



CHARACTERISTIC	DESCRIPTION
<b>Analyze the quality of requirements (RQA)</b>	The tool RQA can create a connector to this source of requirements. This enables the core set of features of RQA
<b>Filter by views</b>	RQA can filter objects/items in the main view only to consider as requirements those meeting the filter (those contained in the view) and skipping the analysis of the rest of objects (headings, comments...)
<b>Filter by artifact type</b>	RQA can filter objects/items based on a logical expression to consider as requirements only those meeting the filter (having a value of True in the logical expression) and skipping the analysis of the rest of objects (headings, comments...)
<b>Assessment by baseline</b>	RQA allows selecting a baseline (other than the current baseline) for the quality assessment
<b>Store quality results back in the tool</b>	RQA can be configured to store back in the Requirements Management System the result of the quality assessment
<b>CCC with RQA</b>	RQA can perform not only a correctness analysis, but also consistency and completeness
<b>RAT.exe. Correctness</b>	The RAT tool can connect to this requirements repository to add or edit new requirements using the quality rules defined in RQA for this document
<b>RAT.exe. CCC</b>	RAT can perform not only a correctness analysis, but also consistency and completeness
<b>RAT.exe. Pattern-based authoring</b>	RAT enables a writing assistant to help authors write following a given set of patterns and following the content of the dictionaries or glossaries
<b>RAT Plug-in. Correctness</b>	RAT acts as a plugin inside the Requirements Management System
<b>RAT Plug-in. CCC</b>	RAT plug-in provides not only a correctness analysis, but also consistency and completeness
<b>RAT Plug-in. Pattern-based authoring</b>	RAT plug-in enables a writing assistant to help authors write following a given set of patterns and following the content of the dictionaries or glossaries
<b>Capture vocabulary</b>	Concepts defined in this tool can be used for requirements writing and/or for the execution of metrics
<b>Extract info from class/block diagrams</b>	The names of the classes/blocks can be used for authoring purposes and/or to create metrics based on this information
<b>Classes and Interfaces</b>	The names of the classes, interfaces, methods and attributes can be used for authoring purposes and/or to create metrics based on this information
<b>Extract properties</b>	The names of the properties can be used for authoring purposes and/or to create metrics based on this information
<b>Extract relationships: hierarchical, aggregation...</b>	The relationships between elements can be used for consistency or completeness purposes
<b>Extract info from state machines</b>	The information from a state chart can be used either for requirements authoring and/or for the elaboration of quality metric
<b>Extract states</b>	The names of the states can be used for authoring purposes and/or to create metrics based on this information
<b>Extract transitions</b>	The transitions among states can be used for consistency or completeness purposes
<b>Capella Operational Architecture</b>	The names of the operational activities in an operational entity can be used for authoring purposes and/or to create metrics based on this information
<b>Capella Capabilities</b>	The names of the capabilities can be used for authoring purposes and/or to create metrics based on this information
<b>Capella Dataflows</b>	The names of the components can be used for authoring purposes and/or to create metrics based on this information
<b>Capella Architecture</b>	The names of the logical, behavioral and implementation components can be used for authoring purposes and/or to create metrics based on this information
<b>Capella Trees</b>	The names of the tree components can be used for authoring purposes and/or to create metrics based on this information
<b>Extract info from sequence diagrams</b>	The names of the actors and lifelines can be used for authoring purposes and/or to create metrics based on this information
<b>Extract info from packages elements</b>	Packages names can be used for authoring purposes and/or to create metrics based on this information
<b>Extract info from use cases and actors</b>	Use case and actor names can be used for authoring purposes and/or to create metrics based on this information
<b>Extract info from activities</b>	Activities names can be used for authoring purposes and/or to create metrics based on this information
<b>Extract signals</b>	The names of the signals can be used for authoring purposes and/or to create metrics based on this information
<b>Signal values</b>	The actual values assigned or checked to a signal in a requirement can be compared (consistency checking) to a range of possible limits defined in these models
<b>Simulink Blocks</b>	The names of the Simulink blocks can be used for authoring purposes and/or to create metrics based on this information
<b>Simulink Common Block Properties</b>	The names of the properties can be used to create metrics based on this information
<b>Simulink Block-Specific Parameters</b>	The names of the parameters can be used to create metrics based on this information
<b>Safety case diagrams</b>	The names of the claims, arguments and evidence elements can be used for authoring purposes and/or to create metrics based on this information
<b>Feature models vocabulary</b>	The names of the feature elements can be used for authoring purposes and/or to create metrics based on this information
<b>Export feature models</b>	Feature models managed within KNOWLEDGE Manager (KM), can be exported to XML files with VEL (Variability Exchange Language) format.
<b>Ontological information</b>	Classes, entities, relationships and properties can be used for authoring purposes and/or to create metrics based on this information