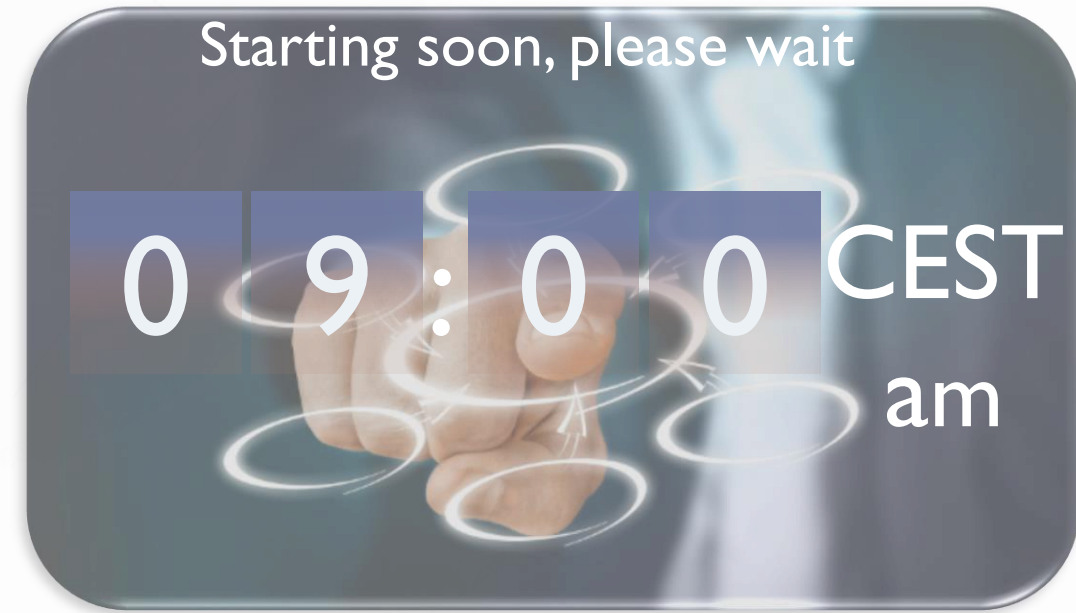


Introduction: Webinar rules

EARS

A practical approach

- Webinar rules:
 - You'll be muted all along the Webinar
 - There's a chatting box to ask your questions or send your comments when you want
 - Please address these comments and questions to the user "The REUSE Company" and not to the presenters directly
 - If you have any technical issue please use this chatting box, or mail us at: support@reusecompany.com
 - The Webinar will be recorded. A link to the recording will be sent to you within few days



EARS: Easy Approach for Requirements Syntax:

A practical approach



Alistair Mavin (Mav)
mav@alistairmavin.com



José M. Fuentes
The REUSE Company
jose.fuentes@reusecompany.com



THE
REUSE
COMPANY

Contents

- Brief introduction to The REUSE Company
- Brief introduction to the presenters
- Introduction to EARS: Easy Approach to Requirements Syntax
- Mapping EARS into a knowledge library
- Possible uses cases
- Live demo
- Q&A

Introduction to The REUSE Company



The presenters



Alistair Mavin (Mav)

- **Current position:** Independent Requirements Consultant
- Over 20 years' experience in requirements engineering
- Lead author of EARS notation
- Over 20,000 reads of papers on Researchgate
- Experience in RE and Systems Engineering in aerospace, automotive, defence, industrial plant design, rail and software systems
- Provides training, coaching and consulting in requirements engineering and EARS

The presenters



José M. Fuentes

- **Current position:** Chief Operating Officer at The REUSE Company
- Product manager of the Systems Engineering Suite tools during the last 5 years
- INCOSE CSEP Certified
- Graduated in the INCOSE Institute for Technical Leadership
- Member of the board of AEIS – the Spanish chapter of INCOSE
- Active contributor to the INCOSE Guide for Writing Requirements

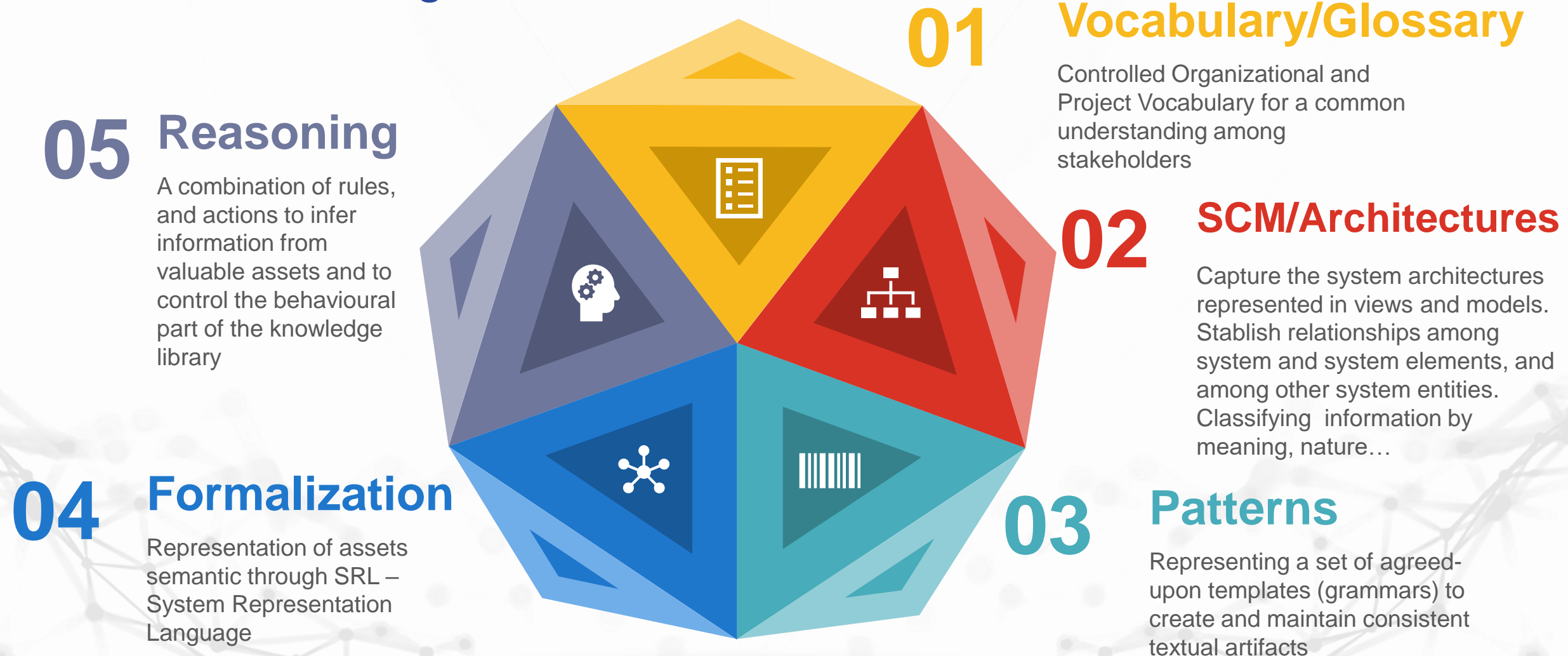
EARS

Introduction to **EARS**



Mapping EARS in a Knowledge library

What is a Knowledge Base



Vocabulary

Shuttle

Columbia

Discovery

System

Operate

Temperature

Environment

Pressure

shall

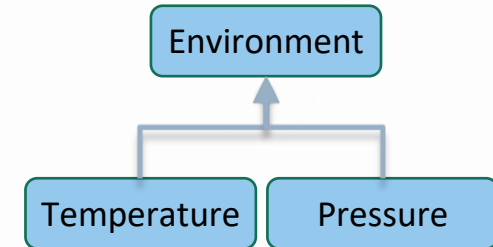
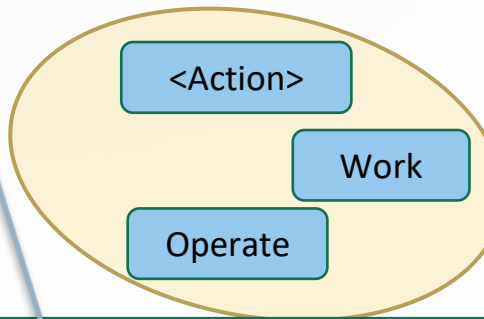
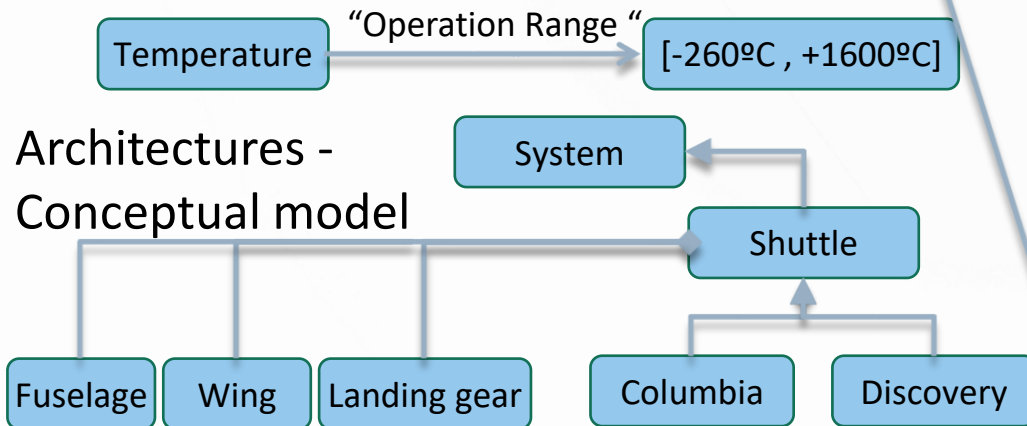
The

of

at

Lower

Architectures - Conceptual model



"Greater than (>)"

Patterns

<System>

Shall

<Action>

At

«Minimum»

<Environment>

Of

NUMBER

MEASUREMENT
UNIT

Formalization

The Columbia shall be able to operate at
a minimum temperature of 10° K

Temperature

"Greater than (>)"

-263,15

°C

Reasoning

If

NUMBER

Lower than (<)

-260°

°C

Or

NUMBER

Greater than (>)

+1600°

°C



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.



Layer LI: vocabulary

- › The library includes all the *common English* content
- › Only the domain vocabulary needed for the examples
- › How to provide your domain specific vocabulary?:
 - › Use the import options of KM



Layer L2: SCM

- › Almost empty wrt relationships
- › But most of the terms used in the examples are classified thanks to clusters
- › To distinguish *<system>* names from *<actors>*, *<properties>*, *<states>*...
- › How to provide your domain specific structures:
 - › Use the import options of KM to import relationships : e.g. PBS
 - › When importing terms, remember to add items to clusters



Layer L3: Patterns

- Patterns are based on small reusable *sub-patterns*
- Two pattern groups in the library: *patterns vs templates*
- Thus, two full KM patterns for each of the patterns defined in EARS:
 - First pattern (template): just including the main keywords, and free blocks between them

Pattern - Syntax

Pattern fields:

Identifier: 1,366 Name: TEMPLATE - State-driven requirement

Original example:

While the aircraft is in in-flight, the control system shall maintain engine fuel flow above 25 lbs/sec.

Current example:

While the aircraft is in in-flight, the control system shall maintain engine fuel flow above 25 lbs/sec.

Syntax:

While [TEMPLATE - In a specific state] COMMA [TEMPLATE - Ubiquitous requirements]

while While "ACTION ACTIVATION" "WHILE ACTIVATION" "STATE Keywords" NUMBER: INVARIANT

[TEMPLATE - In a specific state] WILDCARD SLOT NUMBER: INVARIANT GENDER: N/A

COMMA COMMA NUMBER: INVARIANT GENDER: N/A

the The DEFINITE ARTICLE NUMBER: INVARIANT GENDER: N/A

shall Shall MODAL VERB "MODAL COMPULSORY" PERSON: GERUND VERBAL FORM: INVARIANT

abandon Abandon VERB PERSON: GERUND VERBAL FORM: INVARIANT

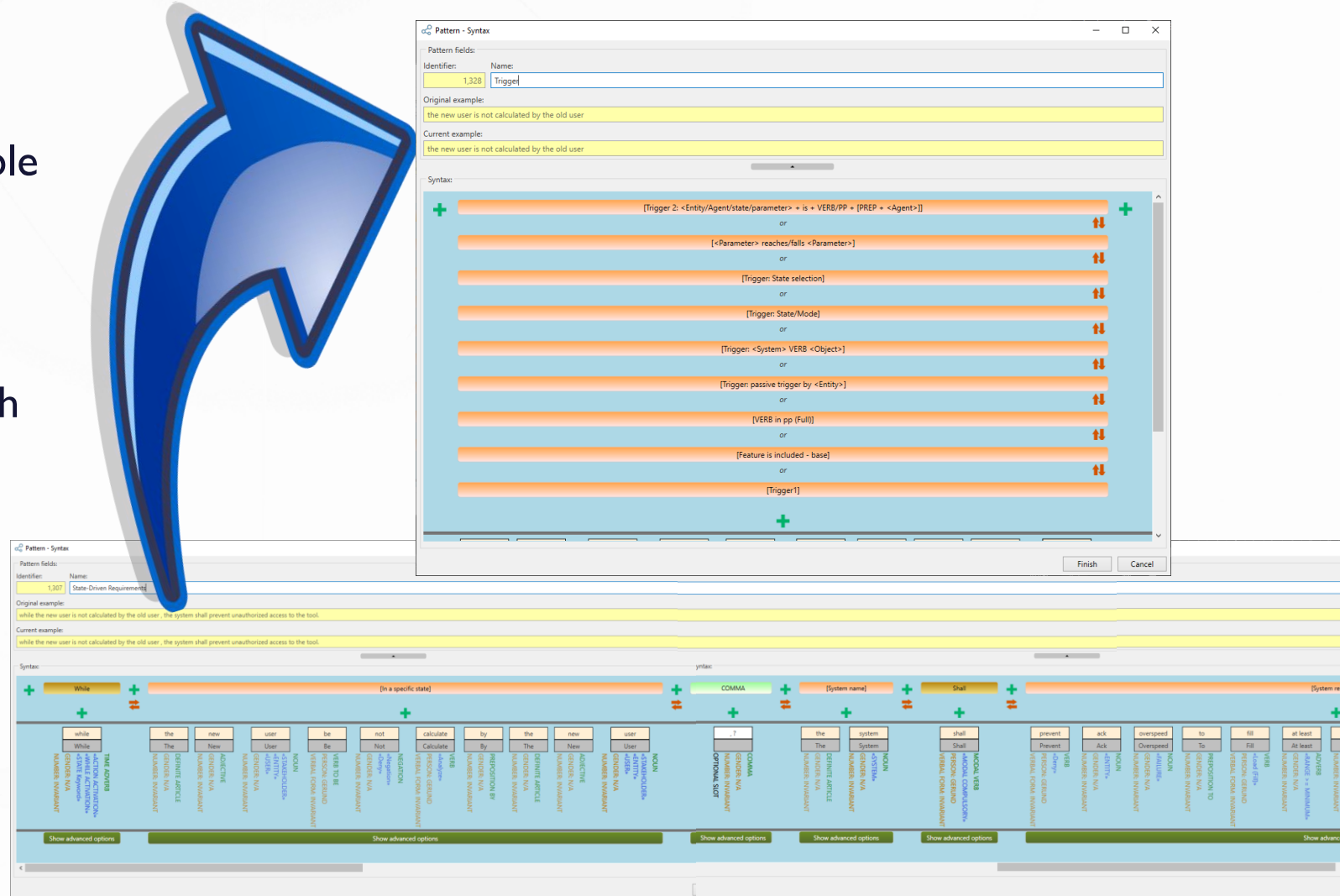
NOT_PUNCTUATION_MARK "NOT_PUNCTUATION_MARK" NUMBER: INVARIANT GENDER: N/A

Show advanced options Show advanced options Show advanced options Show advanced options

Finish Cancel

Layer L3: Patterns

- Patterns are based on small reusable *sub-patterns*
- Two pattern groups in the library: *patterns vs templates*
- Thus, two full KM patterns for each of the patterns defined in EARS:
 - First pattern (template): just including the main keywords, and free blocks between them
 - Second pattern: including guided content for each block/slot



Layer L3: Patterns - Tailoring the EARS patterns

- Why tailoring:
 - To adapt to other levels of abstraction or other types of requirements
 - To come out with other types of triggers, conditions, responses... (domain specific)
 - To allow other modal verbs
 - ...
- Where:
 - KM - Knowledge Manager
- How:
 - Adding more sub-patterns to an existing pattern
 - Adding or removing terms from clusters: e.g. the cluster of *<Modal compulsory>*
 - ...



Layer L5: Inference

➤ Metrics coping with the requirement problems identified in Big EARS (by Mav):



- Untestability
- Inappropriate Implementation
- Wordiness
- Duplication
- Omission
- Complexity
- Vagueness
- Ambiguity

Reference: BIG EARS

(The Return of “Easy Approach to Requirements Syntax”)

By A. Mavin & P. Wilkinson

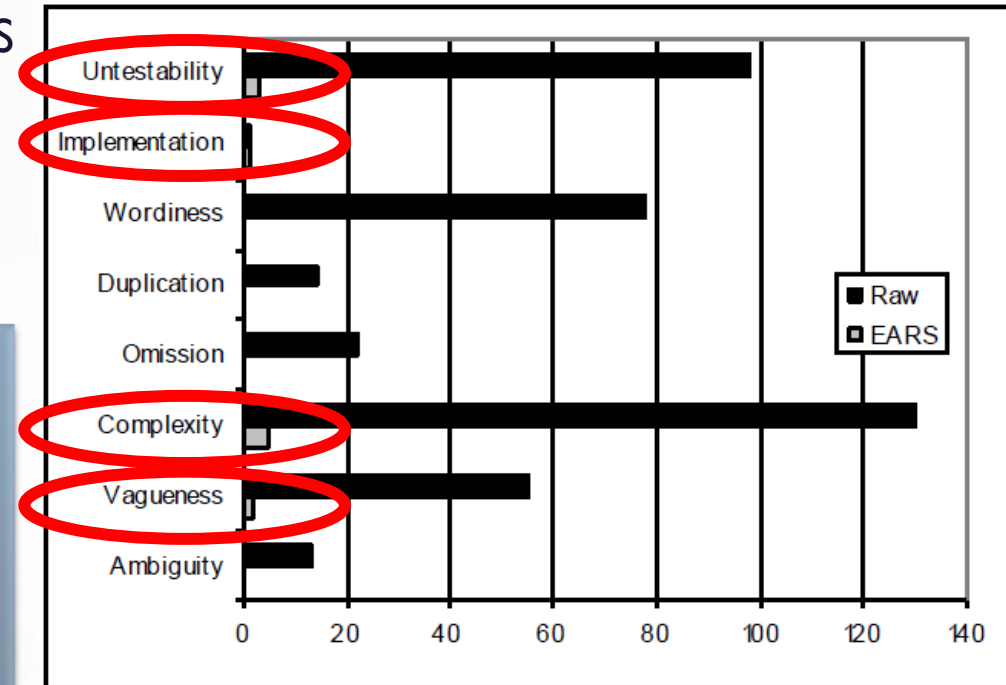
RE10

<div> <div>EARS</div> <div>  <div>THE REUSE COMPANY</div> </div> <div>  </div> </div>				
Big EARS Rule	SES Suite Metric	Metric Type	Uses filter	Domain specific
Untestability				
Requirements that, when implemented, couldn't be proven true or false.	TRC-M140: Ensure Numbers are followed by Units or noun qualifications TRC-M060: Avoid the use of Vague Verbs	Term-tag Cluster	No No	No No
Inappropriate Implementation				
Requirements that specified how the system should be	TRC-M490 - Avoid stating a solution	Special sentence	No	No
Wordiness				
Raw requirements were classified as wordy if they contained more words than each interpreted requirement. A cumulative total for all requirements in each document set was also calculated.	TRC-M330: Check the text length by counting words TRC-M390: Avoid the use of Parenthesis out of the condition block	Non-parameterized Cluster	No Yes	No No
	TRC-M630: Avoid verbose expressions	Special sentence	No	No
Duplication				
Requirements with the same meaning.	TRC-M480: Avoid overlapping among the requirements	Consistency	No	Partial
Omission				
Missing requirements, preconditions, triggers and system responses were identified by applying "Necessary and Sufficient Conditions" to determine the required system response.	TRC-M600: A requirement shall include all the mandatory blocks as defined in the pattern TRC-M610: Actors must be identified TRC-M640: Mind all different types of requirements	Pattern group Pattern Completeness	No No No	Partial No No
Complexity				
Requirements that contained two or more interrelated requirements, three or more preconditions, or two or more triggers.	TRC-M370: Multiple subject detection TRC-M375: Multiple verbs detection TRC-M360: Check the number of Modal Verbs TRC-M620: Avoid complex structures TRC-M640: Avoid multiple triggers TRC-M650: Invalid order of keywords	Pattern group Pattern Term-tag Cluster Pattern Pattern	No No No No No No	No No No No No No
Vagueness				
Requirements that lacked precision.	TRC-M540: Avoid the use of imprecise Quantifiers	Cluster	No	No
Ambiguity				
Requirements that could be interpreted in different ways. Requirements were reviewed for lexical, referential or syntactic ambiguity.	TRC-M240: Avoid incorrect spelling TRC-M250: Facilitate readability TRC-M950: Avoid the use of Vague Terms	Non-parameterized Non-parameterized Cluster	No No No	Partial No No

Layer L5: Inference

- Metrics coping with the requirement problems identified in Big EARS (by Mav):
 - Most of these problems are minimized by using EARS
 - Especially when using patterns and not templates
 - But even using patterns, some problems remain

While using EARS, when the author finds problems, the author shall use automatic inspection, in order to improve efficiency



BIG EARS

(The Return of “Easy Approach to Requirements Syntax”)

By A. Mavin & P. Wilkinson

RE10



Layer L5: Inference

- Metrics coping with the requirement problems identified in Big EARS (by Mav):
 - A third set of metrics, for the identification of main blocks in a spec.
 - Individual elements: System manes, state names, ...
 - Statements: conditions, triggers

1. Introduction

This document Will....

2. Scope

Bla, bla, bla...

3. Requirements

SyR-001 - When requested, the RQA shall generate a quality report.

SyR-002 - The RQA shall be capable to generate a quality report based on the selected quality metrics.

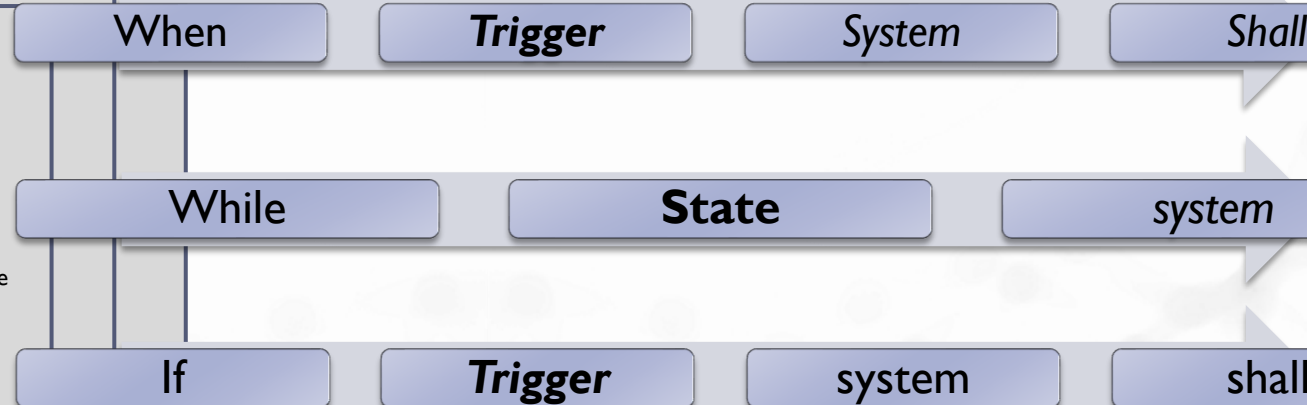
SyR-003 - While RQA is on the main screen , when the author presses the report button, RQA shall generate a quality report including the current content of the main screen.

SyR-004 - While RQA is in real-time mode, the RQA shall analyze the structure of the requirement in order to check if the requirements match with the selected pattern.

SyR-005 - While RAT is in real-time mode, when the author presses the keyboard, the RQA shall analyze the quality of the requirement.

SyR-006 - When the user connects to a requirements repository, the RQA shall retrieve the requirements modules.

SyR-007 - When the user connects to a requirements module, the RQA shall retrieve the requirements of the module.



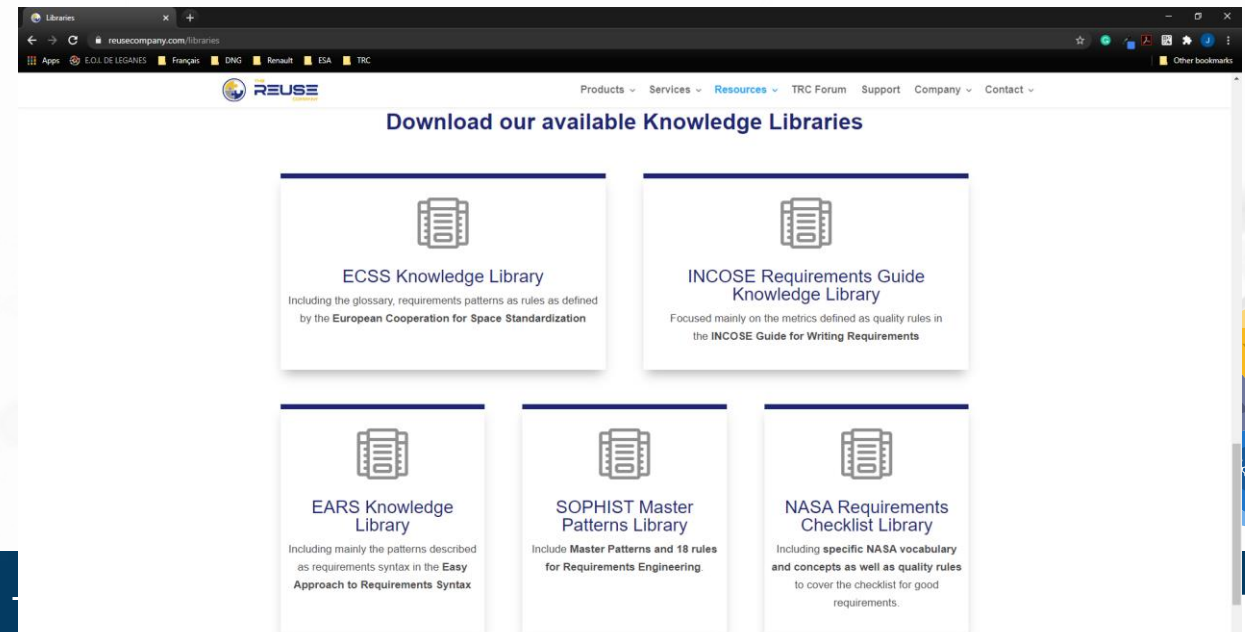
<System>
.RQA
.RAT
...

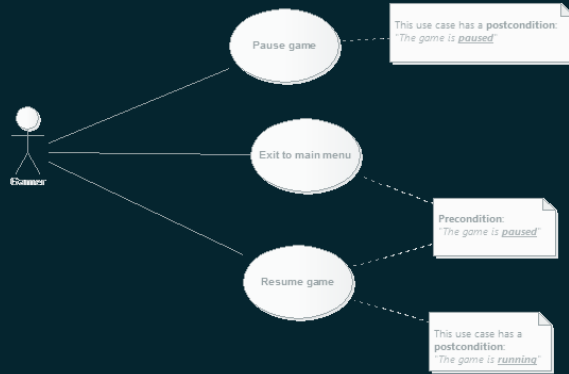
<State>
.RQA is on the
main screen
.RQA is in real-
time mode

<Trigger>
. the user
connects to a
requirements
repository

Layer L5: Inference – Tailoring the quality metrics

- Metrics coping with the requirement problems identified in Big EARS (by Mav):
 - These metrics can be tailored adapting your own rules or checklists
 - Or importing other libraries like:
 - The rules in the INCOSE GfWR
 - The rules in the NASA Systems Engineering Handbook
 - The drafting rules defined in the ECSS standards and used by ESA and the EU space industry
- All these libraries are already available at:
<https://www.reusecompany.com/libraries>





Implemented

use cases

Related use cases: the SES Suite

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



- **RQA / V&V Studio:** to setup, check and manage the quality of a requirements specification
- **Rich Authoring Tool (RAT):** to assist authors while they are creating or editing requirements
- **Knowledge Manager (KM):** to manage knowledge around a requirements specification: dictionaries, glossaries, concept maps, knowledge models, ontologies, patterns...
- **TRACEABILITY Studio:** to link together all the different types of artifacts managed with the rest of the tools

TRC WEBINARS 2020

Use cases implemented with this library

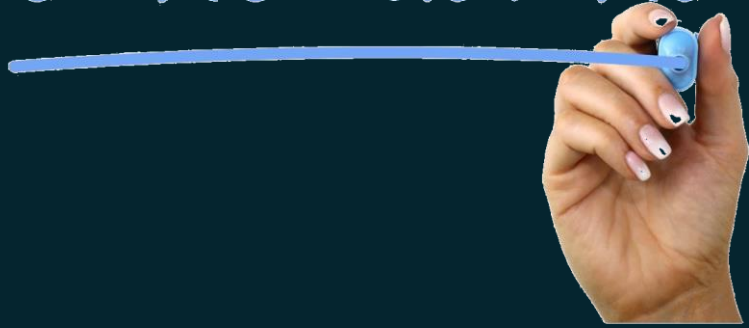
- Checking conformance of legacy requirements
- Writing new requirements following the EARS patterns
- Checking quality rules of a legacy document
- Involving other types of rules for the assessment of the quality of the requirements
- Real-time quality checking
- Tailoring the EARS patterns
- Automatic parsing of non-structured documentation
- Extraction of the main building blocks (conditions, states, system names...)



The screenshot displays an Excel spreadsheet with a table of requirements. A modal dialog titled 'Editing SyR-011 - RQA' is open, showing a list of 'EARS Patterns' and a text area for editing the requirement description. A large blue play button is overlaid on the center of the spreadsheet.

ID	Title	Description	Author	Type	CreationDate	LastModificationDate	QualityLevel
Head-001	RQA Specification		José Fuentes	Other	16/10/2020	16/10/2020	
Head-002		The current version of RQA – QUALITY Studio extends the quality analysis concept and now covers all the engineering items generated during the systems	José Fuentes	Other	16/10/2020	16/10/2020	
Head-003	RQA Requirements		José Fuentes	Other	16/10/2020	16/10/2020	
SyR-001		When requested, the RQA shall generate a quality report.	José Fuentes	Event-Driven	16/10/2020	19/10/2020	Low
SyR-002		The RQA shall be capable to generate a quality report based on the selected quality metrics.	José Fuentes	Ubiquitous	16/10/2020	19/10/2020	High
SyR-003		While RQA is on the main screen, when the author presses the report button, RQA shall generate the current content of the	José Fuentes	State-Driven	16/10/2020	19/10/2020	High
SyR-004						19/10/2020	Low
SyR-005						19/10/2020	High
SyR-006						19/10/2020	High
SyR-007						19/10/2020	High
SyR-008						19/10/2020	High
SyR-009						19/10/2020	High
SyR-010						19/10/2020	Low
SyR-011						19/10/2020	High
SyR-012						19/10/2020	High
SyR-013						19/10/2020	High
SyR-014						19/10/2020	High
SyR-015						19/10/2020	Low
SyR-016						19/10/2020	High
SyR-017						19/10/2020	High
SyR-018						19/10/2020	High
SyR-019						19/10/2020	High
SyR-020						19/10/2020	High
SyR-021						19/10/2020	Low
SyR-022						19/10/2020	Low
SyR-023		Where the RQA is executed with a trial license, the RQA shall show the label "Trial" in the status bar.	José Fuentes	Optional feature	16/10/2020	19/10/2020	High
SyR-024		When the user creates a new requirement, if the type of the requirement doesn't match the filter, the RQA shall not perform the quality analysis	José Fuentes	Event-Driven	16/10/2020	19/10/2020	Low
SyR-025		If the license of RQA is expired, while RQA is in real-time mode, then the RQA shall show a warning message "license expired".	José Fuentes	Unwanted behaviour	16/10/2020	19/10/2020	Low
SyR-026		When a requirement module is opened by an author, the RAT shall gather the requirements of the module	José Fuentes	Event-Driven	16/10/2020	19/10/2020	High

CONCLUSIONS



Main

Conclusions

Main conclusions

- › A knowledge library has been created, covering layers L3 and L5 of the ontology
- › This library is already available at:
 - › <https://www.reusecompany.com/ears-knowledge-library>
- › The SES Suite covers all the use cases described

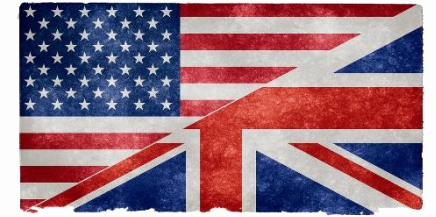


A gift for you!

- › **Orders before end of November** using the promotional code: EARS-RAT...
- › ...will get the EARS library,
- › ...and a discount of 10% on the price of the licenses.



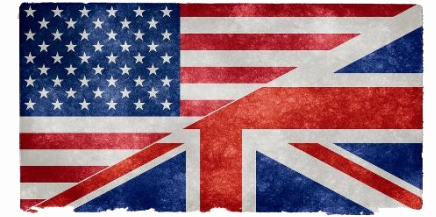




Next webinar

- **The INCOSE Guide for Writing Requirements: Raising the Ante**
- Given the importance of starting out with high-quality project documentation (e.g. concepts, needs and requirements specifications) and the benefit of the INCOSE GfWR (Guide for Writing Requirements) asset, this webinar will address an approach to enhance the implementation of the INCOSE GfWR utilizing a tailoring process together with some automation of the assessment activities.
- Even though the GfWR is a universally used and recognized reference for requirements engineers, its implementation poses various challenges. For example, requirements can be expressed at different levels of abstraction, a document might contain requirements of different types, the components addressed by those requirements may not be equally critical, ... Additionally, because the guide is thoroughly extensive, implementation of the rules in a single step turns out to be very complicated, especially considering the divergent skills of different team members, not to mention the tediousness of manual inspection based on such a large number of quality rules.
- The enhancement is based on the gradual implementation of sets of incrementally more demanding quality rules, that could address different levels of abstraction and criticality, as well as cater for different skill sets of engineers. Besides, the approach aims not only at quality inspection but also at the authoring stage, thus reducing rework and enhancing reusability while boosting performance of the requirements engineering teams. Requirements patterns (or boilerplates) represent another key factor ensuring structural consistency among requirements, while enabling uniformity of structure and content (based on domain-specific dictionaries).
- Such enhancements are not achievable without a tailored process and tool set, which will be discussed in the webinar.
- **Date:**
 - October the 21st, 2020 at 11:00 am ET
- **Hosted by:**
 - <https://www.incose.org/products-and-publications/webinar/2020/10/21/default-calendar/webinar-144-the-incose-guide-for-writing-requirements-raising-the-ante>

Next webinar



- **Textual and model requirements: working together towards the success**
- While Capella is an MBSE environment, it's also clear that those models, or model requirements as we should call them, represent a means to complement other types of requirements, textual requirements. When dealing with both, textual and model requirements, means to ensure consistency and completeness are key to the final success of our systems.
- This webinar introduces a new addon for Capella, the so-called RAT for Capella. RAT is part of the Systems Engineering Suite (by The REUSE Company), so the main goals of this tool are:
 - to ensure the correctness of the requirements that are managed within Capella,
 - to help authors follow a pattern to end up with a well-formed requirement,
 - to ensure naming consistency between the model elements as represented in the different Capella diagrams, and the way those elements are named within the textual requirements,
 - to provide a complete round-trip when your requirements are developed in other RMS

Date:

- November the 5th, 2020 at 4:00 CET

Hosted by:



Next webinar



- **Einstieg in die Qualitätsanalyse von Anforderungen**
- Anforderungen von niedrigerer Qualität stellen eine der wichtigsten Ursachen dar, warum technische Projekte scheitern, egal welche Branche man betrachtet. Mehrdeutige und an Klarheit mangelnde Formulierungen, widersprüchliche Anforderungen und Inkohärenz in den verwendeten Messeinheiten sind einige Beispiele von den zahlreichen Fehlern, die in der Ausfertigung von Anforderungen eintreten können.
- Dieses Webinar leitet die Benutzung von Anwendungen und Praktiken ein, die das Feststellen von Fehlerquellen zu Beginn des Projekts erlauben, um die daraus resultierende Fehler in den Anforderungen zu beseitigen und damit die Erfolgswahrscheinlichkeit Ihres Projekts deutlich zu erhöhen. Alle diese grundlegende Aspekte der Qualität von Anforderungen werden von der Standardkonfiguration unserer Werkzeuge *RQA – Quality Studio* und *RAT – Authoring Tools* abgedeckt.

Dates:

- November, 26th 2020 9.00 AM CET



Next webinar

- **Introducción al análisis de calidad de requisitos**
- Los requisitos de baja calidad constituyen una de las principales causas de fracasos en proyectos de ingeniería, indiferentemente del sector de actividad. Las formulaciones ambiguas y a las que les falta claridad, los requisitos contradictorios o la incoherencia de unidades de medida son algunos ejemplos de los numerosos errores que puedan ocurrir durante la documentación de requisitos.
- Este webinar introduce al uso de herramientas y técnicas que permiten detectar desde las primeras fases de proyectos las fuentes de errores, con el fin de eliminar esos errores en los requisitos y aumentar así la probabilidad de éxito de estos mismos. Todos esos aspectos fundamentales de la calidad de requisitos están cubiertos por una instalación por defecto de las herramientas *RQA – Quality Studio* y *RAT – Authoring Tools*.

Dates:

- November, 26th 2020 4.00 PM CET



References (all available on Researchgate)

“EARS (Easy Approach to Requirements Syntax)”

Mavin, A., Wilkinson, P., Harwood, A. and Novak, M., Proceedings of 17th International Requirements Engineering Conference (RE2009), IEEE, 2009

“BIG EARS (The Return of Easy Approach to Requirements Syntax)”

Mavin, A. and Wilkinson, P., Proceedings of 18th International Requirements Engineering Conference (RE2010), IEEE, Sydney, September 2010

“Listen, then use EARS”

Mavin, A., IEEE Software, March/April 2012, pp 33-34, IEEE, 2012

“Listens learned (8 lessons learned applying EARS)”

Mavin, A., Wilkinson, P., Gregory, S. and Uusitalo, E., Proceedings of 23rd International Requirements Engineering Conference (RE2015), IEEE, 2015

“Ten years of EARS”

Mavin, A. and Wilkinson P., IEEE Software, September/October 2019, pp 10-14, IEEE, 2019

Contact

www.alistairmavin.com

mav@alistairmavin.com

https://www.researchgate.net/profile/Alistair_Mavin





THE
REUSE
COMPANY

