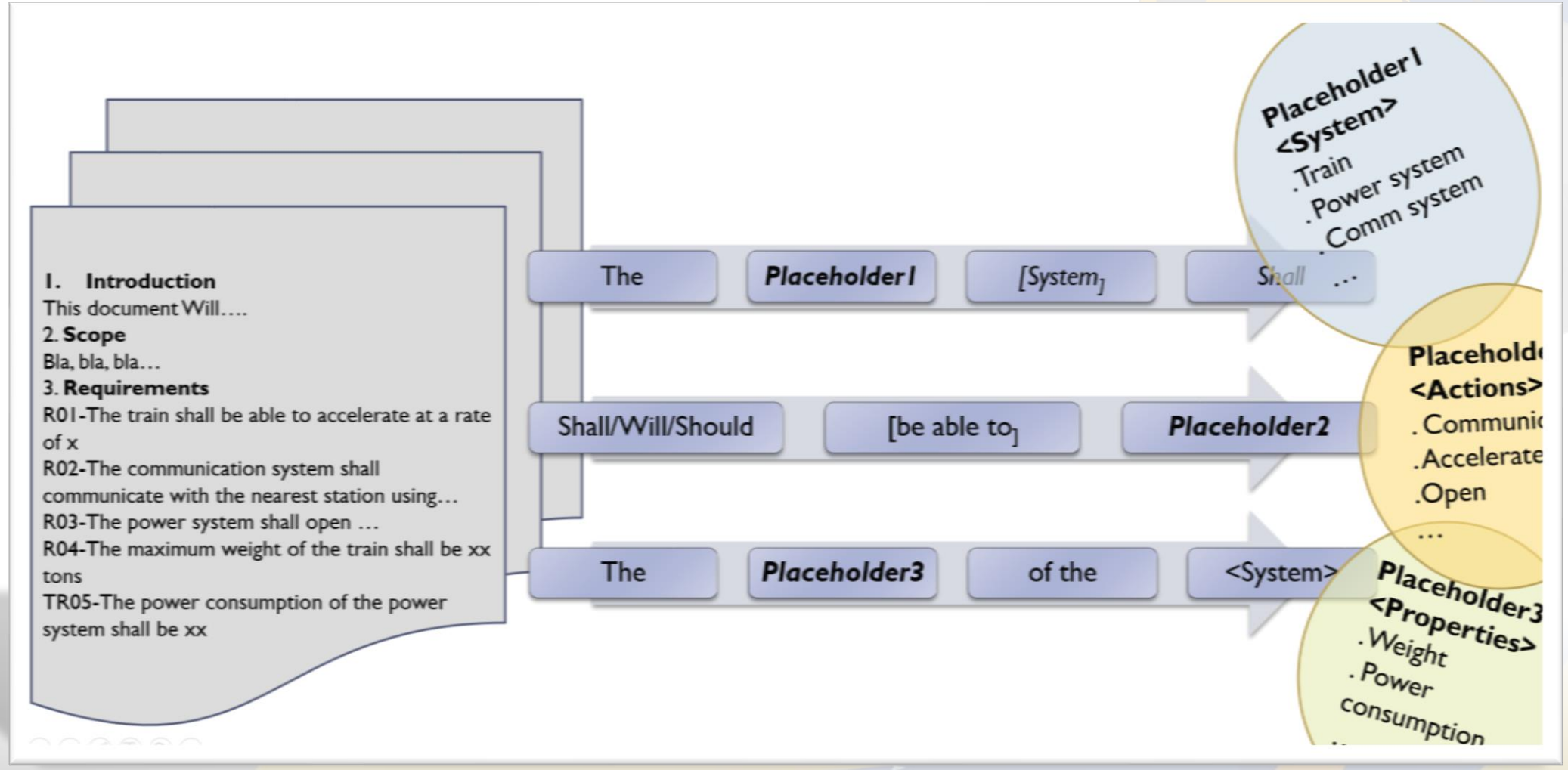
Some examples : Semantic processing of requirements to identify similar requirements in the meaning



Some examples : Extraction of requirements following a given pattern and integration into RM tools

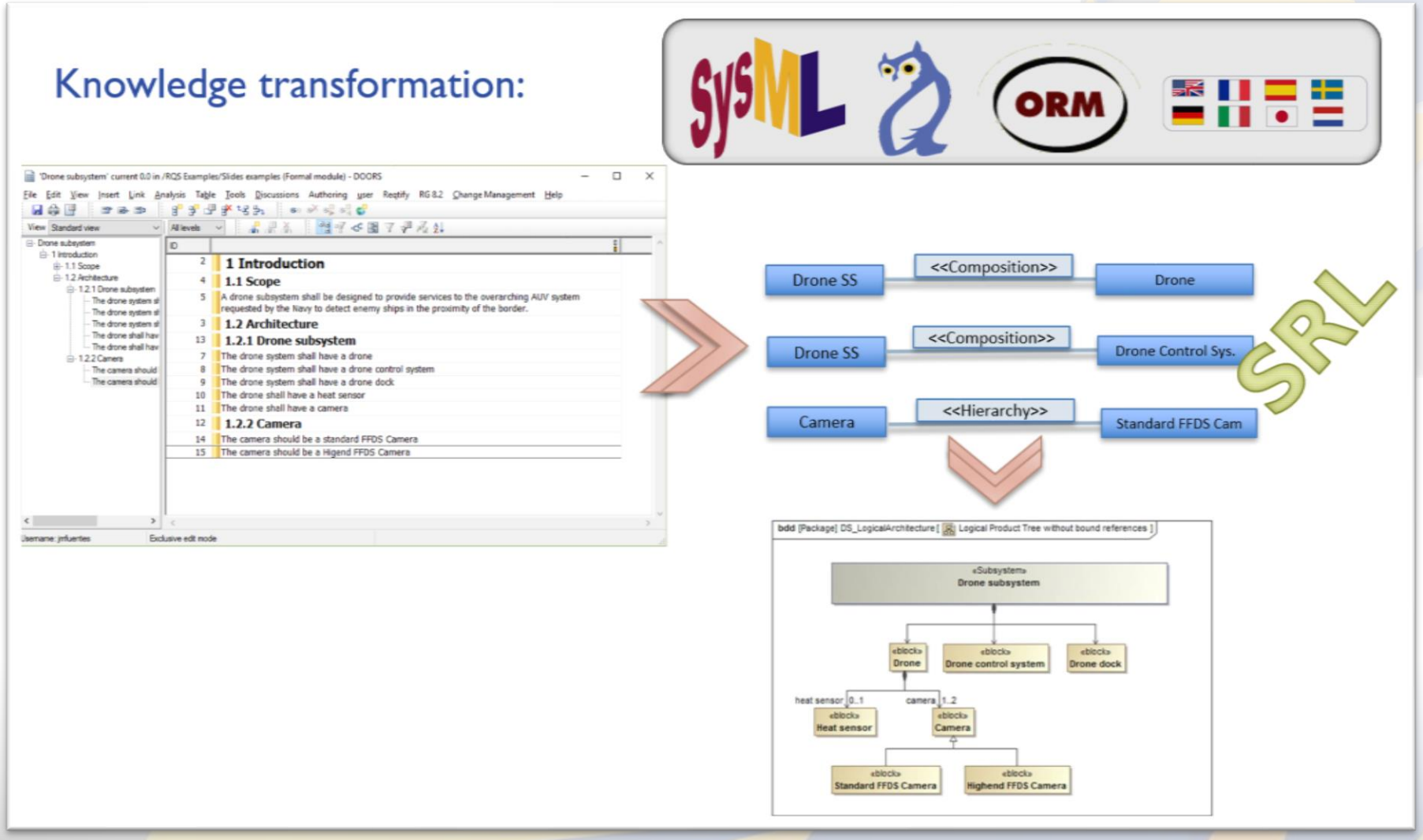


Some examples : Extraction of specific terms using placeholders of a given pattern to feed the ontology



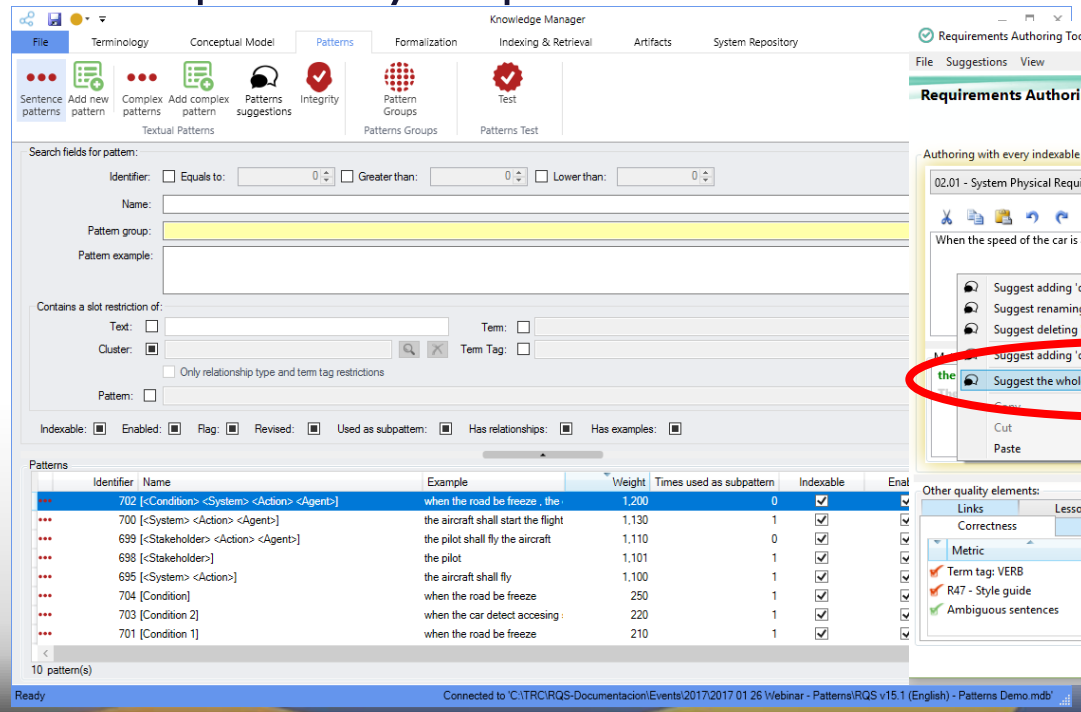
Some examples : Knowledge transformation from system requirements to system modeling

Knowledge transformation:



Patterns inside our Systems Engineering Suite (SES)

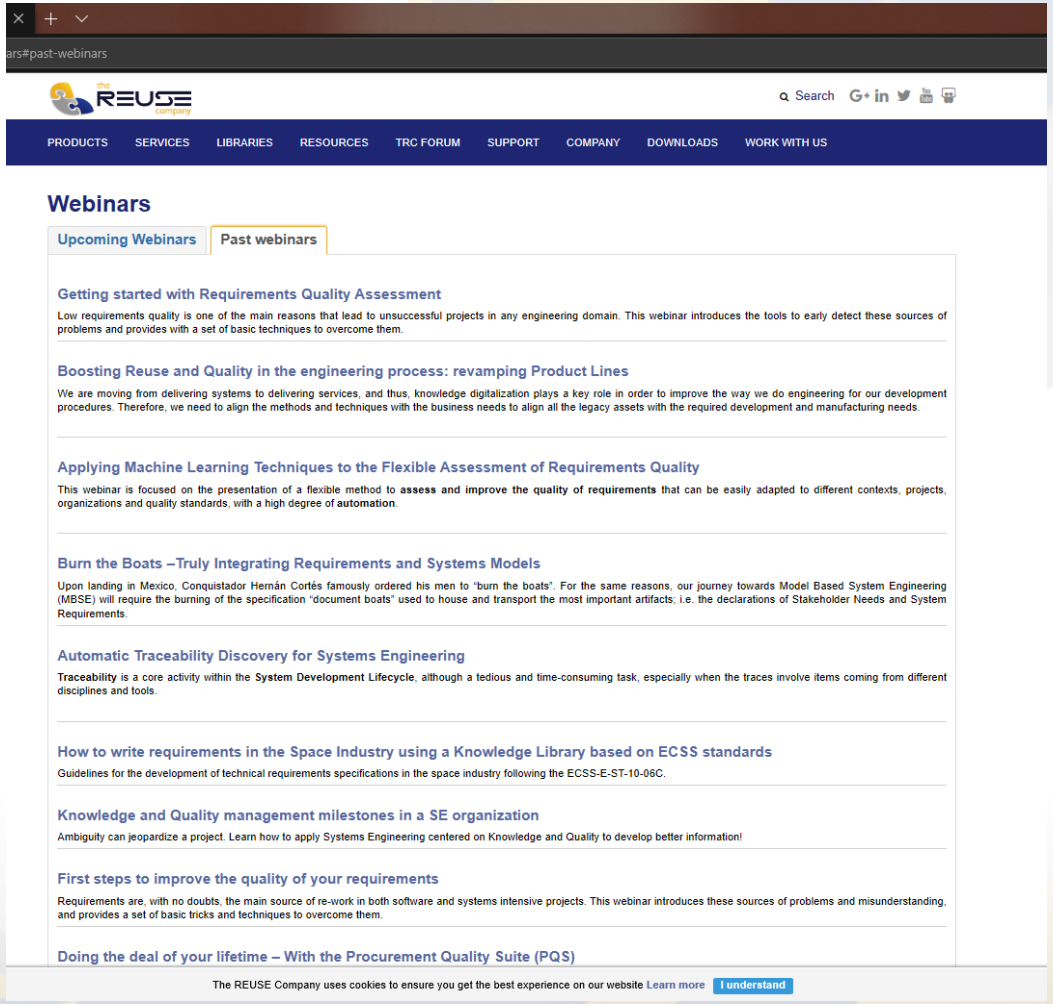
- Some included out-of-the-box database with SES Server installation
- Other included in specific SES Libraries (ECSS, EARS, …)
- Managed in Knowledge Manager (KM)
- Managed by the Knowledge Architect…
- … but also requested by Requirements Authors :



TRC WEBINARS 2020

More content on requirements patterns…

- Webinars focused on patterns definition and application for requirements quality and authoring
  - Quality measuring using patterns : https://www.reusecompany.com/webinar/requirements-patterns-for-requirements-quality-analysis-and-requirements-writing
  - Analyse documents and capture content to feed the knowledge base : https://www.reusecompany.com/webinar/capturing-content-for-your-knowledge-base-with-km-rqa-and-rat
  - Quality metrics : Correctness
    - Basic level : https://www.reusecompany.com/webinar/capturing-content-for-your-knowledge-base-with-km-rqa-and-rat
    - Advanced level : https://www.reusecompany.com/webinar/advanced-requirements-verification-using-parameterized-metrics-in-rqa
  - Quality metrics – Consistency : https://www.reusecompany.com/webinar/how-to-check-requirements-consistency-with-rqs-and-ibm-doors
  - Quality metric – Completeness : https://www.reusecompany.com/webinar/checking-requirements-completeness-with-rqa-and-ibm-doors



Implementing the MASTER patterns inside TRC tools

Who are the SOPHISTs ?

SOPHIST

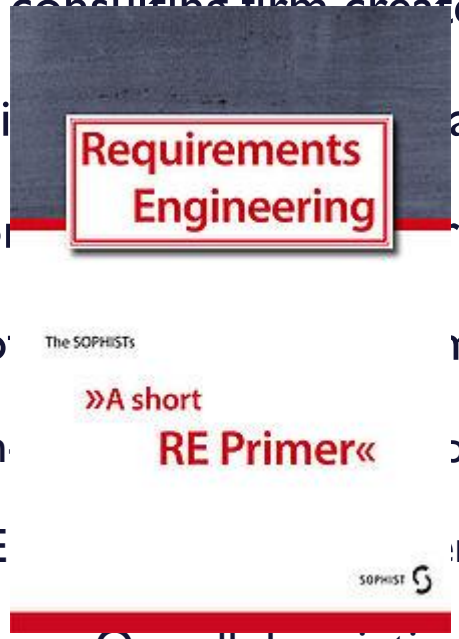
- Training & consulting firm created in 1996 – today 60 members
- Specialization in Requirements and Systems Engineering
- Co-creator of the IREB Standard
- Provider of 100% tool neutral methods and knowledge
  - Wissen-for-free section : free publications of requirements engineering practices
  - MASTER patterns : Requirements patterns to enhance the structuration of requirements documentation
  - RE Primer : Overall description of the requirements engineering phase and set of rules (SOPHISTen-Regelwerk)

WISSEN for free

Source : Sophist.de

Who are the SOPHISTs ?

- Training & consulting firm created in 1996 – today 60 members
- Specializati and Systems Engineering
- Co-creator d
- Provider o methods and knowledge
  - Wissen ublications of requirements engineering prac
  - MASTE nts patterns to enhance the structuration of ntation
  - RE Primer : Overall description of the requirements engineering phase and set of rules (SOPHISTen-Regelwerk)



Source : Sophist.de

SOPHIST MASTER patterns

- Cross-domain patterns to express system requirements :
  - Functional requirements
  - Non-functional requirements
  - With or without introducing a condition to the main sentence of requirements

- The use of the MASTER patterns enhances :
  - The structuration of the syntax of requirements
  - The uniformity of sentence structures (linking words in conditions…)
  - The scope of each set of pattern (functional, non-functional,…).
  - The scope of conditions (time-related, logical, triggered by an event)

https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/MASTeR_Broschuere_5-Auflage_Komplett_Lesezeichen_Update_web.pdf

SOPHIST MASTER patterns : Functional requirements (FunctionMASTER)

- Objective : write functional requirements
- 3 different cases to express the FunctionMASTER
  - Independent system activity
  - User interaction
  - Interface requirement

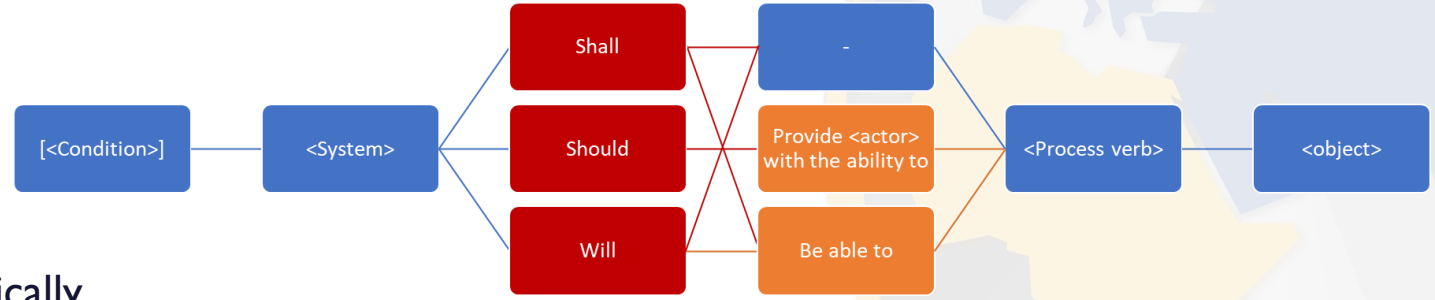
- The FunctionMASTER also has a detailed version to give more context to the functionality described

FunctionMASTER : 3 different paths

Independent system activity

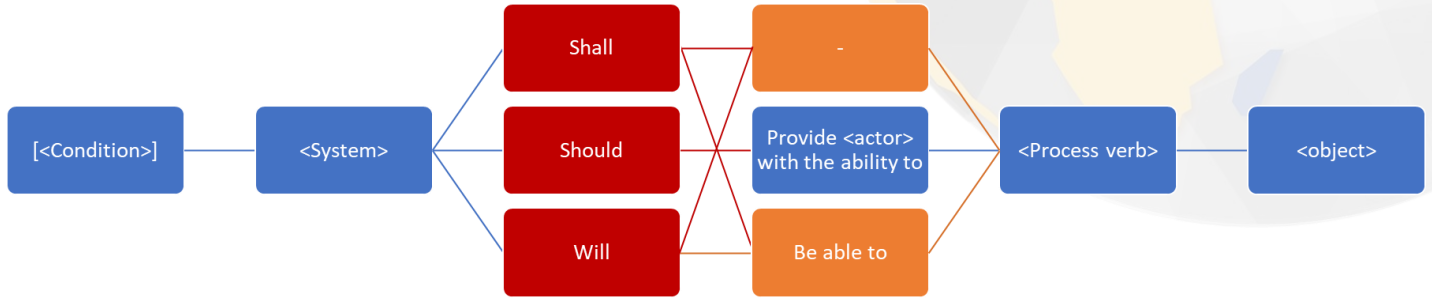
- Function started and perform automatically

by the system



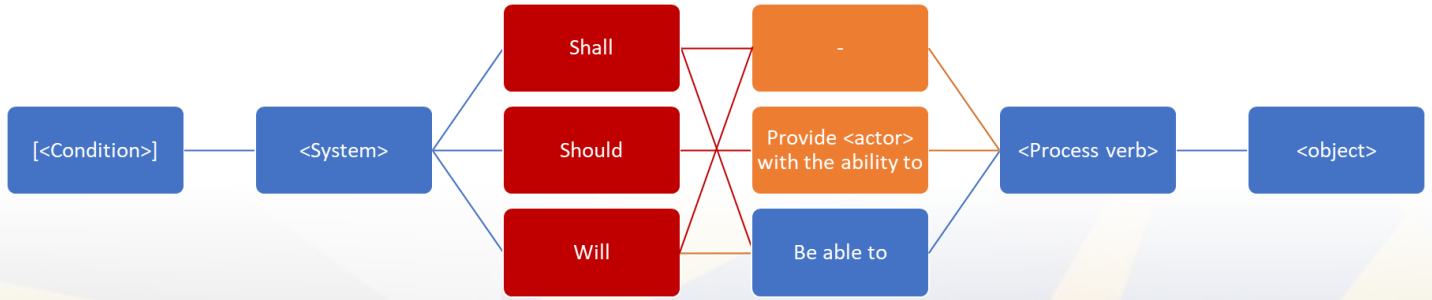
User interaction

- The system enables the user to perform a function to achieve a goal. The system relies on the user to perform the function



Interface requirement

- Cases when the system relies on information coming from a third party (other than a user)



FunctionMASTER : Independent system activity – Implementation inside the TRC Knowledge Manager

Name:
[[SOPHIST]: FunctionalMASTER/Independent system activity]

Description:
The system starts the function automatically and therefore performs it automatically.
This type of requirement is built up with a legal commitment (modal verb shall, should or will) and a process word in the infinitive form.
In this case the process word describes a function automatically performed by the system, where no interaction of the user is expected.

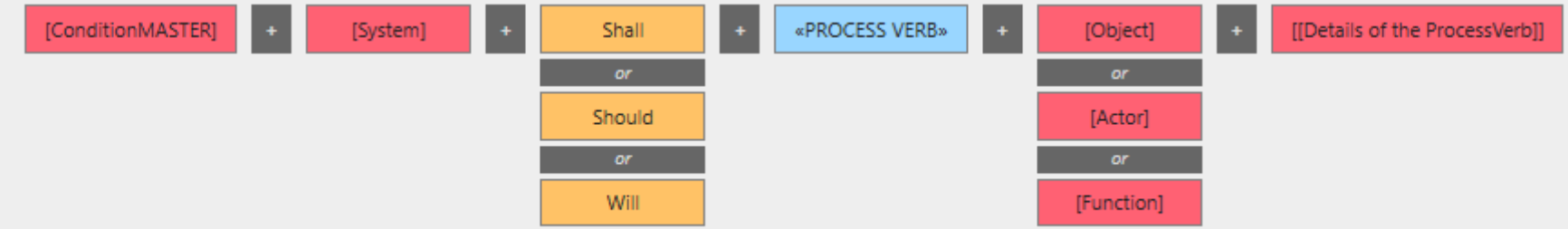
Extracted and translated from : "Die SOPHISTen Schablonen für alle Fälle"
https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/Wissen_for_free/MASTeR_Broschuere_5-Auflage_Komplett_Lesezeichen_Update_web.pdf

Pattern group(s):
- 01. SOPHIST: Functional Requirements (372)

Example:
- The system shall design postcards

Indexable:	Enabled:	Weight:
Yes	Yes	1,010

Syntax:



FunctionMASTER : User interaction – Implementation inside the TRC Knowledge Manager

Name:
[[SOPHIST]: FunctionalMASTER/User interaction]

Description:
N/A

Pattern group(s):
- 01. SOPHIST: Functional Requirements (372)

Example:
- The system shall provide the user with the ability to print selected colored photos at the net-work printer.

Indexable:	Enabled:	Weight:
Yes	Yes	1,020

Syntax:

FunctionMASTER : Interface requirement – Implementation inside the TRC Knowledge Manager

Name:
[[SOPHIST]: FunctionalMASTER/Interface requirement]

Description:
Functional MASTER / Interface requirement

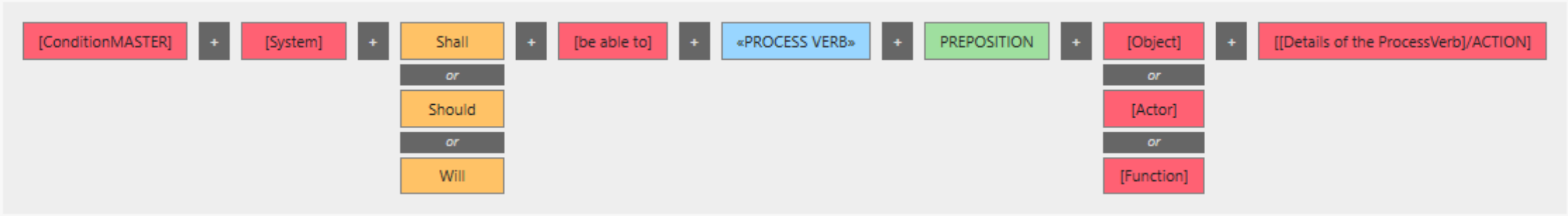
Interface requirements comprise all the cases when the system performs a given function relying on information coming from a third party (other than the User), which could be a neighboring or external system.
The supply of information can unpredictedly occuer and in irregular frequency.
These requirements are specified with the phrase "BE ABLE TO"

Pattern group(s):
- 01. SOPHIST: Functional Requirements (372)

Example:
- The system shall be able to register new customers
- The library system shall be able to archive customer data in the database

Indexable:	Enabled:	Weight:
Yes	Yes	1,030

Syntax:

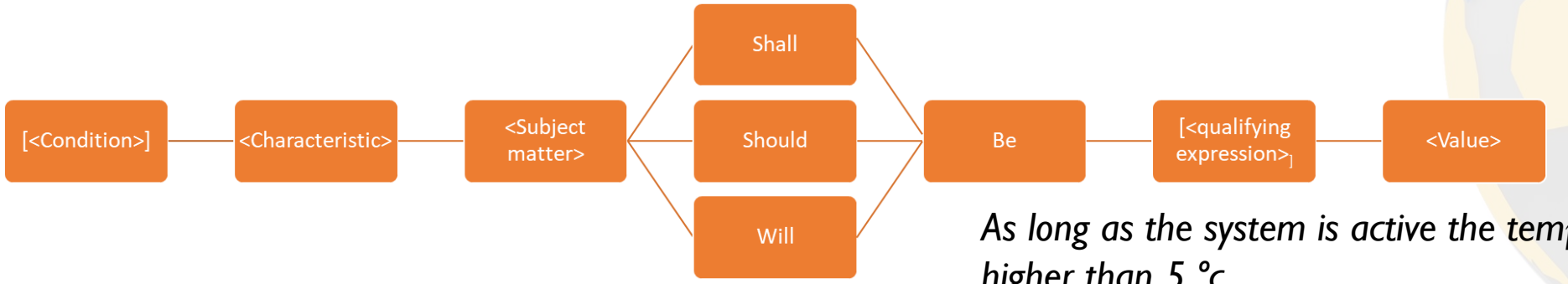


SOPHIST MASTER patterns : Non-functional requirements

- Requirements which do not address the system functionality but contain elements with impact on the addressed functions
- 6 sub-categories of non-functional requirements

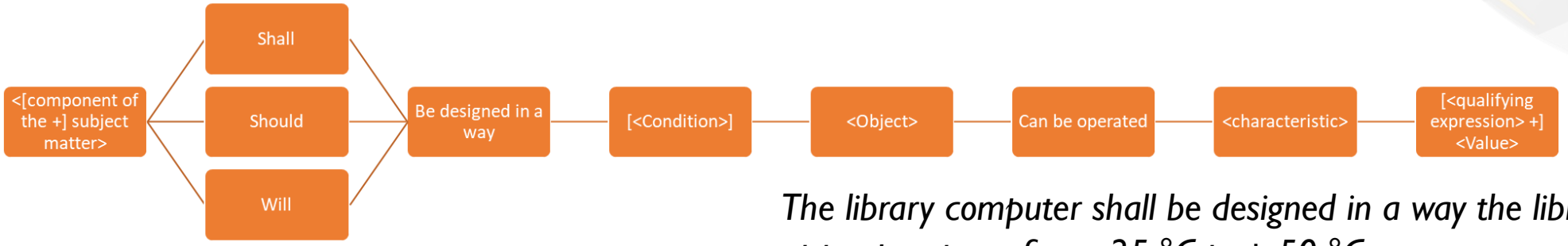
	Addressed content	PropertyMASTER	EnvironmentMASTER	ProcessMASTER
Quality requirements	Qualitative property of the system of interest (performance requirement)	X		
Technological requirements	Efficient way to give more accuracy to the scope of a system functionality	X	Environmental requirements Quantity requirement	
User interface requirements	Focus on the user interface of the system. Details on the visual, acoustic presentation of the functional operations	X		
Requirements for other delivery components	Delivery components : training documents, installation software, tools for assembling components, …	X		
Requirements for activities to be carried out	Description of the process, that is to say the way the system is operated	X		X
Legal-contractual requirements	Agreed rights and obligations with regards to the development and use of the product to be created.	X		X

SOPHIST MASTER patterns : Non-functional MASTER Templates



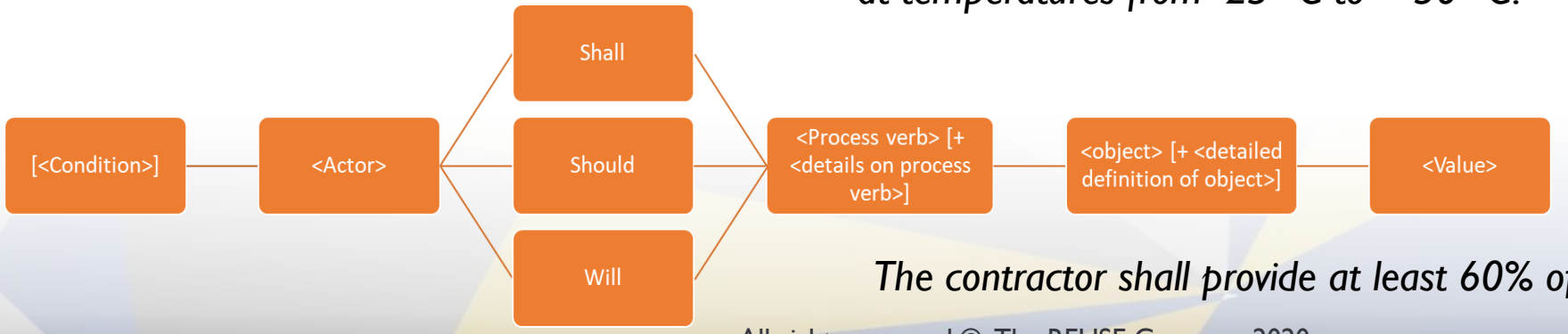
PropertyMASTER

As long as the system is active the temperature of the system shall be higher than 5 °c.



EnvironmentMASTER

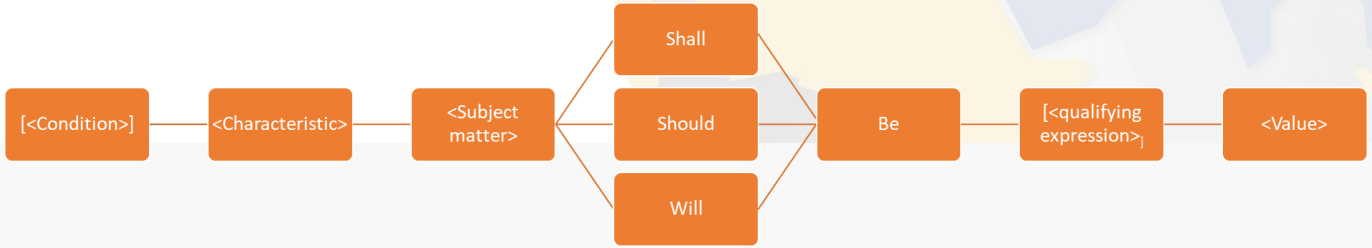
The library computer shall be designed in a way the library system can be operated at temperatures from -25 °C to + 50 °C.



ProcessMASTER

The contractor shall provide at least 60% of the service with its own employees.

SOPHIST MASTER patterns : PropertyMASTER



Name:
[[SOPHIST]: PropertyMASTER]

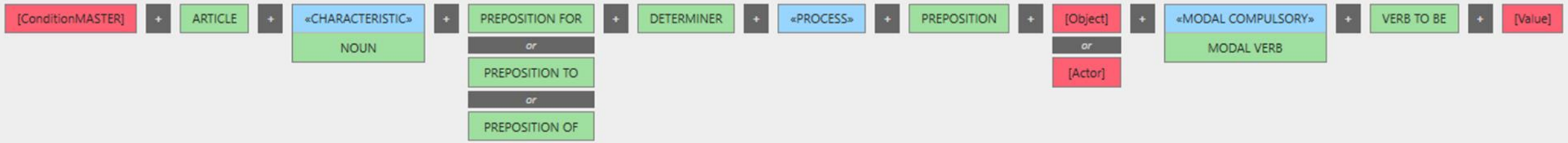
Description:
N/A

Pattern group(s):
- 01 - SOPHIST (391)

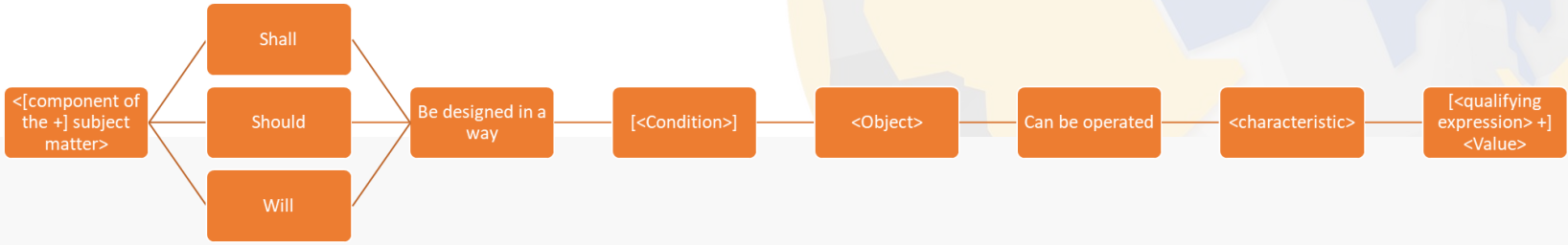
Example:
- The weight of the library system shall be lower than 10 kg

Indexable:	Enabled:	Weight:
Yes	Yes	1,100

Syntax:



SOPHIST MASTER patterns : EnvironmentMASTER



Name:
[[SOPHIST]: EnvironmentMASTER]

Description:
Pattern aimed at describing technical requirements, more specifically requirements related to the environment of the system and the properties of the equipment

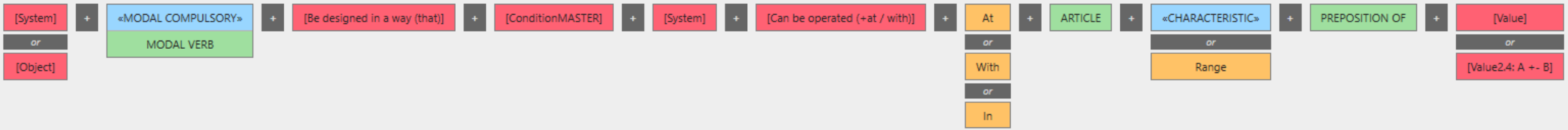
Extracted from : Chris Rupp and die SOPHISTen. Schablonen für alle Fälle. SOPHIST GmbH, 2014

Pattern group(s):
- 01 - SOPHIST (391)

Example:
- The system shall be designed in a way the system can be operated at a voltage of 220 V +- 10 V

Indexable:	Enabled:	Weight:
Yes	Yes	3,361

Syntax:



SOPHIST MASTER patterns : ProcessMASTER

Name:
[[SOPHIST]: ProcessMASTER]

Description:
Patterns aimed at writing legal-contractual requirements and reqirements for activities to be carried out.

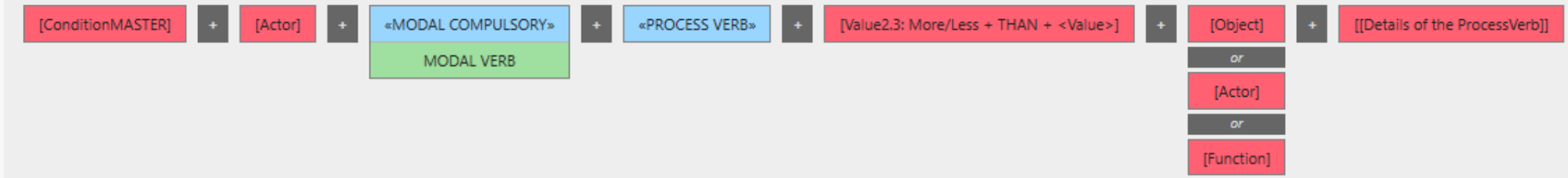
Extracted from : Chris Rupp and die SOPHISTen. Schablonen für alle Fälle. SOPHIST GmbH, 2014

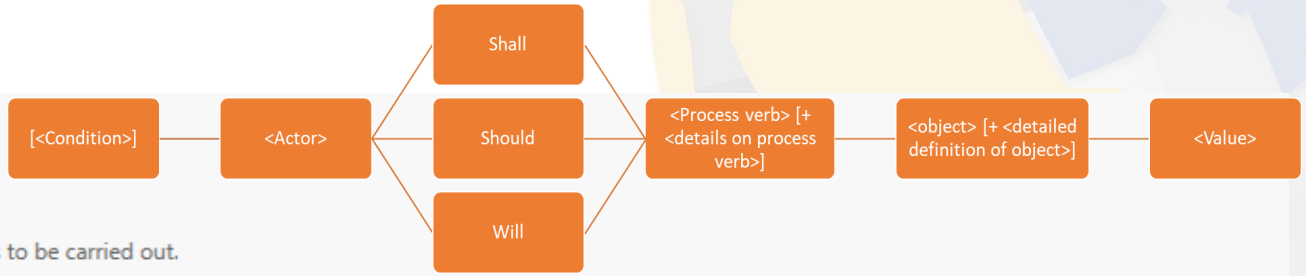
Pattern group(s):
- 01 - SOPHIST (391)

Example:
- The contractor shall prepare an operating manual for the MP3 player.
- The contractor shall update the risk list weekly.

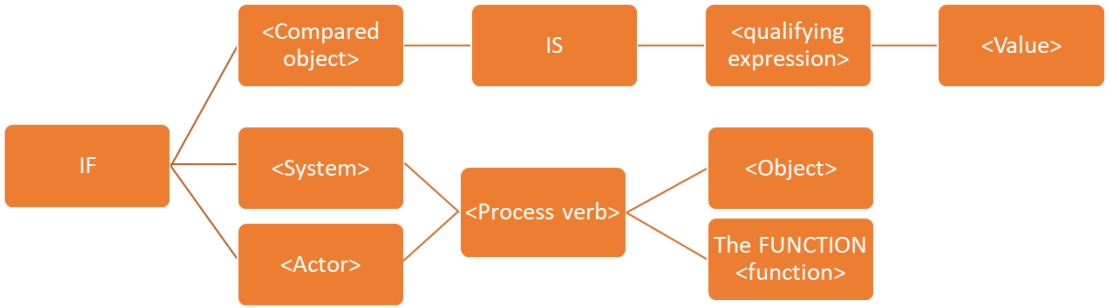
Indexable:	Enabled:	Weight:
Yes	Yes	1,300

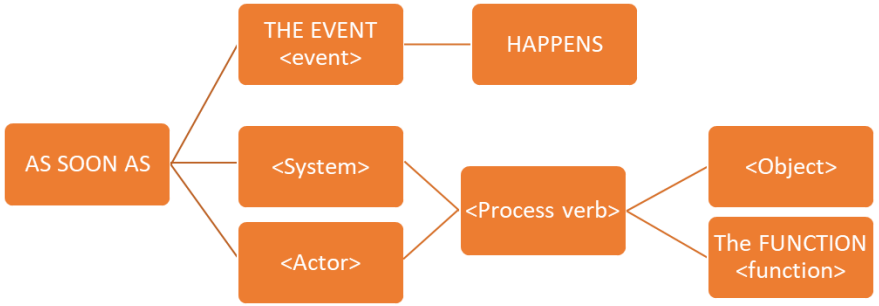
Syntax:

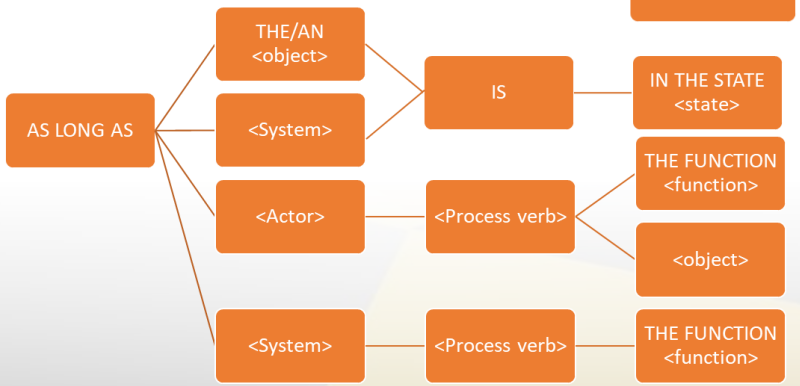




SOPHIST MASTER patterns : Conditional MASTER patterns







LogicMASTER

If the temperature is below -10 °c, …
If the librarian deletes the database, …

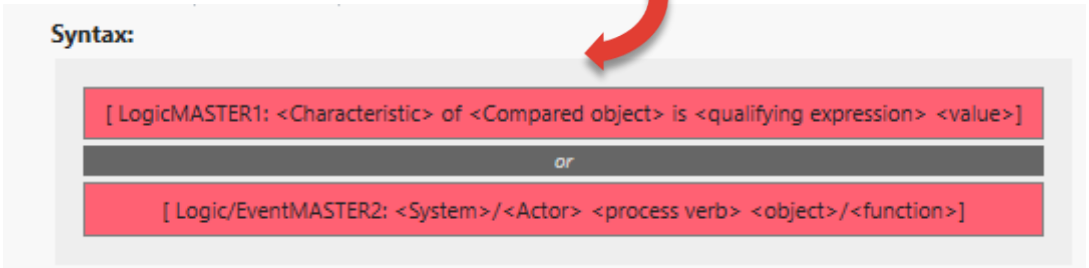
EventMASTER

As soon as the event Evacuation happens, …
As soon as the librarian activates the function Register customer, …

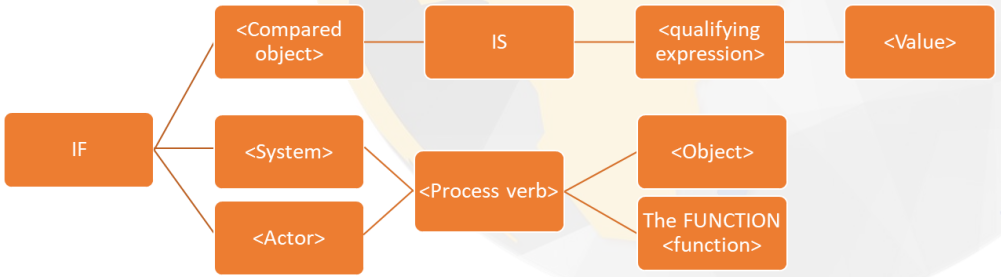
TimeMASTER

As long as the smartphone is in the state Low Battery, …
As long as the customer borrows a book from the library,…

SOPHIST MASTER patterns : Example of structure with LogicMASTER

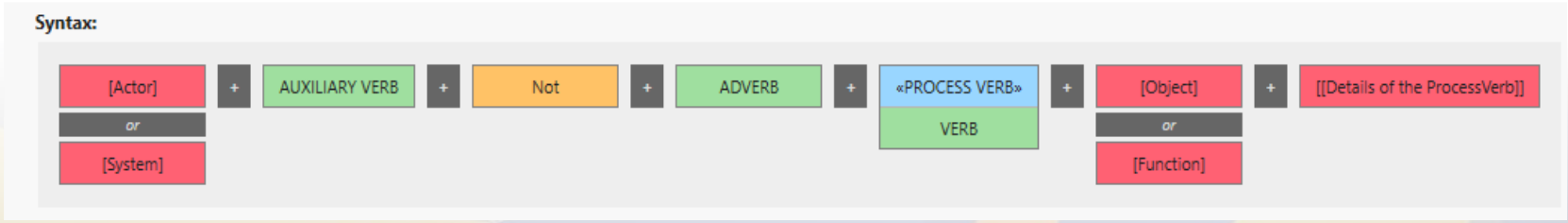


LogicMASTER1



OR

LogicMASTER2

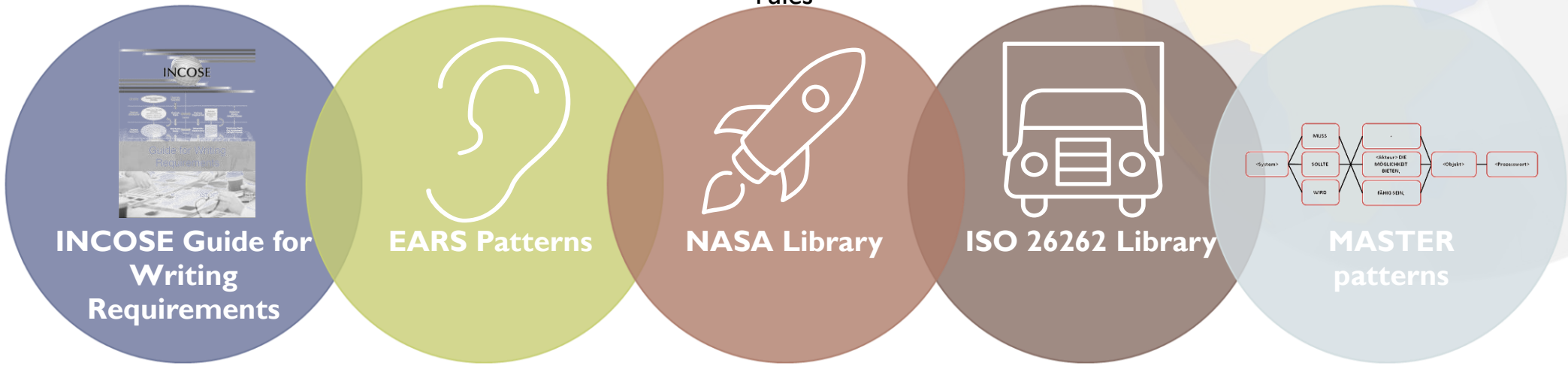


Concept of knowledge libraries in TRC tools

- A combination of knowledge items,
  - of different nature,
  - at different levels of abstraction
- Representing (or not) a specific business domain or **area of knowledge**
- With the aim of improving the way projects are managed, including:
  - the promotion of the principle: **quality** *right the first time,*
  - enabling semantic search portals to archive and retrieve assets,
  - thus providing tools to **reuse** assets at different level,
  - and reducing **time** to market,
  - improving the way engineers generate (**author**) new assets,
  - enhancing the way items are inspected and **verified,**
  - Enabling real **interoperability** mechanisms and services,
  - reducing **time** to elaborate documents, systems and projects.

Knowledge Libraries

NASA
Glossary, patterns and rules



INCOSE
Quality rules for the analysis of textual requirements

EARS
Requirements patterns

ISO 26262
Glossary, patterns and rules

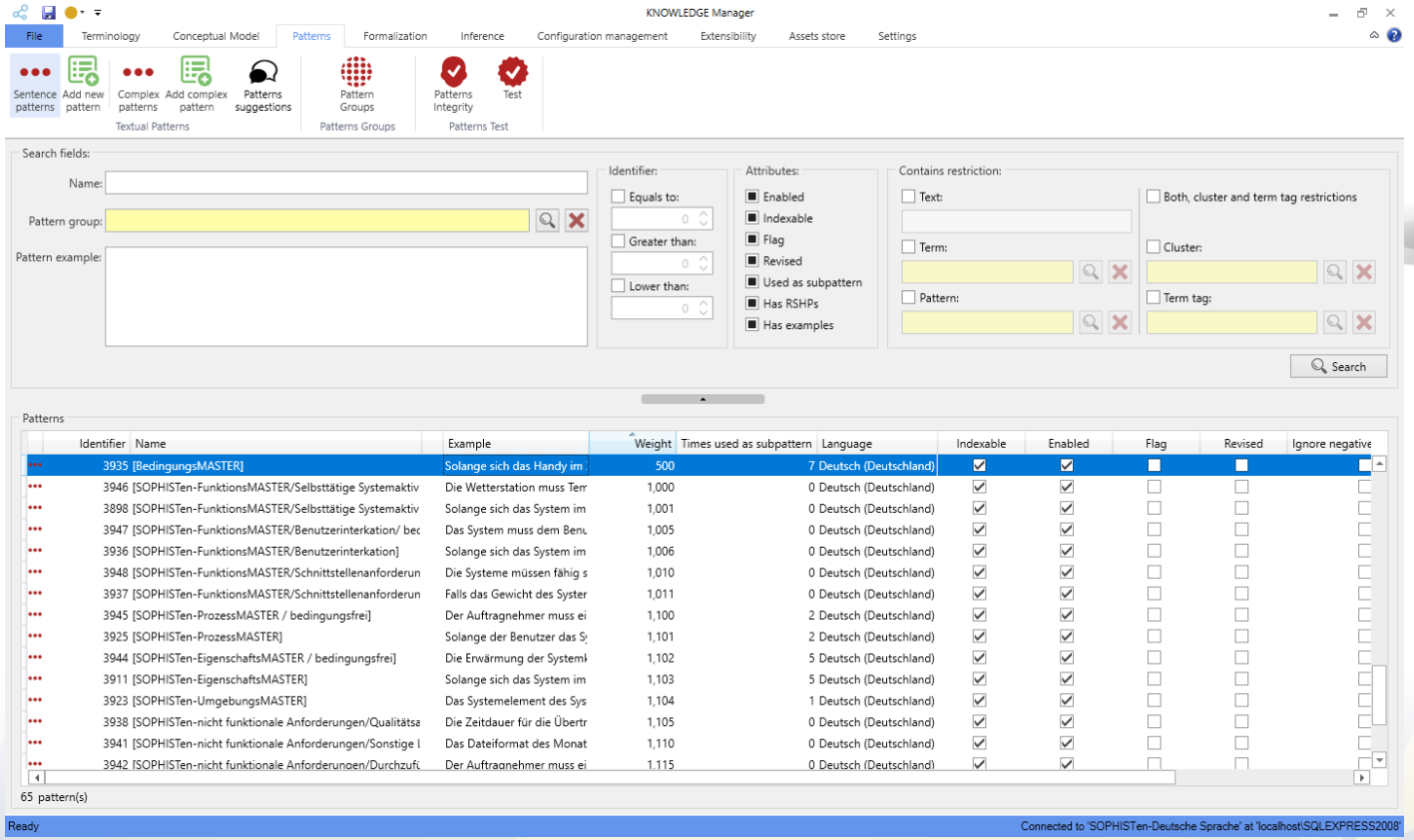
MASTER
patterns

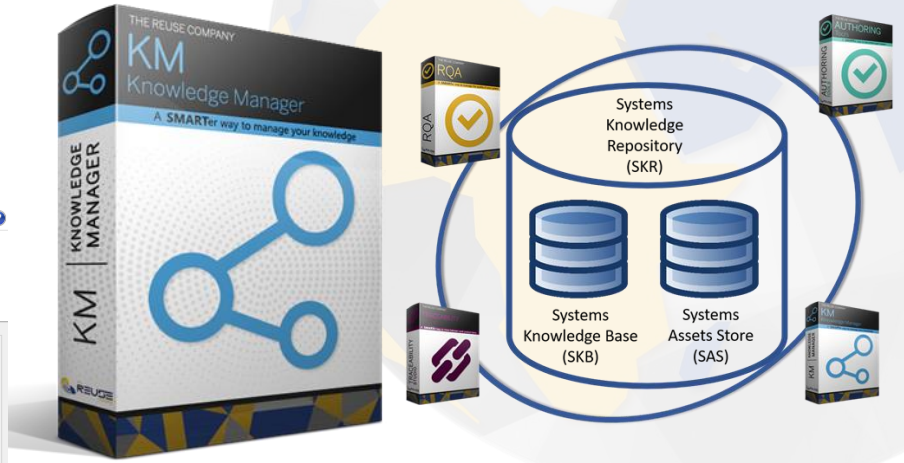
Knowledge Base

Users

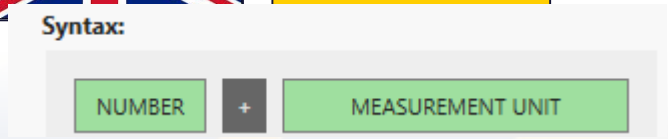
Management of the knowledge base to create the patterns

- Using the **SES Knowledge Manager (KM)**

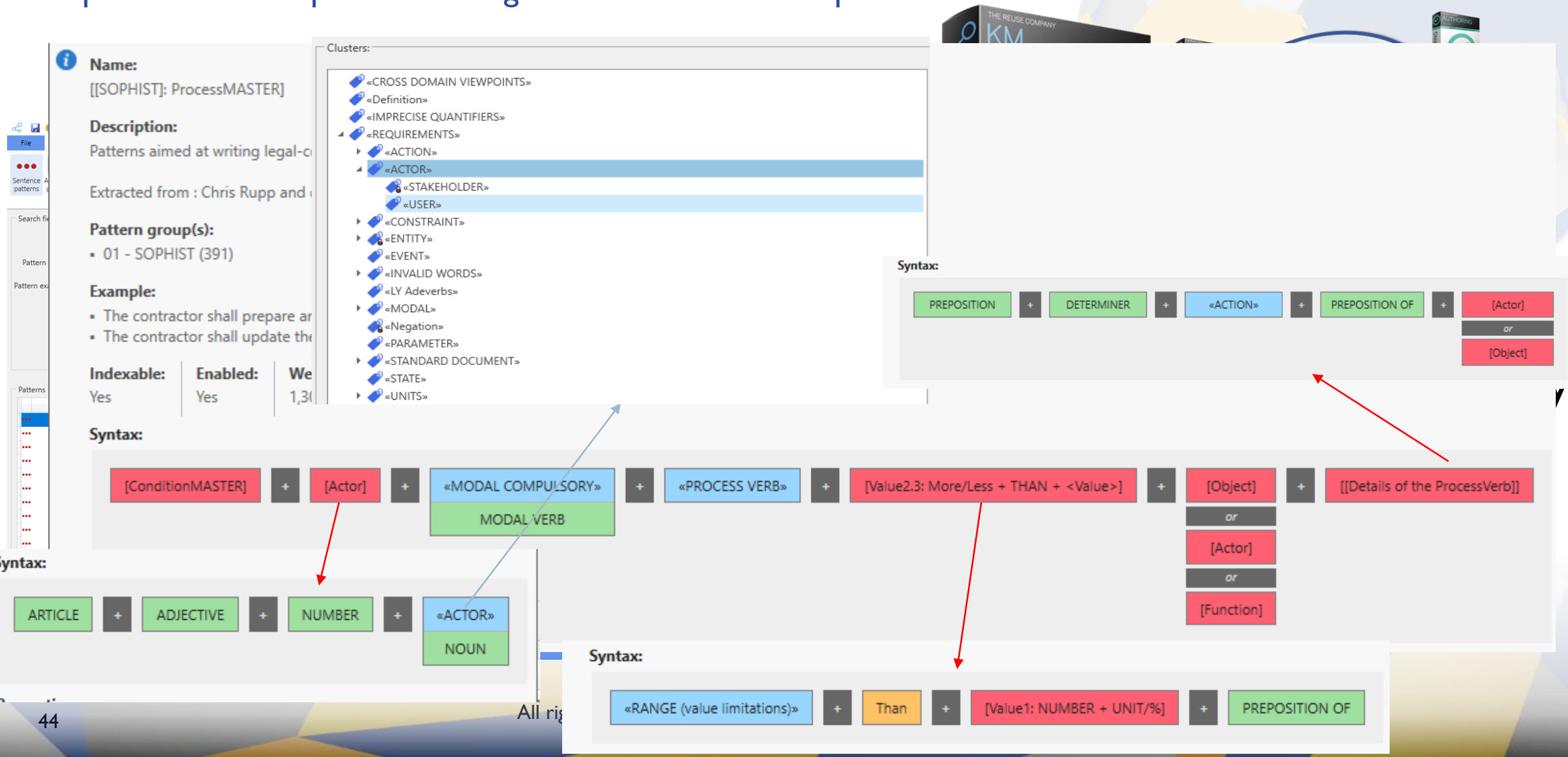




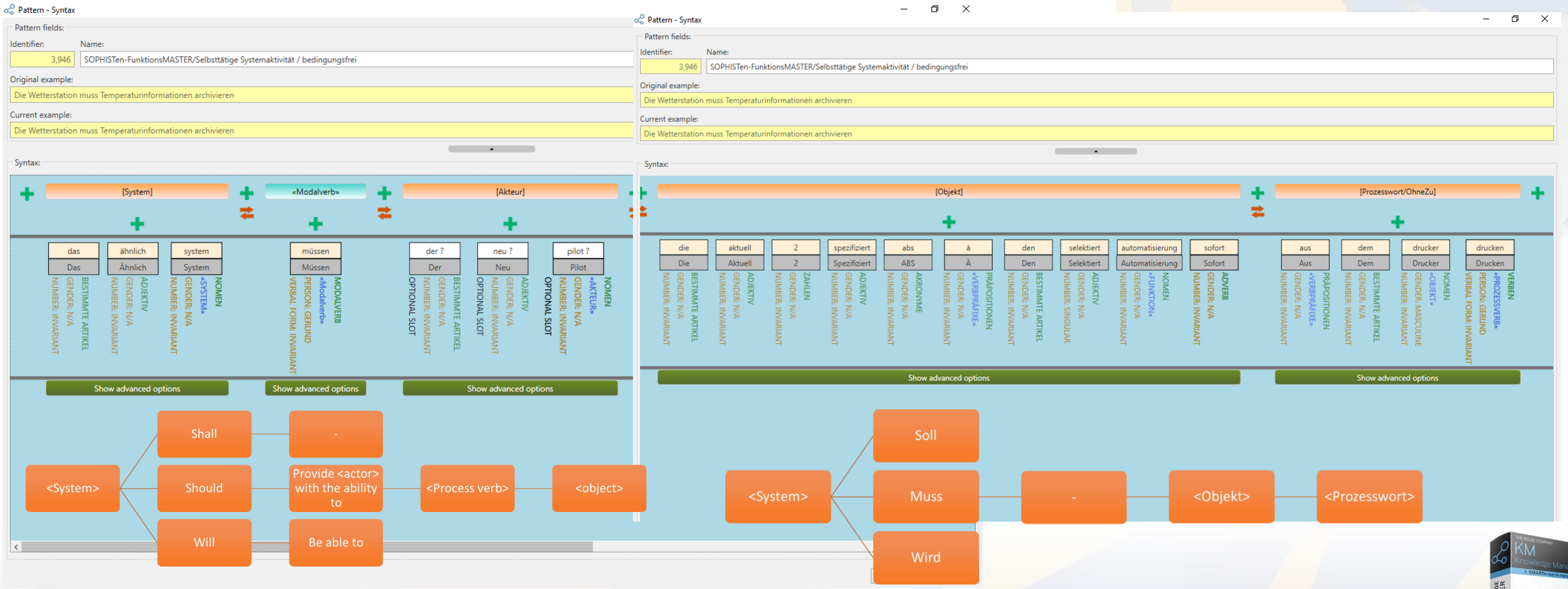
Available languages of the Sophist library



Implementation of patterns : using different levels of sub-patterns and semantic clusters



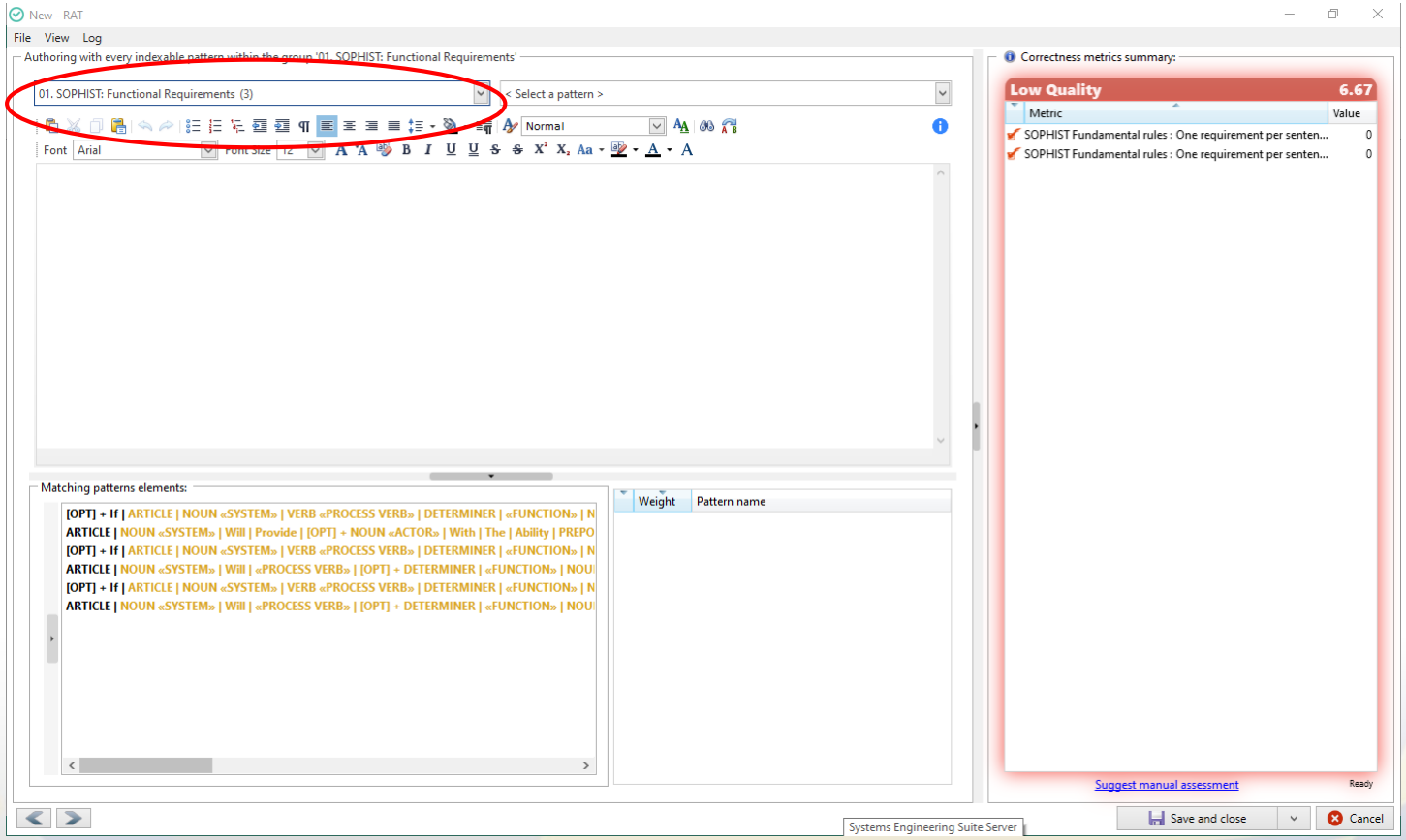
Example of view in KM Pattern wizard : Configuration of paths and optional terms



Following patterns in the Authoring tool

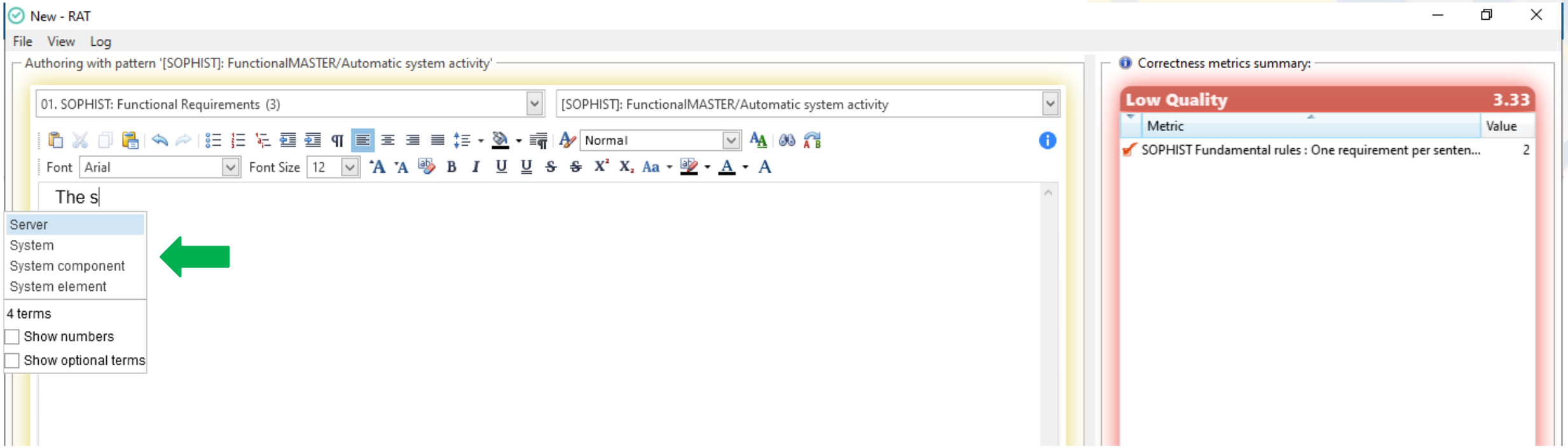
Authoring with the SOPHIST library

- Using the **SES Rich Authoring Tool (RAT)**



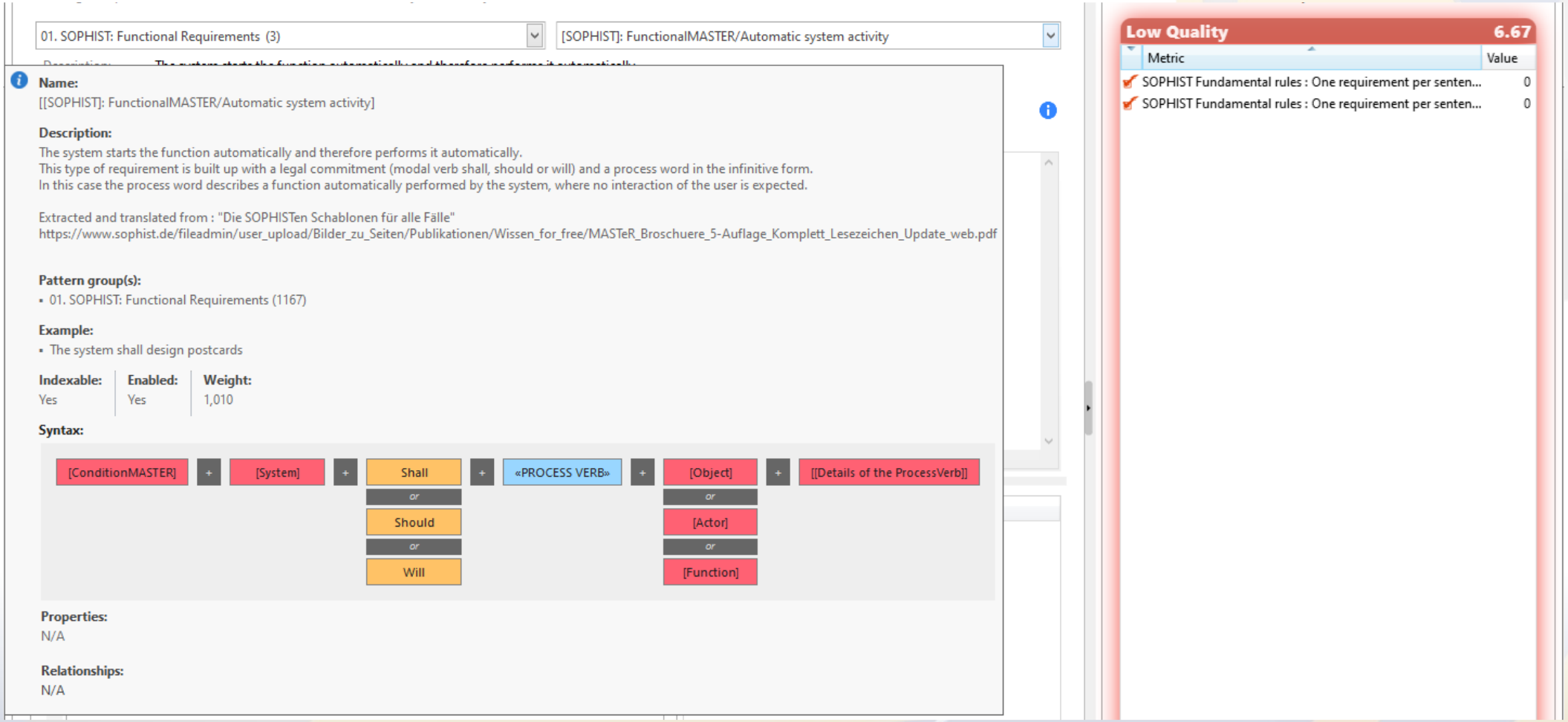
Authoring with SOPHIST patterns

- "on the fly" writing assistance to help following the pattern :



Authoring with SOPHIST patterns

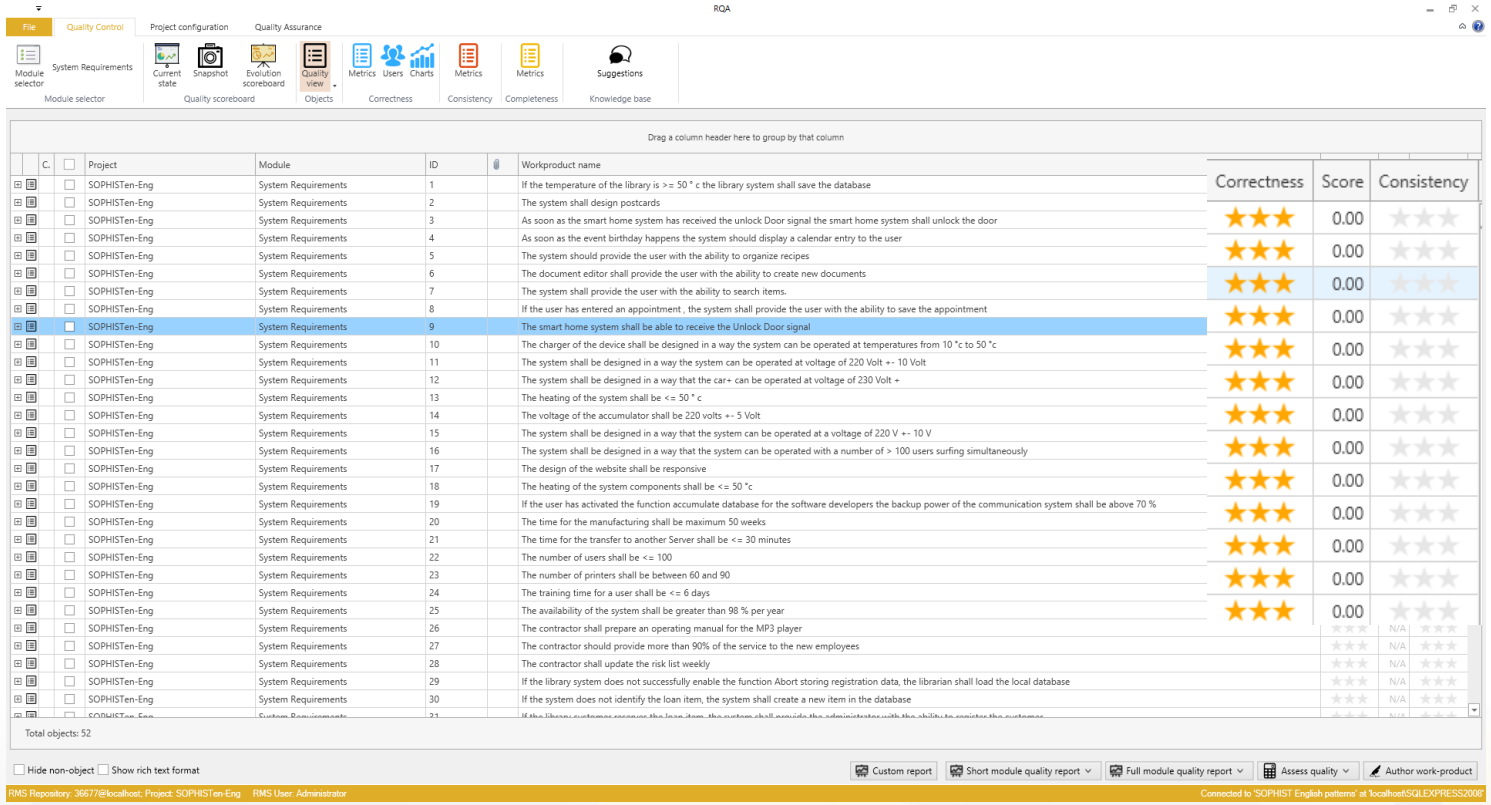
- Display of the pattern when moving the pointer of the mouse on the selected pattern to be followed




Implementing the
SOPHIST RE-Rules

Measuring quality with the SOPHIST library

- Using the **SES Requirements Quality Analyzer (RQA)**



Quality assessment of the requirements using a configured set of rules

The SOPHIST RE-Rules

- SOPHIST RE-Rules : Sophist 18 rules to enhance requirements documentation and enable requirements verification
- 3 priority levels
- Implementation into the CCC model of the TRC Systems Engineering Suite RQA (quality metrics)

The SOPHIST RE-Rules

Priority HIGH

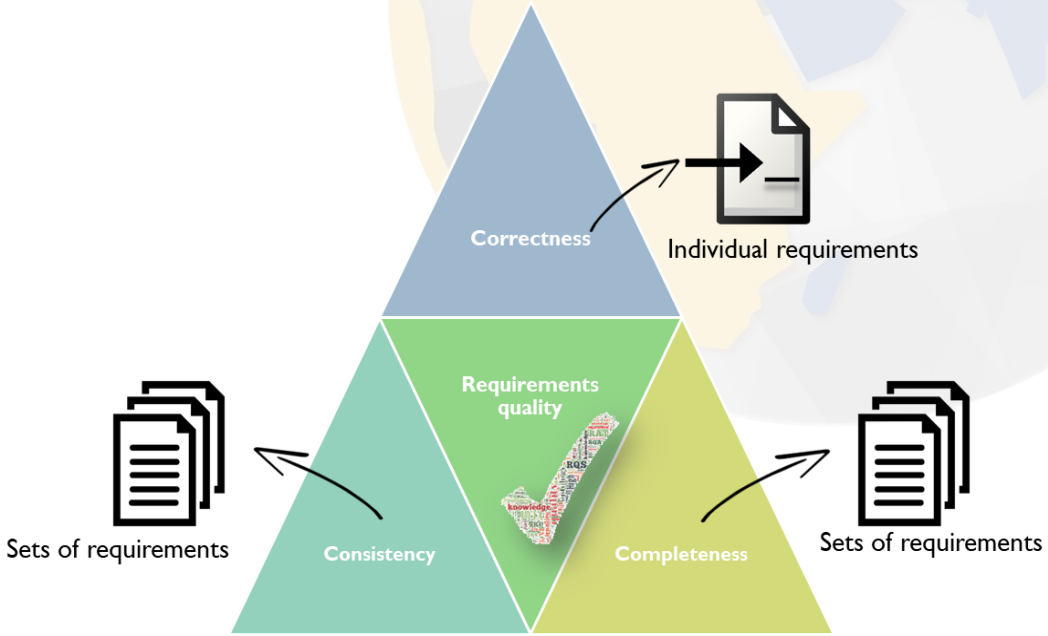
Resolve nominalizations that are not exactly defined and write one or several new requirements with a "good" main verb for every nominalization.	3
Ask wh-questions about the main verb.	6
Question vague nouns.	12
Requirements with incomplete conditional structures should be checked and formulated or described by another requirement.	17
Write one or more requirements for every implicit assumption not described.	18

Priority MEDIUM

Write every requirement in active voice.	1
Express every process using unambiguous main verbs.	2
Write exactly one requirement statement for each main verb.	5
Analyze missing information on the adjective or adverb which is derived from a process verb and add information if necessary.	7
Formulate adjectives in a way that can be measured or tested.	8
Question the used numerals and quantifiers.	10
Clarify missing numerals and quantifiers.	11
Analyze exceptions to the usual behavior of the system and extend the requirement resp. write an additional requirement.	16

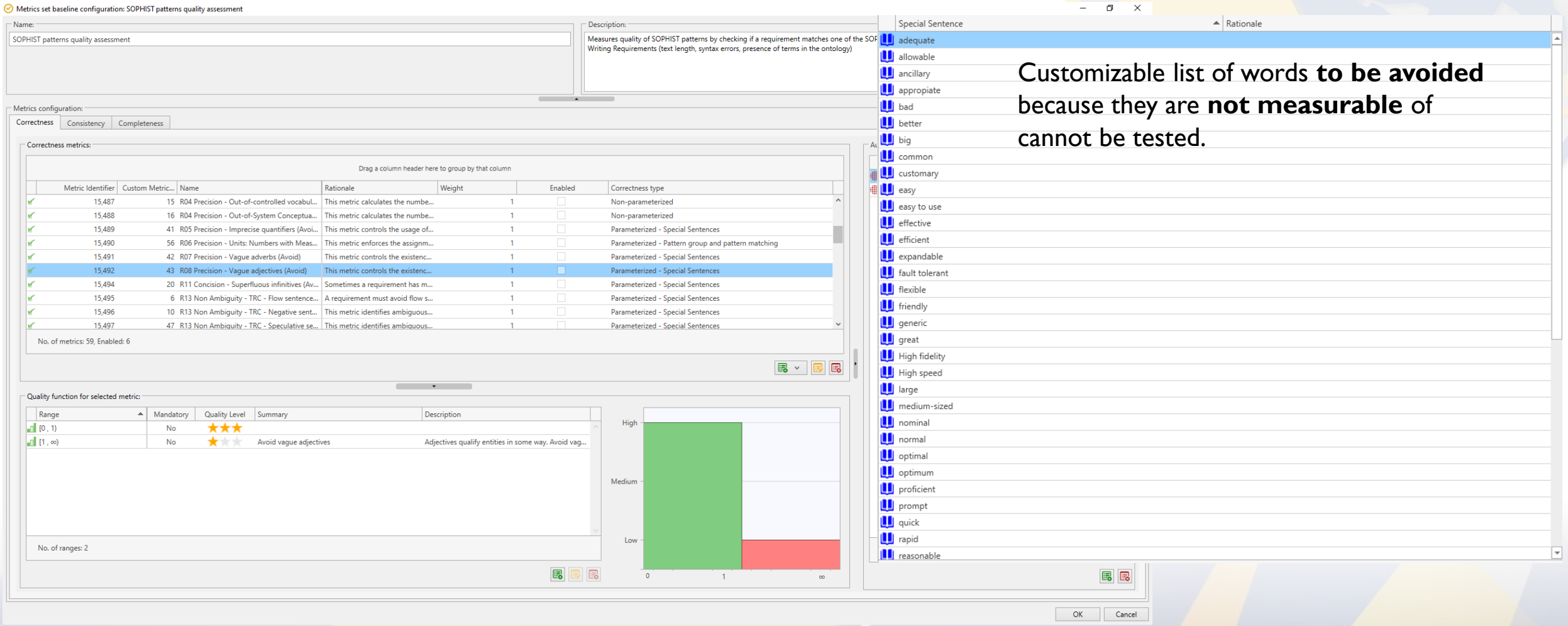
Priority LOW

Dissolve light-verb constructions and define the required process, which represents the system's behavior using a "good" main verb.	4
Formulate separate requirements for non-functional aspects if these aspects are independent or needed as a constraint for several functionalities.	9
Replace formulations that describe possible or impossible situations.	13
Remove subordinate clauses that contain irrelevant information for the requirement.	14
Shorten or eliminate flowery expressions or phrases that are irrelevant for your requirement.	15



The implementation of the Sophist RE-Rules into the TRC tools consists in converting each rule into **quantifiable sets of metrics** in order to create quality assessment baseline.

Implementation inside RQA : Example of Rule #8 : Formulate adjectives in a way that can be measured or tested



Customizable list of words **to be avoided** because they are **not measurable** of cannot be tested.

SOPHIST RE-Rules vs. TRC metrics mapping

SOPHIST

Mapping SOPHIST rules with TRC tool metrics

the REUSE company
http://www.reusecompany.com

(*1) Thanks to intelligent searches, using pattern matching, our tool can differentiate the section of the requirement where the rule is applied

# rule	Priority	Rule Description	TRC interpretation / Comments	Metrics	Metric type	Metric description
1	Medium	Write every requirement in active voice	This metric controls the existence of passive voice.	TRC-M-02	Parameterized - Pattern matching	Passive voice (Avoid). Intelligent passive detection (*1).
2	Medium	Express every process using unambiguous main verbs.	This metric controls the use of vague verbs in the requirement statement.	TRC-M-04	Parameterized - Cluster	Avoid vague verbs for technical documents (Avoid) type: support, process, handle, track, manage, flag, safe...
3	High	Resolve nominalizations that are not exactly defined and write one or several new requirements with a "good" main verb for every nominalization.	This metric controls the use of compounded actions in the same requirement statement.	TRC-M-33	Parameterized - Pattern matching	Write a single sentence that contains a single thought conditioned and qualified by relevant sub-clauses. Number of <shall>
				TRC-M-04	Parameterized - Cluster	Avoid vague verbs for technical documents (Avoid) type: support, process, handle, track, manage, flag, safe...
4	Low	Dissolve light-verb constructions and define the required process, which represents the system's behaviour using a "good" main verb	This metric enforces the use of adequate action verb depending of the target document. A pattern get the: shall + <action>	TRC-M-08	Parameterized - Pattern matching	Specific pattern to force the main verb is a controlled action verb: shall + <action>
5	Medium	Write exactly one requirement statement for each main verb	Write a single, simple, single-thought, affirmative, declarative sentence, conditioned and qualified by relevant sub-clauses.	TRC-M-33	Parameterized - Pattern matching	Write a single sentence that contains a single thought conditioned and qualified by relevant sub-clauses. Number of <shall>
				TRC-M-28	Parameterized - Pattern matching	This metric checks that the requirement statement is written following an acceptable, pre-defined sequence of clauses, composed in different Requirement Patterns, and grouped in a Patterns Group
				TRC-M-32	Parameterized - Cluster	Count the number of action verbs (1 : High quality, 2-4 : Medium ; >4 : Low) except when analysing conditions
				TRC-M-34	Parameterized - Pattern matching	The presence or combinators in a requirement usually indicates that multiple requirements should be written. Too many combinators must be avoided in a requirement out of the conditions. Intelligent detection (*1).
6	High	Ask wh-questions about the main verb	This metric calculates if the requirement specification matches any of the configured set of patterns or pattern groups.	TRC-M-57	Parameterized - Pattern matching	Style guide (Enforce) check if the requirement specification matches any of the configured set of patterns
7	Medium	Analyse missing information on the adjective or adverb which is derived from a process verb and add information if necessary.	The requirement matches configured set of patterns	TRC-M-60	Parameterized - Pattern matching	Pattern (Enforce) check if the requirement specification matches a specific pattern
8	Medium	Formulate adjectives in a way that can be measured or tested	This metric controls the existence of vague adjectives in the requirement statement. Adjectives qualify entities (Agents) in some way. Avoid vague adjectives.	TRC-M-14	Parameterized - Special Sentences	Avoid vague adjectives (Avoid) type: "relevant", "common", "generic", "flexible", "typical", "sufficient", "adequate", "efficient", "effective"...
9	Low	Formulate separate requirements for non-functional aspects if these aspects are independent or needed as a constraint for several functionalities	The requirement matches any of the configured set of patterns	TRC-M-57	Parameterized - Pattern matching	Style guide (Enforce) check if the requirement specification matches any of the configured set of patterns
10	Medium	Question the used numerals and quantifiers.	This metric checks the existence of ambiguous Universal keywords in the requirement statement	TRC-M-49	Parameterized - Cluster	Quantifiers - Ambiguous Universal Keywords (Avoid)
				TRC-M-60	Parameterized - Pattern matching	Pattern (Enforce) check if the requirement specification matches a specific pattern
11	Medium	Clarify missing numerals and quantifiers.	Ensure adequate used of quantities and units	TRC-M-11	Parameterized - Pattern group and pattern matching	This metric enforces the assignment of Measurement Units or noun qualifications to all numbers in a requirement statement.
				TRC-M-13	Pattern matching and consistency	Detect inadequate unit for a characteristic
				TRC-M-12	Parameterized - Pattern group and pattern matching	Avoid use of different unit systems for the same characteristic
				TRC-M-61	Parameterized - Pattern matching	Metric based on pattern and parameter. Force the tolerance value for the unit, members of a cluster of units that required tolerances.
				TRC-M-50	Parameterized - Pattern matching	Metric based on pattern and parameter for Tolerance/Value range applied to a cluster of units that required

SOPHIST RE-Rules vs. TRC metrics mapping

Mapping SOPHIST rules with TRC tool metrics

SOPHIST

the REUSE company
http://www.reusecompany.com

(*1) Thanks to intelligent searches, using pattern matching, our tool can differentiate the section of the requirement where the rule is applied

# rule	Priority	Rule Description	TRC interpretation / Comments	Metrics	Metric type	Metric description
1	Medium	Write every requirement in active voice	This metric controls the existence of passive voice.	TRC-M-02	Parameterized - Pattern matching	Passive voice (Avoid). Intelligent passive detection (*1).
2	Medium	Express every process using unambiguous main verbs.	This metric controls the use of vague verbs in the requirement statement.	TRC-M-04	Parameterized - Cluster	Avoid vague verbs for technical documents (Avoid) type: support, process, handle, track, manage, flag, safe...
3	High	Resolve nominalizations that are not exactly defined and write one or several new requirements with a "good" main verb for every nominalization.	This metric controls the use of compounded actions in the same requirement statement.	TRC-M-33	Parameterized - Pattern matching	Write a single sentence that contains a single thought conditioned and qualified by relevant sub-clauses. Number of <shall>
				TRC-M-04	Parameterized - Cluster	Avoid vague verbs for technical documents (Avoid) type: support, process, handle, track, manage, flag, safe...
4	Low	Dissolve light-verb constructions and define the required process, which represents the system's behaviour using a "good" main verb	This metric enforces the use of adequate action verb depending of the target document. A pattern get the: shall + <action>	TRC-M-08	Parameterized - Pattern matching	Specific pattern to force the main verb is a controlled action verb: shall + <action>
5	Medium	Write exactly one requirement statement for each main verb	Write a single, simple, single-thought, affirmative, declarative sentence, conditioned and qualified by relevant sub-clauses.	TRC-M-33	Parameterized - Pattern matching	Write a single sentence that contains a single thought conditioned and qualified by relevant sub-clauses. Number of <shall>
				TRC-M-28	Parameterized - Pattern matching	This metric checks that the requirement statement is written following an acceptable, pre-defined sequence of clauses, composed in different Requirement Patterns, and grouped in a Patterns Group
				TRC-M-32	Parameterized - Cluster	Count the number of action verbs (1 : High quality, 2-4 : Medium ; >4 : Low) except when analysing conditions
				TRC-M-34	Parameterized - Pattern matching	The presence or combinators in a requirement usually indicates that multiple requirements should be written. Too many combinators must be avoided in a requirement out of the conditions. Intelligent detection (*1).
6	High	Ask wh-questions about the main verb	This metric calculates if the requirement specification matches any of the configured set of patterns or pattern groups.	TRC-M-57	Parameterized - Pattern matching	Style guide (Enforce) check if the requirement specification matches any of the configured set of patterns
7	Medium	Analyse missing information on the adjective or adverb which is derived from a process verb and add information if necessary.	The requirement matches configured set of patterns	TRC-M-60	Parameterized - Pattern matching	Pattern (Enforce) check if the requirement specification matches a specific pattern
8	Medium	Formulate adjectives in a way that can be measured or tested	This metric controls the existence of vague adjectives in the requirement statement. Adjectives qualify entities (Agents) in some way. Avoid vague adjectives.	TRC-M-14	Parameterized - Special Sentences	Avoid vague adjectives (Avoid) type: "relevant", "common", "generic", "flexible", "typical", "sufficient", "adequate", "efficient", "effective"...
9	Low	Formulate separate requirements for non-functional aspects if these aspects are independent or needed as a constraint for several functionalities	The requirement matches any of the configured set of patterns	TRC-M-57	Parameterized - Pattern matching	Style guide (Enforce) check if the requirement specification matches any of the configured set of patterns
10	Medium	Question the used numerals and quantifiers.	This metric checks the existence of ambiguous Universal keywords in the requirement statement	TRC-M-49	Parameterized - Cluster	Quantifiers - Ambiguous Universal Keywords (Avoid)
				TRC-M-60	Parameterized - Pattern matching	Pattern (Enforce) check if the requirement specification matches a specific pattern
11	Medium	Clarify missing numerals and quantifiers.	Ensure adequate used of quantities and units	TRC-M-11	Parameterized - Pattern group and pattern matching	This metric enforces the assignment of Measurement Units or noun qualifications to all numbers in a requirement statement.
				TRC-M-13	Pattern matching and consistency	Detect inadequate unit for a characteristic
				TRC-M-12	Parameterized - Pattern group and pattern matching	Avoid use of different unit systems for the same characteristic
				TRC-M-61	Parameterized - Pattern matching	Metric based on pattern and parameter. Force the tolerance value for the unit, members of a cluster of units that required tolerances.
				TRC-M-50	Parameterized - Pattern matching	Metric based on pattern and parameter for Tolerance/Value range applied to a cluster of units that required

SOPHIST RE-Rules vs. TRC metrics mapping

Mapping SOPHIST rules with TRC tool metrics

(*1) Thanks to intelligent searches, using pattern matching, our tool can differentiate the section of the requirement where the rule is applied

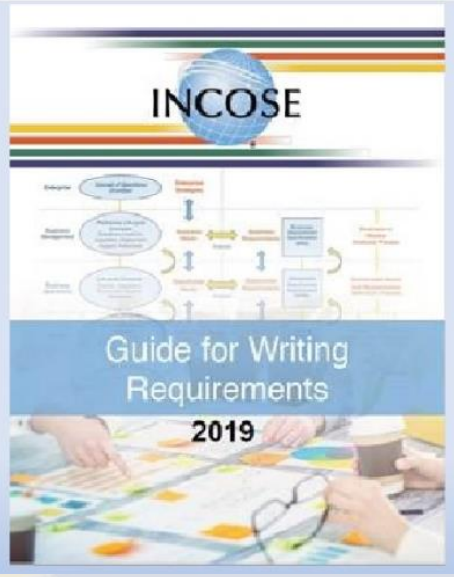
# rule	Priority	Rule Description	TRC interpretation / Comments	Metrics	Metric type
1	Medium	Write every requirement in active voice	This metric controls the existence of passive voice.	TRC-M-02	Parameterized - Pattern matching
2	Medium	Express every process using unambiguous main verbs.	This metric controls the use of vague verbs in the requirement statement.	TRC-M-04	Parameterized - Cluster
3	High	Resolve nominalizations that are not exactly defined and write one or several new requirements with a “good” main verb for every nominalization.	This metric controls the use of compounded actions in the same requirement statement.	TRC-M-33	Parameterized - Pattern matching
				TRC-M-04	Parameterized - Cluster
4	Low	Dissolve light-verb constructions and define the required process, which represents the system’s behaviour using a “good” main verb	This metric enforces the use of adequate action verb depending of the target document. A pattern get the: shall + <action>	TRC-M-08	Parameterized - Pattern matching
5	Medium	Write exactly one requirement statement for each main verb	Write a single, simple, single-thought, affirmative, declarative sentence, conditioned and qualified by relevant sub-clauses.	TRC-M-33	Parameterized - Pattern matching
				TRC-M-28	Parameterized - Pattern matching
				TRC-M-32	Parameterized - Cluster
				TRC-M-34	Parameterized - Pattern matching
11	Medium	Clarify missing numerals and quantifiers.	requirement statement Ensure adequate used of quantities and units	TRC-M-60	Parameterized - Pattern matching	Pattern (Enforce) check if the requirement specification matches a specific pattern
				TRC-M-11	Parameterized - Pattern group and pattern matching	This metric enforces the assignment of Measurement Units or noun qualifications to all numbers in a requirement statement.
				TRC-M-13	Pattern matching and consistency	Detect inadequate unit for a characteristic
				TRC-M-12	Parameterized - Pattern group and pattern matching	Avoid use of different unit systems for the same characteristic
				TRC-M-61	Parameterized - Pattern matching	Metric based on pattern and parameter. Force the tolerance value for the unit, members of a cluster of units that required tolerances.
				TRC-M-50	Parameterized - Pattern matching	Metric based on pattern and parameter for Tolerance/Value range applied to a cluster of units that required

Live Demo


Q&A

Next webinar

- **INCOSE Guide for Writing Requirements: Real-Time Quality Assessment of the INCOSE Rules**
- While other standards just define a number of nice-to-have, but yet very abstract set of quality characteristics, the **INCOSE Guide for Writing Requirements** also includes a number of comprehensible, and SMART (specific, measurable and easy to automatize), set of quality rules for requirements and for requirements documents.
- No matter what tool you use to manage your requirements, RQA - QUALITY Studio and RAT - Authoring Tools from The REUSE Company offer an easy-to-use library including the quality rules described in the GfWR; allowing both quality control of your existing documents, and real-time help during the authoring stage of the requirements. Writing high-quality requirements has never been so easy.
  - **Dates:**
  - March 24 and 26




Q&A

contact@reusecompany.com