

ReqCyle overview

OpenETCS – WP7 December 2013

Raphaël FAUDOU raphael.faudou@samares-engineering.com

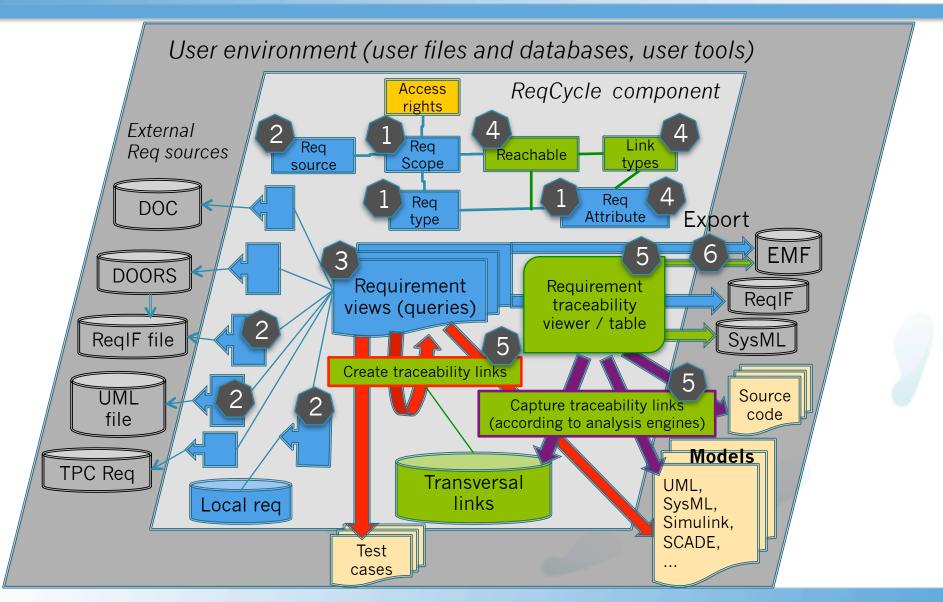
Agenda



- ReqCycle : Why ?
- Big picture
- Steps
 - 1. Data model scopes and types
 - 2. Requirement creation & import
 - 3. Visualization of requirements
 - 4. Definition of traceability links
 - 5. Creation and capture of traceability links
- TeamWork and configuration mgt
- Work to do
- Discussion/questions

ReqCycle big picture and steps

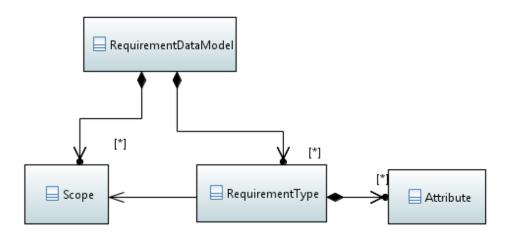




1. Requirement data model



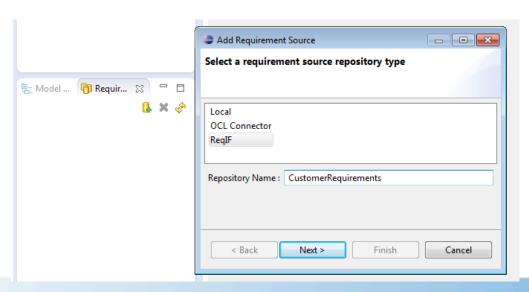
- One ReqCycle data model is defined with scopes (organization) and requirement types (structure of requirements)
- Scopes allow creating classification and allocation of teams
 - Ex: SSRS, VV-SSRS, SSHA, SubS1, SubS2, SW-A, SW-B, SW-C...
- Types allow managing different sources of requirements
 - Ex: Functional Req, Safety Req...



2. Import or reference requirements



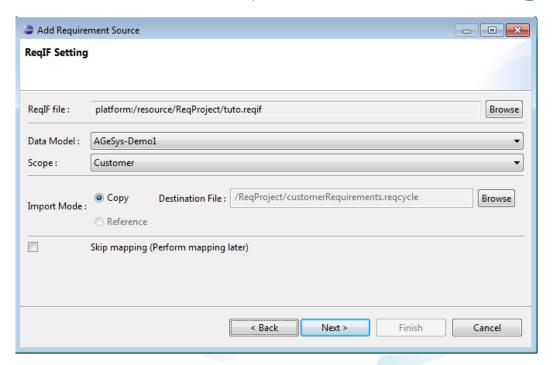
- Can import external requirements (external source)
 - Note: can reference them in future release
- Go to requirement sources view
 - Create requirement source
 - Choose « ReqIF »
 - Note: « Local » will be used to create own requirements



2. ReqIF import - parameters



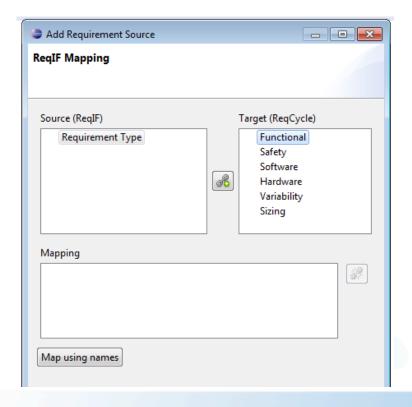
- Select ReqIF file (without spaces in file path)
- Select one data model and one scope
- Define destination file (for teamwork usage see later)

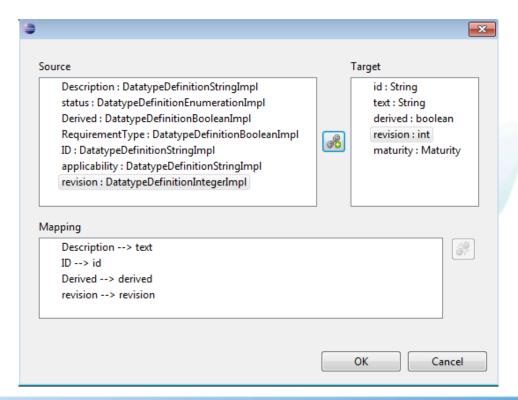


2. ReqIF import - mapping



- Map requirement types
- Map attributes

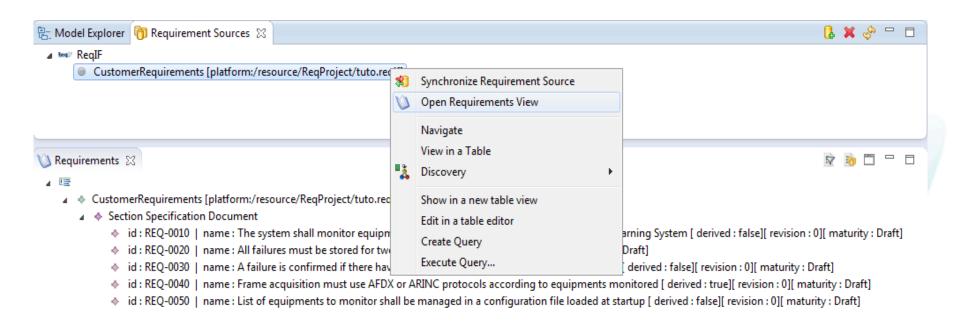




2. See import result



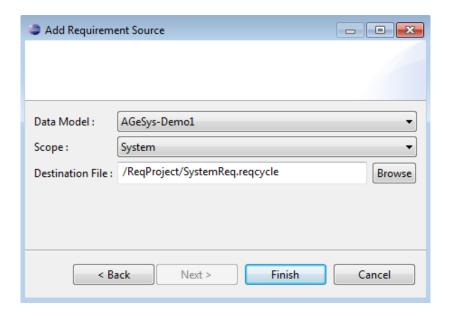
- Requirement source is created
- Right click and « open in Requirement view »



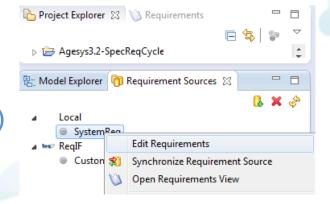
2. Create own requirements



- Can use "Local" requirement source connector
 - Fewer input parameters



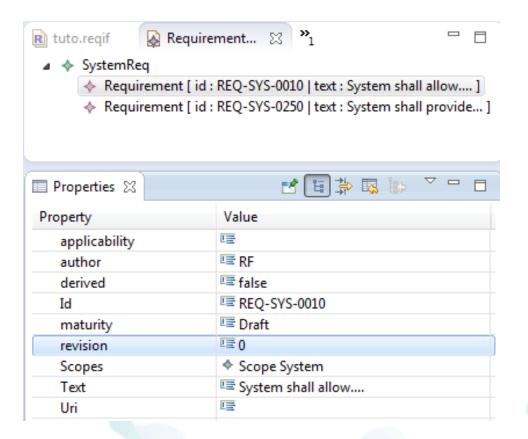
Can edit requirements (see next)



2. Create own requirements



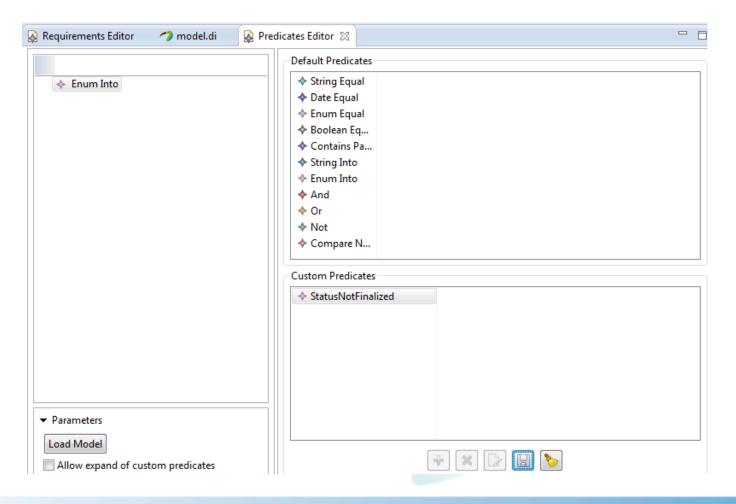
Requirement editor (minimalist)



3. Define requirement queries – future



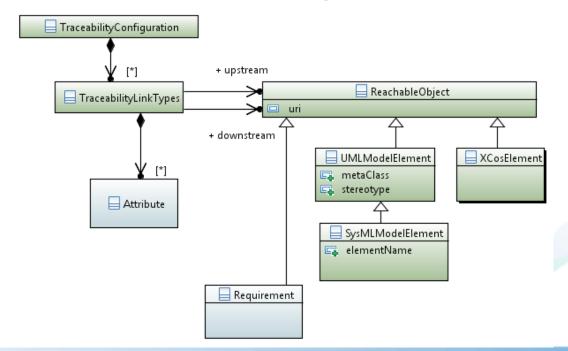
Ability to filter requirements on predicates



4. Traceability configuration overview



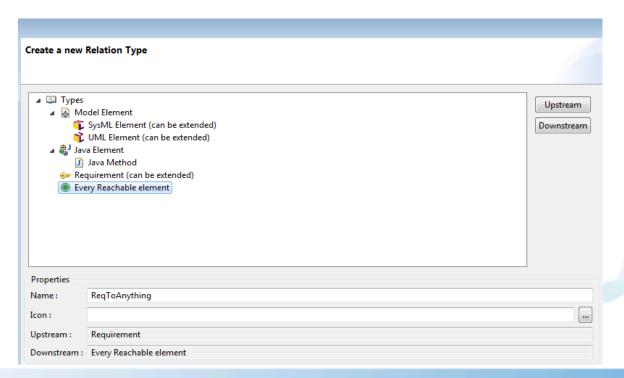
- By default, reachable objects known by ReqCycle are
 - UML and SysML model elements
- Can capture other reachable objects with new plugins
 - Just have to implement a few given extensions



4. Create traceability link type



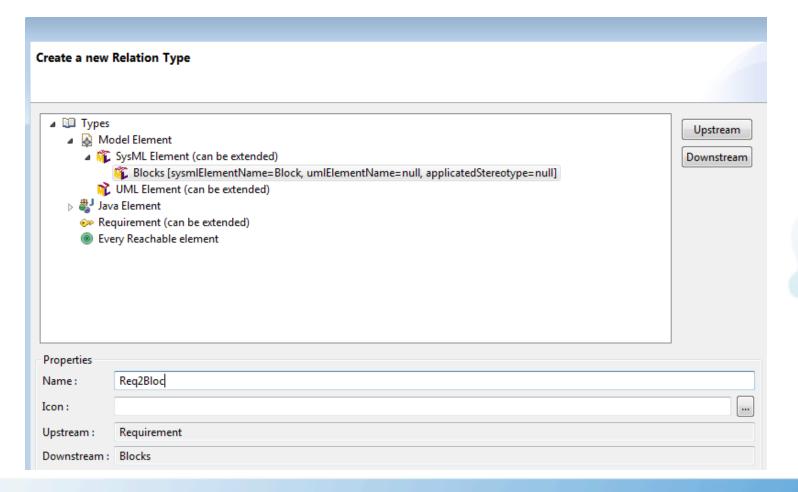
- Choose upstream data type
 - Requirement
- Choose downstream data type
 - Can be any reachable data



4. Restrict link target



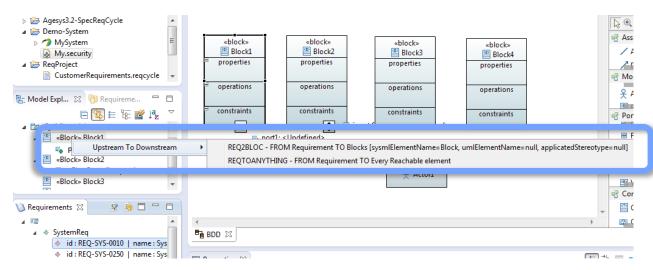
Ex: create link type from requirement to SysML block



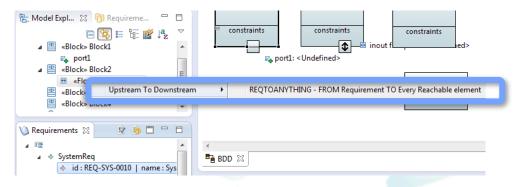
4. Restrict links to blocks - usage



Drag req on a block: two link types available



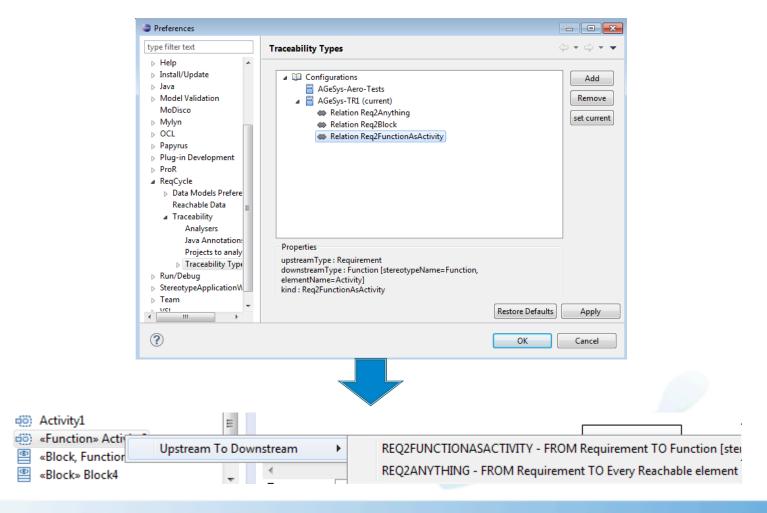
Drag req on else: Req2Blocs is not available



4. Restrict links to UML stereotype



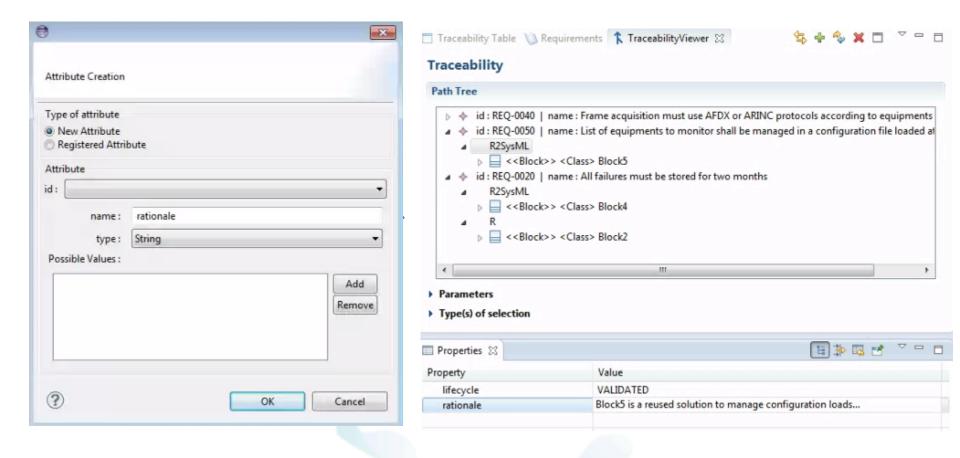
Example: Function as UML Activity specialization



4. Create links with attributes - future



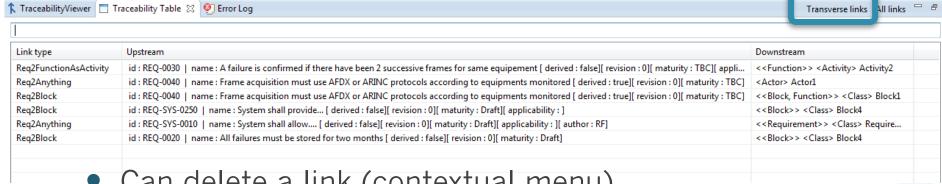
Possibility to manage traceability link attributes



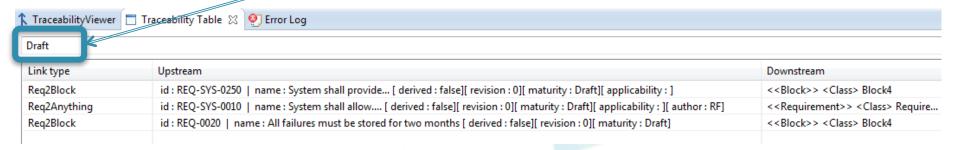
5. Traceability table



- Apply "Transverse link" button
 - Requires selecting an Eclipse project



- Can delete a link (contextual menu)
- Can search on any part of text

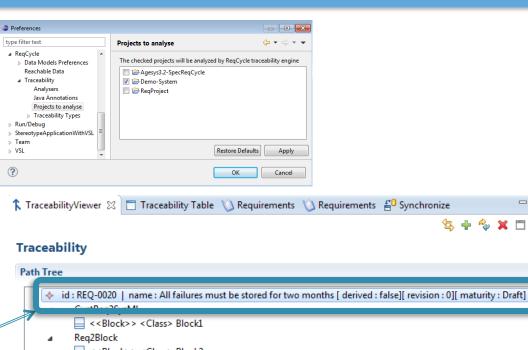


18

5. Traceability viewer



- In ReqCycle preferences, select projects to analyse
- Then open traceability Viewer view
- DnD requirements in Path Tree area
 - Traceability is captured in real time and displayed



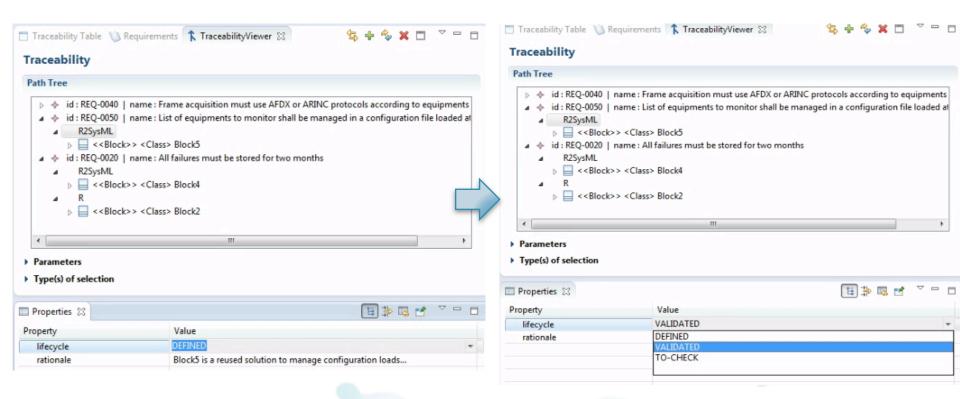
© 2013 SAMARES ENGINEERING – All rights reserved

Type(s) of selection

5. Manage link attributes – future



- Ability to define and edit link attributes
 - A first step toward traceability validation...

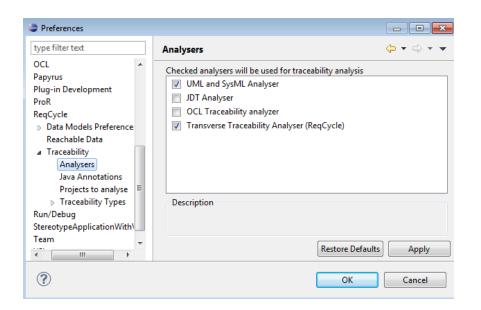


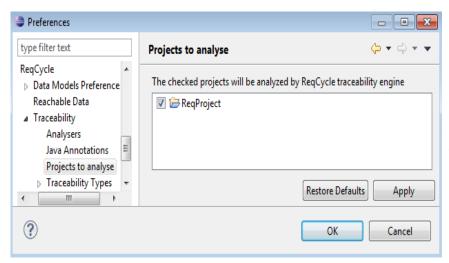
Ability to use predicates to filter or underline some links

5. Capture existing links



- ReqCycle provides analysis engines
 - Can select some engines



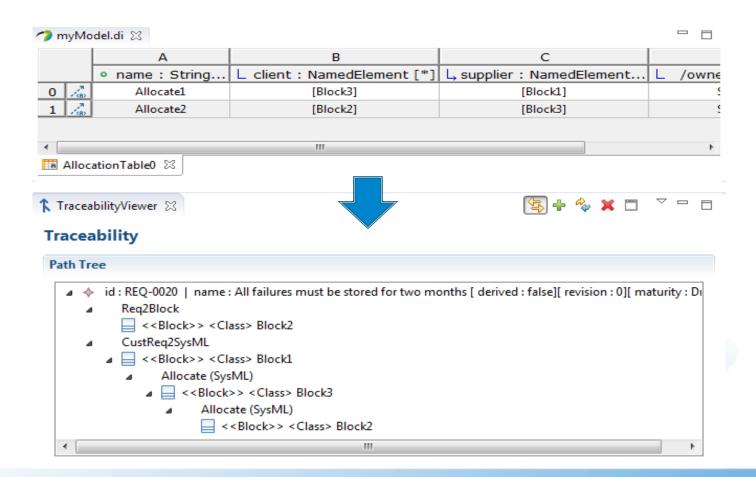


And can choose projects to analyse

5. Extended traceability



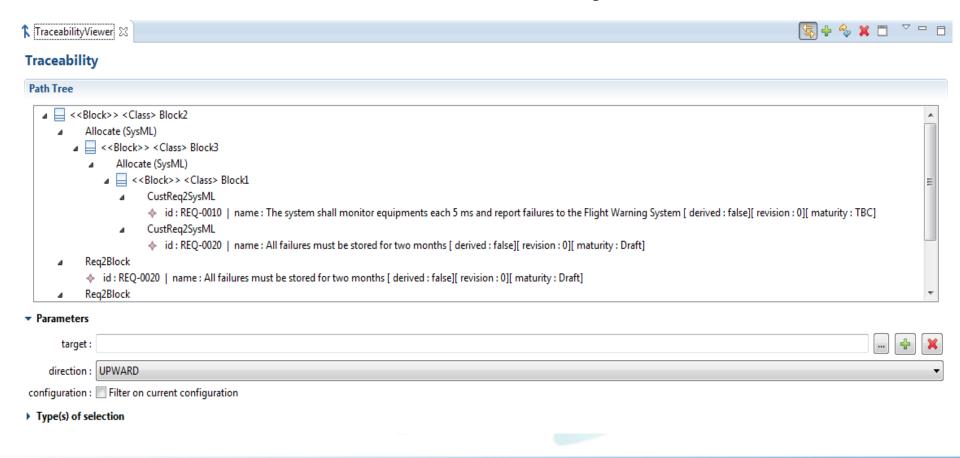
 Capture other links (SysML, Xcos) and complete transverse links



5. Upward traceability



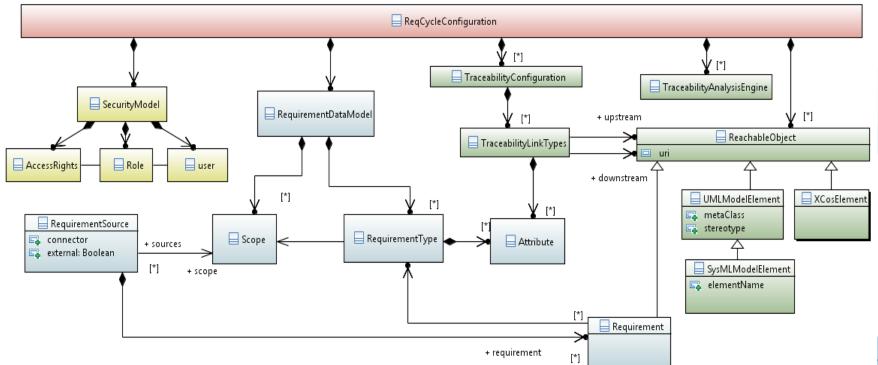
- Can change direction to upward
 - And then start from a block or any model element...



Summary



- Requirements data model
- Traceability creation and analysis
- Security model = to specify



Teamwork overview

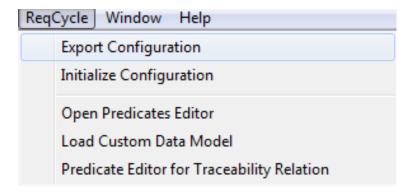


- Can share ReqCycle configuration
 - Export/Import
- Can put requirements file and traceability file to a SVN repository
 - Commit/update manage versions
- Will be possible (in the future) to synchronize requirements and traceability links at fine-grained level with diff/merge approach
 - Not yet operational (bugs)

Share ReqCycle configuration



- Can export ReqCycle full configuration as a file
 - Data saved = Data models + reachable types + traceability configurations + requirement sources
 - Menu "ReqCycle">export configuration
 - Save it as a file can then be distributed to others



Can then initialize configuration in an other workspace

Still a lot of work to do



- Consistency between requirement data model and traceability links
- Consolidate teamwork architecture
- New features to finalize
 - Export, predicate, attributes...
- Ergonomics improvements
- New connectors
 - Word source, XCos traceability...
- Combine several link types to define semantics

Questions?

