SOPHIST Master Patterns Knowledge Library

V1.0 - May 2020

Contents

All rights reserved © The REUSE Company 2020

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)

Source : Sophist.de

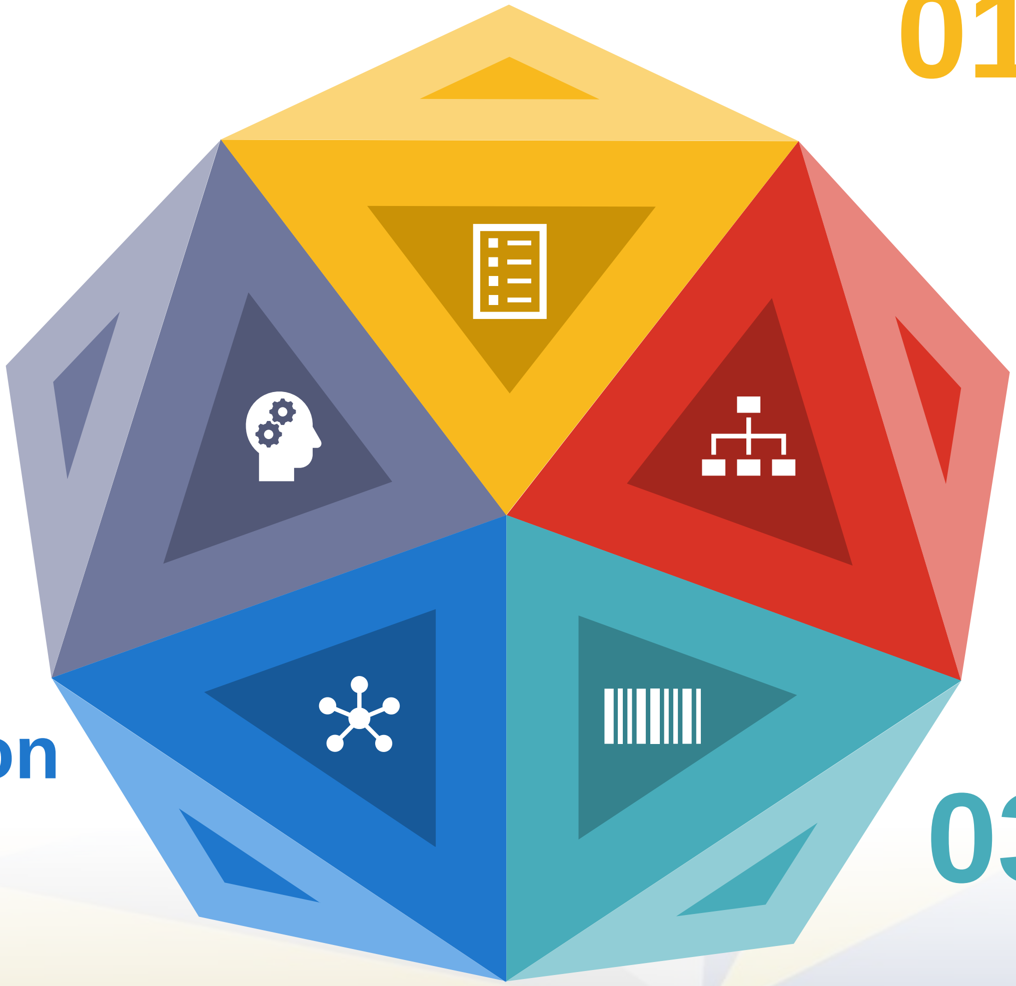
All rights reserved © The REUSE Company 2020

What is a Knowledge Library

- A combination of Knowledge items,
  - of different nature,
  - at different levels of abstraction
- Representing 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.

All rights reserved © The REUSE Company 2020

What is a Knowledge Library



01 Vocabulary/Glossary

Controlled Organizational and Project Vocabulary for a common understanding among stakeholders

02 SCM/Architectures

Capture the system architectures represented in views and models. Stablish relationships among system and system elements, and among other system entities. Classifying information by meaning, nature…

03 Patterns

Representing a set of agreed-upon templates (grammars) to create and maintain consistent textual artifacts

04 Formalization

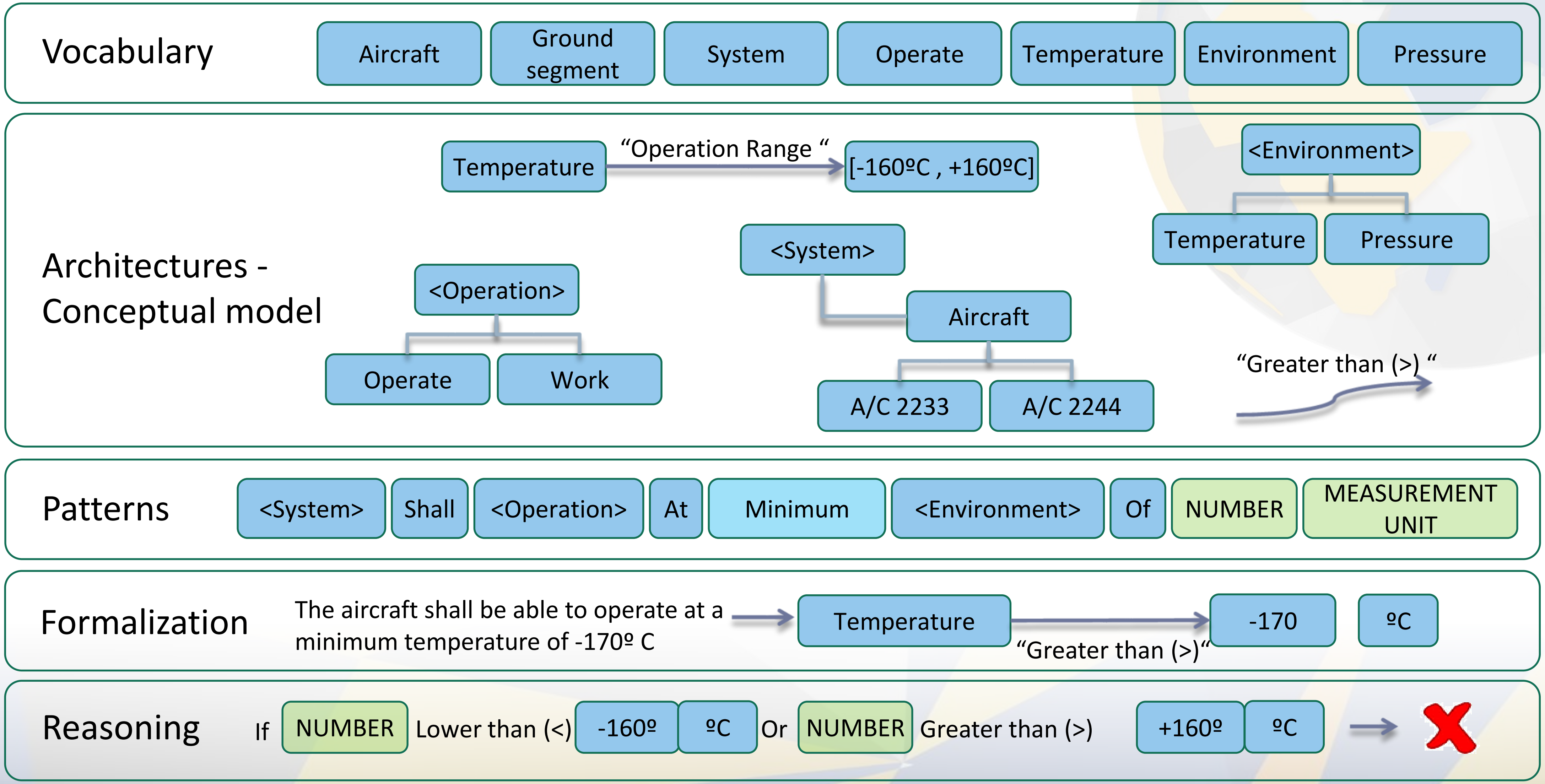
Representation of assets semantic through SRL – System Representation Language

05 Reasoning

A combination of rules, and actions to infer information from valuable assets and to control the behavioural part of the knowledge library

All rights reserved © The REUSE Company 2020

Example of a Knowledge Library



All rights reserved © The REUSE Company 2020

SOPHIST MASTER patterns

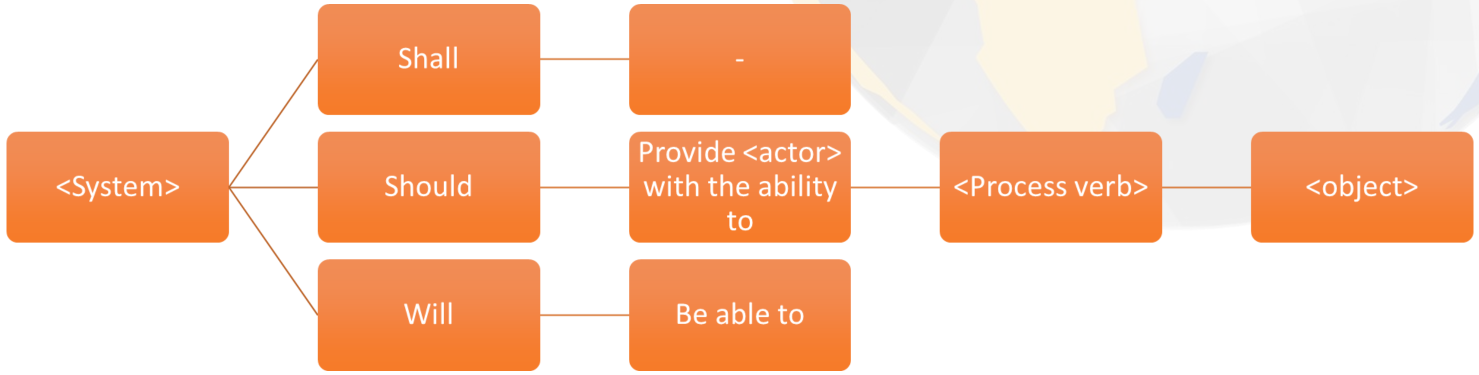
- Cross-domain patterns to express system requirements :
  - Functional requirements
  - Non-functional requirements
  - With or without introducing a condition to the main sentence of requirements
- The use of the MASTER patterns enhances :
  - The structuration of the syntax of requirements
  - The 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

All rights reserved © The REUSE Company 2020

SOPHIST MASTER patterns : Functional requirements (FunctionMASTER)

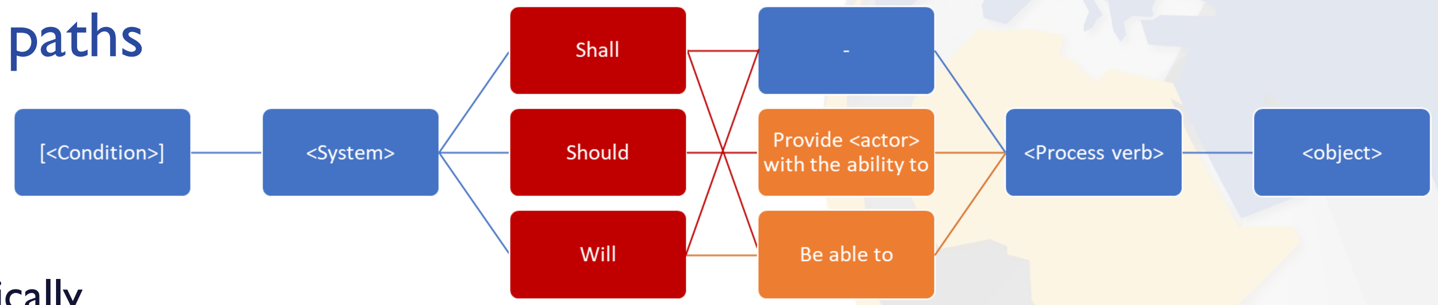
- Objective : write functional requirements
- 3 different cases to express the FunctionMASTER
  - Independent system activity
  - User interaction
  - Interface requirement



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

All rights reserved © The REUSE Company 2020

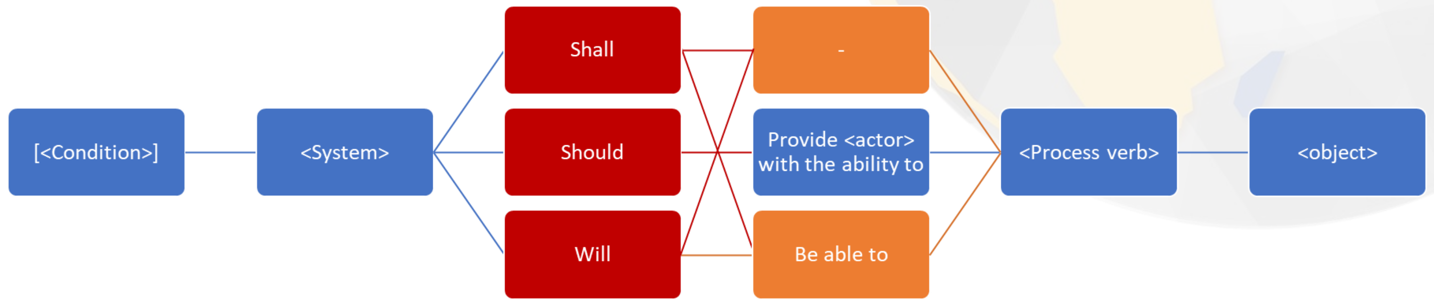
FunctionMASTER : 3 different paths



Independent system activity

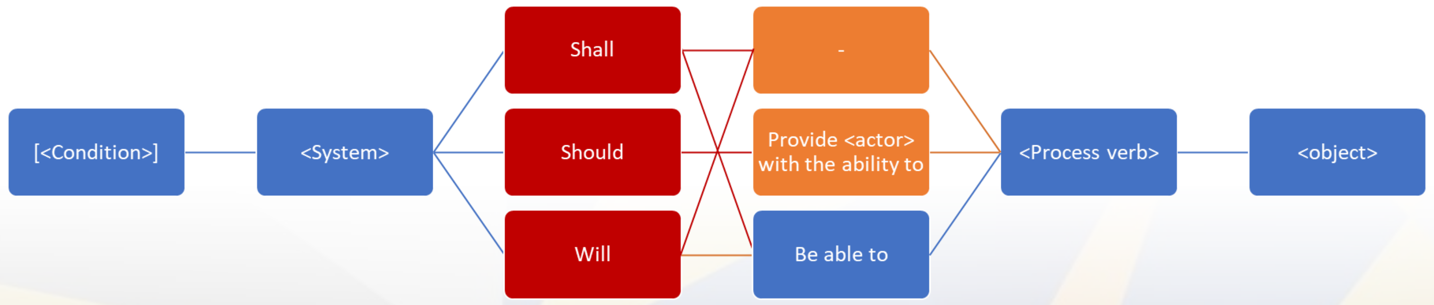
- Function started and perform automatically

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)

All rights reserved © The REUSE Company 2020

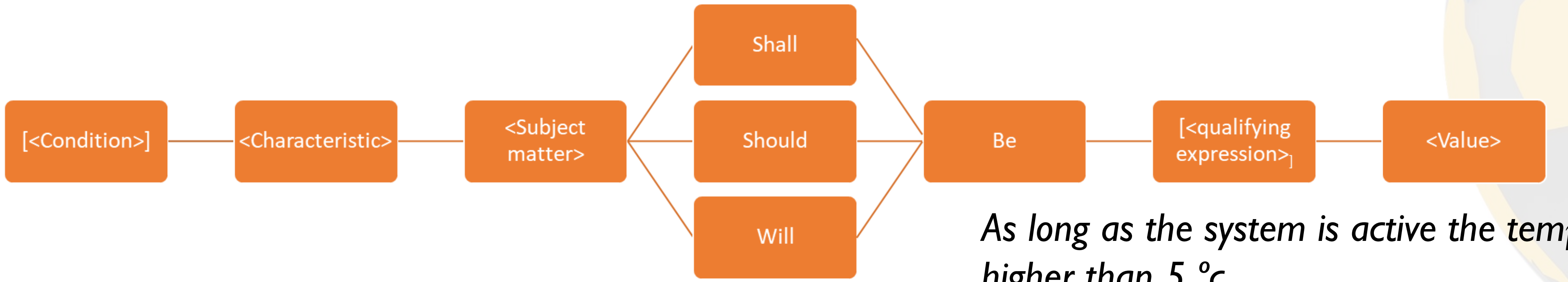
SOPHIST MASTER patterns : Non-functional requirements

- 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

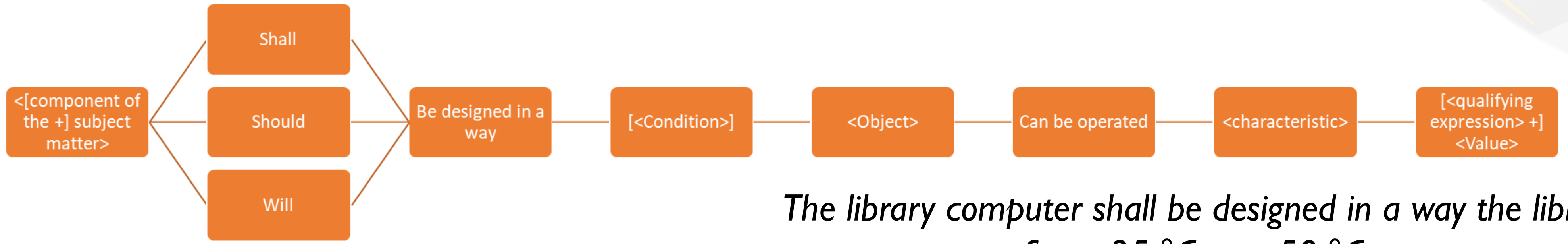
All rights reserved © The REUSE Company 2020

SOPHIST MASTER patterns : Non-functional MASTER Templates



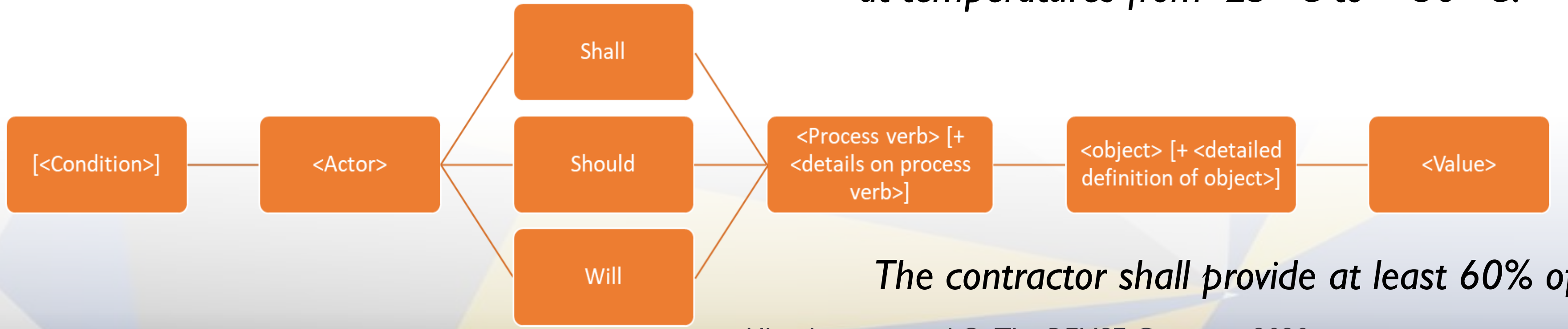
PropertyMASTER

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.

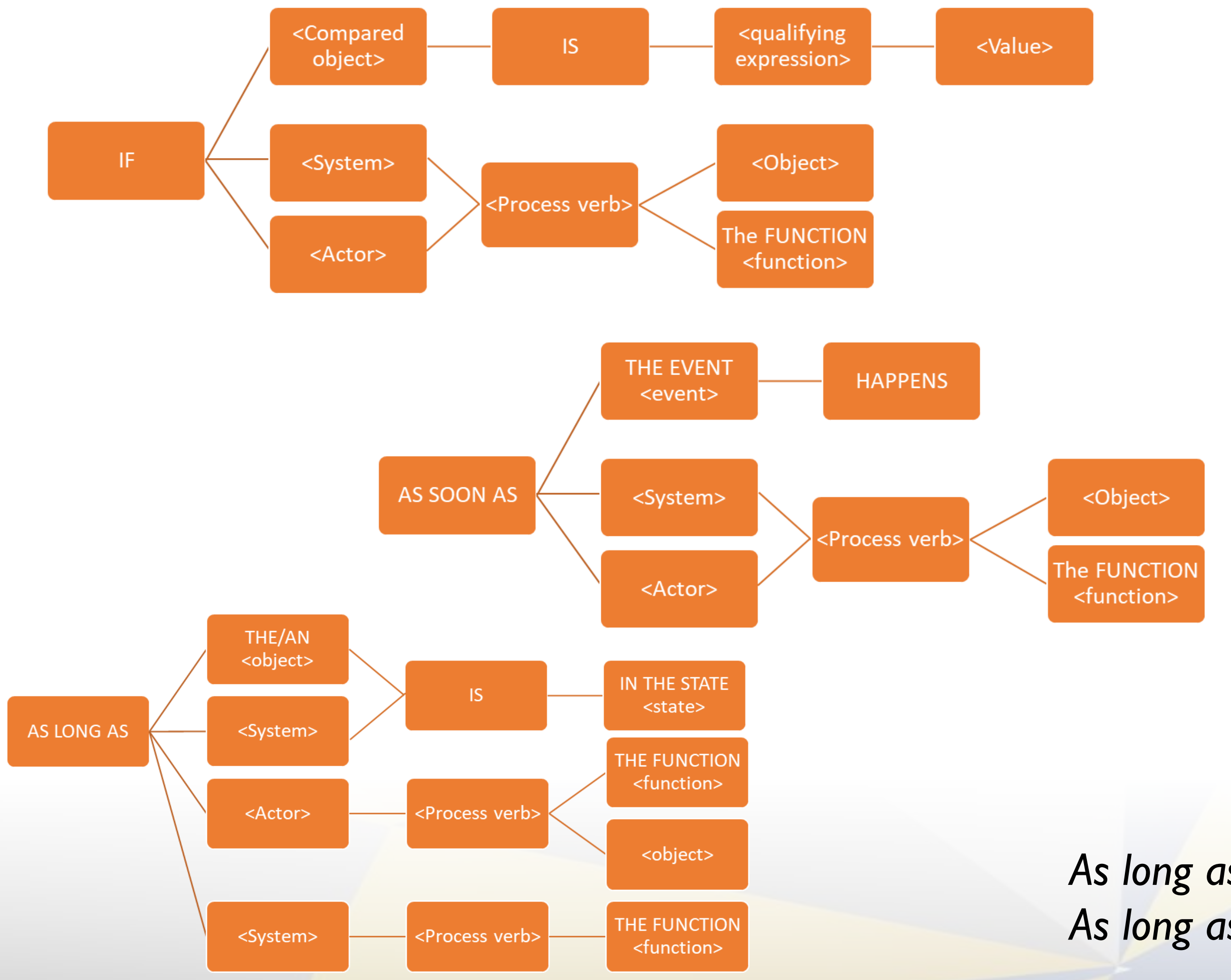

[<Condition>]
<Actor>
Shall
Should
Will
<Process verb> [+ <details on process verb>]
<object> [+ <detailed definition of object>]
<Value>

ProcessMASTER

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

All rights reserved © The REUSE Company 2020

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

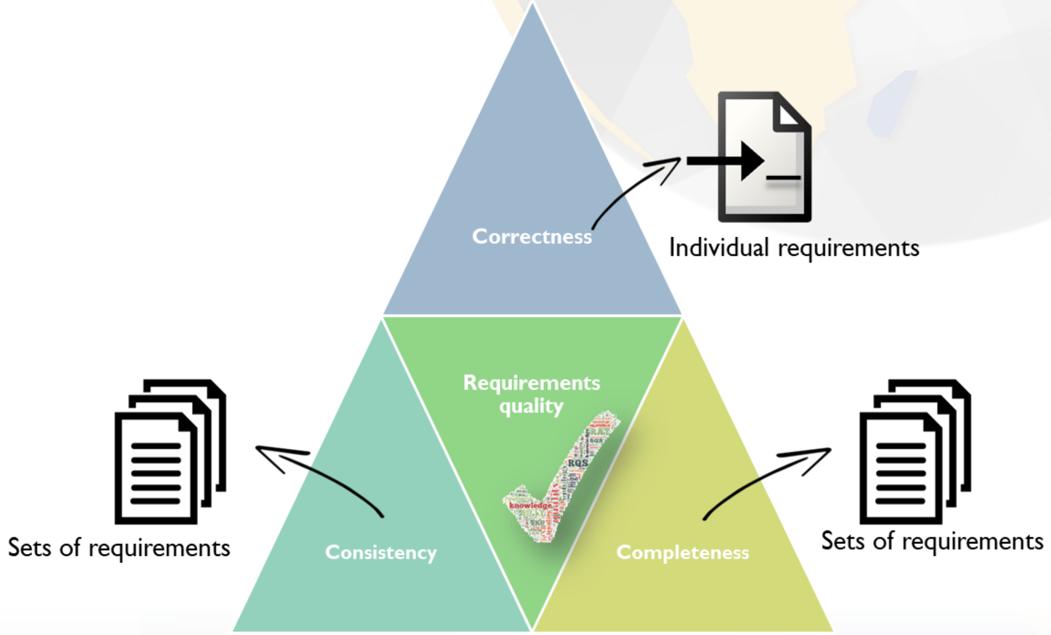
As long as the smartphone is in the state Low Battery, …
As long as the customer borrows a book from the library,…

All rights reserved © The REUSE Company 2020

The SOPHIST RE-Rules

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

SOPHIST **R**equirements **E**ngineering 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.

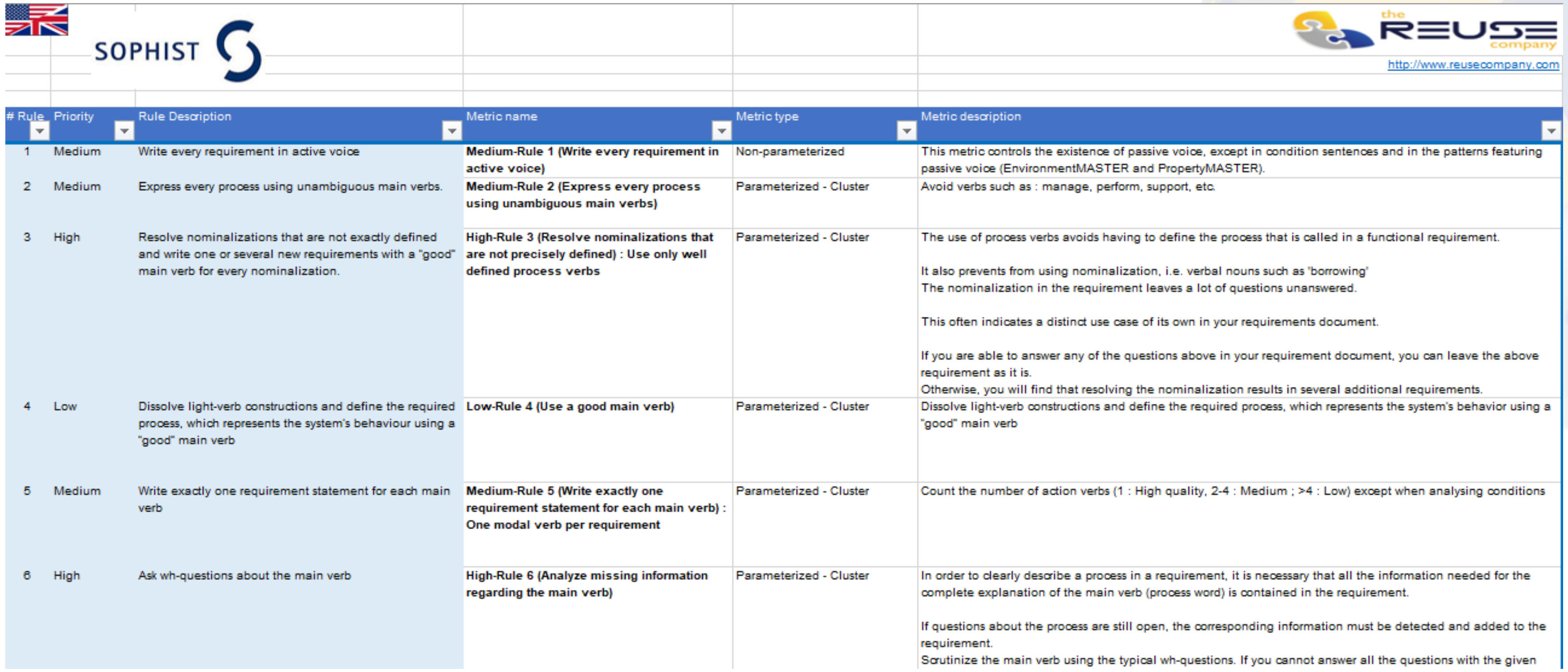
All rights reserved © The REUSE Company 2020

The set of 18 rules

#Rule	Priority Level	Rule Description
1	Medium	Write every requirement in active voice
2	Medium	Express every process using unambiguous main verbs.
3	High	Resolve nominalizations that are not exactly defined and write one or several new requirements with a "good" main verb for every nominalization.
4	Low	Dissolve light-verb constructions and define the required process, which represents the system's behaviour using a "good" main verb
5	Medium	Write exactly one requirement statement for each main verb
6	High	Ask wh-questions about the main verb
7	Medium	Analyse missing information on the adjective or adverb which is derived from a process verb and add information if necessary.
8	Medium	Formulate adjectives in a way that can be measured or tested
9	Low	Formulate separate requirements for non-functional aspects if these aspects are independent or needed as a constraint for several functionalities
10	Medium	Question the used numerals and quantifiers.
11	Medium	Clarify missing numerals and quantifiers.
12	High	Question vague nouns.
13	Low	Replace formulations that describe possible or impossible situations.
14	Low	Remove subordinate clauses that contain irrelevant information for the requirement
15	Low	Shorten or eliminate flowery expressions or phrases that are irrelevant for your requirement.
16	Medium	Analyse exceptions to the usual behaviour of the system and extend the requirement resp. write an additional requirement.
17	High	Requirements with incomplete conditional structures should be checked and formulated or described by another requirement
18	High	Write one or more requirements for every implicit assumption not described.

All rights reserved © The REUSE Company 2020

Implementation of the rules into RQA Quality metrics



http://www.reusecompany.com

# Rule	Priority	Rule Description	Metric name	Metric type	Metric description
1	Medium	Write every requirement in active voice	**Medium-Rule 1 (Write every requirement in active voice)**	Non-parameterized	This metric controls the existence of passive voice, except in condition sentences and in the patterns featuring passive voice (EnvironmentMASTER and PropertyMASTER).
2	Medium	Express every process using unambiguous main verbs.	**Medium-Rule 2 (Express every process using unambiguous main verbs)**	Parameterized - Cluster	Avoid verbs such as : manage, perform, support, etc.
3	High	Resolve nominalizations that are not exactly defined and write one or several new requirements with a "good" main verb for every nominalization.	**High-Rule 3 (Resolve nominalizations that are not precisely defined) : Use only well defined process verbs**	Parameterized - Cluster	The use of process verbs avoids having to define the process that is called in a functional requirement. It also prevents from using nominalization, i.e. verbal nouns such as 'borrowing' The nominalization in the requirement leaves a lot of questions unanswered. This often indicates a distinct use case of its own in your requirements document. If you are able to answer any of the questions above in your requirement document, you can leave the above requirement as it is. Otherwise, you will find that resolving the nominalization results in several additional requirements.
4	Low	Dissolve light-verb constructions and define the required process, which represents the system's behaviour using a "good" main verb	**Low-Rule 4 (Use a good main verb)**	Parameterized - Cluster	Dissolve light-verb constructions and define the required process, which represents the system's behavior using a "good" main verb
5	Medium	Write exactly one requirement statement for each main verb	**Medium-Rule 5 (Write exactly one requirement statement for each main verb) : One modal verb per requirement**	Parameterized - Cluster	Count the number of action verbs (1 : High quality, 2-4 : Medium ; >4 : Low) except when analysing conditions
6	High	Ask wh-questions about the main verb	**High-Rule 6 (Analyze missing information regarding the main verb)**	Parameterized - Cluster	In order to clearly describe a process in a requirement, it is necessary that all the information needed for the complete explanation of the main verb (process word) is contained in the requirement. If questions about the process are still open, the corresponding information must be detected and added to the requirement. Scrutinize the main verb using the typical wh-questions. If you cannot answer all the questions with the given

Mapping of the Sophist RE-Rules vs. the TRC Systems Engineering Suite Quality metrics

- ➢ In this version of the library, most of the rules are covered by one single quality metric, except for rules **12, 14 and 15**, each covered by 2 quality metrics. Rules 16 and 18 can be addressed by the tools but were not implemented because they rely on specific context for the requirements to be analysed.

All rights reserved © The REUSE Company 2020

Mapping of the rules with RQA Quality metrics

#Rule	Priority Level	Rule Description	Quality metric name	Metric type
1	Medium	Write every requirement in active voice	**R010-M : Write every requirement in active voice**	Non-parameterized
2	Medium	Express every process using unambiguous main verbs.	**R020-M : Express every process using unambiguous main verbs**	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.	**R030-H : Use only well defined process verbs**	Parameterized - Cluster
4	Low	Dissolve light-verb constructions and define the required process, which represents the system's behaviour using a "good" main verb	**R040-L : Use a defined main verb**	Parameterized - Cluster
5	Medium	Write exactly one requirement statement for each main verb	**R050-M : Use one modal verb per requirement**	Parameterized - Cluster
6	High	Ask wh-questions about the main verb	**R060-H : Analyze missing information regarding the main verb**	Parameterized - Cluster
7	Medium	Analyse missing information on the adjective or adverb which is derived from a process verb and add information if necessary.	**R070-M : Missing one adjective in the information derived from the process verb**	Parameterized - Term tag
8	Medium	Formulate adjectives in a way that can be measured or tested	**R080-M : Avoid vague adjectives**	Parameterized - Cluster
9	Low	Formulate separate requirements for non-functional aspects if these aspects are independent or needed as a constraint for several functionalities	**R090-L : Write only one paragraph for each requirement**	Non-parameterized

All rights reserved © The REUSE Company 2020

Mapping of the rules with RQA Quality metrics

#Rule	Priority Level	Rule Description	Quality metric name	Metric type
10	Medium	Question the used numerals and quantifiers.	**R100-M : Use consistent quantifiers**	Parameterized - Relationship not SCM compliant
11	Medium	Clarify missing numerals and quantifiers.	**R110-M : Avoid using numbers without quantifiers**	Parameterized - Pattern group and pattern matching
12	High	Question vague nouns.	**R120-H : Avoid incorrect spelled words**	Non-parameterized
			R121-H : Only use controlled and defined vocabulary	Non-parameterized
13	Low	Replace formulations that describe possible or impossible situations.	**R130-L : Avoid universal and absolute expressions**	Parameterized - Cluster
14	Low	Remove subordinate clauses that contain irrelevant information for the requirement	**R140-L : Avoid open ended clauses**	Parameterized - Cluster
			R141-L : Avoid speculative sentences	Parameterized - Cluster
15	Low	Shorten or eliminate flowery expressions or phrases that are irrelevant for your requirement.	**R150-L : Avoid flowery expressions**	Parameterized - Cluster
			R151-L : Avoid imprecise quantifiers	Parameterized - Cluster
17	Medium	Analyse exceptions to the usual behaviour of the system and extend the requirement resp. write an additional requirement.	**R170-M : Define system states in all possible cases***	Completeness metric : Terminology coverage metric

*This metric requires the definition of the system states in the Knowledge Manager tool

All rights reserved © The REUSE Company 2020

Quality metrics configuration : Weights

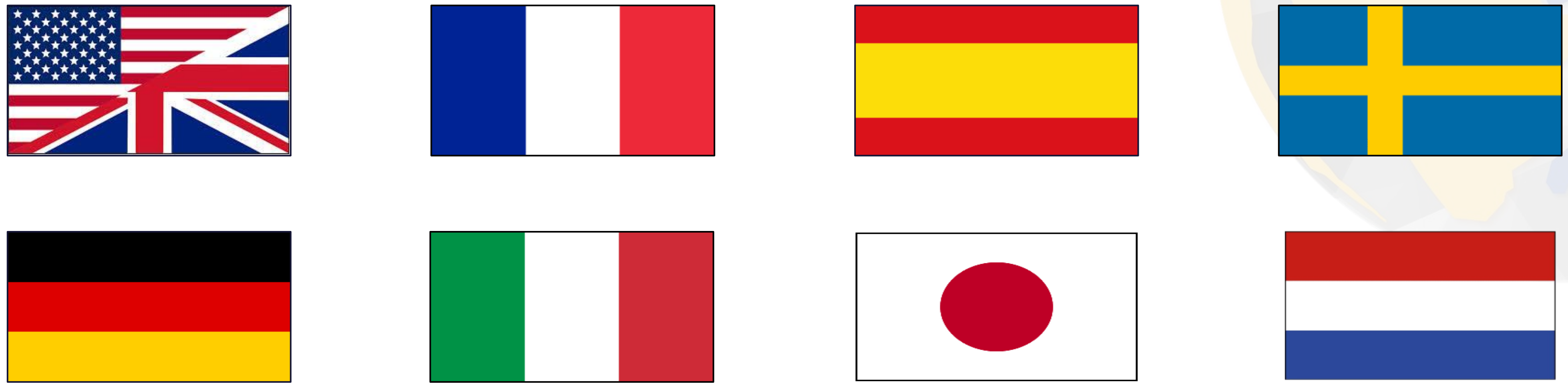
- 3 priority levels in the SOPHIST RE-Rules
  - High
  - Medium
  - Low

- Default weight configuration of the quality metrics to take into account priority levels :

High	Weight = 4
Medium	Weight = 2
Low	Weight = 1

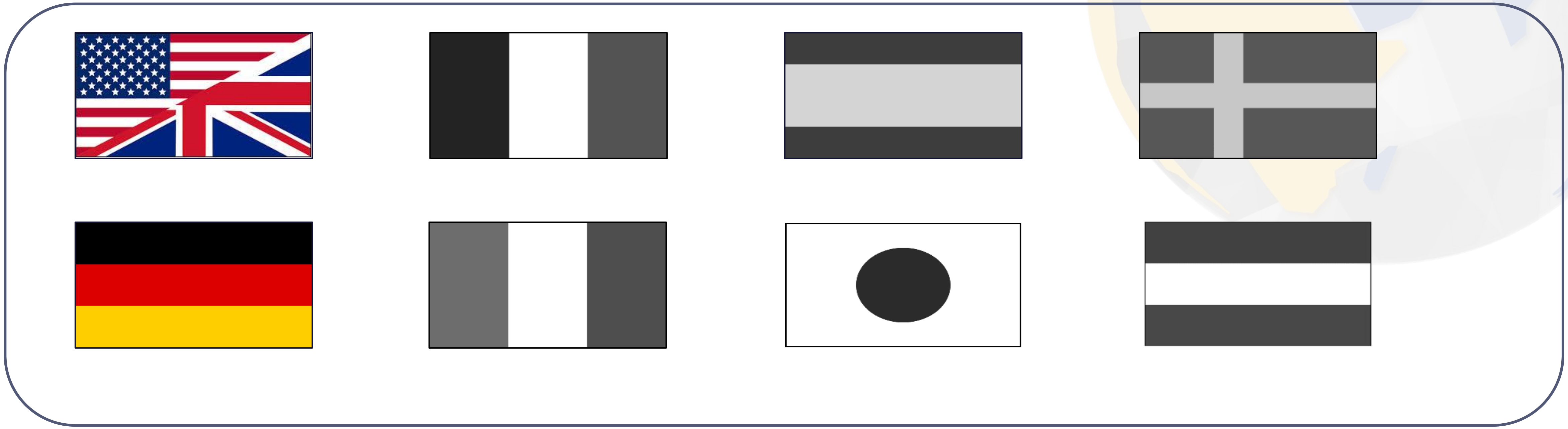
All rights reserved © The REUSE Company 2020

Languages supported by the TRC Systems Engineering Suite



All rights reserved © The REUSE Company 2020

Languages available for this Knowledge Library



The SOPHIST MASTER patterns Knowledge library is available in <u>English</u> and <u>German</u>

All rights reserved © The REUSE Company 2020

Demo of the library

- Extracted from the webinar “Ensuring completeness, consistency, and correctness with the MASTER patterns by Sophist and RAT – Authoring Tools”

- Youtube video link : https://youtu.be/LvUhKSirusE?t=1810

All rights reserved © The REUSE Company 2020

the REUSE company