



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

- › Description of The Reuse Company
- › Presenter's profile
- › Why using patterns to increase the quality of requirements ?
- › The MASTER patterns by SOPHIST
- › RAT (Rich Authoring Tool) : Requirements authoring to improve quality – application to the MASTER patterns
- › Implementing the Sophist RE-Rules inside RQA (Requirements Quality Analyzer)
- › Live Demo
- › Q&A

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



REVaMP²

Past

ARTEMIS CRYSTAL
Requirements
Engineering



AMASS
Assurance and Certification of CPS



ARROWHEAD

Current



Celtic+: IoD
Celtic-Plus
Smart Connected World



ITEA3

Future
ITEA3: EMBRACE

New Control



ECSEL JU

T (he) R (euse) Q (ompany)y

Leveraging Systems Engineering activities



TRACEABILITY



QUALITY



REUSABILITY

Who is using our technology?

	Aerospace and defense
	Energy
	Automotive
	Healthcare
	Other industries



Aerospace and defense: AIRBUS DEFENCE & SPACE, AIRBUS GROUP INNOVATIONS, SAFRAN AIRCRAFT ENGINES, THALES, THALES Communications & Security S.A.S., HENSOLDT, aces, arianeGROUP.

Energy: FUSION FOR ENERGY, iter the way to new energy, EDF, REPSOL.

Automotive: RENAULT.

Healthcare: THINK SURGICAL, Health Net[®].

Other industries: 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**.



the
REUSE
company

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



Why using patterns ?

accepting (word
article).

focus n point

converging rays of light,
heat, waves of sound, meet;

centre of activity or
intensity; pl focuses, foci;

adjust; cause to converge;

concentration of force;
pertaining to focus;

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

05 Reasoning

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

04 Formalization

Representation of assets semantic through SRL – System Representation Language



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

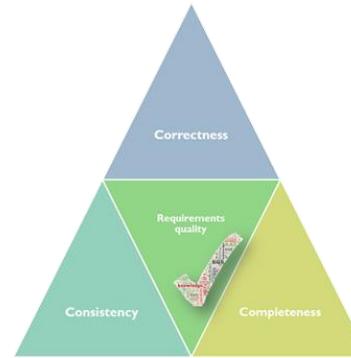
03

Patterns

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

Why using patterns ?

Requirements quality metrics: knowledge needs



Why using patterns ?

SKB – System Knowledge Base

05. Reasoning

04. Formalization

03. Patterns

02. SCM / architectures

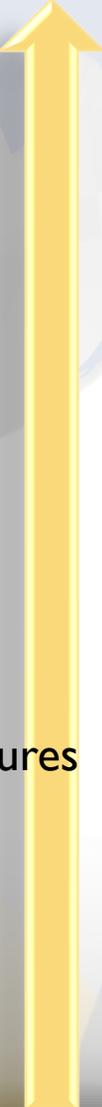
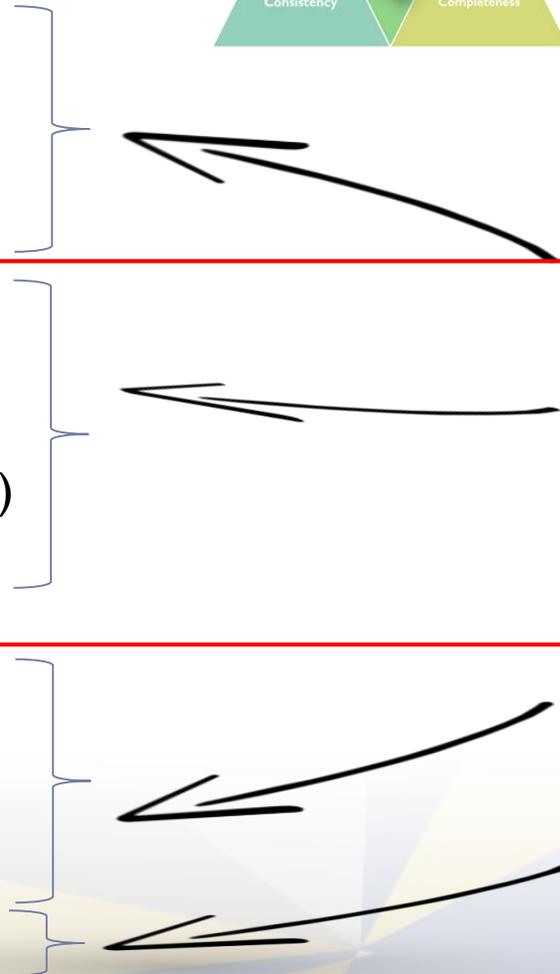
01. Vocabulary

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

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

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

CCC Metrics (most)



Some examples : Dictionary references (acronyms)

The screenshot displays the RAT (Requirements Authoring Tool) interface. The main window shows a requirement titled "02. SOPHIST: Non-functional requirements (6)" with a description: "Quality requirements define the qualitative properties of the system under consideration or of individual functions. These are: changeability, usability, efficiency, functionality, ...". The requirement text is "The availability of the ECU shall be 400000 Kilo|". A dropdown menu is open, listing various units: Kilometre, Kilomole, Kilonewton, Kiloohm, Kilopascal, Kilosecond, Kilosiemens, Kilosievert, and KiloTesla. The menu also includes "30 terms", "Show numbers", and "Show optional terms" (checked).

On the right side, a "Correctness metrics summary" panel is visible, showing a "Low Quality" score of 1.43. A table lists the metrics:

Metric	Value
✓ SOPHIST Fundamental rules : Clear ambiguous references	1

At the bottom of the metrics panel, there is a "Suggest manual assessment" link and a "Ready" status. The main window has a "Save and close" button and a "Cancel" button at the bottom right.

Some examples : enforcing the tolerance when using measurement units

The screenshot shows a software interface for editing a pattern. The main window is titled "Editing 326f1a8e-a044-4b85-823d-00fa197ca90b - RAT". The interface is divided into two main sections: a pattern editor on the left and a correctness metrics summary on the right.

Pattern Editor:

- System: System Physical Requirements (2)
- Pattern Name: Physical property
- Description: Use this pattern to provide a value for a specific property of a system or component. [Maximum/Minimum] <Property> of the <System element> shall be NUMBER <Unit>
- Text: The maximum length of the antenna shall be 12 cm
- Annotation: A tooltip points to the text "12 cm" with the message "Metric: Missing tolerance" and "N/A".

Correctness metrics summary:

Medium Quality		0.56
Metric		Value
Missing tolerance		1

Buttons at the bottom: [Suggest manual assessment](#), [Save and close](#), [Cancel](#)

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

The screenshot displays the TRC software interface with the following components:

- Requirement Editor:** Shows a requirement text "When the ABS is activated, the display shall be shown" with a red box highlighting "be shown". A tooltip indicates the metric: "Metric: R02 Precision - Passive voice (avoid) N/A".
- Correctness metrics summary:** A table showing the overall quality score and specific metrics.

Metric	Value
R02 Precision - Passive voice (avoid)	1
R44 Uniformity Of Language - Style guide (Enforce)	0
- Matching patterns elements:** Shows a pattern rule: "[OPT] + NOUN «STAKEHOLDER» | MODAL VERB «MODAL COMPUL»".
- Other quality elements:** A table listing various quality metrics.

Metric	Correctness	Value	Summary	Mandatory	Weight
R02 Precision - Passive voice (...)	☆☆☆	1	Avoid passive voice	<input checked="" type="checkbox"/>	1
R44 Uniformity Of Language - ...	☆☆☆☆	0	Requirement's structure should be conforming with an...	<input type="checkbox"/>	0
Avoid connectors	☆☆☆☆	0	N/A	<input type="checkbox"/>	1
Check of possible Values for a g...	☆☆☆☆	0	N/A	<input type="checkbox"/>	1
Pattern group: METRIC - Anti-P...	☆☆☆☆	0	N/A	<input type="checkbox"/>	1
Properties measured with the ri...	☆☆☆☆	0	N/A	<input type="checkbox"/>	1
R02 Precision - TRC - Condition...	☆☆☆☆	0	N/A	<input type="checkbox"/>	1
R02 Precision - TRC - Imperativ...	☆☆☆☆	1	N/A	<input type="checkbox"/>	1
R05 Precision - Imprecise quant...	☆☆☆☆	0	N/A	<input type="checkbox"/>	1
R05 Precision - Imprecise quant...	☆☆☆☆	0	N/A	<input type="checkbox"/>	1
- Bottom Panel:** Includes a "Save and close" button and a "Cancel" button.

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

Requirements specification incompleteness
20.00% of the properties selected in the metric are not covered in the specification

Metric configuration incompleteness
86.21% of the properties found in the specification have not been selected in the metric

Properties coverage metric results - Power consumption allocation metric

This metric results are obsolete due to changes in the specification. Please recalculate the metric to update the results

Properties selected in the metric vs Properties found in the specification:

- Properties selected in the metric
- Properties selected in the metric not found in the requirements specification
- Properties selected in the metric found in the requirements specification
- Properties selected in the metric found in the requirements specification without value
- Properties found in the specification but not selected in the metric
- Properties found in the specification without value and not selected in the metric
- All properties found in the specification

Properties selected in the metric not found in the requirements specification:

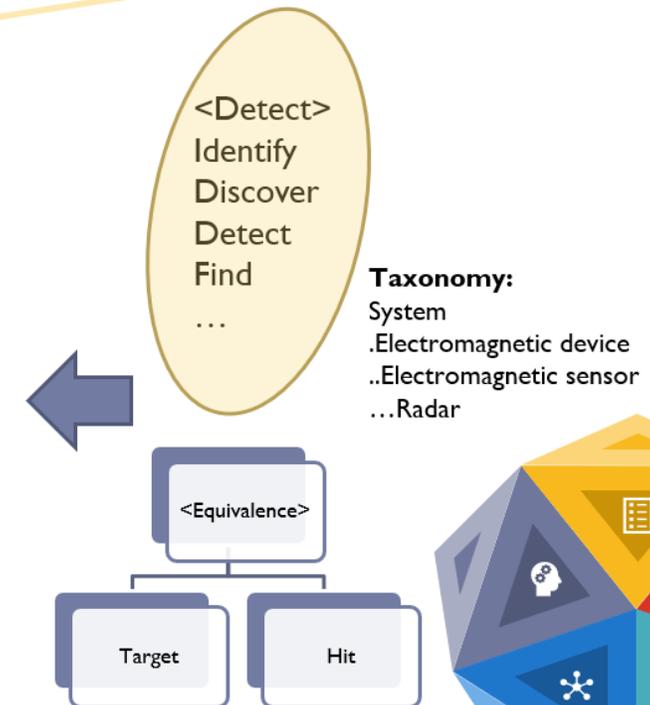
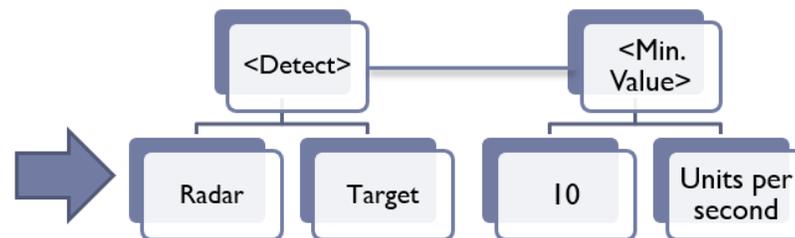
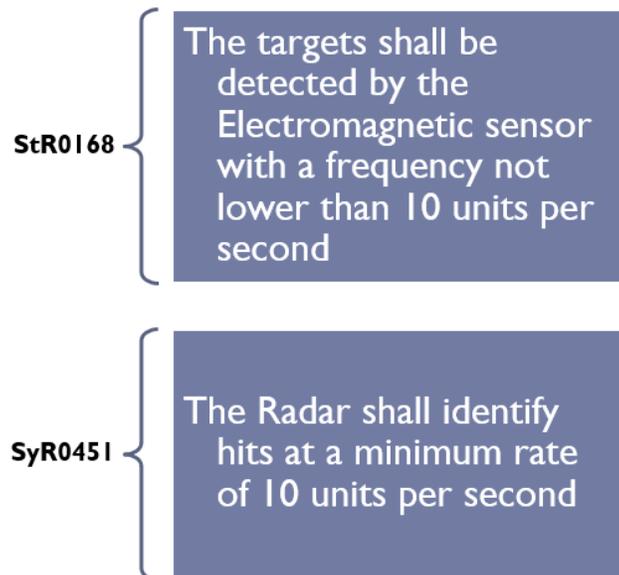
Properties	Instances in the specification
{ Air conditioning system } { Power consumption } { W }	0

Total: 1

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

StR0168 - The targets shall be detected by the Electromagnetic sensor with a frequency not lower than 10 units per second

SyR0451 - The Radar shall identify hits at a minimum rate of 10 units per second



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

➤ Automatic extraction of requirements from unstructured textual sources



When / After / If ... [Condition] <Subject> Shall <Action> <Object> [Constraint]

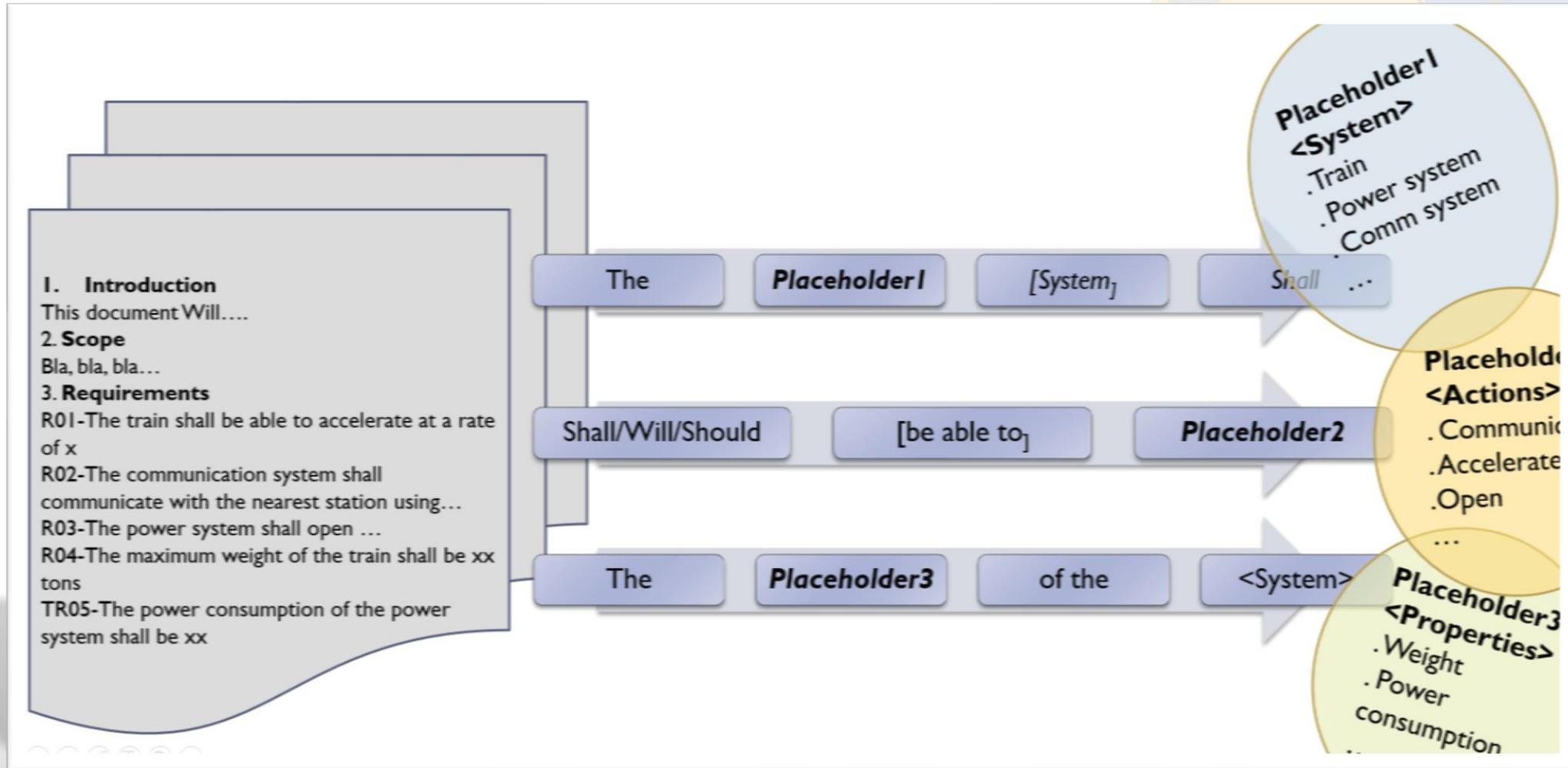
<Entity> Shall Have aNUMBER <Entity>

The <Property> Of <Entity> Shall be <Value>

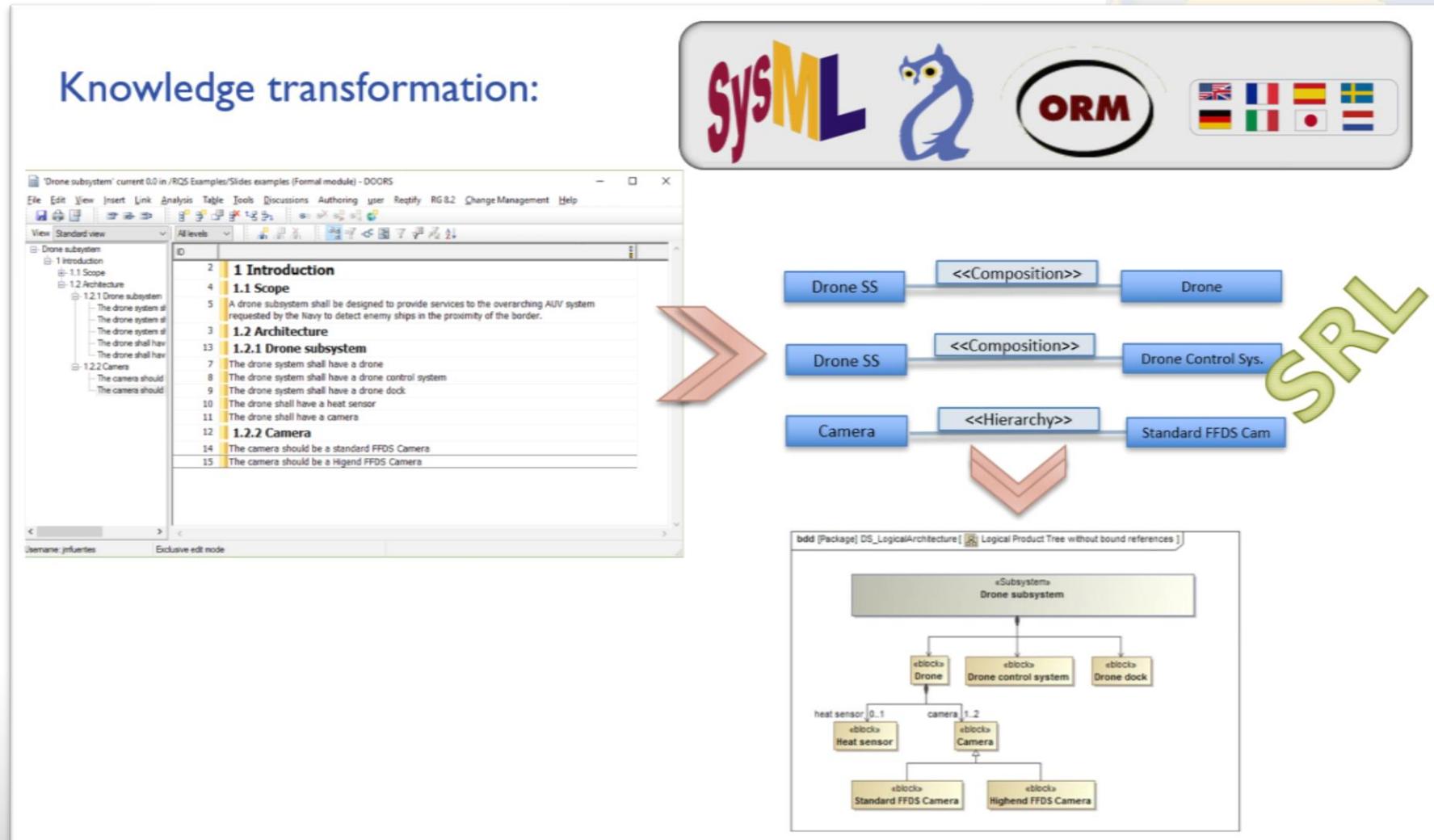


... and more

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



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 :

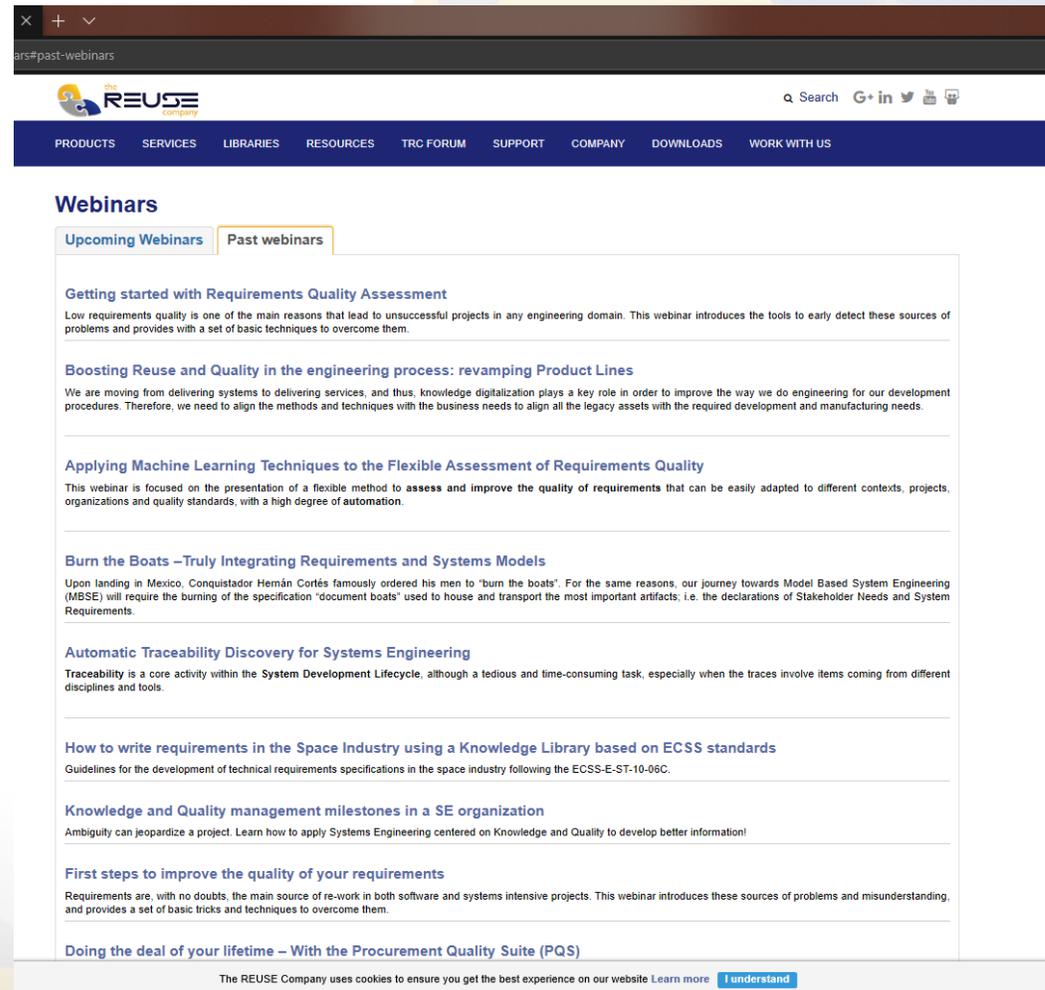
The screenshot displays the 'Requirements Authoring Tool Plugin for DOORS' interface. The main window shows a 'Patterns' tab with a search field and a list of patterns. A context menu is open over a pattern, with the option 'Suggest the whole requirement as a new pattern' circled in red. The interface also includes a 'Correctness metrics summary' panel on the right and a 'Other quality elements' panel at the bottom.

Identifier	Name	Example	Weight	Times used as subpattern	Indexable	Enabled
702	[<Condition> <System> <Action> <Agent>]	when the road be freeze , the	1,200	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
700	[<System> <Action> <Agent>]	the aircraft shall start the flight	1,130	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
699	[<Stakeholder> <Action> <Agent>]	the pilot shall fly the aircraft	1,110	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
698	[<Stakeholder>]	the pilot	1,101	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
695	[<System> <Action>]	the aircraft shall fly	1,100	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
704	[Condition]	when the road be freeze	250	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
703	[Condition 2]	when the car detect accessing :	220	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
701	[Condition 1]	when the road be freeze	210	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Metric	Correctness	Value	Summary
Term tag: VERB	☆☆☆	3	Too many verbs
R47 - Style guide	☆☆☆	0	The structure of the requirement must follow one of th...
Ambiguous sentences	☆☆☆	0	N/A

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>



The screenshot shows the REUSE Company website's 'Webinars' page. The page has a dark blue header with the REUSE logo and navigation links: PRODUCTS, SERVICES, LIBRARIES, RESOURCES, TRC FORUM, SUPPORT, COMPANY, DOWNLOADS, and WORK WITH US. Below the header, there are two tabs: 'Upcoming Webinars' and 'Past webinars'. The 'Past webinars' tab is selected, displaying a list of webinar titles and brief descriptions:

- Getting started with Requirements Quality Assessment**
Low requirements quality is one of the main reasons that lead to unsuccessful projects in any engineering domain. This webinar introduces the tools to early detect these sources of problems and provides with a set of basic techniques to overcome them.
- Boosting Reuse and Quality in the engineering process: revamping Product Lines**
We are moving from delivering systems to delivering services, and thus, knowledge digitalization plays a key role in order to improve the way we do engineering for our development procedures. Therefore, we need to align the methods and techniques with the business needs to align all the legacy assets with the required development and manufacturing needs.
- Applying Machine Learning Techniques to the Flexible Assessment of Requirements Quality**
This webinar is focused on the presentation of a flexible method to assess and improve the quality of requirements that can be easily adapted to different contexts, projects, organizations and quality standards, with a high degree of automation.
- Burn the Boats – Truly Integrating Requirements and Systems Models**
Upon landing in Mexico, Conquistador Hernán Cortés famously ordered his men to 'burn the boats'. For the same reasons, our journey towards Model Based System Engineering (MBSE) will require the burning of the specification 'document boats' used to house and transport the most important artifacts, i.e. the declarations of Stakeholder Needs and System Requirements.
- Automatic Traceability Discovery for Systems Engineering**
Traceability is a core activity within the System Development Lifecycle, although a tedious and time-consuming task, especially when the traces involve items coming from different disciplines and tools.
- How to write requirements in the Space Industry using a Knowledge Library based on ECSS standards**
Guidelines for the development of technical requirements specifications in the space industry following the ECSS-E-ST-10-06C.
- Knowledge and Quality management milestones in a SE organization**
Ambiguity can jeopardize a project. Learn how to apply Systems Engineering centered on Knowledge and Quality to develop better information!
- First steps to improve the quality of your requirements**
Requirements are, with no doubts, the main source of re-work in both software and systems intensive projects. This webinar introduces these sources of problems and misunderstanding, and provides a set of basic tricks and techniques to overcome them.
- Doing the deal of your lifetime – With the Procurement Quality Suite (PQS)**

At the bottom of the page, there is a footer: 'The REUSE Company uses cookies to ensure you get the best experience on our website [Learn more](#) [I understand](#)'

A green folding ruler is laid out horizontally across the top half of the image. The ruler has black markings and numbers, with '25 FEET' and '7.5m' printed in red. Below the ruler, two US coins are placed on a wooden surface. One is a 1988 Lincoln cent (penny) and the other is a one-cent coin (penny) featuring the Lincoln Memorial. The background is a dark wood grain.

Implementing the MASTER patterns inside TRC tools

Who are the SOPHISTs ?

- 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)



Who are the SOPHISTS ?



- > Training & consulting firm created in 1996 – today 60 members
- > Specialization in Requirements Engineering and Systems Engineering
- > Co-creator of the Requirements Engineering standard
- > Provider of training, consulting, and methods and knowledge



- > Wissen: »A short RE Primer« publications of requirements engineering practice



- > MASTE: Master patterns to enhance the structuration of the requirements engineering process



- > RE Primer : Overall description of the requirements engineering phase and set of rules (SOPHISTen-Regelwerk)



Die SOPHISTen



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



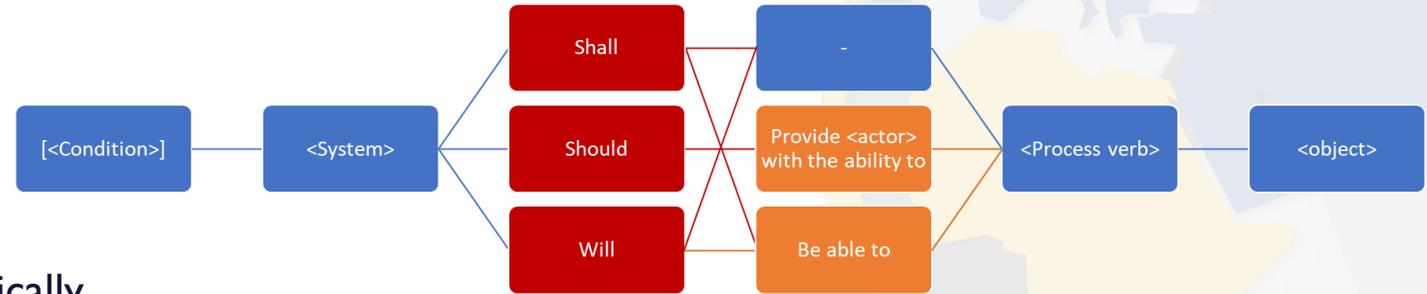
- The FunctionMASTER also has a detailed version to give more context to the functionality described

FunctionMASTER : 3 different paths

Independent system activity

- > Function started and perform automatically

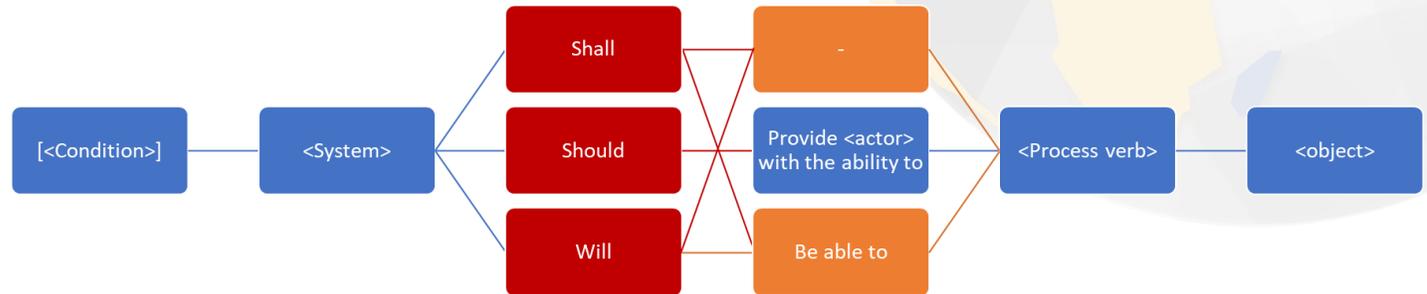
by the system



User interaction

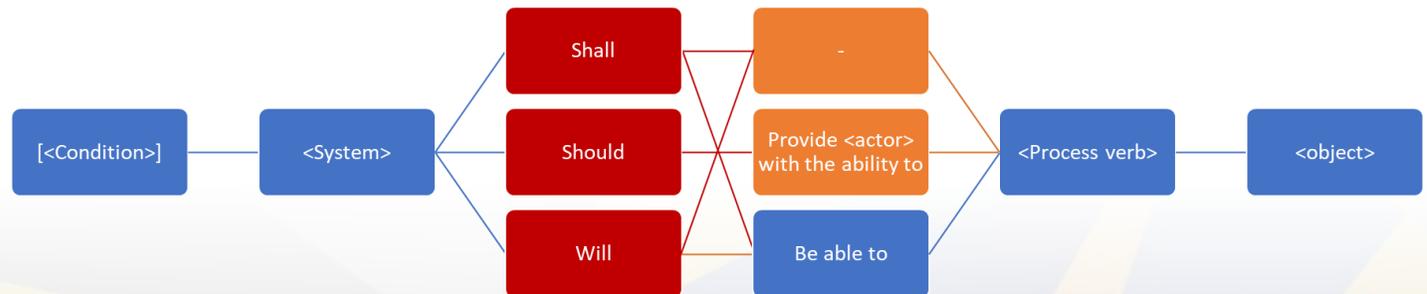
- > The system enables the user to perform a function to achieve a goal. The system relies on the user to perform the

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

i 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:

[ConditionMASTER]	+	[System]	+	Shall	+	«PROCESS VERB»	+	[Object]	+	[[Details of the ProcessVerb]]
				or				or		
				Should				[Actor]		
				or				or		
				Will				[Function]		

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 unpredictably occur 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:

Yes

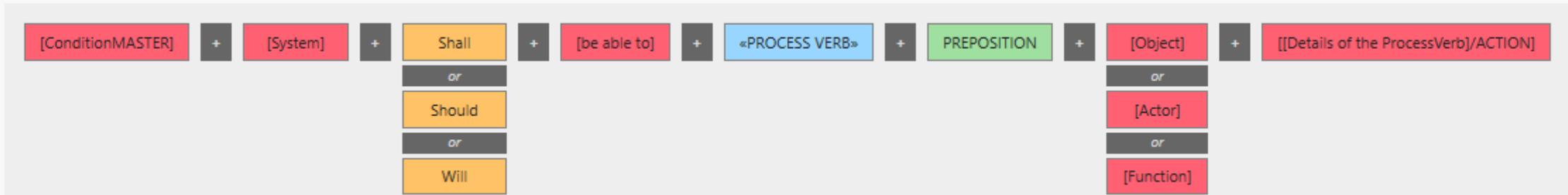
Enabled:

Yes

Weight:

1,030

Syntax:

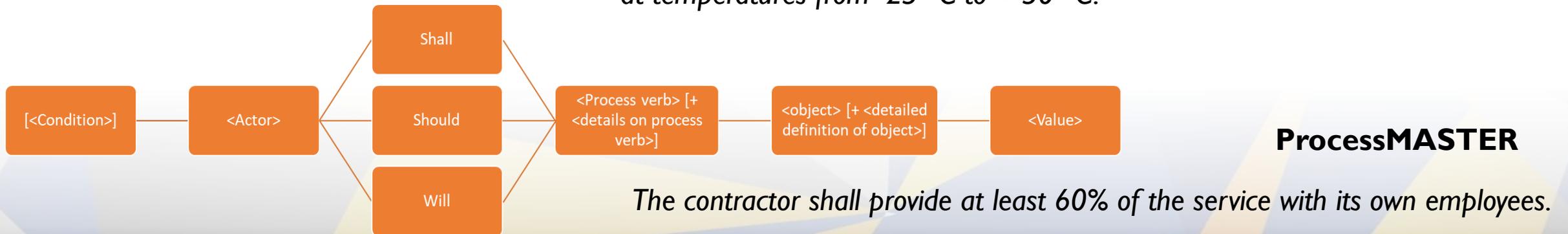
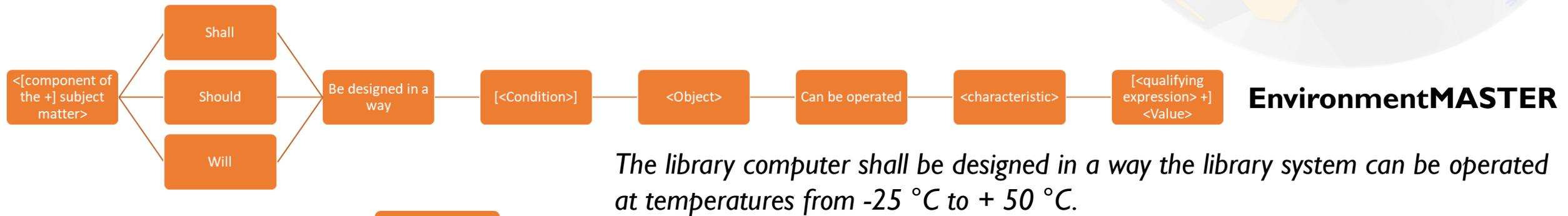
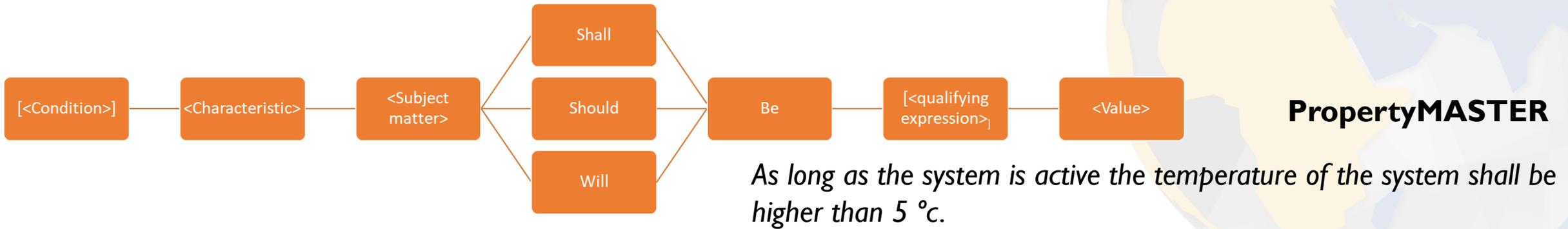


SOPHIST MASTER patterns : Non-functional requirements

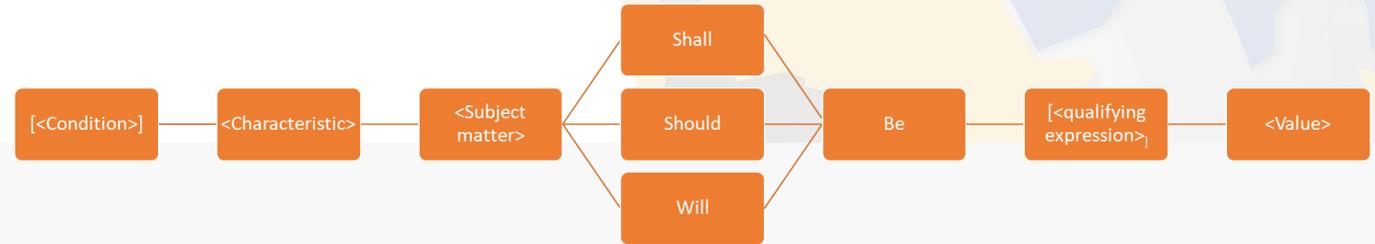
- Requirements which do not address the system functionality but contain elements with impact on the addressed functions
- 6 sub-categories of non-functional requirements

	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



SOPHIST MASTER patterns : PropertyMASTER



Name:
[[SOPHIST]: PropertyMASTER]

Description:
N/A

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

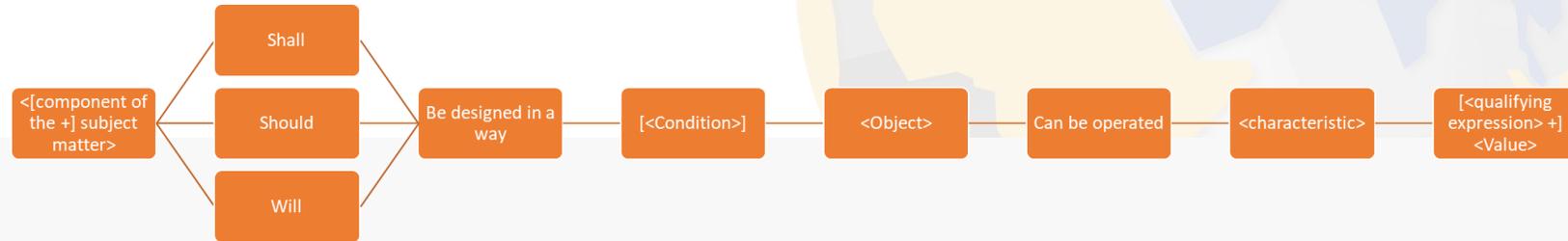
Example:
• The weight of the library system shall be lower than 10 kg

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

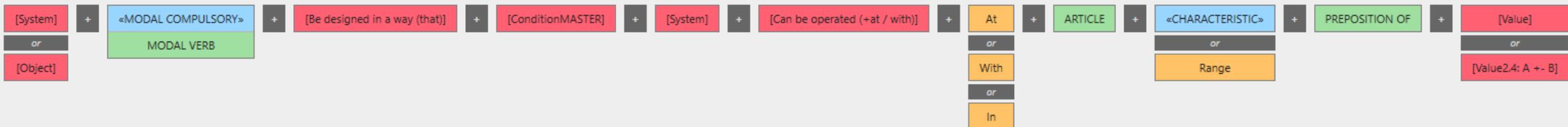
Extracted from : Chris Rupp and die SOPHISTen. Schablonen für alle Fälle. SOPHIST GmbH, 2014

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 requirements for activities to be carried out.

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

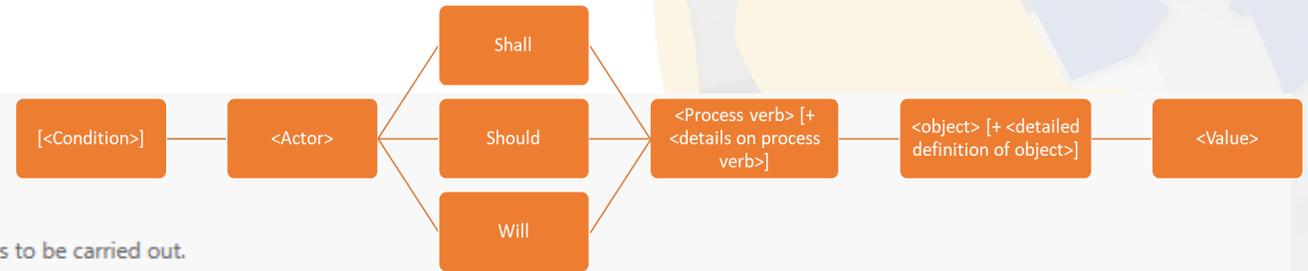
Pattern group(s):
• 01 - SOPHIST (391)

Example:

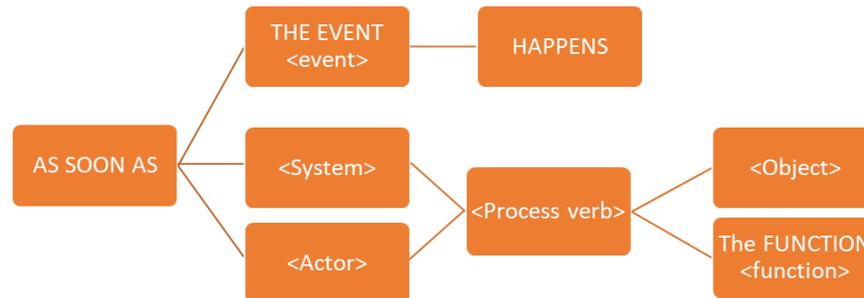
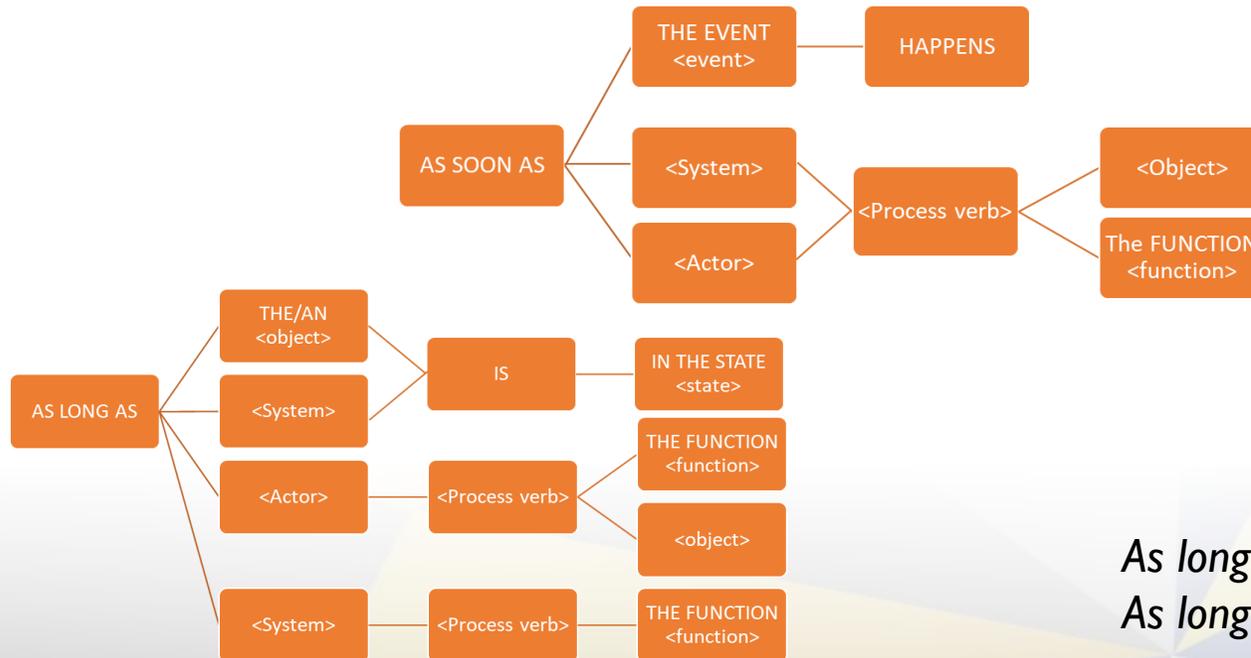
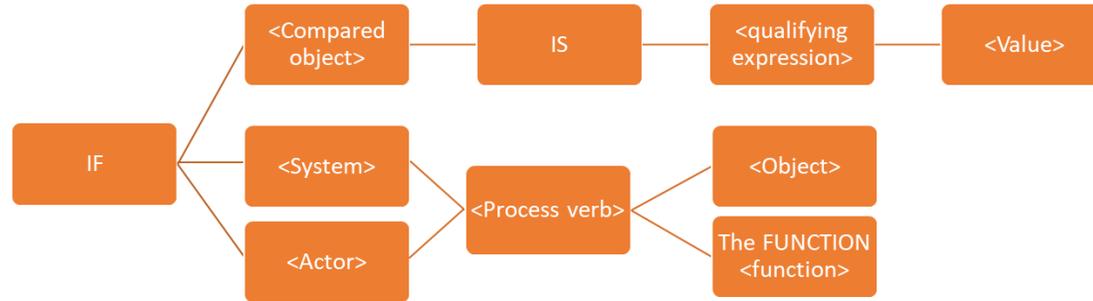
- The contractor shall prepare an operating manual for the MP3 player.
- The contractor shall update the risk list weekly.

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

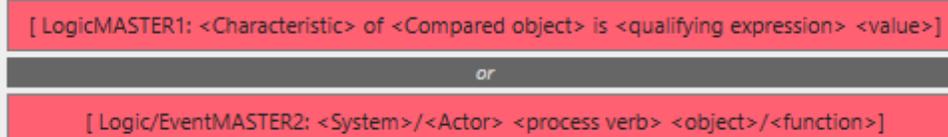
*As long as the smartphone is in the state Low Battery, ...
As long as the customer borrows a book from the library, ...*

SOPHIST MASTER patterns : Example of structure with LogicMASTER

Syntax:



Syntax:



LogicMASTER I

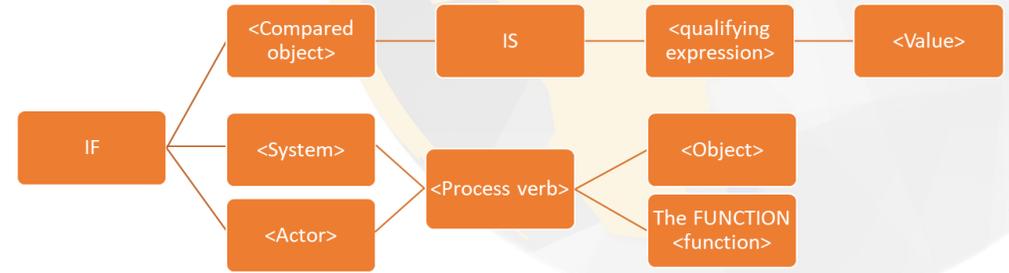
Syntax:



OR

LogicMASTER2

Syntax:



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



INCOSE Guide for Writing Requirements

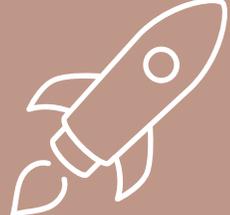
INCOSE
Quality rules for the analysis of textual requirements



EARS Patterns

EARS
Requirements patterns

NASA
Glossary, patterns and rules



NASA Library

Knowledge Base



ISO 26262 Library

ISO 26262
Glossary, patterns and rules



MASTER patterns

MASTER
patterns



Management of the knowledge base to create the patterns

Using the SES Knowledge Manager (KM)

The screenshot shows the Knowledge Manager application window. The top menu includes File, Terminology, Conceptual Model, Patterns, Formalization, Inference, Configuration management, Extensibility, Assets store, and Settings. The main interface is divided into several sections:

- Search fields:** Includes Name, Pattern group, and Pattern example.
- Identifier:** Includes radio buttons for 'Equals to', 'Greater than', and 'Lower than'.
- Attributes:** Includes checkboxes for Enabled, Indexable, Flag, Revised, Used as subpattern, Has RSHPs, and Has examples.
- Contains restriction:** Includes checkboxes for Text, Term, Pattern, Cluster, and Term tag.

Below these sections is a table of patterns:

Identifier	Name	Example	Weight	Times used as subpattern	Language	Indexable	Enabled	Flag	Revised	Ignore negative
3935	[BedingungsMASTER]	Solange sich das Handy im	500	7	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3946	[SOPHISTen-FunktionsMASTER/Selbsttätige Systemaktiv	Die Wetterstation muss Terr	1,000	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3898	[SOPHISTen-FunktionsMASTER/Selbsttätige Systemaktiv	Solange sich das System im	1,001	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3947	[SOPHISTen-FunktionsMASTER/Benutzerinterkation/ bec	Das System muss dem Ben.	1,005	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3936	[SOPHISTen-FunktionsMASTER/Benutzerinterkation]	Solange sich das System im	1,006	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3948	[SOPHISTen-FunktionsMASTER/Schnittstellenanforderun	Die Systeme müssen fähig s	1,010	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3937	[SOPHISTen-FunktionsMASTER/Schnittstellenanforderun	Falls das Gewicht des Syster	1,011	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3945	[SOPHISTen-ProzessMASTER / bedingungsfrei]	Der Auftragnehmer muss ei	1,100	2	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3925	[SOPHISTen-ProzessMASTER]	Solange der Benutzer das S	1,101	2	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3944	[SOPHISTen-EigenschaftsMASTER / bedingungsfrei]	Die Erwärmung der Systemk	1,102	5	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3911	[SOPHISTen-EigenschaftsMASTER]	Solange sich das System im	1,103	5	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3923	[SOPHISTen-UmgebungsMASTER]	Das Systemelement des Sys	1,104	1	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3938	[SOPHISTen-nicht funktionale Anforderungen/Qualitäts	Die Zeitdauer für die Übertr	1,105	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3941	[SOPHISTen-nicht funktionale Anforderungen/Sonstige I	Das Dateiformat des Monat	1,110	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3942	[SOPHISTen-nicht funktionale Anforderunoen/Durchzuf	Der Auftragnehmer muss ei	1,115	0	Deutsch (Deutschland)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

65 pattern(s)

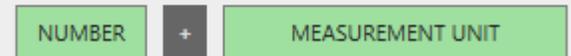
Connected to 'SOPHISTen-Deutsche Sprache' at 'localhost\SQLEXPRESS2008'



Available languages of the Sophist library



Syntax:



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

The screenshot displays the TRC tool interface for a pattern named 'ProcessMASTER'. The interface is divided into several sections:

- Name:** [[SOPHIST]: ProcessMASTER]
- Description:** Patterns aimed at writing legal-c...
Extracted from : Chris Rupp and ...
- Pattern group(s):** 01 - SOPHIST (391)
- Example:**
 - The contractor shall prepare ar...
 - The contractor shall update the...
- Indexable:** Yes
- Enabled:** Yes
- We:** 1,3

The **Clusters:** section shows a hierarchical tree of semantic clusters:

- «CROSS DOMAIN VIEWPOINTS»
- «Definition»
- «IMPRECISE QUANTIFIERS»
- «REQUIREMENTS»
- «ACTION»
- «ACTOR» (highlighted)
 - «STAKEHOLDER»
 - «USER» (highlighted)
- «CONSTRAINT»
- «ENTITY»
- «EVENT»
- «INVALID WORDS»
- «LY Adeverbs»
- «MODAL»
- «Negation»
- «PARAMETER»
- «STANDARD DOCUMENT»
- «STATE»
- «UNITS»

The **Syntax:** section shows the pattern's structure with various components and their relationships:

- [ConditionMASTER] + [Actor] + «MODAL COMPULSORY» + «PROCESS VERB» + [Value2.3: More/Less + THAN + <Value>] + [Object] + [[Details of the ProcessVerb]]
- «MODAL COMPULSORY» is linked to MODAL VERB.
- «PROCESS VERB» is linked to «ACTION».
- [Value2.3: More/Less + THAN + <Value>] is linked to «RANGE (value limitations)».
- [Object] is linked to «PREPOSITION OF».
- [[Details of the ProcessVerb]] is linked to «ACTOR».

Additional syntax components shown in callouts:

- «ACTOR» is linked to «USER» and «STAKEHOLDER».
- «ACTOR» is linked to NOUN.
- «RANGE (value limitations)» + Than + [Value1: NUMBER + UNIT/%] + «PREPOSITION OF».
- «ACTION» + «PREPOSITION OF» + [Actor] or [Object].
- «ACTOR» + «PREPOSITION OF» + [Actor] or [Function].

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

The image displays two screenshots of the KM Pattern wizard interface, showing the configuration of paths and optional terms for a pattern.

Left Screenshot:

- Pattern fields:** Identifier: 3,946; Name: SOPHISTen-FunktionsMASTER/Selbsttätige Systemaktivität / bedingungsfrei
- Original example:** Die Wetterstation muss Temperaturinformationen archivieren
- Current example:** Die Wetterstation muss Temperaturinformationen archivieren
- Syntax:**
 - Path: [System] -> <Modalverb> -> [Akteur]
 - Optional terms:
 - System: das, ähnlich, system
 - Modalverb: müssen
 - Akteur: der ?, neu ?, pilot ?
- Diagram:**

```

graph LR
    S["<System>"] --- Shall
    S --- Should
    S --- Will
    Shall --- P["Provide <actor> with the ability to"]
    Should --- P
    Will --- P
    P --- PV["<Process verb>"]
    PV --- O["<object>"]
    
```

Right Screenshot:

- Pattern fields:** Identifier: 3,946; Name: SOPHISTen-FunktionsMASTER/Selbsttätige Systemaktivität / bedingungsfrei
- Original example:** Die Wetterstation muss Temperaturinformationen archivieren
- Current example:** Die Wetterstation muss Temperaturinformationen archivieren
- Syntax:**
 - Path: [Objekt] -> [Prozesswort/OhneZu]
 - Optional terms:
 - Objekt: die, aktuell, 2, spezifiziert, abs, à, den, selektiert, automatisierung, sofort
 - Prozesswort/OhneZu: aus, dem, drucker, drucken
- Diagram:**

```

graph LR
    S["<System>"] --- Soll
    S --- Muss
    S --- Wird
    Soll --- P["-"]
    Muss --- P
    Wird --- P
    P --- O["<Objekt>"]
    O --- PW["<Prozesswort>"]
    
```

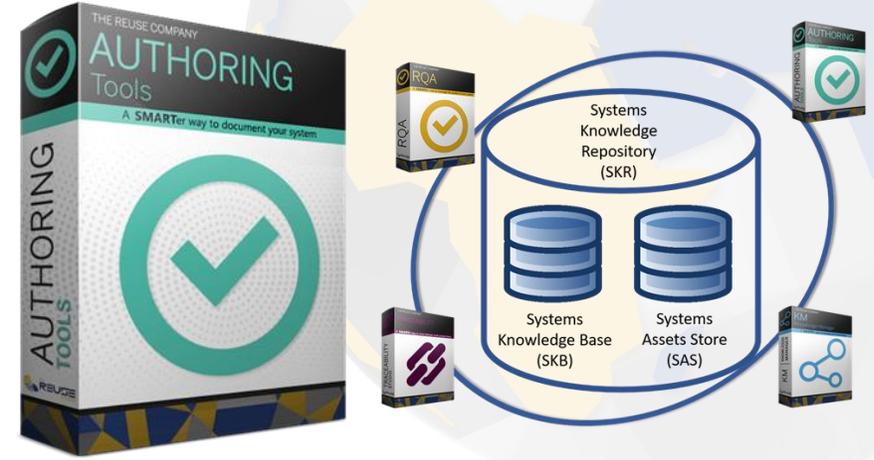
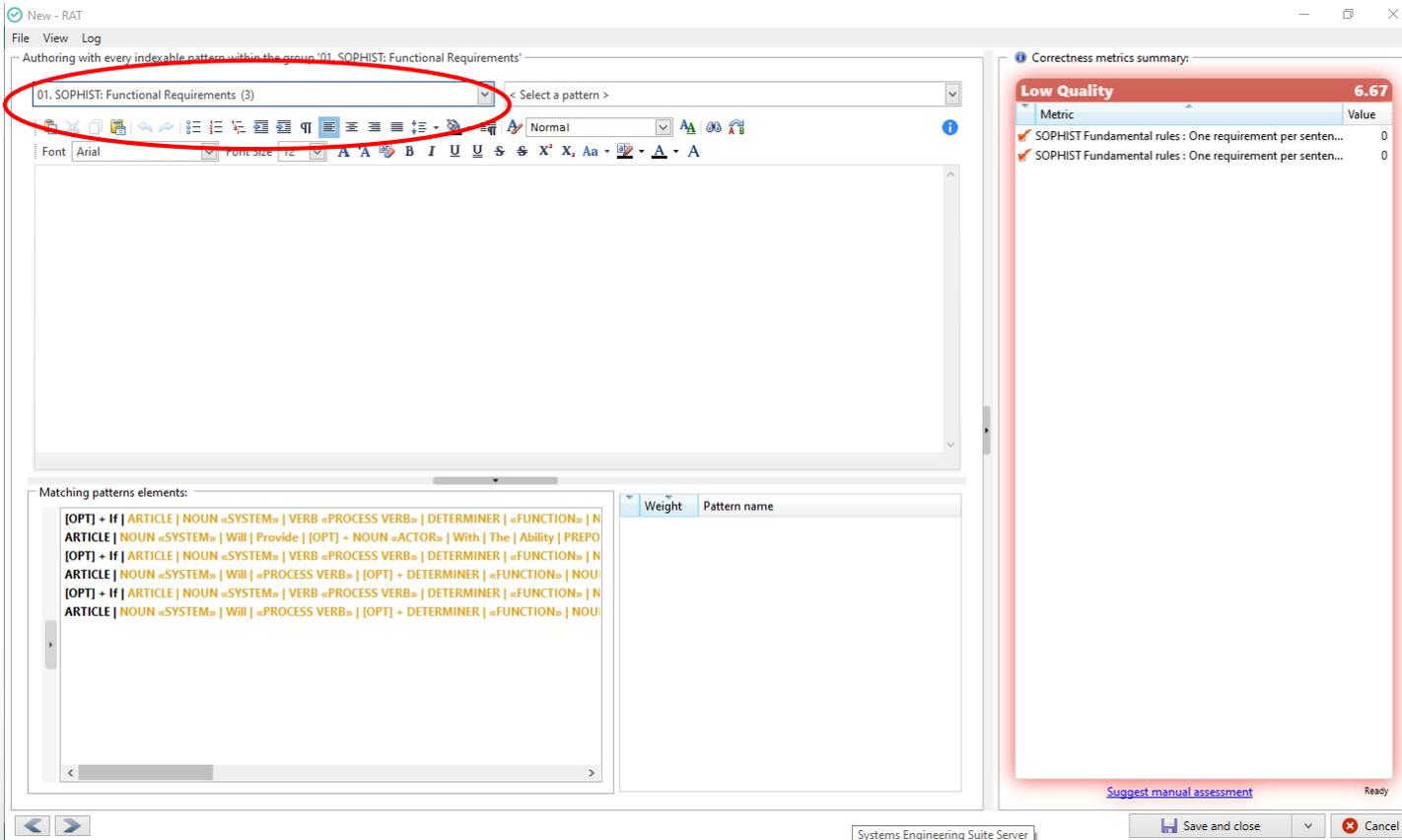


IMPROVE
-
MENT

**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 :

The screenshot shows the Authoring tool interface. The main window is titled 'New - RAT' and contains a menu bar (File, View, Log) and a toolbar. The document title is '01. SOPHIST: Functional Requirements (3)'. The text area contains the text 'The s' and a dropdown menu is open, showing options: 'Server', 'System', 'System component', and 'System element'. A green arrow points to the 'Server' option. Below the dropdown menu, there are checkboxes for '4 terms', 'Show numbers', and 'Show optional terms'. On the right side, a 'Correctness metrics summary' panel is visible, showing a 'Low Quality' status with a score of 3.33. The panel lists a metric: 'SOPHIST Fundamental rules : One requirement per senten...' with a value of 2.

Metric	Value
✓ SOPHIST Fundamental rules : One requirement per senten...	2



Authoring with SOPHIST patterns

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

01. SOPHIST: Functional Requirements (3) [SOPHIST]: FunctionalMASTER/Automatic system activity

Name:
[[SOPHIST]: FunctionalMASTER/Automatic 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 (1167)

Example:

- The system shall design postcards

Indexable:	Enabled:	Weight:
Yes	Yes	1,010

Syntax:

[ConditionMASTER] + [System] + Shall + «PROCESS VERB» + [Object] + [[Details of the ProcessVerb]]

or

Should

or

Will

[Actor]

or

[Function]

Properties:
N/A

Relationships:
N/A

Low Quality 6.67

Metric	Value
<input checked="" type="checkbox"/> SOPHIST Fundamental rules : One requirement per senten...	0
<input checked="" type="checkbox"/> SOPHIST Fundamental rules : One requirement per senten...	0





Implementing the SOPHIST RE-Rules

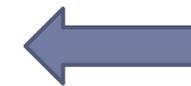


Measuring quality with the SOPHIST library

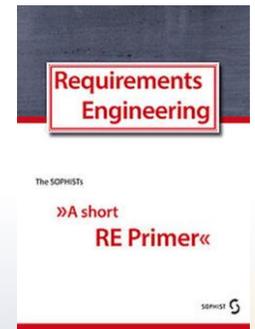
Using the SES Requirements Quality Analyzer (RQA)

C.	Project	Module	ID	Workproduct name	Correctness	Score	Consistency
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	1	If the temperature of the library is $\geq 50^\circ\text{C}$ the library system shall save the database	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	2	The system shall design postcards	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	3	As soon as the smart home system has received the unlock Door signal the smart home system shall unlock the door	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	4	As soon as the event birthday happens the system should display a calendar entry to the user	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	5	The system should provide the user with the ability to organize recipes	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	6	The document editor shall provide the user with the ability to create new documents	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	7	The system shall provide the user with the ability to search items.	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	8	If the user has entered an appointment, the system shall provide the user with the ability to save the appointment	★★★★	0.00	☆☆☆☆
<input checked="" type="checkbox"/>	SOPHISTen-Eng	System Requirements	9	The smart home system shall be able to receive the Unlock Door signal	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	10	The charger of the device shall be designed in a way the system can be operated at temperatures from 10°C to 50°C	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	11	The system shall be designed in a way the system can be operated at voltage of 220 Volt \pm 10 Volt	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	12	The system shall be designed in a way that the car+ can be operated at voltage of 230 Volt +	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	13	The heating of the system shall be $\leq 50^\circ\text{C}$	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	14	The voltage of the accumulator shall be 220 volts \pm 5 Volt	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	15	The system shall be designed in a way that the system can be operated at a voltage of 220 V \pm 10 V	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	16	The system shall be designed in a way that the system can be operated with a number of > 100 users surfing simultaneously	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	17	The design of the website shall be responsive	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	18	The heating of the system components shall be $\leq 50^\circ\text{C}$	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	19	If the user has activated the function accumulate database for the software developers the backup power of the communication system shall be above 70 %	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	20	The time for the manufacturing shall be maximum 50 weeks	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	21	The time for the transfer to another Server shall be ≤ 30 minutes	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	22	The number of users shall be ≤ 100	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	23	The number of printers shall be between 60 and 90	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	24	The training time for a user shall be ≤ 6 days	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	25	The availability of the system shall be greater than 98 % per year	★★★★	0.00	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	26	The contractor shall prepare an operating manual for the MP3 player	☆☆☆☆	N/A	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	27	The contractor should provide more than 90% of the service to the new employees	☆☆☆☆	N/A	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	28	The contractor shall update the risk list weekly	☆☆☆☆	N/A	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	29	If the library system does not successfully enable the function Abort storing registration data, the librarian shall load the local database	☆☆☆☆	N/A	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	30	If the system does not identify the loan item, the system shall create a new item in the database	☆☆☆☆	N/A	☆☆☆☆
<input type="checkbox"/>	SOPHISTen-Eng	System Requirements	31	If the library system does not successfully enable the function Abort storing registration data, the librarian shall load the local database	☆☆☆☆	N/A	☆☆☆☆

Implementing the SOPHIST RE-Rules



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)

TRC WEBINARS 2020

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

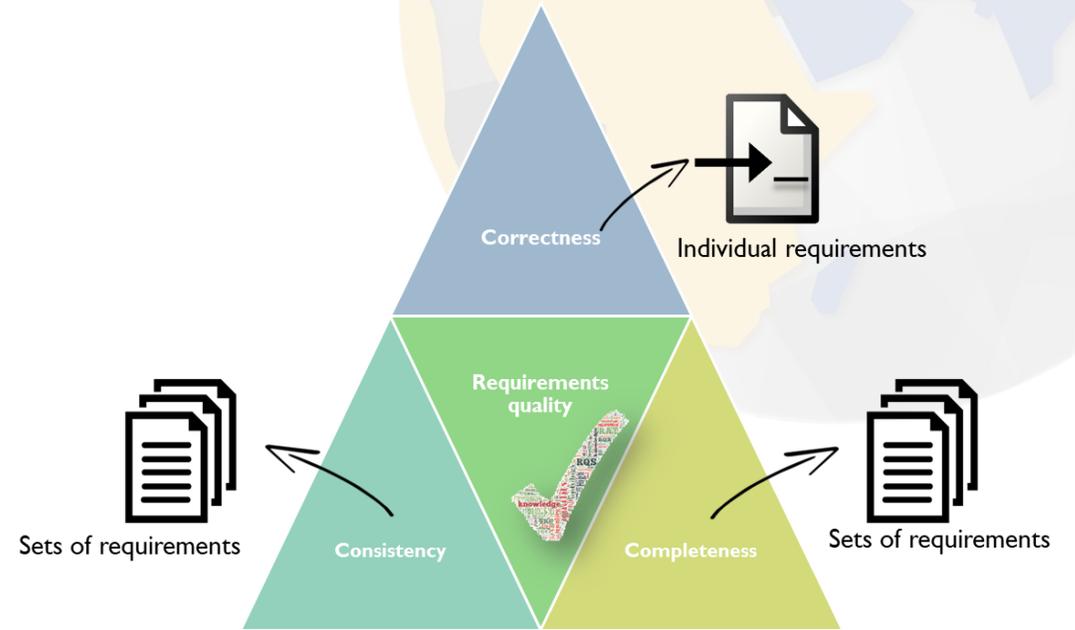
Priority LOW

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



Implementing the SOPHIST RE-Rules

The SOPHIST RE-Rules



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

Metrics set baseline configuration: SOPHIST patterns quality assessment

Name: SOPHIST patterns quality assessment

Description: Measures quality of SOPHIST patterns by checking if a requirement matches one of the SOPHIST Writing Requirements (text length, syntax errors, presence of terms in the ontology)

Metrics configuration:

Correctness Consistency Completeness

Correctness metrics:

Metric Identifier	Custom Metric...	Name	Rationale	Weight	Enabled	Correctness type
15,487	15	R04 Precision - Out-of-controlled vocabul...	This metric calculates the numbe...	1	<input type="checkbox"/>	Non-parameterized
15,488	16	R04 Precision - Out-of-System Conceptua...	This metric calculates the numbe...	1	<input type="checkbox"/>	Non-parameterized
15,489	41	R05 Precision - Imprecise quantifiers (Avoi...	This metric controls the usage of...	1	<input type="checkbox"/>	Parameterized - Special Sentences
15,490	56	R06 Precision - Units: Numbers with Meas...	This metric enforces the assignm...	1	<input type="checkbox"/>	Parameterized - Pattern group and pattern matching
15,491	42	R07 Precision - Vague adverbs (Avoid)	This metric controls the existenc...	1	<input type="checkbox"/>	Parameterized - Special Sentences
15,492	43	R08 Precision - Vague adjectives (Avoid)	This metric controls the existenc...	1	<input type="checkbox"/>	Parameterized - Special Sentences
15,494	20	R11 Concision - Superfluous infinitives (Av...	Sometimes a requirement has m...	1	<input type="checkbox"/>	Parameterized - Special Sentences
15,495	6	R13 Non Ambiguity - TRC - Flow sentence...	A requirement must avoid flow s...	1	<input type="checkbox"/>	Parameterized - Special Sentences
15,496	10	R13 Non Ambiguity - TRC - Negative sent...	This metric identifies ambiguous...	1	<input type="checkbox"/>	Parameterized - Special Sentences
15,497	47	R13 Non Ambiguity - TRC - Speculative se...	This metric identifies ambiguous...	1	<input type="checkbox"/>	Parameterized - Special Sentences

No. of metrics: 59, Enabled: 6

Quality function for selected metric:

Range	Mandatory	Quality Level	Summary	Description
[0, 1)	No	★★★★		
[1, ∞)	No	★★★☆☆	Avoid vague adjectives	Adjectives qualify entities in some way. Avoid vag...

No. of ranges: 2

Special Sentence

- adequate
- allowable
- ancillary
- appropriate
- bad
- better
- big
- common
- customary
- easy
- easy to use
- effective
- efficient
- expandable
- fault tolerant
- flexible
- friendly
- generic
- great
- High fidelity
- High speed
- large
- medium-sized
- nominal
- normal
- optimal
- optimum
- proficient
- prompt
- quick
- rapid
- reasonable

Rationale

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



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	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

#	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	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	
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

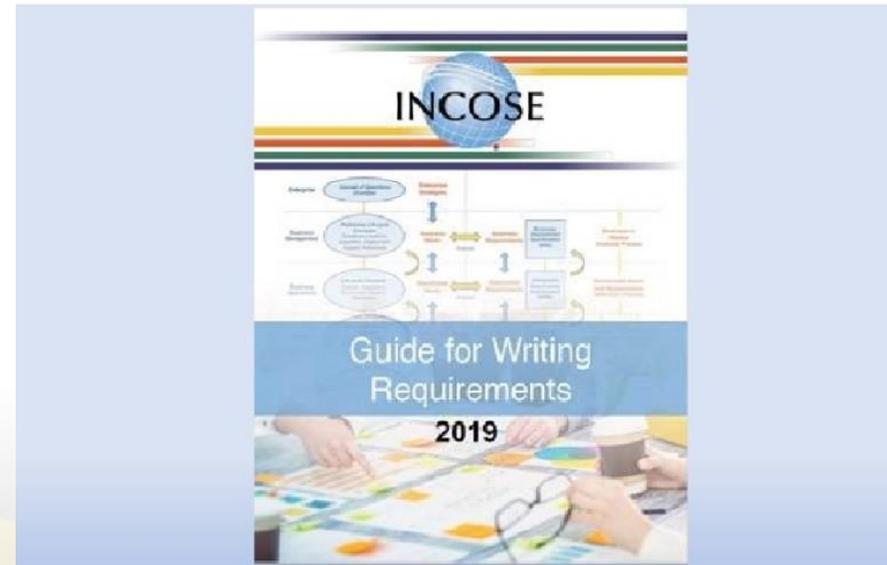


Live Demo



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







contact@reusecompany.com