PROPRIETARY RIGHTS STATEMENT

THIS DOCUMENT CONTAINS INFORMATION, WHICH IS PROPRIETARY TO THE CRYSTAL CONSORTIUM. NEITHER THIS DOCUMENT NOR THE INFORMATION CONTAINED HEREIN SHALL BE USED, DUPLICATED OR COMMUNICATED BY ANY MEANS TO ANY THIRD PARTY, IN WHOLE OR IN PARTS, EXCEPT WITH THE PRIOR WRITTEN CONSENT OF THE CESAR CONSORTIUM THIS RESTRICTION LEGEND SHALL NOT BE ALTERED OR OBLITERATED ON OR FROM THIS DOCUMENT. THE RESEARCH LEADING TO THESE RESULTS HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S SEVENTH FRAMEWORK PROGRAM (FP7/2007-2013) FOR CRYSTAL – CRITICAL SYSTEM ENGINEERING ACCELERATION JOINT UNDERTAKING UNDER GRANT AGREEMENT N° 332830 AND FROM SPECIFIC NATIONAL PROGRAMS AND / OR FUNDING AUTHORITIES.



CRitical SYSTem Engineering AcceLeration

Requirements Authoring Tool D607.031



DOCUMENT INFORMATION

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	Requirements Authoring Tool
Deliverable No.	D607.031
Dissemination Level	СО
Nature	Ρ
Document Version	V3.00
Date	2013-11-29
Contact	José Fuentes
Organization	REUSE
Phone	+34912172596
E-Mail	jose.fuentes@reusecompany.com



AUTHORS TABLE

Name	Company	E-Mail
José Fuentes	REUSE	jose.fuentes@reusecompany.co m

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected
1.0	30.Oct.2013	Initial version	All
2.0	7.Nov.2013	Internal Review	Minor changes all over the document
			Brief introduction to WP607.
3.0	28.Nov.2013	External Review	Clarifications to the architecture figure. Reference to specific versions of some tools. Clarification on some metrics. Typo in the reference to other deliverable.
			Added more elements to the acronym list



CONTENT

1	INTR	ODUCTION	6
	1.1 F 1.2 F 1.3 S	ROLE OF DELIVERABLE RELATIONSHIP TO OTHER CRYSTAL DOCUMENTS STRUCTURE OF THIS DOCUMENT	6 6 6
2	CUR	RENT TECHNICAL FEATURES	7
	2.1 [2.2 / 2.3 L 2.3.1 2.3.2 2.3.3 2.3.4 2.3.5 2.3.6 2.3.7 2.3.8 2.3.9	DESCRIPTION OF THE TOOL ARCHITECTURE OF THE SUITE IST OF FEATURES RQS connectors RAT modes The CCC approach Correctness Consistency based on measurement units Coupling analysis Writing assistance Additional information Suggestion management: communication with the domain experts	
3	TRAI	NING OFFERED TO PARTNERS	19
4	MAIN	I DEVELOPMENT GOALS DURING THE CRYSTAL PROJECT	
5	TER	IS, ABBREVIATIONS AND DEFINITIONS	
6	REFE	ERENCES	



Content of Figures

Figure 2-1: RQS Architecture	8
Figure 2-2. Selection of the desired mode	9
Figure 2-3: Quality functions	10
Figure 2-4: Correctness	10
Figure 2-5: Correctness details in RATon mode	11
Figure 2-6: Correctness in Full CCC mode	12
Figure 2-7: Consistency of units	13
Figure 2-8: Coupling analysis	14
Figure 2-9: Coupled requirements	15
Figure 2-10: Writing assistance	16
Figure 2-11: Additional attributes	17
Figure 2-12: Syntactic analysis	18

Content of Tables

Table 4-1: Terms, Abbreviations and Definitions	22
---	----



1 Introduction

1.1 Role of deliverable

The goal of this deliverable is to describe the features included in the Requirements Authoring Tool (RAT), brick (B2.30).

This brick is part of WP607. The main goals of this WP are the following:

- CCC approach for quality requirements:
 - Correctness Consistency Completeness
- Requirements reuse
- Integration with IOS
- Support to the development supply chain and shareable content

In order to fulfil with the goals of the deliverable, the main goal of this brick is to provide a kind of wizard (assistant) that could help authors while they're actually writing their requirements.

1.2 Relationship to other CRYSTAL Documents

This deliverable is related to the rest of deliverables and bricks created in WP6.7 as well as the main deliverables from UC2.4 (Electrical Flight Control System - the UC which WP6.7 is based on). The level of relationship is the following:

- D607.011: since this document will describe the further needs and goals, the development and finally the assessment of the brick Requirements Authoring Tool (B2.30)
- D607.021: since the way RAT analyses quality is customized in Requirements Quality Analyzer (RQA), brick B2.29
- D607.041: since part of the analysis performed by relies on ontologies and boilerplates managed in knowledgeMANAGER (kM), brick 2.37
- D204.013: since this deliverable represents the needs of the industrial partners involved in WP6.7 (UC2.4).

1.3 Structure of this document

The structure of the document is the following:

- Chapter 2 Current technical features: first we start with the description of the current state of the brick
- Chapter 3 Training offered to end-users: this chapter lists the training sessions that have been held related to this brick, a link to the training material is also included
- Chapter 4 Main goals for the brick during the CRYSTAL project: finally, this chapter summarizes what seems to be the most important goals for the industrial partners related to the WP



2 Current technical features

This chapter describes the current technical features of the tool Requirements Authoring Tool. As a first iteration for this brick/deliverable, the set of features described hereinafter correspond to the status of the tool as it is today in the commercial version (version 4.1), available at <u>http://www.reusecompany.com</u>.

2.1 Description of the tool

Requirements Authoring Tool (RAT) belongs to Requirements Quality Suite (RQS), a set of tools aimed to customize, manage and improve the quality of a set of requirements (see also deliverables D607.021 and D607.041 for a more detailed description of the other tools included in the suite).

More specifically, the main goals of RAT are the following (see section 2.3 for a more detailed description):

- Typing (either adding or editing) requirements on top of a RMS
- Generating correctness information on the fly (see section 2.3.4.2 for a detailed list of correctness metrics)
- Highlighting the defects (or order relevant information) found during the quality analysis
- Accessing the details of the quality metrics: actual quantitative value, qualitative value, expressions found in the requirement which raised the metric...
- Assistance in writing requirements by following a set of agreed upon boilerplates
- Use of the right vocabulary by showing suggestions coming from domain ontologies
- Consistency information based on measurement units
- Similar requirements based on their semantic graphs
- Suggestion management: that allows to send suggestions to the "owners" of the ontology about new concepts or even new boilerplates

2.2 Architecture of the suite

This picture represents the architecture of the whole Requirements Quality Suite. The rest of the section describes all the boxes in the architecture and how and why RAT is connected to the other tools.

This picture shows the dependencies among different components as blue arrows. Those components may or may not be installed in the same physical node (a Windows based computer), but all of them must be connected to the same LAN.





Figure 2-1: RQS Architecture

The components of this architecture are the following:

- RQA Server Requirements Quality Analyzer Server: in charge of the main configuration of the whole suite. Database connection, licensing and low-level database management...
- RQA Client Requirements Quality Analyzer Client: provides the quality reporting but, what is more important for RAT, it also provides the quality configuration used by RAT
- RAT Requirements Authoring Tool: this module allows quality analysis on the fly. More info in other sections of this document
- kM knowledgeMANAGER: this tool is needed to customize the ontology and boilerplates needed during the requirements authoring
- SKR System Knowledge Repository: this is a relational database where we can find two clearly different parts:
 - SKB System Knowledge Base: represents the main ontology behind all the quality analysis as well as all the information needed to perform a Natural Language Process to generate a semantic graph out of a textual requirement
 - Assets: represents the formal representation (mainly as a semantic graph) generated out of every textual requirement once the requirement has been created with RAT

2.3 List of features

2.3.1 RQS connectors

RQS is currently connected to some of the most wide used requirements management tools in the market:

- RQA: connected to IBM Rational DOORS (versions 8.x and 9.x), Dassault Systèmes Reqtify (version 2.13), Visure Requirements (version 4.x) and MS Excel (versions 2003, 2007 and 2010)
- RAT: connected as a plugin to IBM Rational DOORS (versions 8.x and 9.x) and MS Excel (versions 2003, 2007 and 2010)

For all of the aforementioned connectors, the corresponding APIs were used, i.e. no interoperable connector has been created yet.

Version	Nature	Date	Page
V03.00	Р	2013-11-29	8 of 23



2.3.2 RAT modes

RAT screen can be configured in three different modes, each of them with different amount of information in the screen. Those modes are the following:

- RATon:
 - Focussed on correctness
 - \circ Simplified layout: less amount of information in the screen
- Correctness RAT:
 - o Includes all RATon features
 - o Add assistance in requirements writing based on boilerplates
- CCC RAT:
 - Includes all Correctness RAT features
 - o Full correctness information
 - o Inconsistent units
 - Couples/overlapped requirements
 - o Extended list of attributes of the selected requirement
 - o Grammatical information of the requirement

The changes of mode are performed by using the *Mode* menu. In order to show/hide the main menu, the *<Alt>* key must be pressed.

M RAT - Requirements	Authorir	ng Tool
Suggestions Quality	Mode	
Requirement By The REUSE Compar	R C C	ATon RAT CC RAT
Select your boilerplat	e to help	you writing your requirement:
< Select a pattern group >		

Figure 2-2. Selection of the desired mode

2.3.3 The CCC approach

RQS is based, as CESAR was, in the CCC approach: Consistency, Completeness and Correctness.

While Correctness is measured individually, for every single requirement, one by one, Consistency and Completeness are analysed for the whole project or requirements module.

The most developed "C" so far is the Correctness one, while the main goal for the CRYSTAL project is to enhance the Consistency and Completeness analysis of the suite.

2.3.4 Correctness

2.3.4.1 Introduction to the correctness metric

In this approach, RAT is able to analyse the quality of a new (or edited) requirement on the fly, in order to provide enough information to enhance its quality even before the requirement is stored back into the RMS.

Version	Nature	Date	Page
V03.00	Р	2013-11-29	9 of 23



The correctness metric is computed as the sum of more than 30 different indicators (the number and weight of those indicators shall be configured in RQA). For every indicator, RAT generates its quantitative value (a positive number). An example of such kind of indicators could be the text length, readability... Finally, every indicator is transformed into a qualitative value by the associated quality function.



Figure 2-3: Quality functions

During the correctness checking process, every metric rated as medium or low quality will generate a hint in RAT. This hint leads the requirement author in the best way to get rid of the problem and enhance the quality of the requirement.

All the quality hints are shown in the right hand side of the RAT form, inside a coloured frame where:

- Green: represents a high quality requirement, but it's important to remark that still some minor problems can be reported by the tool
- Yellow: some problems have been detected
- Red: major problems have been detected

SAT - Requirements Authoring Tool	
Original requirement: ID: 6 - the system shall be able to fly high	Quality Assessment Summary
the system shall be able to fly high	Metric Value At least one domain term must be used 0 At least one domain verb must be used 0 The number of verbs not recognized by t 1 The number of concepts not recognized 1
	Save and close Cancel

Figure 2-4: Correctness

2.3.4.2 Quality metrics for correctness

RQA and RAT include more than 30 different metrics that allow checking correctness for individual requirements. Some of these metrics are the following:

- Size: expressed in paragraphs, chars, nouns or verbs. Long requirements will be difficult to understand
- Readability: number of letters between punctuation marks and some other formulas than indicate whether the requirement will be easy to read. Ease to read requirements generates less problems all over the project
- Conditional sentences vs. imperative sentences: avoid would and use Shall, should and will in the right way
- Active vs. passive voice: avoid using passive voice to increase the readability of the requirement

Version	Nature	Date	Page
V03.00	Р	2013-11-29	10 of 23



- Optional sentences: maybe... Optional requirements must be stated by an attribute, never in the body of the requirement
- Ambiguous sentences: fast, user-friendly... What do the analyst, the coder and the customer understand by the same ambiguous sentence
- Subjective sentences: in my opinion, I think that... Don't show your ideas, but what the system should do
- Implicit sentences: it must be provided by them... Too many pronouns make your requirements difficult to understand
- Abuse of connectors: and, or. Many times connectors reveal different needs enclosed within the same requirement, losing the atomic characteristic
- False friends: customized according to "mother language" of your project
- Negations: no, never... Two or more negations in the same sentence make it difficult to understand
- Speculative sentences: usually, almost always... Make the requirement imprecise
- Design terms: loop, hash... Remember, avoid How, concentrate on What
- Flow terms: while, if, else... Remember avoid How, concentrate on What
- Number of domain nouns and verbs: domain terms and verbs should be involved into the requirement specification, nevertheless, too many different terms in the same requirement many times mean multiple needs
- Acronyms: avoid those that don't belong to the domain representation
- Hierarchical levels: don't complicate your specification with too many indentation levels
- Volatility: if a requirement suffers many changes, you must be very careful with it
- Number of dependences: the same if your requirement is the source of too many dependences

2.3.4.3 Correctness details

In *RATon mode*, correctness details are highlighted in red when the metric is clicked in the right hand side of the screen:

KAT - Requirements Authoring Tool		3
Quality Mode		
Original requirement: ID: 233 -	Quality Assessment Summary	
the system shall be able to fly to the moon with a rocket moved with solid	Metric Value	
	The number of verbs not recognized by t 1	
	The number of concepts not recognized 3	
	Save and close Cancel	

Figure 2-5: Correctness details in RATon mode

When in *Full CCC mode*, the highlighted terms are shown in the *Textual assessment* tab; while the *Quality metrics assessment* provides much more information about every individual metric.



Requirements Authoring Tool By The REUSE Company Select your boilerplate to help you writing your requirement: 01 - System Functionality	elect a pat	tern >			
elect your boilerplate to help you writing your requirement: 01 - System Functionality	elect a pat	tern >			•
Boilerplate example: < none > Driginal requirement: ID: 233 - the system shall be able to fly to the moon with a rocket moved w	ect a pat	tern >			•
Boilerplate example: < none >)riginal requirement: ID: 233 - the system shall be able to fly to the moon with a rocket moved w					
Original requirement: ID: 233 - the system shall be able to fly to the moon with a rocket moved w					
the system shall be able to fly to the moon with a rocket moved w				Quality Assessment Summary	
	ith solid			Metric Unclassified verbs Unclassified concepts Boilerplates matching	Value
urrent boilerplate elements: NOUN (0) MODAL VERB «MODAL COMPULSORY» (1) VER NOUN (0) MODAL VERB «MODAL COMPULSORY» (1) Be (B (2) NC 2) Able (DUN (4) 3) To (4) 1	VERB (5) NOUN (7)		
flatching boilerplates Syntax Similar requirements Inconsister	nt measur	ement units	Quality metrics assessment	Textual assessment Additional a	attributes
		Quality	Recommendation		Affects overall
Metric	Value	C			quality
Metric Out-links	Value) High	Out-links must be used accor	ding to the guidelines and polici	quality

Figure 2-6: Correctness in Full CCC mode

2.3.5 Consistency based on measurement units

RAT can easily find two or more requirements, in the same project, using inconsistent measurement units. Examples of this lack of consistency could be two requirements where one of them is using *yards* in order to represent the precision an altimeter must take the measurement; and the other requirement is using *meters* to represent the minimum distance a target must be in order to be represented in the screen.

Then this kind of inconsistency is generated because of the requirements that RAT is currently editing, all the requirements that may lead to the lack of consistency are shown in the *Inconsistent Measurement Unit* tab, highlighting in red the occurrences of the inconsistent units.



equirements Aut	horing Tool				V
lect your boilerplate to help y	you writing your requirement:				
01 - System Functionality	•	< Select a pattern >			•
Boilerplate example: < none	>				
riginal requirement: ID: 199 -				Quality Assessment Summary	
The radar shall provide a spec	cial warning about objects at l	ess than 100 meters		Metric Unclassified concepts Ambiguous sentences Domain concepts Domain verbs	Value 4 1 0 2
urrent boilerplate elements:					
urrent boilerplate elements:	Similar requirements Inco	nsistent measurement units	Quality metrics assessment	Textual assessment ¹ Additional attri	ibutes
urrent boilerplate elements: atching boilerplates Syntax Code Name	Similar requirements Inco Description	nsistent measurement units	Quality metrics assessment The radar shall detect object	Textual assessment Additional attra ts bellow 10000 miles high	ibutes (
urrent boilerplate elements: atching boilerplates Syntax Code Name 138 152 198	Similar requirements Inco Description The distance Requiremen The radar sh	nsistent measurement units e to the authors must Is must be written by	Quality metrics assessment The radar shall detect object	Textual assessment Additional attri ts bellow 10000 miles high	ibutes

Figure 2-7: Consistency of units

2.3.6 Coupling analysis

Redundancy and inconsistency are two big issues in the requirements engineering process. Thanks to RAT, different requirements, even with very different wording, could be matched in case they could share a suspicious similar meaning.

To do that, every requirement is transformed into its semantic representation. Such a representation is a semantic graph.

Thus, two apparently different requirements could be easily identified, allowing the author to take the proper steps with both requirements. An example of such a semantic analysis is depicted in the following picture, where two requirements are eventually represented with the same semantic graph.



Requirements Authoring Tool





Figure 2-8: Coupling analysis

In order to get such a result, the ontology (see knowledgeMANAGER deliverables, D607.041) must be populated with enough information to allow the tool to identify the knowledge behind both requirements as similar. Examples of this kind of knowledge in the ontology could be the following:

- Radar *is_a_kind_of* Electromagnetic sensor
- To detect and To identify both have the same semantics (meaning)
- Two different boilerplates have been represented both with a different grammar (structure) but both with a similar formalization to represent the rate > 10 units per second

In the Full CCC mode, RAT shows the overlapped (coupled) requirements in the Similar requirements tab:



ggestions Q	Quality <u>M</u> ode							
Requiren By The REUSE	ments Auth	oring Tool	ment:					V
01 - System I	Functionality	sa winang your requirer	▼ < Select a p	pattern >				
Boilerplate ex	example: < none >							
)riginal requir	rement: ID: 233 -					Quality Assessmen	nt Summary	
The software	e shall be able to is	sue metric information	to the authors			Metric		Value
urrant hailar	misto elemente							
Current boilers Natching boile	rplate elements: Ierplates Syntax	Similar requirements	Inconsistent mea	isurement units	Quality metrics assessment	Textual assessment	Additional attributes	
Current boiler Matching boile Code N	rplate elements: lerplates Syntax Module	Similar requirements Author 1	Inconsistent meas	isurement units Description	Quality metrics assessment	Textual assessment	Additional attributes	Similarity
Current boiler; Matching boile Code N 43	rplate elements: Ierplates Syntax Module	Similar requirements Author N	Inconsistent meas	surement units Description The system sha	Quality metrics assessment Il mail metric information to a	Textual assessment uthors	Additional attributes	Similarity 63
Autrent boilerp Vatching boile Code N 43 147 150	rplate elements: lerplates Syntax Module	Similar requirements Author 1	Inconsistent meas	isurement units Description The system sha The system sha The system sha	Quality metrics assessment Il mail metric information to a Il send metric information to i Il send metric information to a	Textual assessment authors the authors authors	Additional attributes	Similarity 63 63 63

Figure 2-9: Coupled requirements

As shown in this screenshot, those requirements semantically similar to the current requirement are listed here.

2.3.7 Writing assistance

Based on the proper definition of a taxonomy of requirements types, the user is able to create a set of patterns or boilerplates representing the structure of every type of requirement.

For example, the following requirement:

While in landing mode, in case the button x is pressed, the emergency engine must start.

Will match in a boilerplate with the following structure:

While + in + <**mode>** + , + in case + the + <**component>** + to_be + <**trigger>** + the + <**component>** + must + <**action>**

By customizing in the ontology the full set of boilerplates (see knowledgeMANAGER, brick number B2.37) the RAT users will be able to select one of the requirements types and therefore, the corresponding associated set of patterns. Once a pattern is selected:

- 1. RAT shows the grammar or items involved in the selected pattern (see example above in this section)
- 2. RAT suggests the correct working from the controlled vocabulary. This process is based on the proper tag and proper semantic selected for any of the items (slots) of the pattern



RAT - Requirements Authoring Tool gestions Quality Mode	ng assistat	are .		
Requirements Authoring Tool By The REUSE Company				V
01 - System Functionality	Stakeholder Functional Re	equirement		•
Boilerplate example: [the manager of the airplane	e shall be able to accelerate the accel	lerometer]		
Driginal requirement: ID: 198 -			Quality Assessmen	t Summary
The manager shall be able to abnonnally ad Acce Acce Acce Acce Acce Accu Accu Accu	de lerate pt ss mulate irre arke alise alize		Metric Unclassified c Domain conce Domain verbs	vncepts 1 pts 0 0
NOUN «STAKEHOLDER» (0) MODAL VERB «M NOUN «STAKEHOLDER» (0) MODAL VERB «M NOUN «STAKEHOLDER» (0) MODAL VERB «M (ODAL COMPULSORY» (1) VERB IODAL COMPULSORY» (1) VERB IODAL COMPULSORY» (1) VERB IODAL COMPULSORY» (1) VERB IODAL COMPULSORY (1) VERB	TO BE (0) Able (1) To (TO BE (0) Able (1) To (s Quality metrics assessme	2) ADVERB «LY Adeverbs» (4) VERB T(2) ADVERB «LY Adeverbs» (4) VERB «L 2) ADVERB «LY Adeverbs» (4) VERB «L 3) ADVERB «LY Adeverbs» (4) VERB «L 3) ADVERB «LY Adeverbs» (4) VERB «L 3) ADVERB «LY Adeverbs» (4) VERB «L 4) ADVERB «LY Adeverbs» (4) ADVERB «L 4) ADVERB «LY Adeverbs» (4) VERB «L 4) ADVERB «LY Adeverbs» (4) ADVERB «L 4) ADVERB «L ADVERB «L 4) ADVERB «L ADVERB	D BE (0) «F-DUTY ACTION» (1) -MODES AND STATES : TRANS -CAPABILITY» (0) NOUN (0) INTERFACE (0) NOUN (0) THE STATES : S
Metric	Value	Quality Recommendat	ion	Affects overall
Implicit sentences Connectors Rationale sentences In-links	0 0 0 0	High High High High In-links must b	e used according to the guidelines and polici	es of your

Figure 2-10: Writing assistance

In this screenshot we can see how the controlled vocabulary is always shown as a dropdown list representing the suitable concepts from the ontology matching with the current item in the requirement (i.e. whether or not it's a noun or a verb, whether or not a fixed semantic is expected...).

By using the tool, the requirements written by RAT will always follow the agreed upon structure/grammar and will always use the right terminology from a controlled vocabulary.

2.3.8 Additional information

The *Full CCC mode* shows two more tabs with additional information.

2.3.8.1 Additional attributes

In this tab, RAT shows all the attributes defined in the RMS for the current requirement. This tab is *read only*, that means that the values of the attributes cannot be changed into RAT.



RAT - Requirements Authoring Tool		
Requirements Authoring Tool By The REUSE Company		
Select your boilerplate to help you writing your requirement:		
01 - System Functionality	< Select a pattern >	•
Boilerplate example: < none >		
Original requirement: ID: 199 -		Quality Assessment Summary
The radar shall provide a special warning about objects at le	ess than 100 meters	Metric Value Unclassified concepts 4 Ambiguous sentences 1 Domain concepts 0 Domain verbs 2
Current boilerplate elements: Matchino boilerolates Syntax Similar requirements Incor	nsistent measurement units Ouality metrics assessmen	nt Textual assessment [Additional attributes]
Attribute	Value	
Attribute Absolute Number Created By Created On Created Thru	Value 199 Administrator 29 August 2013 Manual Input	
		Save and close Cancel

Figure 2-11: Additional attributes

2.3.8.2 Syntax information

The requirement that the user is typing in RAT includes some concepts from the controlled vocabulary or out of the controlled vocabulary. The syntactic analysis represents how the system understands the textual content of the requirement including several pipelined stages:

- Tokenization: initially, every word is separated with spaces or punctuation marks. Nevertheless, sometimes the meaningful concept is represented by more than one word (compound concepts such as *breaking system*); and some other times, a "unit" of information (some characters enclosed into white spaces) shall be split into two or more tokens of information (e.g. 20Km/h is split into 20 as a number, and Km/h as a unit)
- Normalization: this stage deals with the morphology of the words, dealing with (mostly) the ending of the words in examples such us: singular/plural, verb conjugation, or even conjugation of irregular verbs
- Tagging: performed in those cases where the same work could have two different meaning, or could play a different role in the sentence. The word *plane* could be a noun, but also a verb and even an adjective. According to the structure of the sentence, the system must chose the most suitable tag
- Semantic grouping: finally, some terms may have a particular meaning by belonging to one or more semantics. This information is very valuable for further retrieval (coupling) processes
- Boilerplate matching: where different boilerplates are evaluated to find the most suitable one matching with the input requirement



The result of the syntactic analysis is listed in the Syntax tab:

and Acquirements Authoning	Tool		-	-		- Ο -Σ
ggestions Quality Mode						
Requirements Auth By The REUSE Company	oring Tool					
Select your boilerplate to help yo	ou writing your requirement:					
01 - System Functionality	•	< Select a pattern >				•
Boilerplate example: < none >						
Driginal requirement: ID: 233 -				Quality Assessmen	it Summary	
the system shall be able to fly to the moon with a rocket moved with solid				Metric Munclassified v Munclassified c	erbs oncepts	Value 1
urrent boilerplate elements:	·					
latching boilerplates Syntax	Similar requirements Inco	onsistent measurement units	Quality metrics assessment	Textual assessment	Additional attributes	
ndexing results:						
the system	4150					•
Identificador 1/						
be able Término: systen Etiqueta sintáct Semántica: SYST	n nica: NOUN TEM					*

Figure 2-12: Syntactic analysis

2.3.9 Suggestion management: communication with the domain experts

The dynamic nature of the projects will make, for sure, that the ontology should be evolved while requirements are gathered and authored into the RMS. However, uncontrolled changes to the ontology must be avoided, so the ontology management process must be well-controlled.

In order to provide a controlled process, only *domain experts* will have rights to edit ontologies by using knowledgeMANAGER. Nevertheless, the requirements authors have the chance to communicate with domain experts by exchanging suggestions. RAT supports two kinds of suggestions:

- Suggestions of new terms: every term deemed as unclassified (out of the scope of the domain) can be suggested to be included into the domain ontology
- Suggestions of new boilerplates: whenever RAT marks a specific requirement as not fulfilling a valid boilerplate, if the author disagrees with this decision, the current requirement could be sent as a suggestion so that the domain experts team could –if the suggestion is accepted- create a new boilerplate and communicate with the author of the requirement

The suggestions management systems allow the proper communication between authors and domain experts, so that authors could be informed about the current status of their suggestions.



3 Training offered to partners

During the first few months of the project, several training sessions have been scheduled to:

- Train industrial partners on how to use RQA
- Train other technical partners involved in WP6.7 on the fundamentals and details about the semantic approach followed by RQS

The training material used for both sessions is available in the CRYSTAL repository:

- Training for end users (14 October 2013): https://projects.avl.com/11/0154/Data%20Exchange/Forms/AllItems.aspx?RootFolder=%2f11%2f01 <u>54%2fData%20Exchange%2f001_MEETINGS%2f011_SP6_Meetings%2fWP6_7%2fMeetings%2f2</u> <u>013-10-14%20RQS%20Training&FolderCTID=&View=%7bA036B3F1-CA9C-4631-A46F-C55BDA6D5C01%7d</u>
- Training for technical partners (16 September 2013): <u>https://projects.avl.com/11/0154/Data%20Exchange/Forms/AllItems.aspx?RootFolder=%2f11%2f01</u> <u>54%2fData%20Exchange%2f001_MEETINGS%2f011_SP6_Meetings%2fWP6_7%2fMeetings%2f2</u> <u>013-09-</u> <u>16_RBE%20Training%20about%20the%20tool%20bricks%20%28Madrid%29%2fkM%20Document</u> ation&FolderCTID=&View=%7bA036B3F1-CA9C-4631-A46F-C55BDA6D5C01%7d



4 Main development goals during the CRYSTAL project

The set of features described so far have been described as very valuable for the industrial partners related to WP6.7; nevertheless, all the partners involved in that workpackage are working on envisaging a set of new features to improve the tool. Those improvements are pretty much focussed on Consistency and Completeness.

The main improvements to RQS in general and RAT in particular are the following:

- Having in mind that the main goal of this brick, in terms of integration with RM tools, is to integrate RQS with IBM DOORS; an integration in the overall tool chain through the IOS of the CRYSTAL RTP is conceived
- Integration with modelling tools through IOS to check correctness, completeness and consistency
- Support to a set of new techniques for consistency checking, e.g.:
 - Consistency with respect to the system structure. By exploiting the knowledge of the system structure, the referenced concepts of the requirement can not only be identified but checked for their correct usage:
 - E.g. the possible transitions in a state machine described in a specification can be checked about those transitions shown in the corresponding SysML state machine
- Support to a set of new techniques for completeness checking:
 - Structural completeness questions addressable by exploiting knowledge of the system structure; e.g:
 - Each interface is addressed in a requirement
 - All instances of environmental conditions are addressed by the requirements
 - Hazards have been addressed in requirements
 - Range Completeness for interface variables, e.g.:
 - Check for certain variables (e.g. those that are used for conditional statements) whether the whole range of the domain is covered by the requirements. This metric will be based on the value of certain attributes from a PBS (*max_value* and *min_value* for a specific attribute of a component)
- Support to new correctness metrics:
 - Deprecated concepts
 - Use of not preferred concepts (synonyms)
 - Use of concepts identifies as ambiguous because of their list of more specific concept in the ontology
- Support for formal requirements:
 - Boilerplates with formal semantics to enable formal CCC analysis techniques
 - Metrics to support formalization process
- Contracts: Description of guaranteed properties with explicitly stated assumptions on the context in which a component is embedded, e.g.;
 - The guarantee of a component can only restrict the outputs of the component, not the inputs.
 - Virtual integration analysis
- Customized metrics: allowing the end-user to write the code for their own metrics
- Enhance collaborative work with RAT by better clarifying different roles among all the RQS tools, and enhancing the work with RQS along the supply chain
- Enhance the current in and out-links metric with nominal links where the user could identify the name and direction of the link to quantify
- Pre and Post-analysis code: this represents a way for the end-users to write customized code to be executed at different particular moments while the quality analysis:

Version	Nature	Date	Page
V03.00	Р	2013-11-29	20 of 23



- Before the analysis: the code will be able to change any of the attributes of the requirement
- After the analysis: RQA will provide information related to the result of the analysis so that the proper actions could be taken
- Identification of similar requirements
 - o using their semantic graphs in previous projects
 - Syntactic similarity of requirements (e.g. requirements are addressing the same concepts of the system)



5 Terms, Abbreviations and Definitions

CCC	Correctness, Completeness and Consistency
CESAR	Cost-efficient methods and processes for safety relevant embedded systems
СО	Confidential, only for members of the consortium (including the JU).
CRYSTAL	CRitical SYSTem Engineering AcceLeration
D	Demonstrator
IOS	Interoperability Specification
kМ	knowledgeMANAGER
LAN	Local Area Network
Layout	The arrangement of visual elements in the different screens of the tool
NLP	Natural Language Processing
0	Other
Р	Prototype
PBS	Product Breakdown Structure
PP	Restricted to other program participants (including the JU).
PU	Public
R	Report
RAT	Requirements Authoring Tool
RE	Restricted to a group specified by the consortium (including the JU).
RMS	Requirements Management System
RQA	Requirements Quality Analyzer
RQS	Requirements Quality Suite
RTP	Reference Technology Platform
SKB	System Knowledge Base
SKR	System Knowledge Repository
SP	Subproject
UC	Use case
WP	Work Package

Table 5-1: Terms, Abbreviations and Definitions



6 References

[Allain, 2010]	G. Allain et al; <i>Completeness/Consistency/Correctness</i> ; CESAR project deliverable, October 2010 (D_SP2_R3.2_M2 Vol4)
[Hull et al, 2009]	Requirements Engineering
[INCOSE, 2012]	Guide for Writing Requirements