The ECSS Knowledge Library

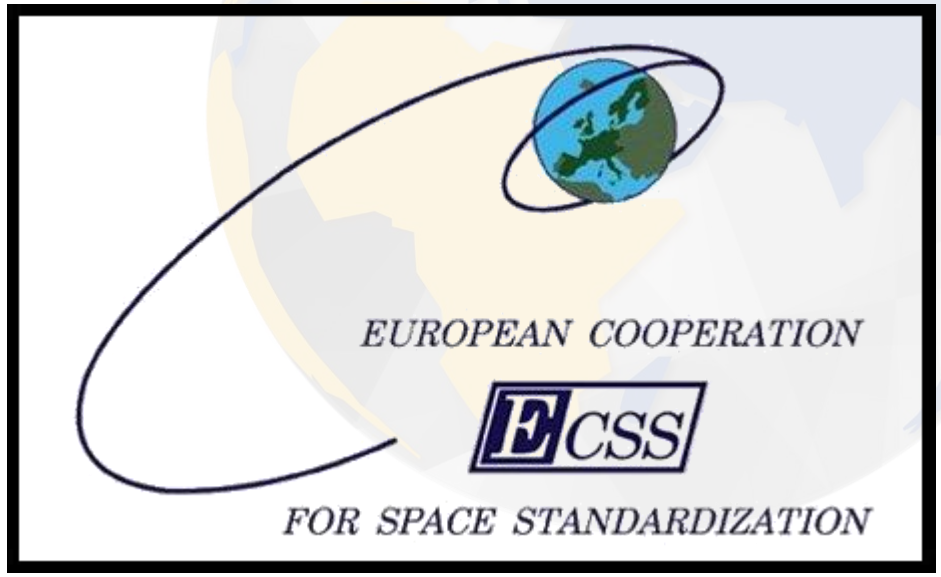
Tuesday, 11 December 2018

Table of contents

- Introduction to the ECSS and the main standards covered in the library
- The content of the ECSS Knowledge Library
- Next steps with the library
- Live demo of the Library

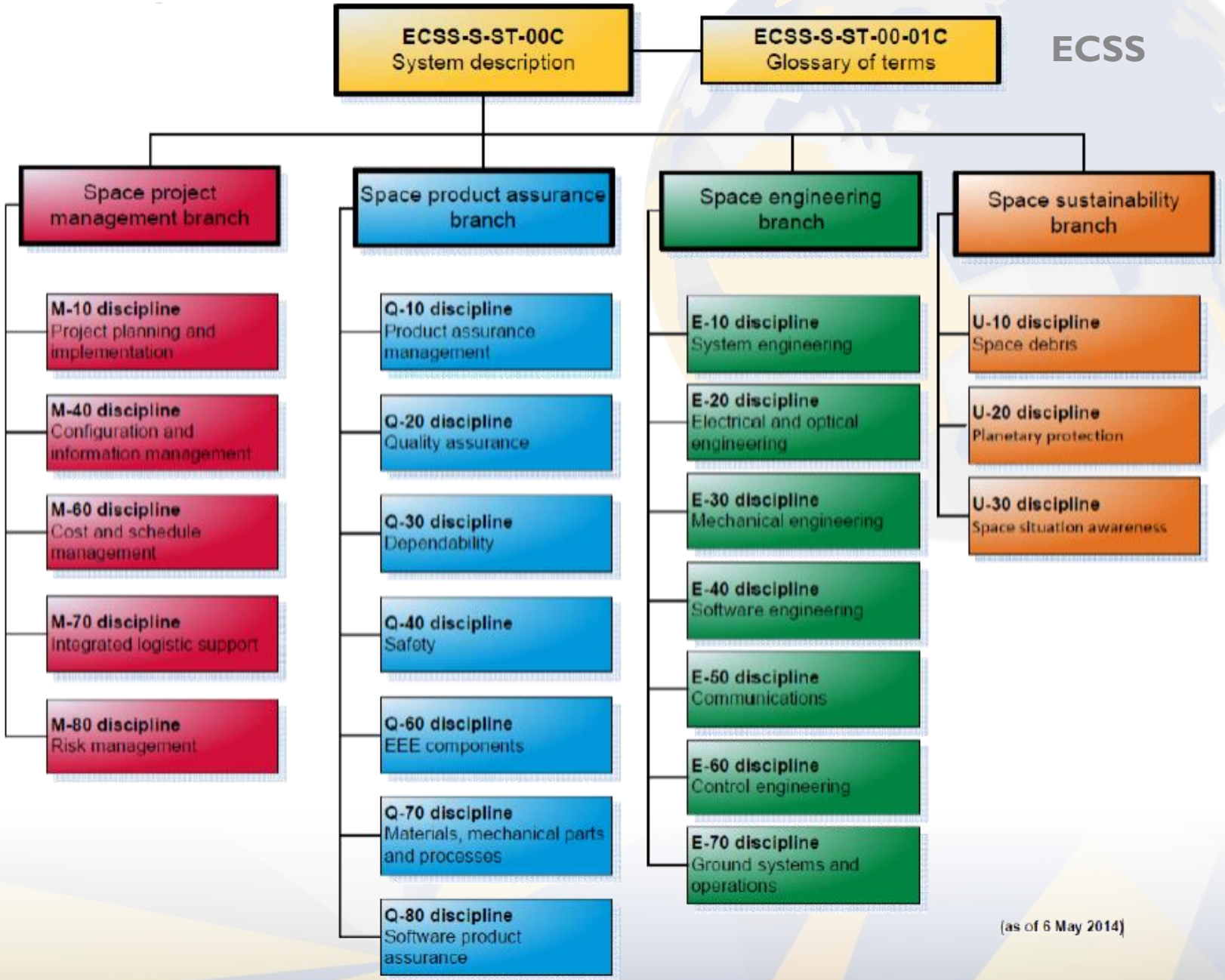
What is the ECSS

- *European Cooperation for Space Standardization (ECSS): ecss.nl*
- *founded in 1996 as an initiative established to develop a coherent, single set of user-friendly standards for use in all European space activities,*
- *thus providing the European space community with an integrated set of space-specific standards*



What is the ECSS

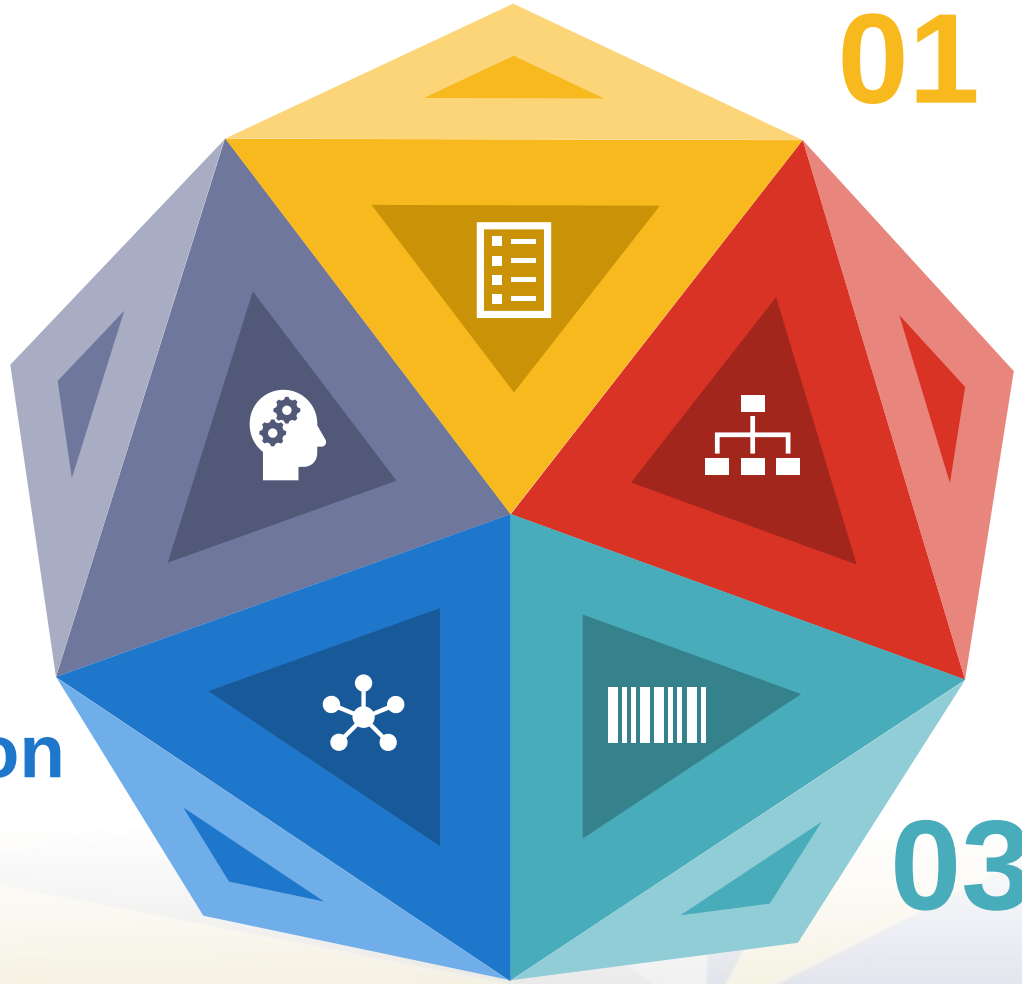
- Standards, Handbooks and Technical Memoranda
- Glossary of terms
- 4 technical disciplines



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

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

Example of Knowledge Library

Vocabulary

Aircraft | Ground segment | System | Operate | Temperature | Environment | Pressure

Architectures - Conceptual model

Temperature → "Operation Range " → [-160ºC , +160ºC]

<Environment>
- Temperature
- Pressure

<Operation>
- Operate
- Work

<System>
- Aircraft
  - A/C 2233
  - A/C 2244

"Greater than (>) "

Patterns

<System> | Shall | <Operation> | At | Minimum | <Environment> | Of | NUMBER | MEASUREMENT UNIT

Formalization

The aircraft shall be able to operate at a minimum temperature of -170º C → Temperature → "Greater than (>)" → -170 | ºC

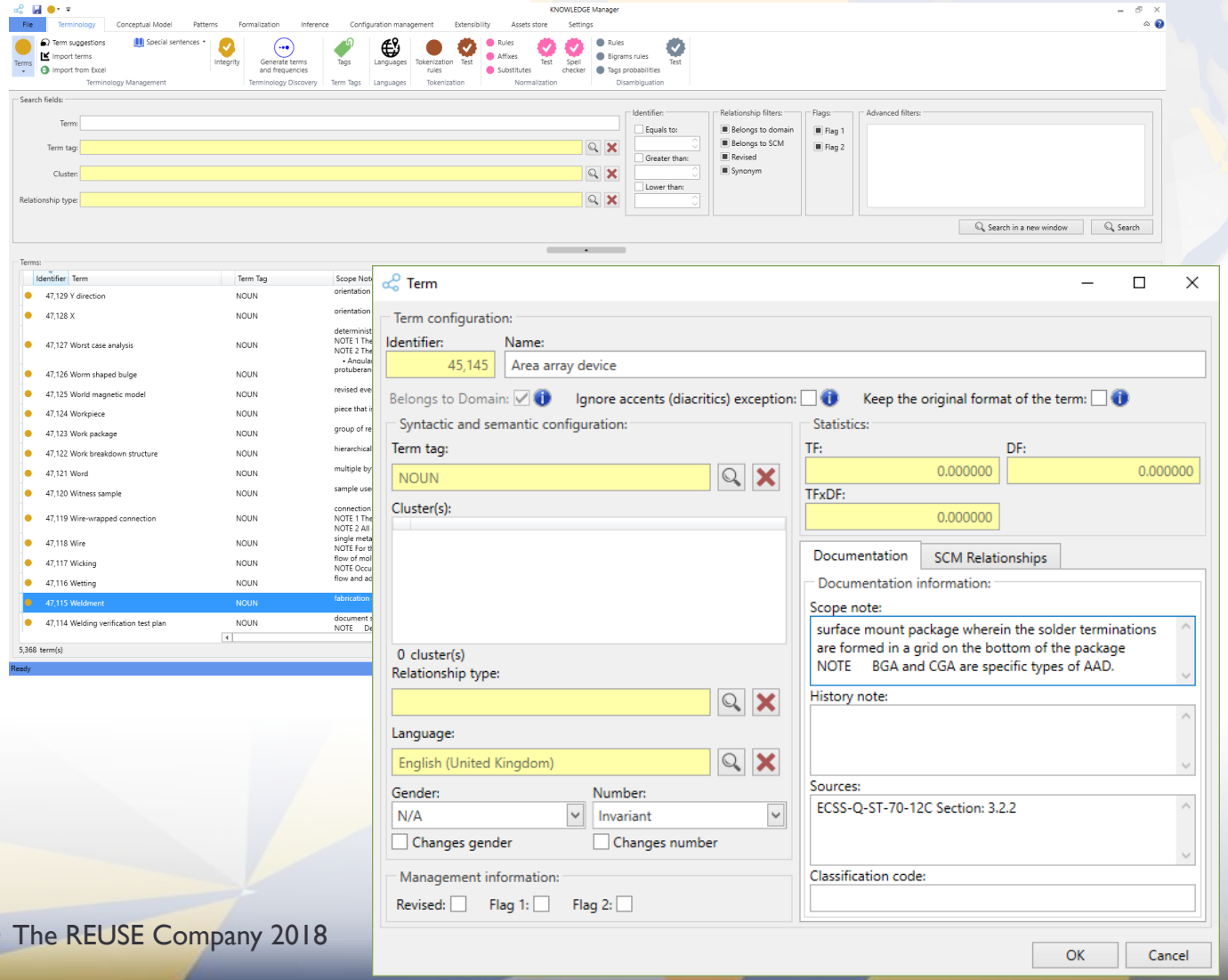
Reasoning

If NUMBER Lower than (<) -160º ºC Or NUMBER Greater than (>) +160º ºC → ✗

ECSS Glossary

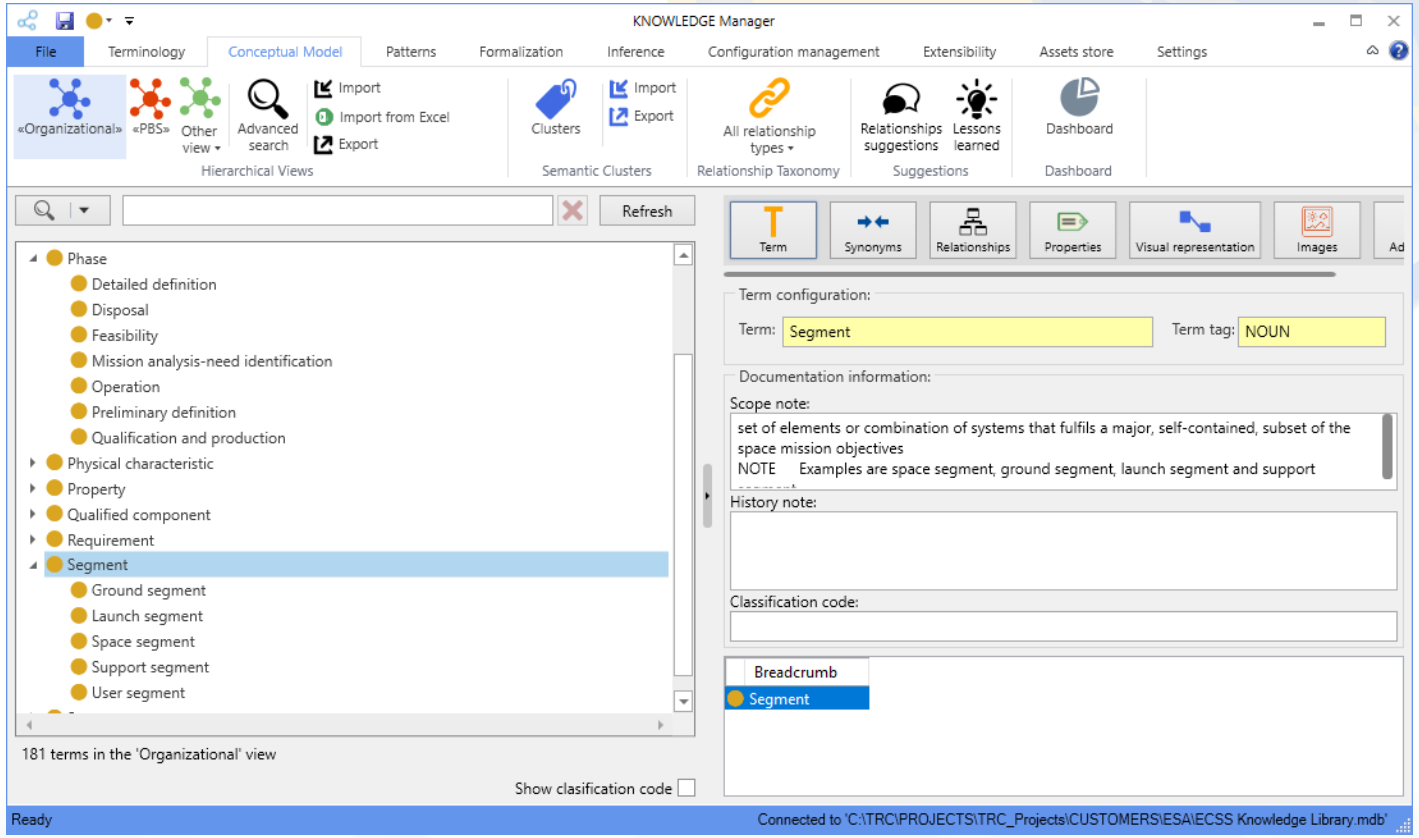
- **ECSS-S-ST-00-01C – Glossary of terms (1 October 2012)**
- This document controls the definition of all common terms used in the European Cooperation for Space Standardization (ECSS) Standards System. Terms specific to a particular ECSS Standard are defined in that standard.

- **Provides a consistent way to name and understand all the concepts across the industry**
- **The system can highlight and link references to these entries in the body of the documents**



Taxonomy of terms and other relationships

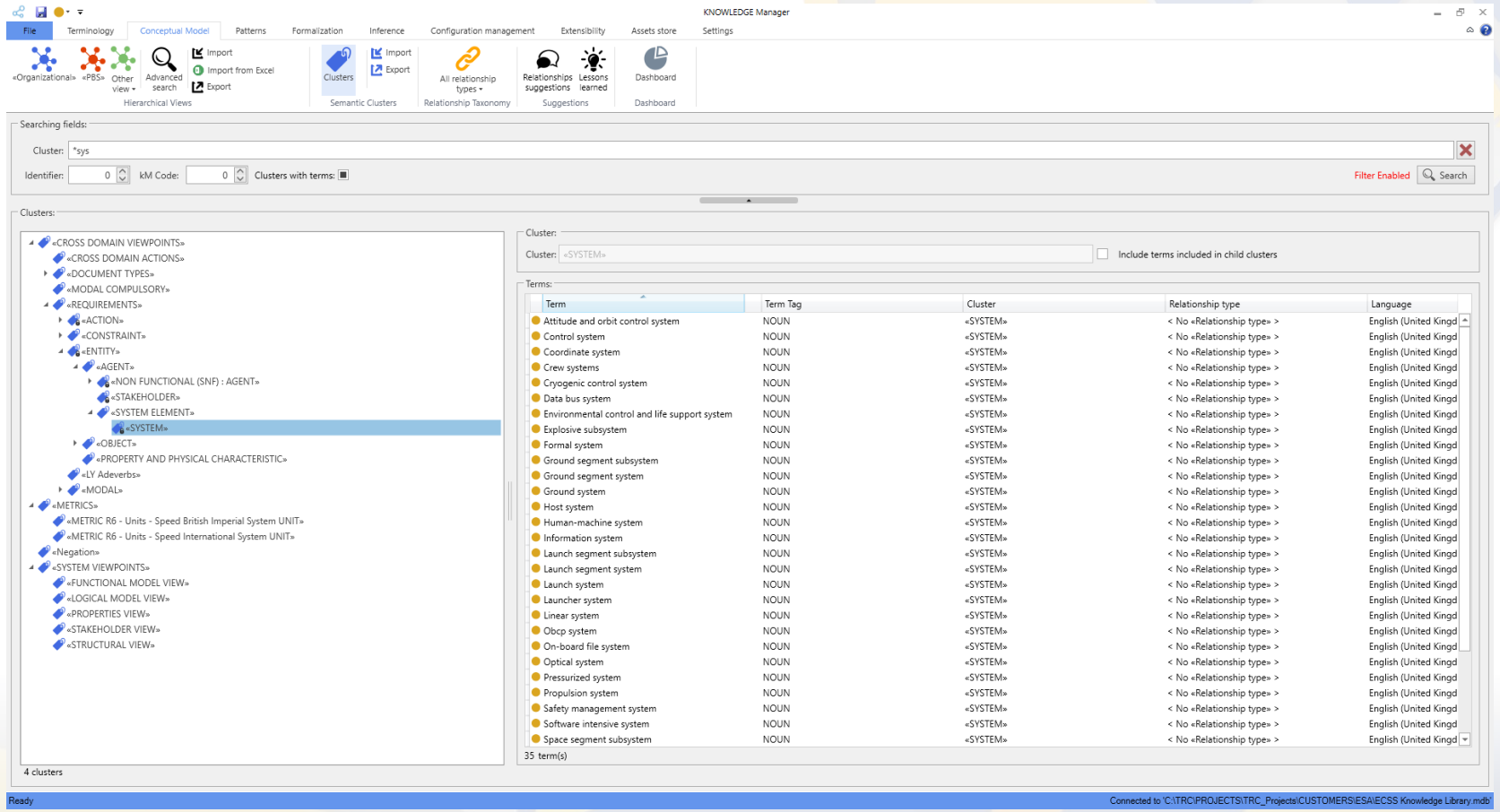
- Based on the glossary standard, and some other standards
- Represented as relationships in the Knowledge Library
- **Provides means to propagate queries in further reuse stages or just for information retrieval**



Clustering the terms of the glossary

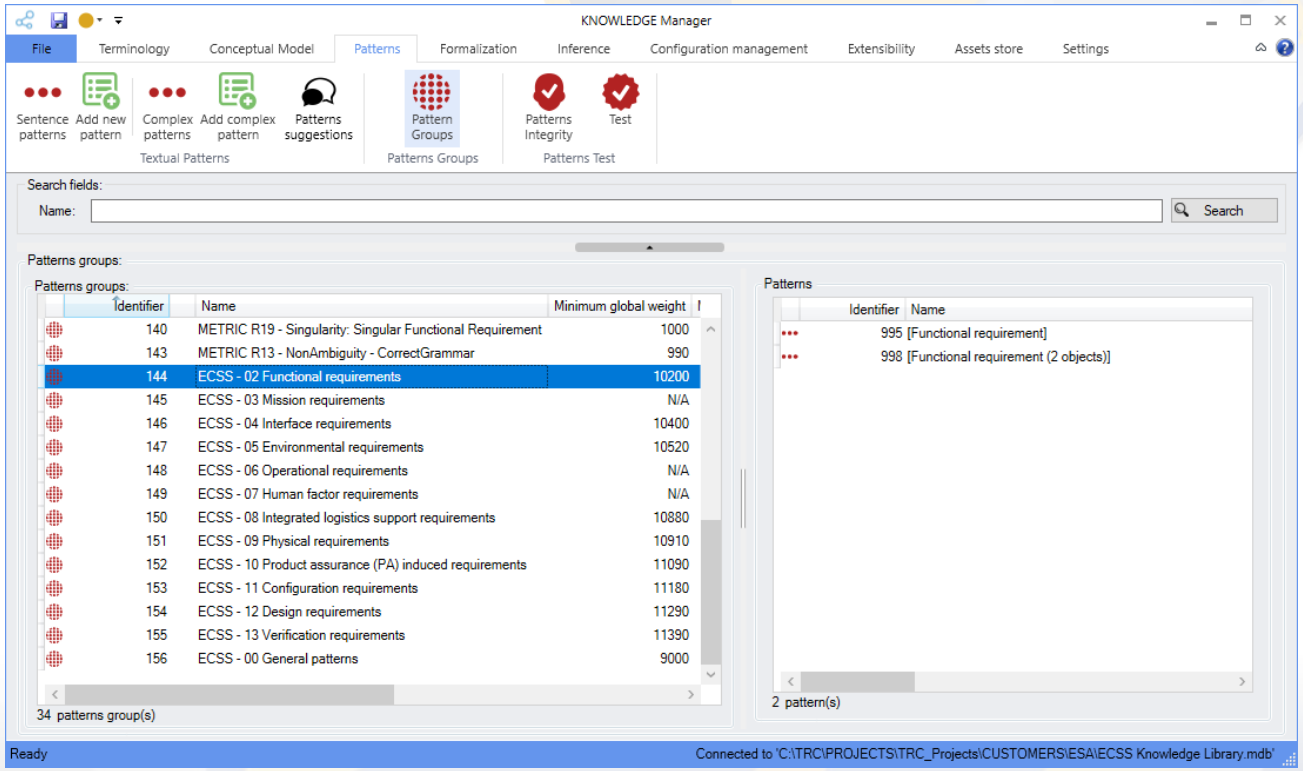
- **Clustering:** according to the semantic of the terms in the library

- **Provides means to fit the textual paterns and help authors while the write requirements or other types of textual assets**



Taxonomy of types of requirements

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
  - This Standard provides an overview of the purposes and positions of the technical requirements specification, defines the different types of requirements, and defines requirements on the TS and on its requirements.
  - This Standard is applicable to all types of space systems, all product elements, and projects.
- **Provides a reference during the requirements authoring phase**



Requirements patterns

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
  - This Standard provides an overview of the purposes and positions of the technical requirements specification, defines the different types of requirements, and defines requirements on the TS and on its requirements.
  - This Standard is applicable to all types of space systems, all product elements, and projects.

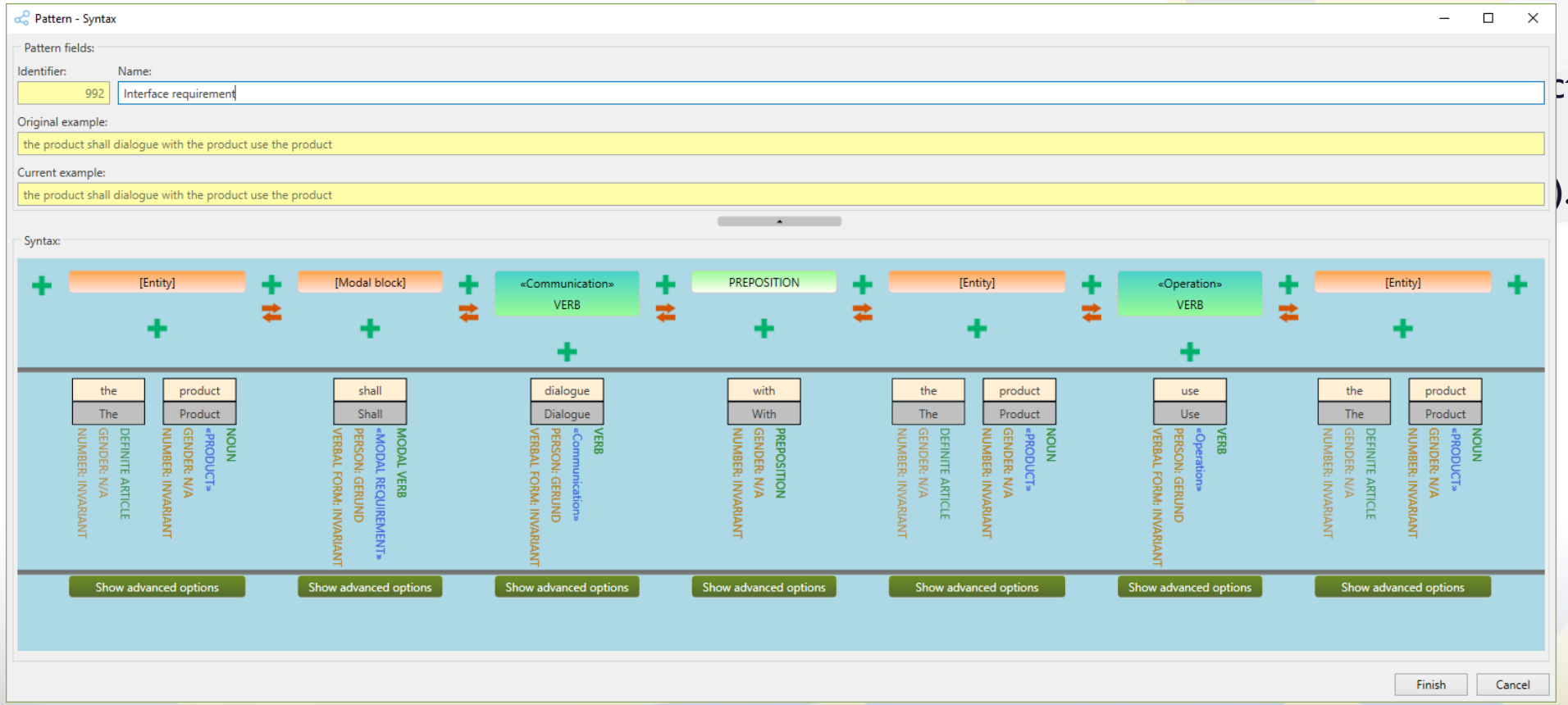
- **Provide a reference during the requirements authoring phase**
- **Allows an automatic parsing of textual and unstructured documents**
- **Some specific patterns allow:**
  - **Automatic extraction of properties: consistency checking**
  - **Automatic generation of models: SysML, ORM (Object Role Modeling)…**

Requirements patterns

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009):**
  - **Example of requirement pattern: Interface requirement**
    - <Entity> <Modal> <Communication/VERB> <Entity> <Operation_VERB> <Entity>
    - Requirements related to the interconnection or relationship characteristics between the product and other items.
    - NOTE 1 This includes different types of interfaces (e.g. physical, thermal, electrical, and protocol).
    - Example: "The product shall dialogue with the ground segment using telemetry"

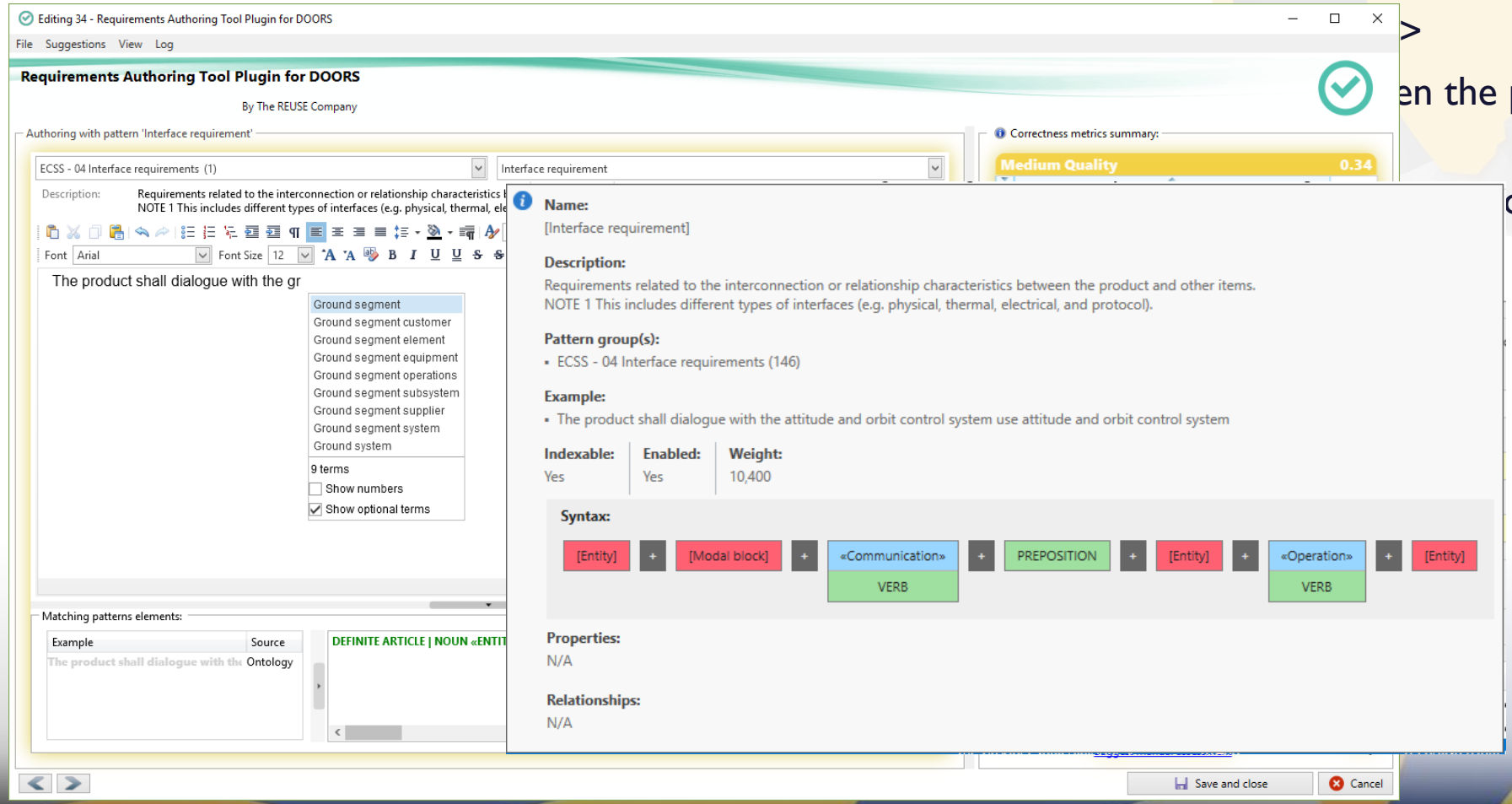
Requirements patterns

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009):**
  - **Example of requirement pattern: Interface requirement**



Requirements patterns

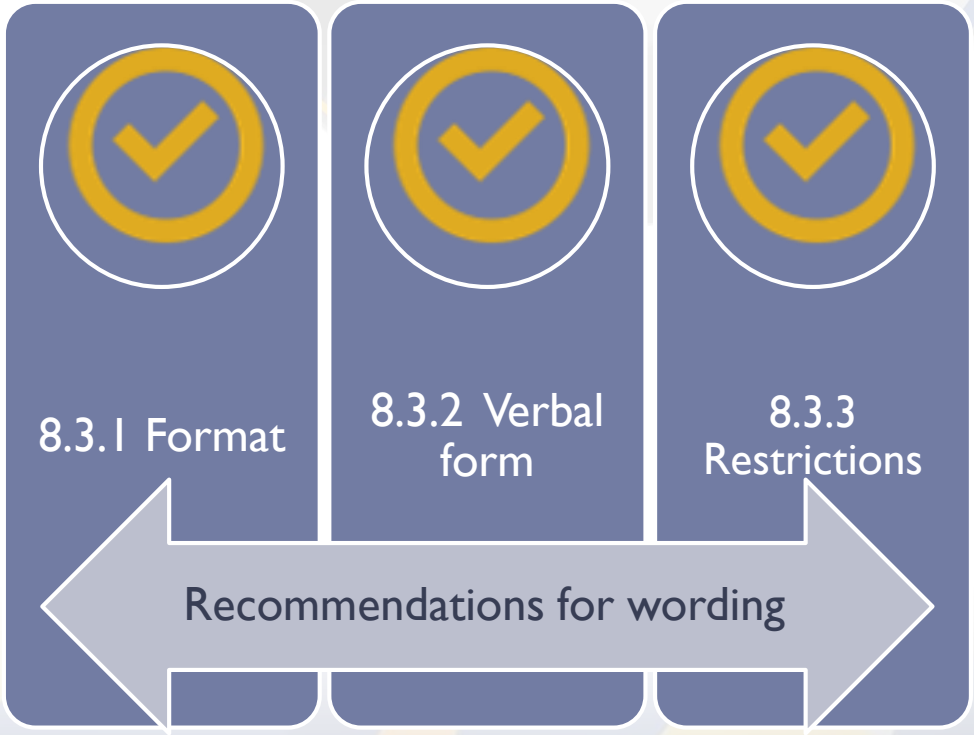
- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009):**
  - **Example of requirement pattern: Interface requirement**



Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**

8.2.1 Performance
8.2.2 Justification
8.2.3 Configuration management and traceability
8.2.4 Ambiguity
8.2.5 Uniqueness
8.2.6 Identifiability
8.2.7 Singularity
8.2.8 Completeness
8.2.9 Verification
8.2.10 Tolerance



Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **Issues:**
  - The definition of the Quality rules in this standard is very abstract

> **8.2.4 Ambiguity**
>
> a. The technical requirements shall be unambiguous.
>
> **8.2.5 Uniqueness**
>
> a. Each technical requirement shall be unique.

- **Solution:**
  - Mixed with the rules in the INCOSE Guide and other standards for Writing Requirements

- **Provides a reference about the rules to be met for each requirement.**
- **Assists the author of the requirements, and provides means and evidences for the inspection of the requirements**

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.1 Performance:**
  - a. Each technical requirement shall be described in quantifiable terms.
  - b. If necessary to remove possible ambiguities of a given performance requirement the method used to determine the required performance shall be indicated in the requirement itself.

- **Rules:**
  - Avoid unprecise quantifiers
  - Force a performance attribute

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.2 Justification:**
  - a. Each technical requirement should be justified.
  - b. The entity responsible of the technical requirement shall be identified.
  - c. The entity responsible of the specification shall define what part of the justification shall be included in the specification as informative material.

- **Rules:**
  - Justification attribute
  - Detection of the entity responsable for the technical requirement

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.3 Configuration management and traceability:**
  - a. Each technical requirement shall be under configuration management.
  - b. All technical requirements shall be backwards-traceable.
  - c. All technical requirements shall be forward-traceable.

- **Rules:**
  - Detection of back and forward traces

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.4 Ambiguity:**
  - a. The technical requirements shall be unambiguous.

- **Rules:**
  - Detection of ambiguous words and expressions
  - Detection of passive voice or conditional voice vs. active voice
  - Force a verb as the main action of the requirement
  - Detection of inconsistent measurement units
  - Each number must be followed by a measurement unit
  - Readability and incorrect punctuation
  - Detection of incorrect spelling
  - Detection of pronouns
  - Use of indefinite terms or acronyms

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.5 Uniqueness:**
  - a. Each technical requirement shall be unique.

- **Rules:**
  - Detection of overlapped requirements

Requirements Quality rules

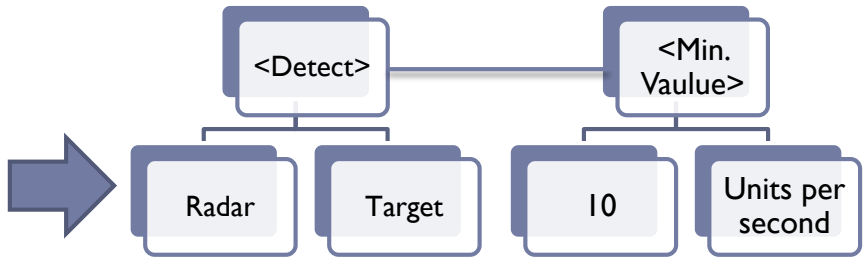
- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.5 Uniqueness:**

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

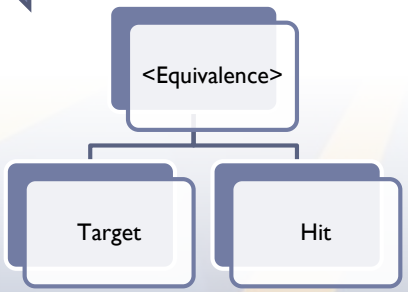
StR0168 The targets shall be detected by the Electromagnetic sensor with a frequency not lower than 10 units per second

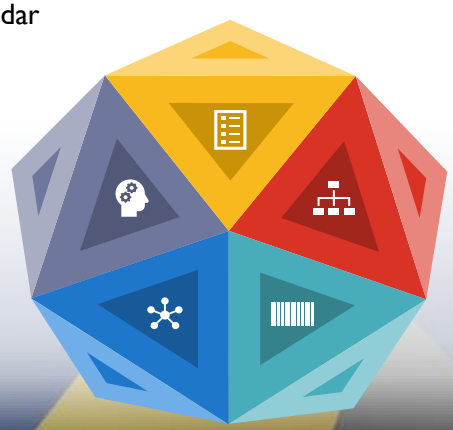
SyR0451 The Radar shall identify hits at a minimum rate of 10 units per second



<Detect>
Identify
Discover
Detect
Find
…

Taxonomy:
System
.Electromagnetic device
..Electromagnetic sensor
…Radar





Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.6 Identifiability:**
  - a. A technical requirement shall be identified in relation to the relevant function, product or system.
  - b. A unique identifier shall be assigned to each technical requirement.
  - c. The unique identifier should reflect the type of the technical requirement.
  - d. The unique identifier should reflect the life profile situation.
  - NOTE In general a technical requirement is identified by, for example, a character or a string of characters, a number, or a name tag or hypertext.

- **Rules:**
  - The subject of the requirement expression must be a: function, product or system
  - Force the unique identifier attribute
  - Check the content of the ID attribute: by means of a regular expression

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.7 Singularity:**
  - a. Each technical requirement shall be separately stated.
  - NOTE Technical requirements are single or separately stated when they are not the combination of two or more technical requirements.

- **Rules:**
  - Forcing a single expression: based on patterns
  - Ensuring the proper length (in words and in paragraphs)
  - Avoiding combinators: *and/or, as well as, but also, on the other hand, then, meanwhile, and, or…*
  - Avoid open-ended expressions: *etc, and so on…*
  - Avoiding enumerations
  - Avoiding too much detail in the requirement expression (avoid parenthesis)
  - Avoid the rationale in the requirement expression: *in order to, justify, so that, thus, thus allowing…*

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.8 Completeness:**
  - a. A technical requirement shall be self-contained.
  - NOTE A technical requirement is self-contained when it is complete and does not require additional data or explanation to express the need.

- **Rules:**
  - Conformance with the suggested patterns

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.9 Verification:**
  - a. A technical requirement shall be verifiable using one or more approved verification methods.
  - NOTE A technical requirement is verifiable when the means to evaluate if the proposed solution meets the requirement are known.
  - b. Verification of technical requirements shall be performed in conformance with ECSS-E-ST-10-02.

- **Rules:**
  - Detection of non-empty value in the verification attribute

Requirements Quality rules

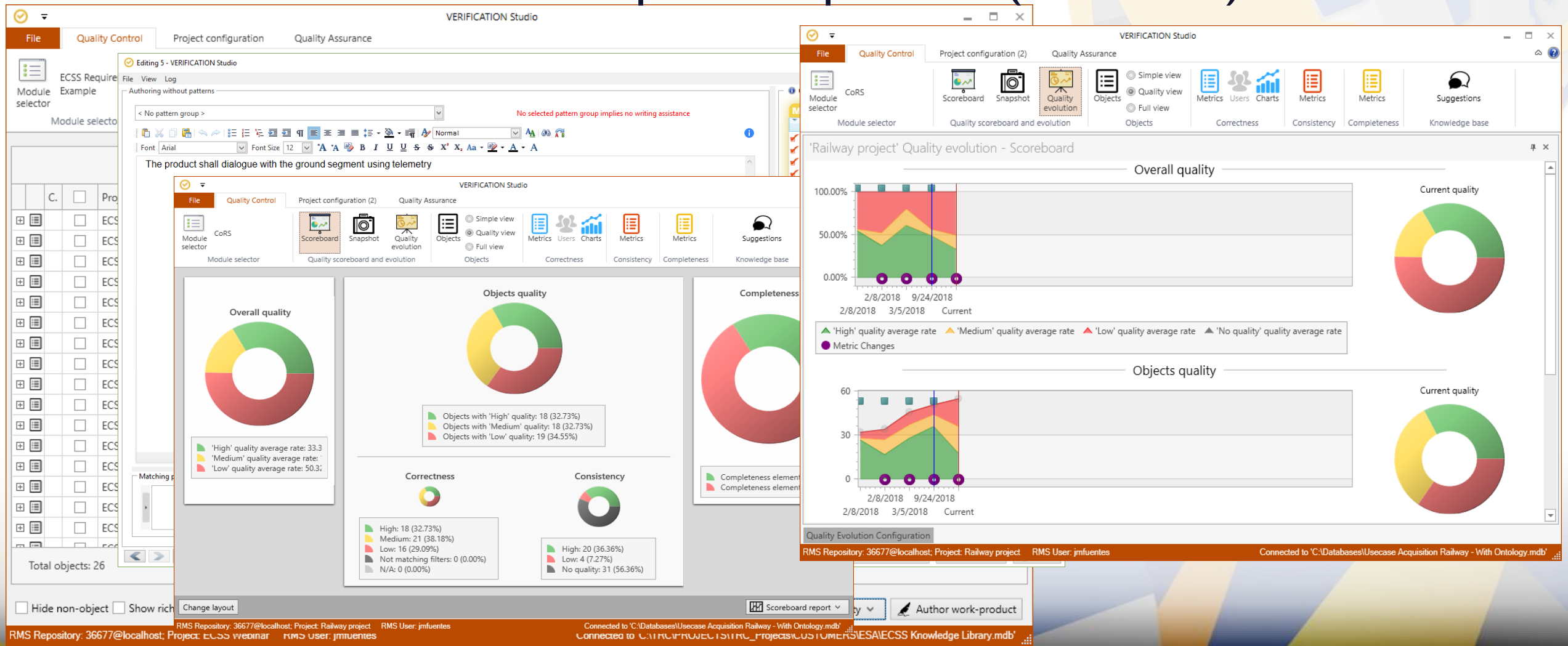
- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**
- **8.2.10 Tolerance:**
  - a. The tolerance shall be specified for each parameter/variable.
  - NOTE The technical requirement tolerance is a range of values within which the conformity to the requirement is accepted.

- **Rules:**
  - Pattern to detect a parameter value without tolerance. For instance:
    - (30 +/- 2) Kg
    - (30 +/- 5%) Kg

the REUSE company

Requirements Quality rules

- **ECSS-E-ST-10-06C – Technical requirements specification (6 March 2009)**

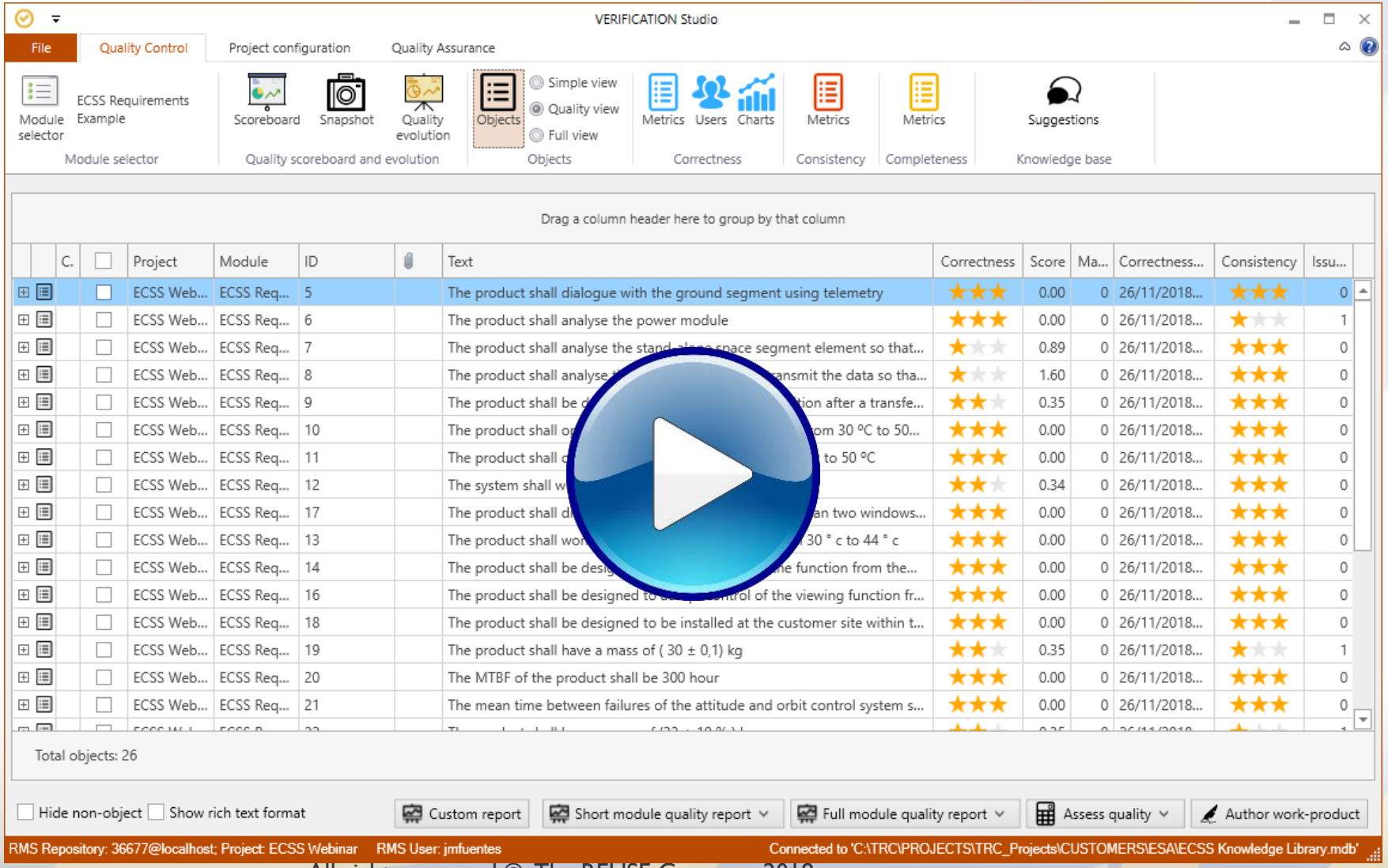


Next steps

- **ECSS-S-ST-00-01C** (ECSS Glossary)**:**
  - Integration of new terms
  - Implementation of relationships among terms and semantic clustering
- **ECSS-E-ST-10-02** and **ECSS-E-HB-10-02A** (Verification Standard and Guidelines Handbook):
  - Definition of the main entities proposed in these documents: verification approach, method, level, stage…
  - Support to the verification process in our tool VERIFICATION Studio: planning, execution, reporting, control and close-out
  - Implementation of all the reports suggested in these standards

- Other capabilities:
  - Semantic search of requirements based on: the information managed in the ontology, the paterns already generated
  - Advanced traceability
  - Generation of models

Live demo



the REUSE company