SAMARES
ENGINEERING
Accelerate Systems Design

ReqCyle starting guide

January2014

Raphaël FAUDOU
raphael.faudou@samares-engineering.com

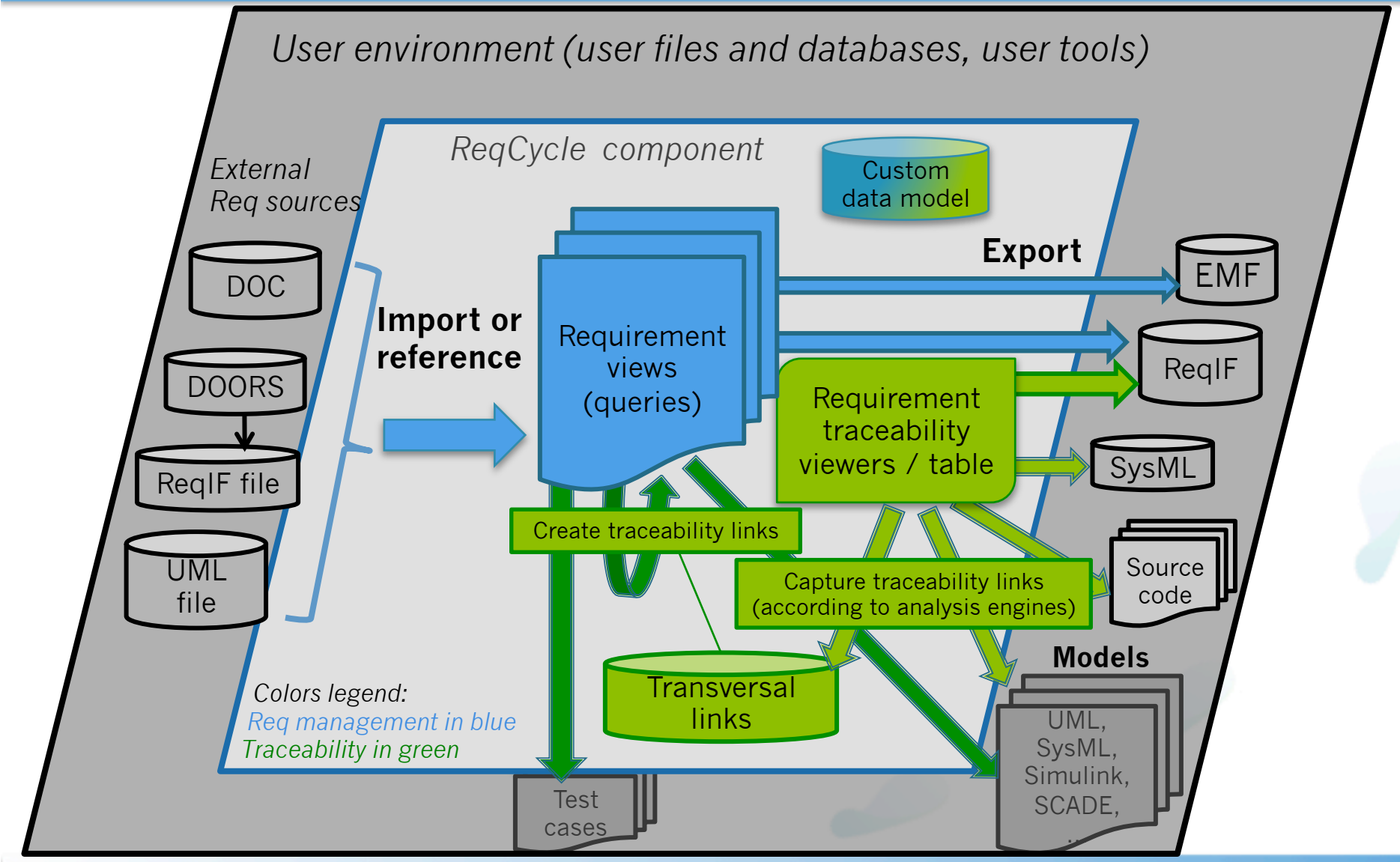
Agenda

- Motivations, overview and current state
- Installation
- Big picture and detailed usage step by step
  1. Data model – scopes and types
  2. Requirement import, creation and visualization
  3. Configuration and creation of traceability links
  4. Capture and management of traceability links
  5. Export of requirements and links (doc, matrix,…)
- Teamwork support
- Questions / contacts

Motivations

- Provide an **open source solution** supporting all activities concerning requirements in the development of an industrial product or system:
  - Requirement engineering (from needs to requirements…)
  - Requirement management (classification, configuration…)
  - Requirement traceability (links between requirements or between one requirement and any development artefact including model element, test case, source code…)
- Provide a flexible solution able to adapt to any organization
  - Can organize requirements with a user-defined data model
  - Can drive traceability through a user-defined process
  - Can import or reference requirements managed in an other tool
  - Can trace requirements to models of any language and tool
  - Can capture existing requirement traceability links

ReqCycle overview



Current state

- ReqCycle does not yet implement requirement engineering (2015 plan)
- ReqCycle provides means to create a custom data model to manage requirements and traceability relationships
- ReqCycle can import requirements coming from different external sources or can create requirements
  - External requirement sources already supported: ReqIF, EMF models
- ReqCycle can create traceability links between requirements or between one requirement and a model element according to the traceability relationships defined in the custom data model
  - EMF UML and SysML models supported for now
- ReqCycle can capture existing traceability links or any link from models and display extended traceability (all links from one element to another)
  - SysML, OCL and Java traceability analysers
- ReqCycle cannot yet export requirement traceability in EMF format (soon)

ReqCycle installation

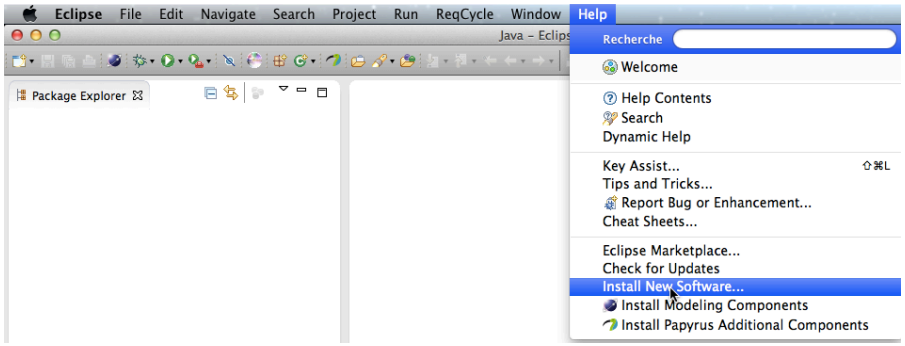
Available for Windows, Linux and MacOS
Checked for Windows 7 64 bits and MacOS 10.8 64 bits

Installation steps – Eclipse

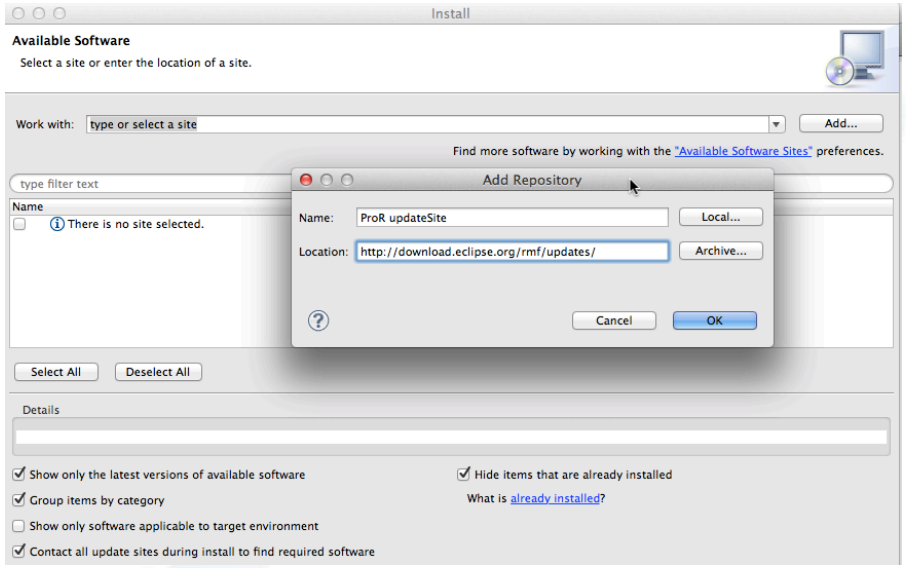
1. Ensure you get a Java Runtime Environment V1.6 or higher installed
   a. In terminal view type and run command line “java –version”
   b. If version is lower than 1.6 or if you do not have java installed, download and install it according to your operating system: http://www.java.com/fr/download/
      - Note: for Windows 64 bits, you might need to install a Java 64 bits runtime to fully benefit from 64 bits
2. Download Eclipse Kepler Modeling SR1
   a. http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/keplerr
   b. Choose your installation package according to your operating system (installation packages differ for 32 bits or 64 bits)

Installation steps – ReqIF support (1)

- Launch Eclipse application
- Go to Help>new software

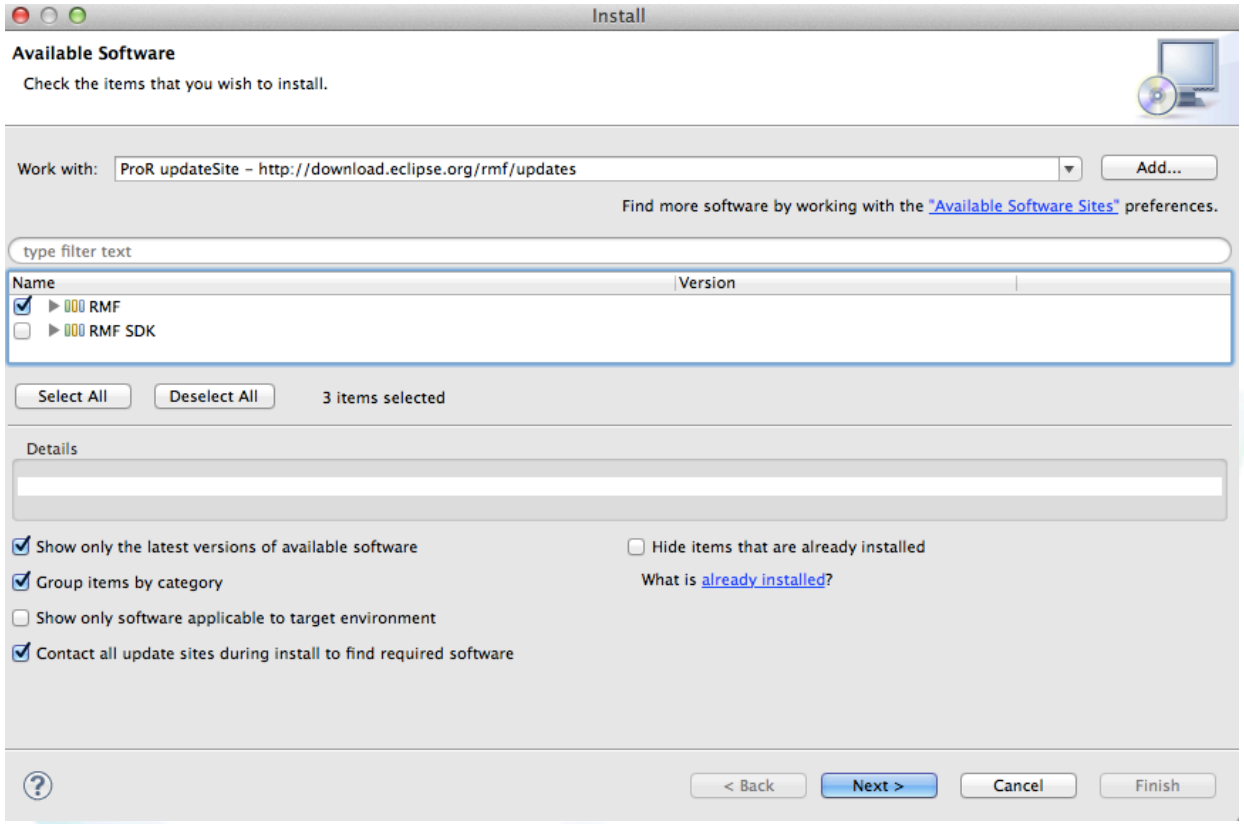


- Click “Add” button
  - Fill name and location
  - http://download.eclipse.org/rmf/updates/
  - Click “OK”



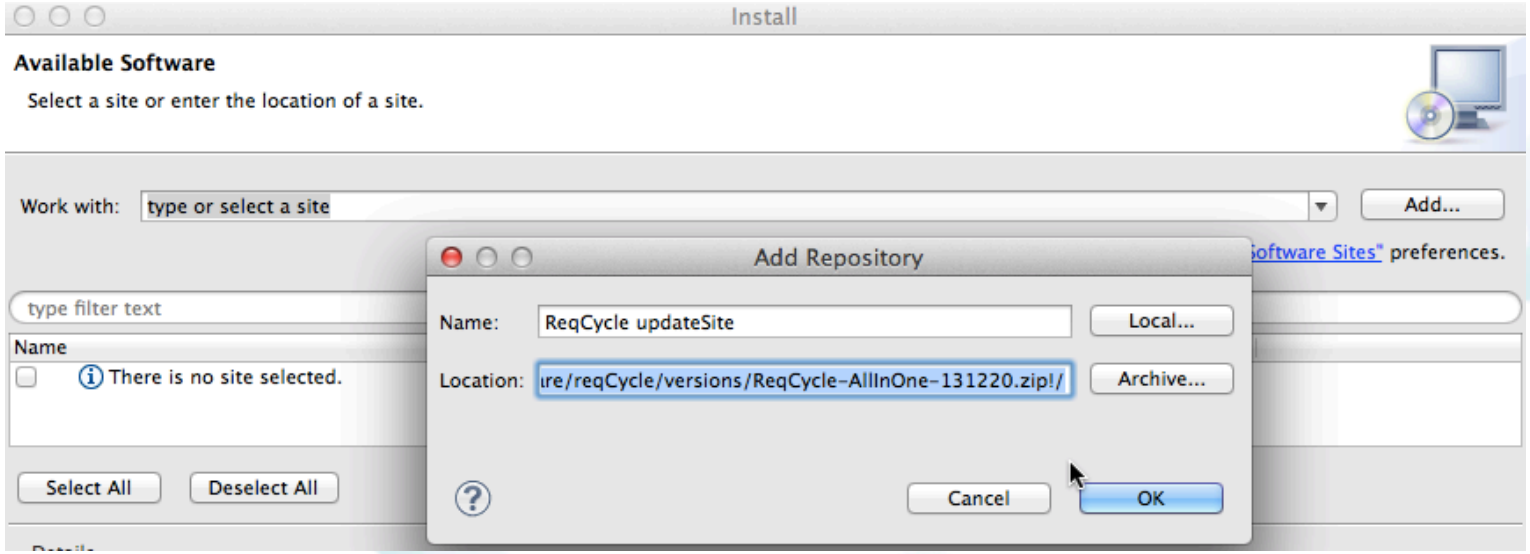
Installation steps – ReqIF support (2)

- Only select « RMF » and follow wizard with check boxes filled as below
  - Next,
  - Accept license
  - Finish
  - OK to install
  - Restart Eclipse



Installation steps – ReqCycle

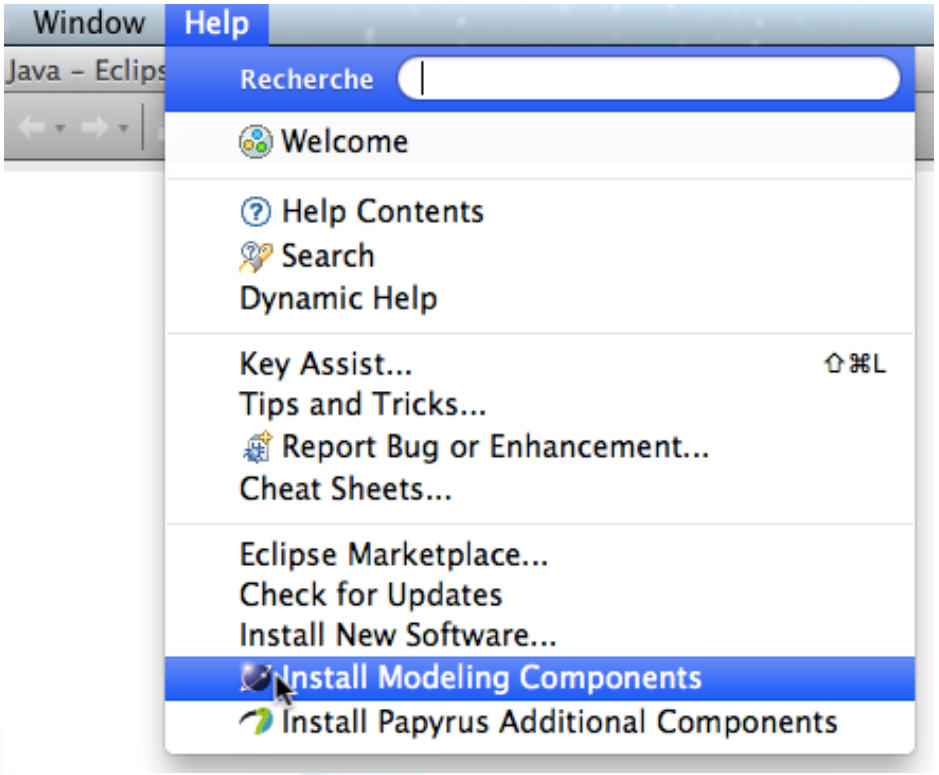
1. Download ReqCycle updateSite
   - https://github.com/raphaelfaudou/ReqCycle/blob/master/releases/ReqCycle-AllInOne-131220.zip
2. In Eclipse, menu Help>new software
   - Add ReqCycle update site (select .zip with “archive” button)



   - OK - follow wizard (includes Eclipse restart) - That’s it ☺

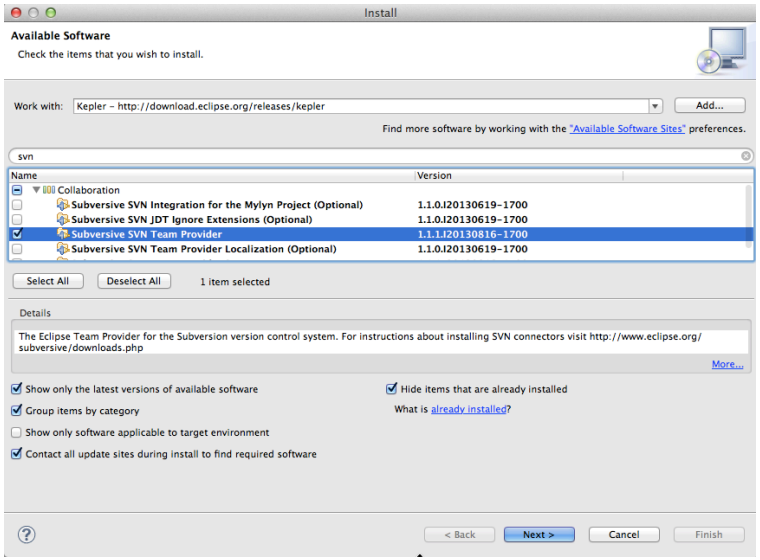
Installation steps – MDT Papyrus

- You can then install MDT Papyrus to create links between requirements and Papyrus models…
  - Menu Help>Install Modeling Components
  - Choose Papyrus
  - Follow wizard



Installation steps – SVN connectors

- You can install SVN team and connectors to put your requirements and models in version management in a SVN repository
  - Team plugins are part of Kepler release: install new software, select Kepler and then “collaboration>subversive SVN team provider)

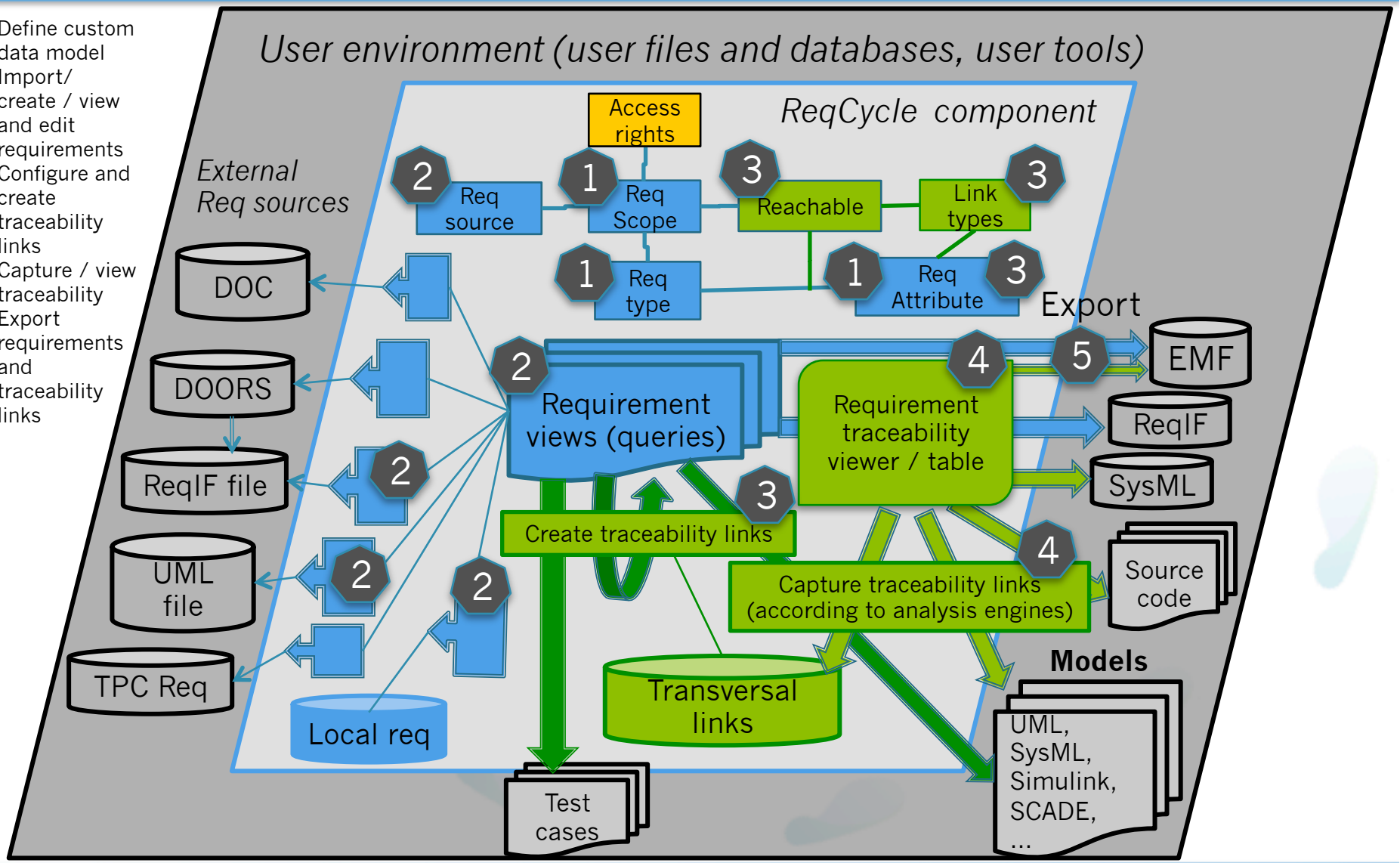


  - Install connectors with update site:
    - http://community.polarion.com/projects/subversive/download/eclipse/3.0/kepler-site/
  - In case of issues, full Installation instructions are detailed here: http://www.polarion.com/products/svn/subversive/download.php

Big picture and steps overview

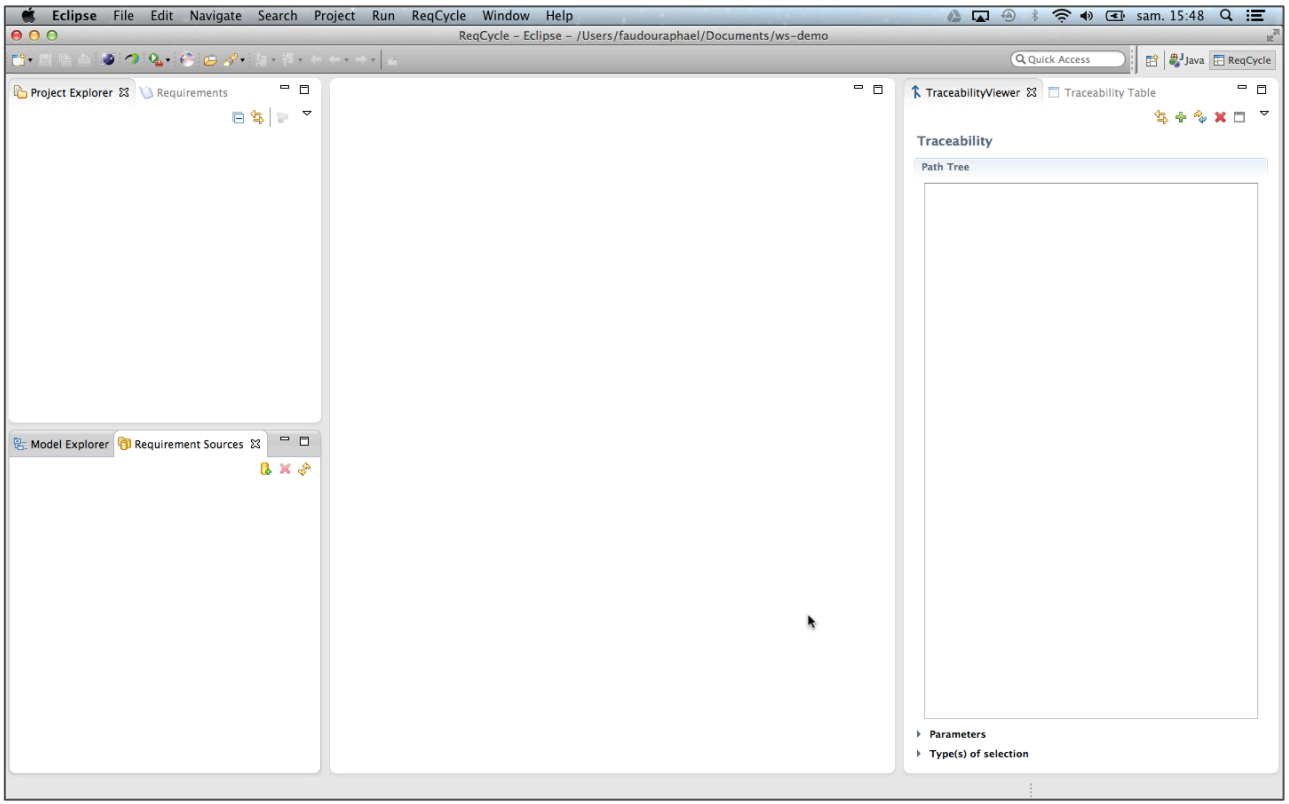
Big picture and steps overview

1. Define custom data model
2. Import/ create / view and edit requirements
3. Configure and create traceability links
4. Capture / view traceability
5. Export requirements and traceability links



Requisites - ReqCycle perspective

- Once ReqCycle is installed (see previous slides), launch Eclipse, choose a workspace and switch to ReqCycle perspective
  - Workbench is then organized in different views (requirement sources, editors, traceability viewer…) that will be presented in next slides



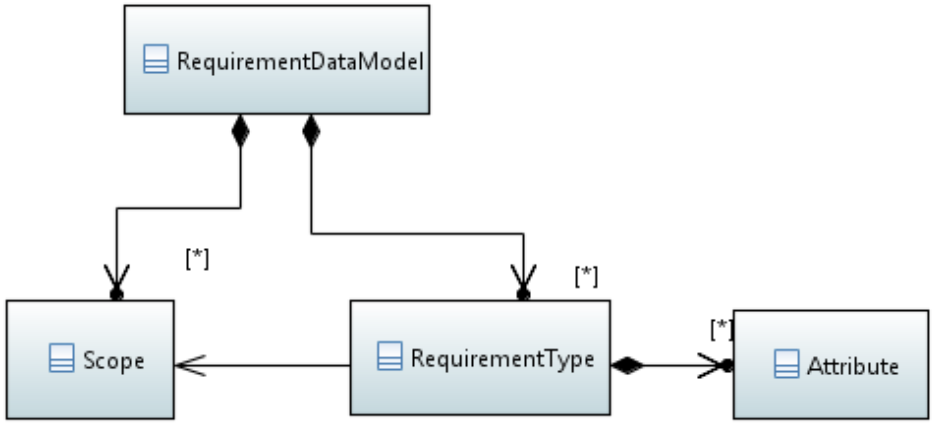
SAMARES
ENGINEERING
Accelerate Systems Design

1st step - Requirement data model

Concepts and screenshots

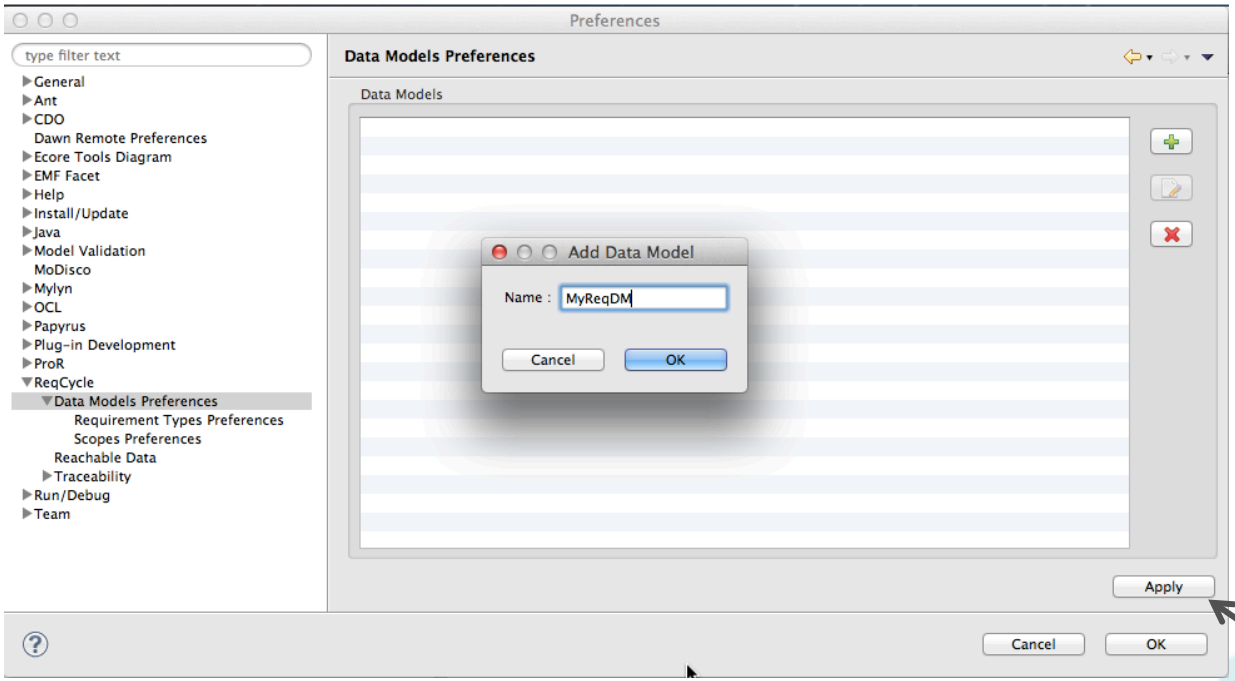
Requirement custom data model

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



Create data model

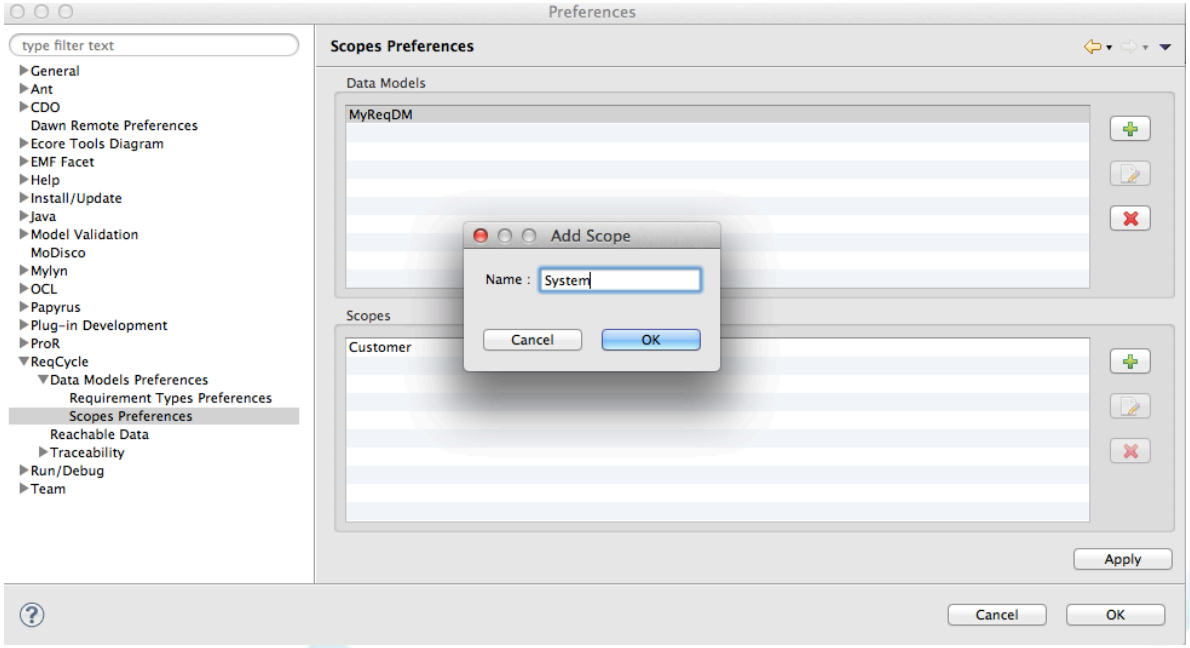
- Go to main menu Preferences>ReqCycle
- In Data Models Preferences page, click “+” button and create a data model “MyReqDm” – click “OK”



- Then select newly created data model and click « apply » button to register it

Create scopes for this data model

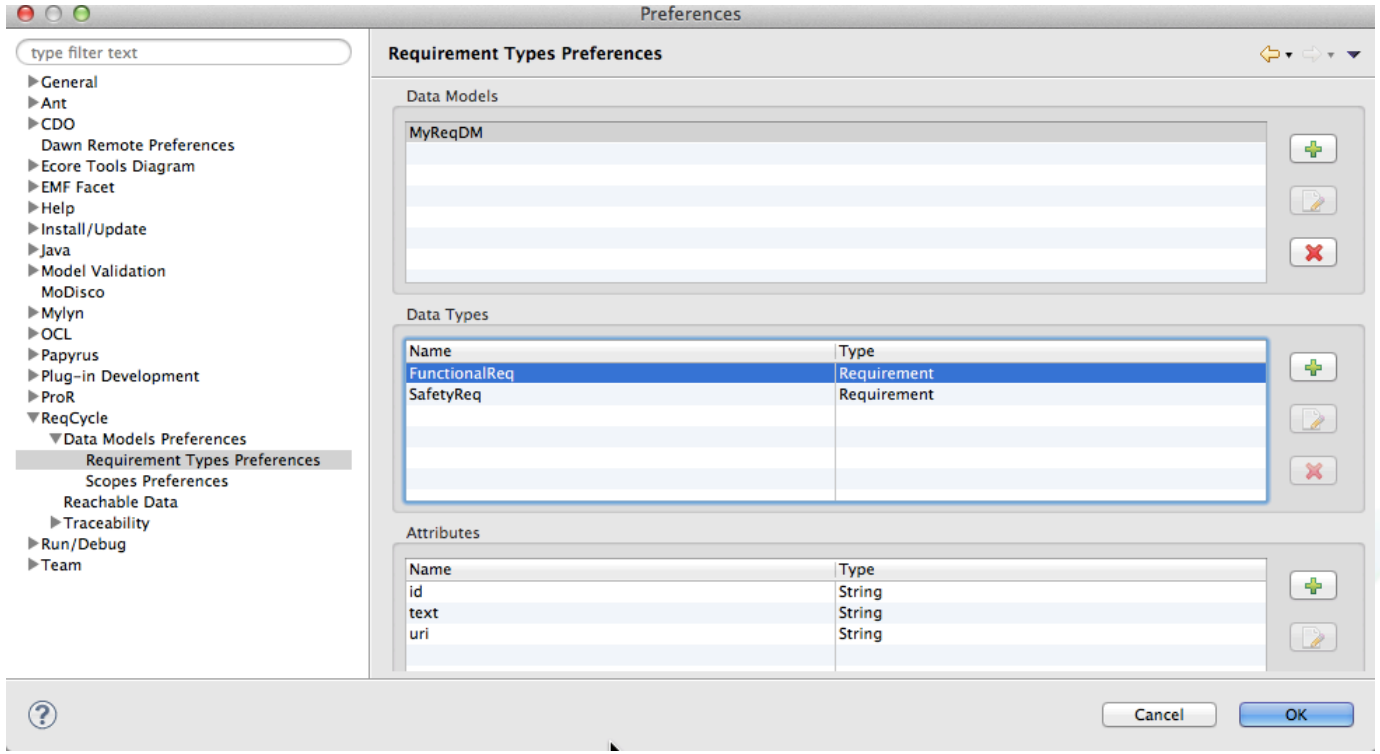
- Go to Scopes Preferences page
- Select “MyReqDM” data model and add scope
  - For example “Customer” and “System”



- Then click « apply » button

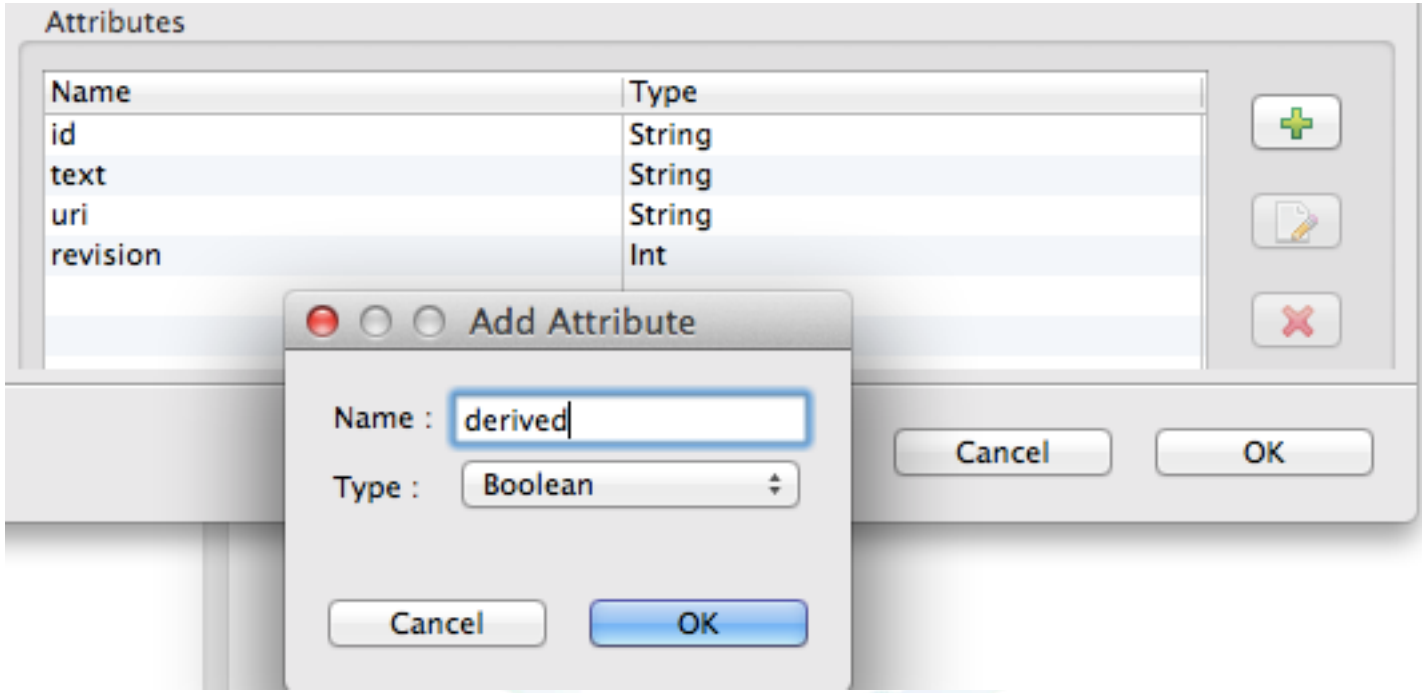
Create requirement types

- Go to Requirement types Preferences page
- Select “MyReqDM” data model and add data types
  - For example “FunctionalReq” and “SafetyReq”



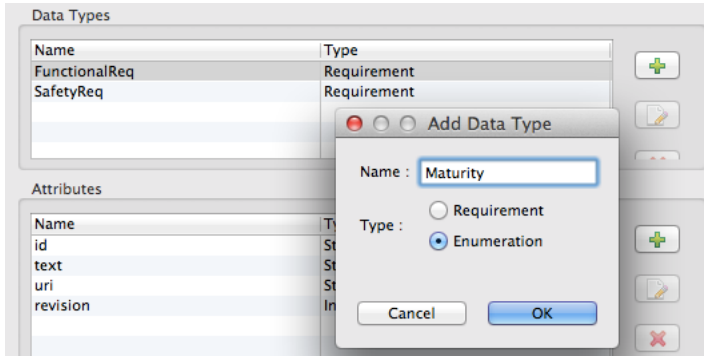
Add requirement types attributes (1)

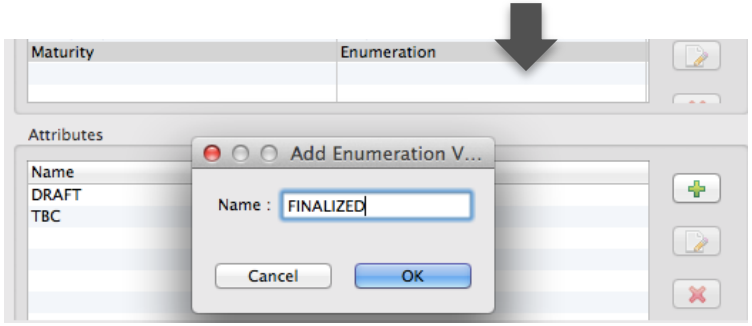
- Select one requirement type (data type)
- Can add attributes of predefined types
  - Examples: “revision” of type “int” and “derived “ (bool)



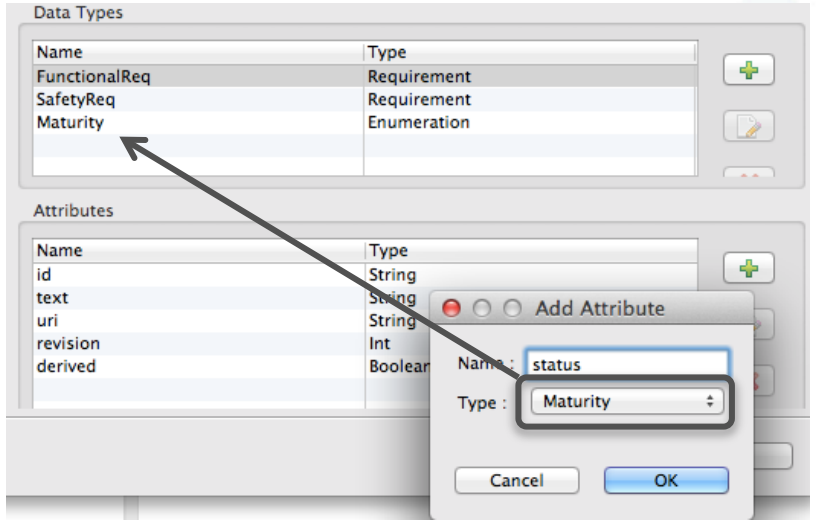
Add requirement types attributes (2)

- Can also define (new) enumerations
  - Add new data type and select “enumeration”
  - Example: Maturity
  - Then select enum data type
  - Add values from attributes area





  - Now can define a new attribute typed with this enum



Step 1 conclusion

- Now we have seen how to create a custom data model with scopes, requirements types and attributes
- In step 2 we will see how to integrate external requirements or create own requirements that conform to this data model

SAMARES ENGINEERING

Accelerate Systems Design

2nd step – Import /reference / create view / edit / filter requirements

Concepts and screenshots

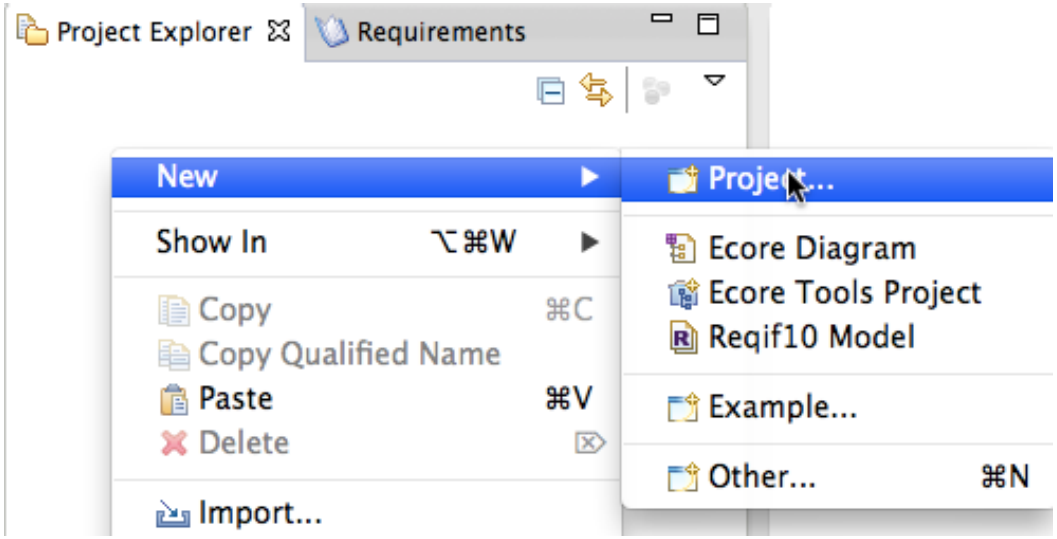
Retrieve customer requirements

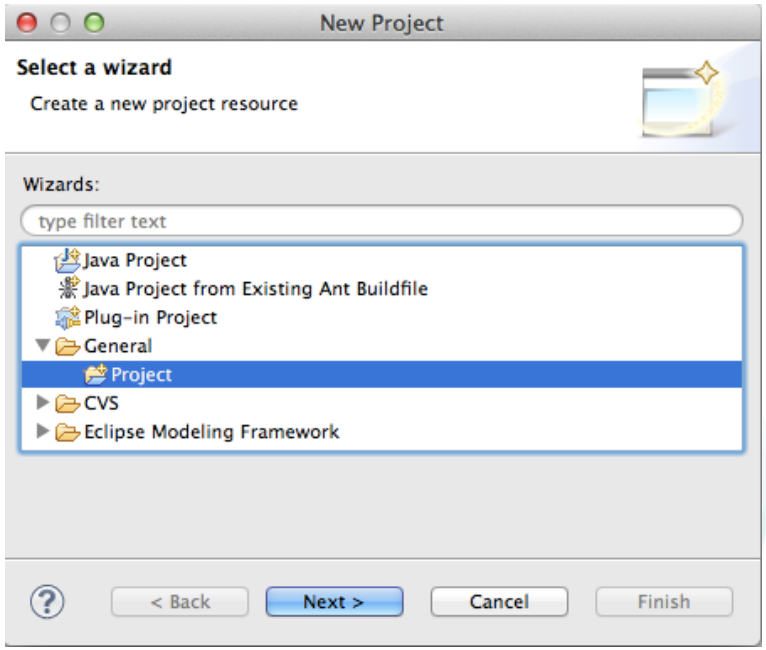
- Generally customer requirements have been edited through a requirement tool (Doors ™, Requesite Pro ™..) or through a document (Word ™, Excel ™, OpenOffice)

- We want to retrieve those requirements and visualize them into ReqCycle so that we can easily formalize them, refine them, implement them and verify them with different system development artefacts (other requirements, model elements, documents, source code…)
  - Next slides show how to do that practically

Eclipse project to host requirements

- We create an Eclipse project to store customer requirements

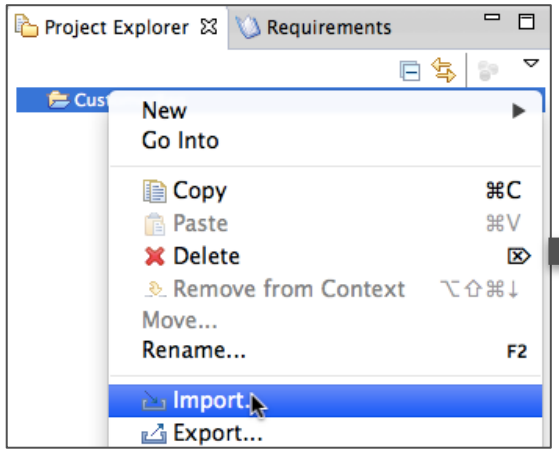


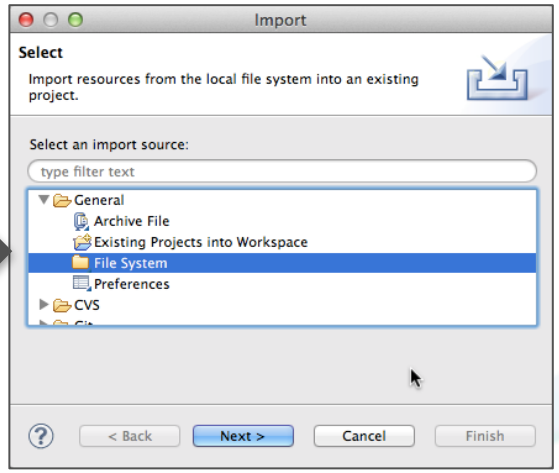


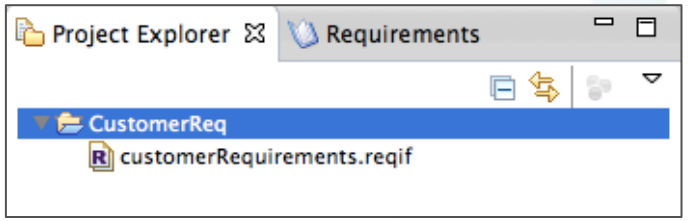
- Give name for project (for instance « CustomerReq »)
- OK – project is created

Store customer requirements

- We start with requirements formalized in ReqIF format
  - ReqIF is OMG standard Interchange format for requirements. All requirement tools should now support ReqIF export
- Select your Eclipse project and then import customer requirements .ReqIF file. Sample file available here:
  - https://github.com/raphaelfaudou/ReqCycle/blob/master/resources/sample/customerRequirements.reqif



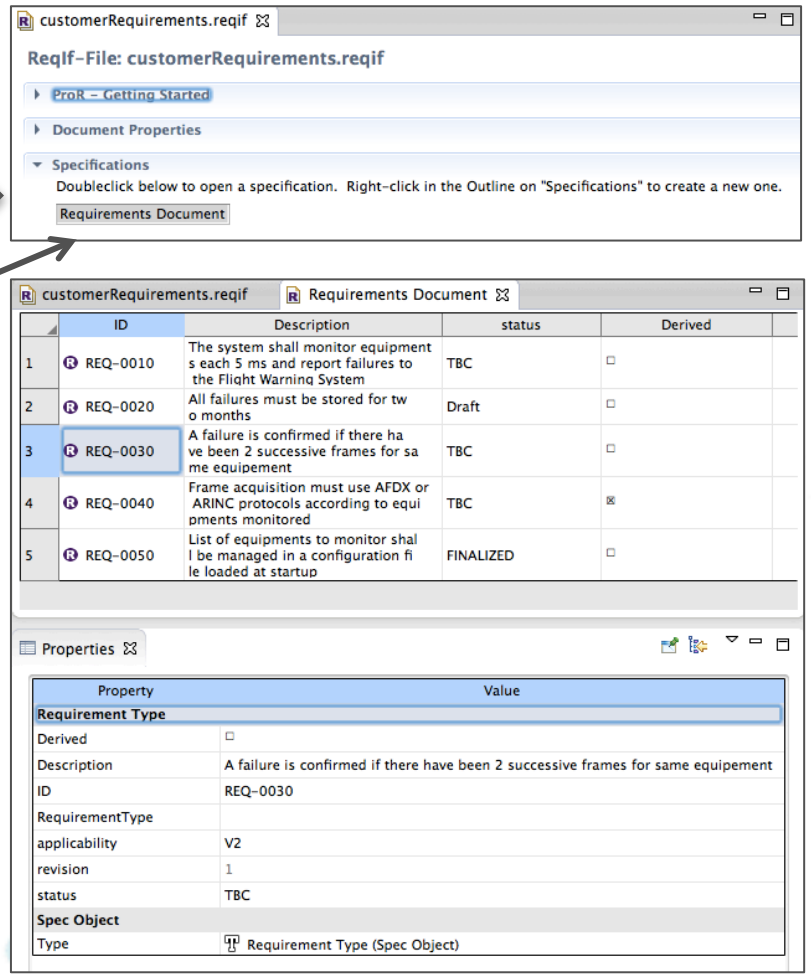




Result after import

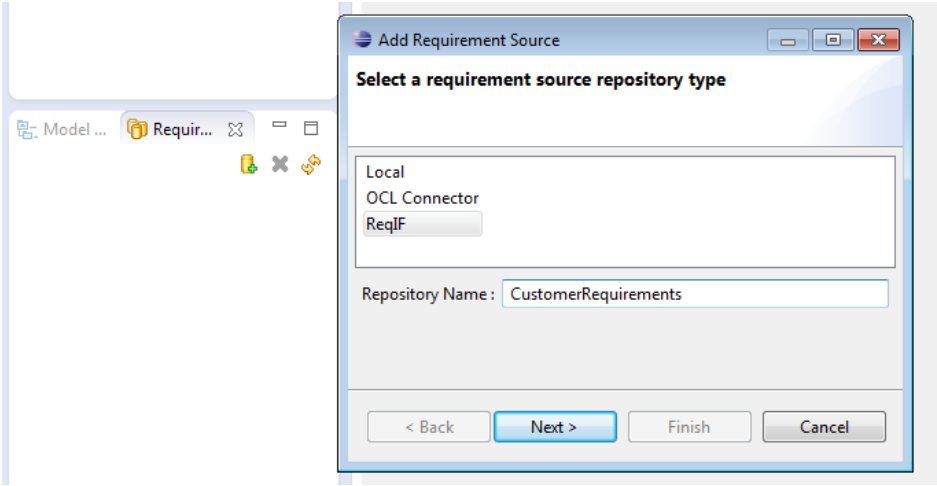
Visualize .ReqIF requirements

- Thanks to ProR Eclipse tool you can see your ReqIF requirements in a table
  - Double click on the .reqIF file and ProR editor opens
  - Double click on a ReqIF specification to see the requirements in a custom table
  - Right click to open the properties view and select one requirement -> all requirement attributes are listed



Import ReqIF requirements in ReqCycle (1)

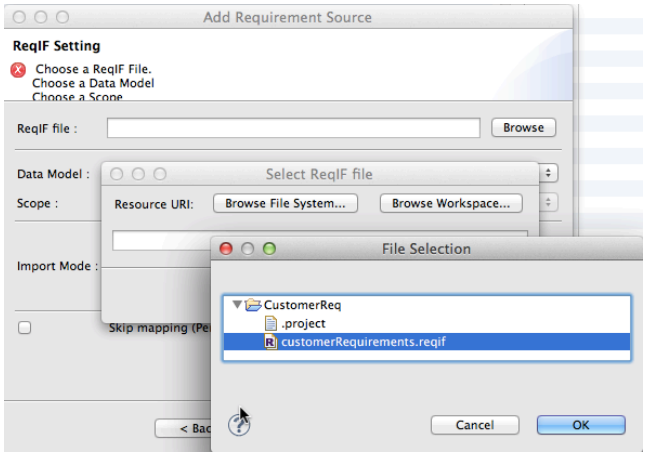
- Now we want to integrate those requirements with our custom requirement data model (MyReqDM – see step 1)
  - ReqIF file does not contain any links with this data model, so we must perform a mapping (import)
- Go to requirement sources view
  - Add requirement source (+) and choose ReqIF connector
  - Give name for this source (ex: customerRequirements)
  - Next



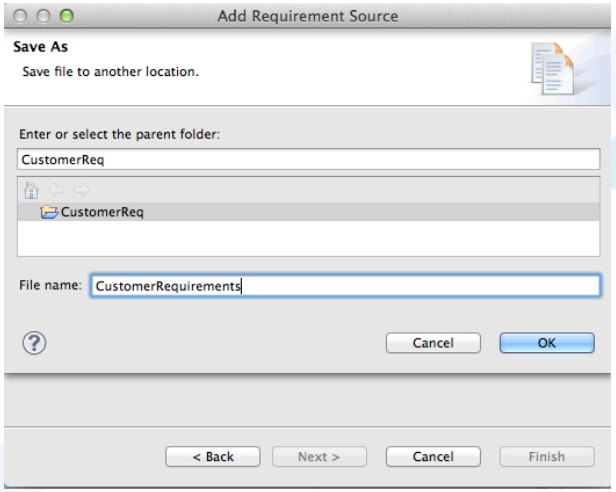
Import ReqIF requirements in ReqCycle (2)

SAMARES ENGINEERING

- Select .reqIF file with “browse” and then “browse workspace”



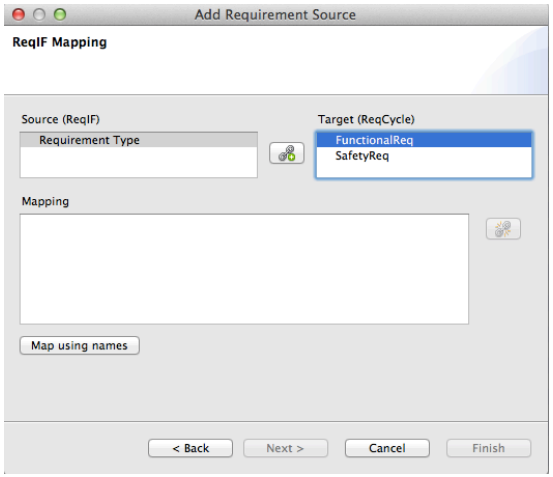
- Select data model and scope
  - “MyReqDM” for data model
  - “Customer” for scope
- Select destination file
  - Select project and then put a name (same than source)
  - Note: will be filled automatically in a future release

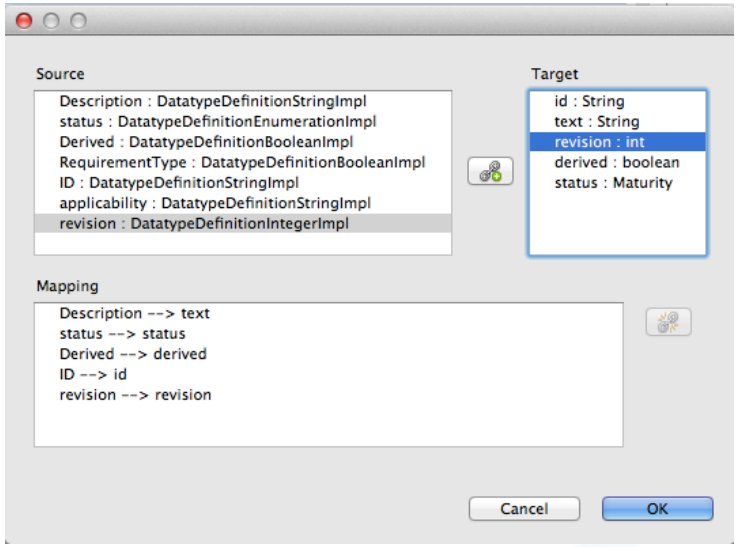


- Next

Import ReqIF requirements in ReqCycle (3)

- Map ReqIF specifications to a requirement type
  - For instance « Requirement Type » to « FunctionalReq »
  - Click the « map » button (between source and target): a new screen opens that lets you map attributes

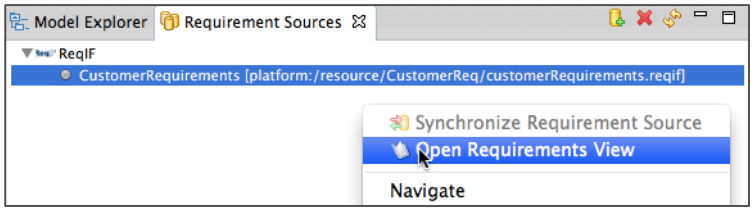




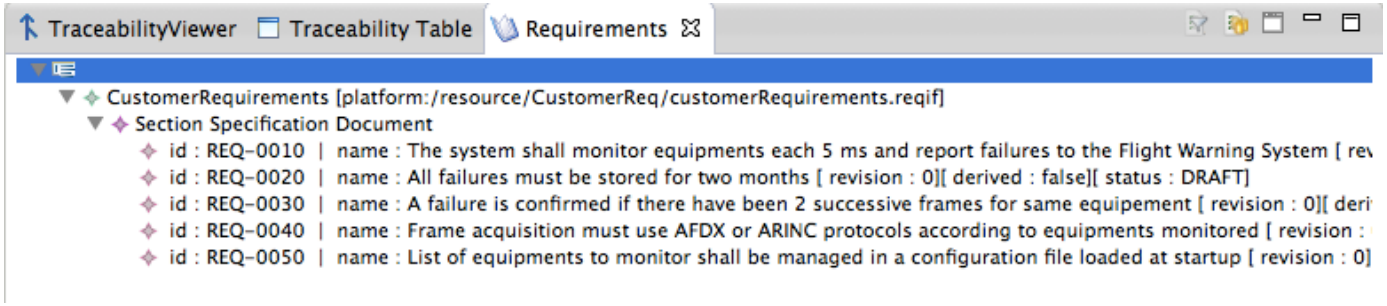
- Note concerning mapping:
  - In this example, « Applicability » has no mapping in target: you can cancel import and update your data model with new attribute ‘applicability’ (string) and then do the mapping (import) again to get applicability mapped

See mapping result

- « CustomerRequirements » source has been created
  - Right click > « open in Requirement view »



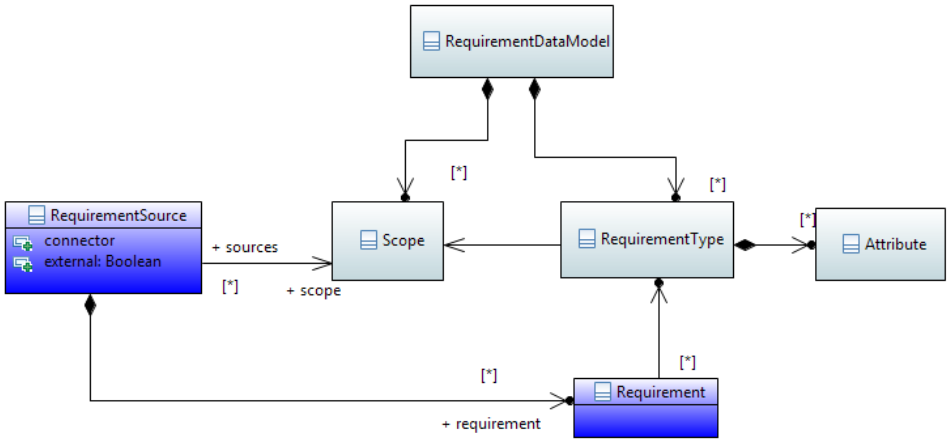
  - A Requirements treeview (readOnly) is created and that can be expanded



  - Note: this editor is poor compared to what ProR can bring but ReqCycle needs an editor able to show all requirements (not only ReqIF) and linked to requirement data model: so we can not use ProR editor for now - This is an area where we intend to find some synergy with ProR team

And now?

- Now we have integrated some customer requirements in our custom data model



- If you need to integrate more customer requirements, you can create a new requirement source
  - New connectors for external sources are planned for future releases: Word, Excel, openOffice, Topcased Req
  - If needed you can update your data model (new scopes, new requirements types, attributes) with current limits: labels cannot yet be renamed (later) and no deletion for now (some rules still to specify)
- Then time has come to trace our customer requirements…

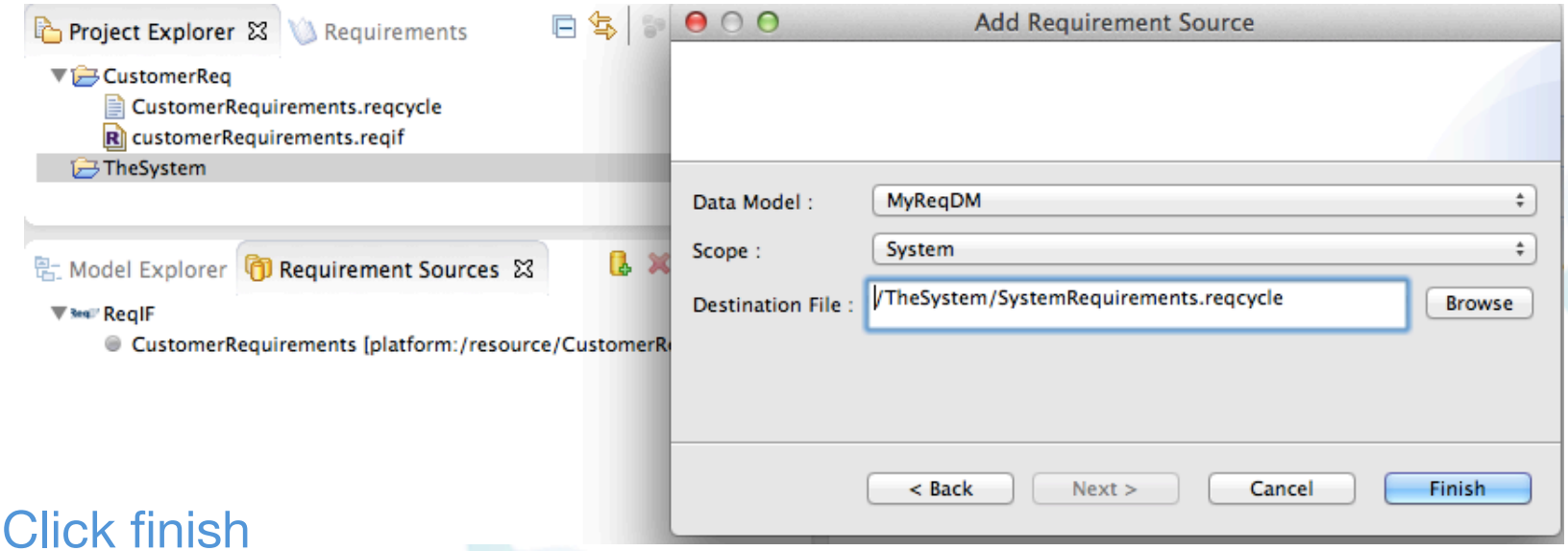
Trace customer requirements: 2 options

- If customer requirements already provide good definition of target system, you can switch to modelling world to refine and implement those requirements. This is first option.
  - To do that, you can jump to step 3 – configure traceability

- Else, you might consider that you need to refine customer requirements through System textual requirements. This is second option.
  - Next slides show how to create your own textual requirements

Create textual requirements (1)

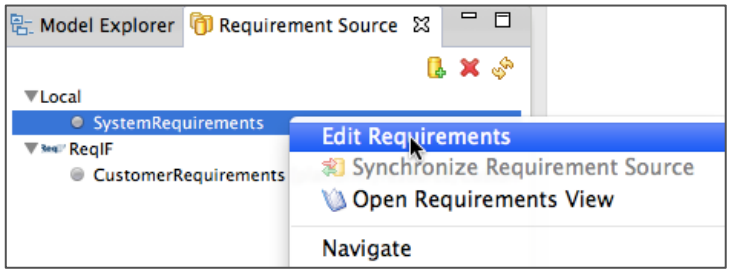
- Create a new Eclipse project “MySystem” to differentiate customer requirements and System requirements that you are about to define
- Go to Requirement sources view and add source
  - Choose “Local” requirement source connector
  - Fill input parameters

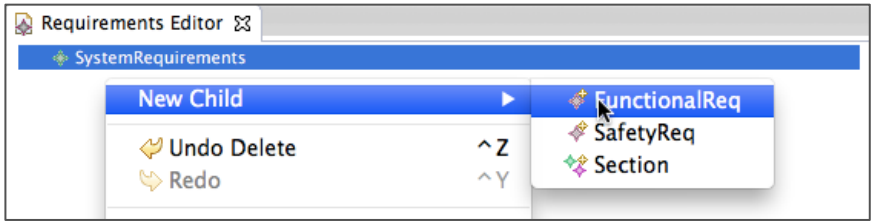


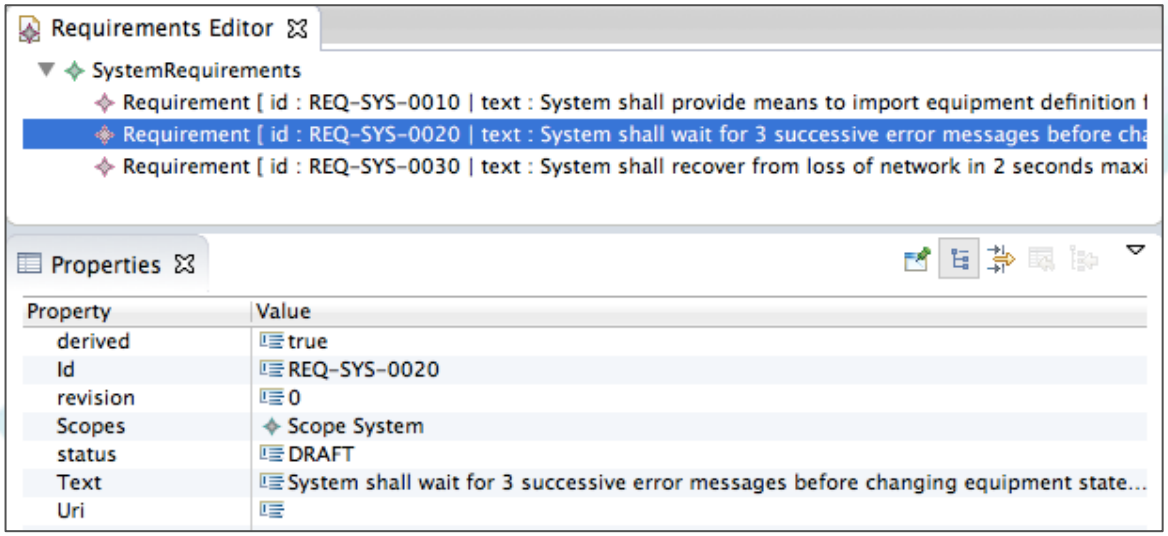
- Click finish
  - That’s it (no mapping)

Create textual requirements (2)

- Now you can edit those requirements (right click)
- A minimalist Requirement editor lets you create requirements according to types defined in data model
- Attribute values are set through the property view
  - Note: contextually to selected requirement

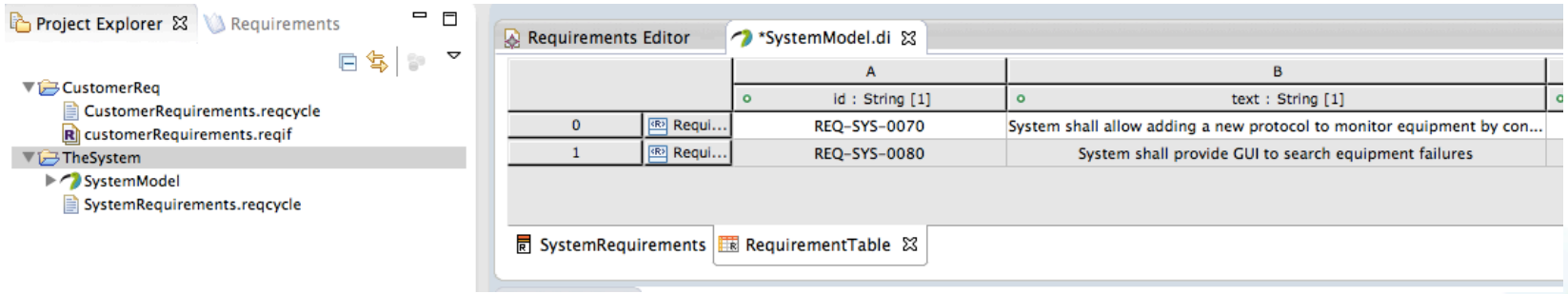






Create textual requirements (3)

- You can also create textual requirements with Papyrus SysML and use OCL requirement source connector to extract those requirements from model

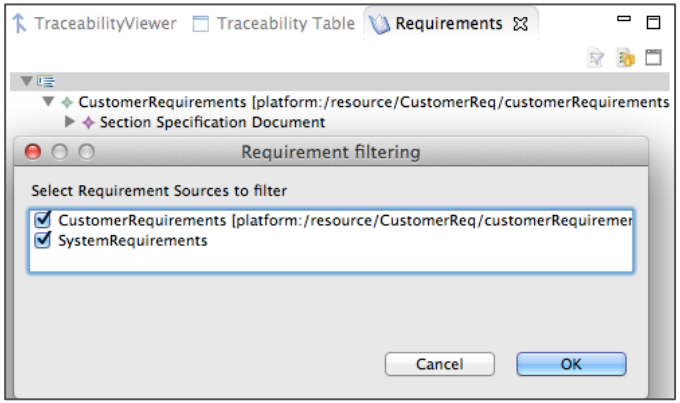


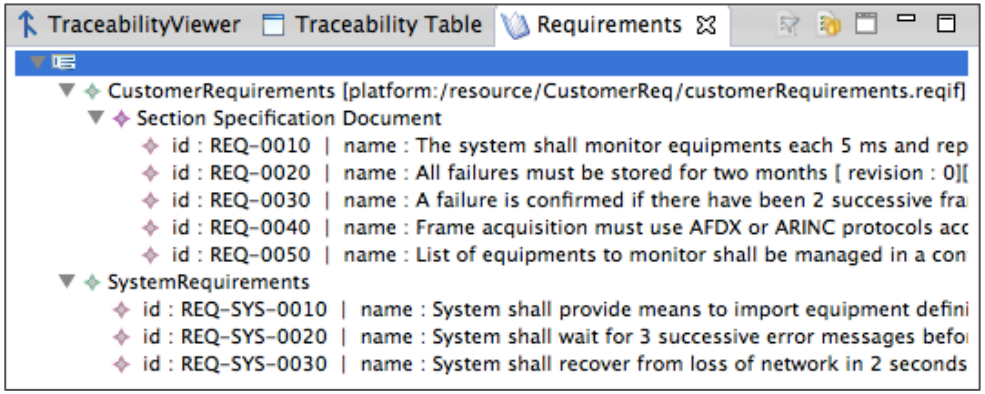
  - This « OCL » connector is configured by providing an OCL file that contains rules to identify requirements and their attributes.

- *Information of how to fill this OCL file will be provided in a future release of this guide.*

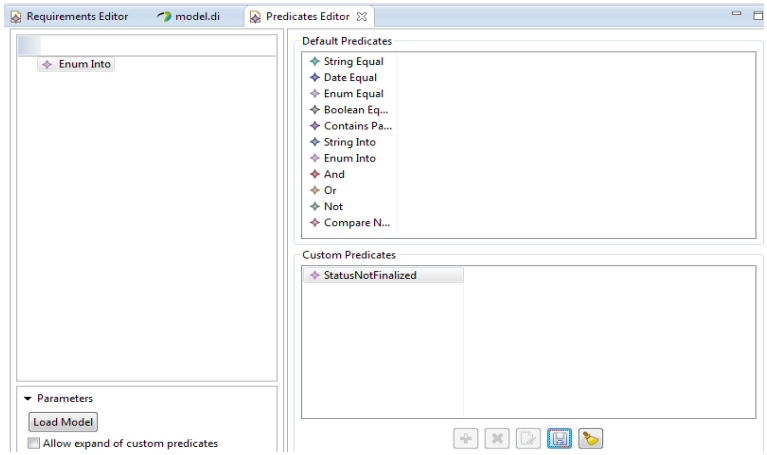
Queries/filters on requirements

- We can already filter requirements on their source



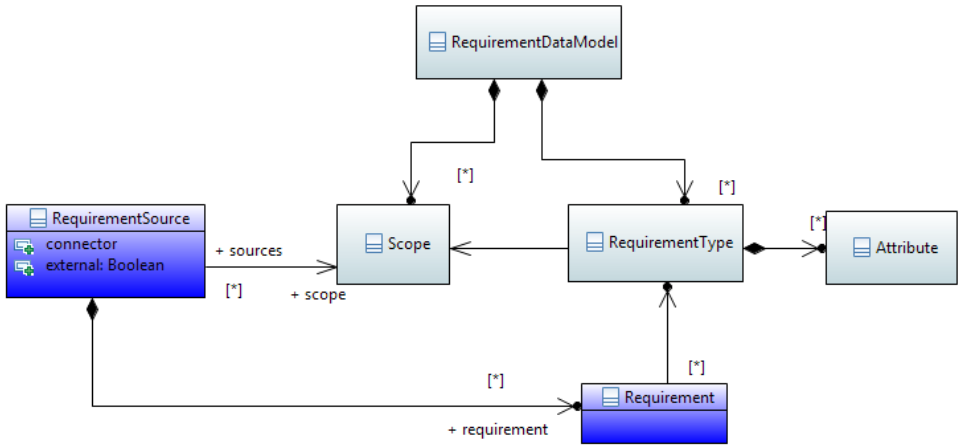


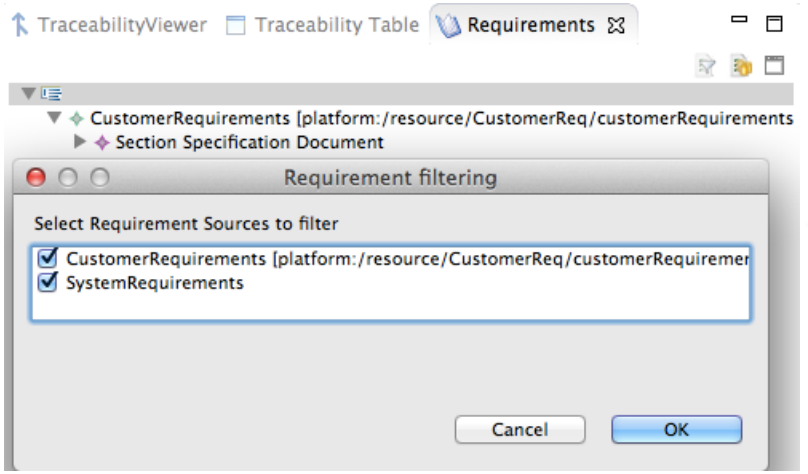
- In a future release we will have ability to define predicates (queries) to filter views
  - Feature partially developed
  - Requires more tests



Step 2 conclusion

- Requirements (external or local, ReqIF or SysML) can be imported or created in ReqCycle so that they conform to the requirement data model defined in step 1
  - We can filter them and show them in requirements views





- Now we are going to show in step 3 how to trace them

SAMARES
ENGINEERING
Accelerate Systems Design

3rd step – configure traceability links

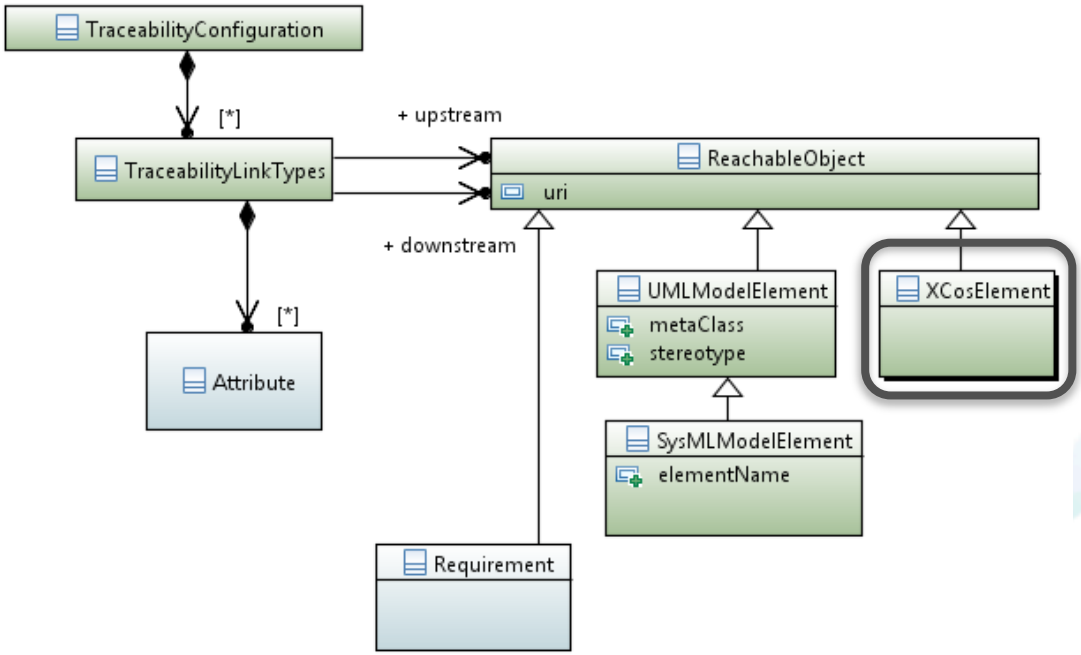
Concepts, screenshots

Traceability overview

- 2 kinds of traceability engines: generic and specific
- Generic engine able to create any link
  - Between reachable data and reachable data (see later)
  - Can restrict possible links by configuration or checks
  - Currently requires specific GUI for Drag and Drop support
- Specific analysis engines able to capture existing links
  - One by existing tool/language
  - First targets are UML/SysML and Java, next is Xcos
  - Adding a target consists in implementing two interfaces
- Note: 3rd step focuses on link creation (generic engine), capture of existing links is presented in step 4.

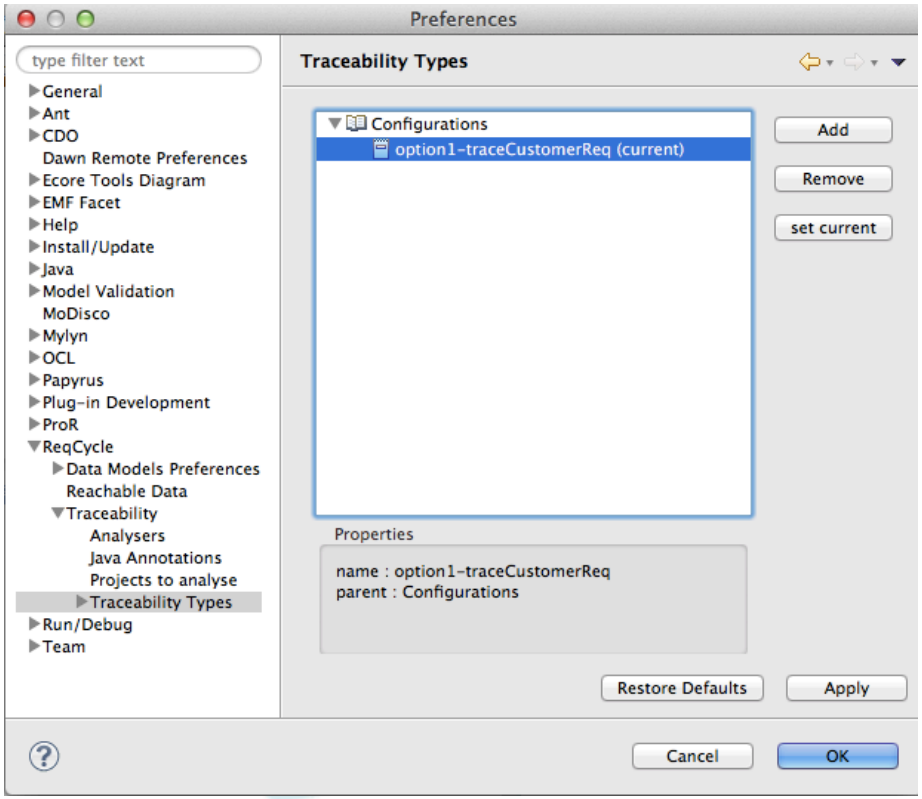
Traceability configuration

- By default, reachable objects known by ReqCycle are
  - Requirements, UML and SysML model elements
- Can capture other reachable objects with new plugins
  - Just have to implement 3 extensions (interfaces)



Configuration to trace customer req.

- In preferences>ReqCycle>Traceability>Traceability types
  - Select “Configurations” root node and click “add”
  - Give name (for instance “option1-traceCustomerReq”) - OK
  - Click “set current”
    - To activate it
  - Click “apply”
    - To register data



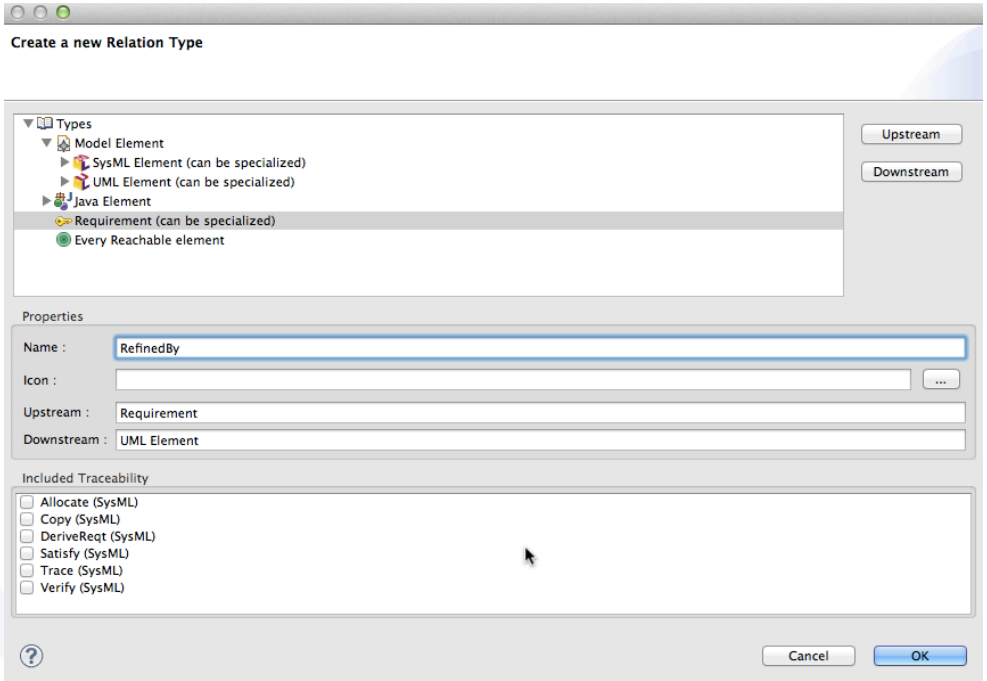
Traceability links for customer Req.

- To trace our customer requirements we want to define several kinds of links. For instance:
  - “RefinedBy” that will concern UML (and SysML) elements
  - “ImplementedBy” that can concern any reachable data
  - “VerifiedBy” that only concerns SysML test cases

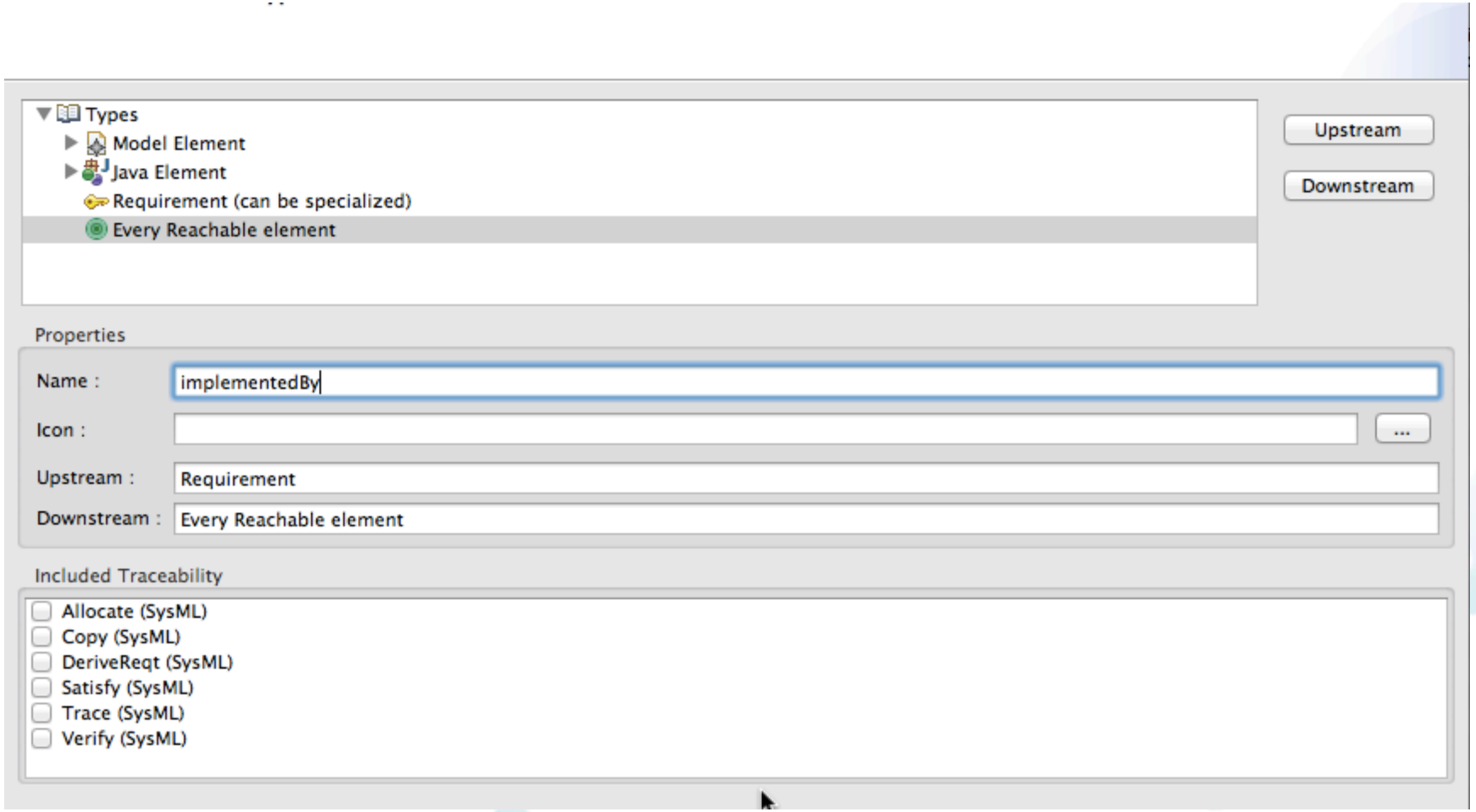
- Next slides show how to do that

Create “RefinedBy” traceability link type

- Select upstream data type - click “Upstream” button
  - Here we choose “Requirement”
- Select downstream data type - click “Downstream” button
  - Here we choose “UML element”
- Give name
  - “RefinedBy”
- OK and then Apply



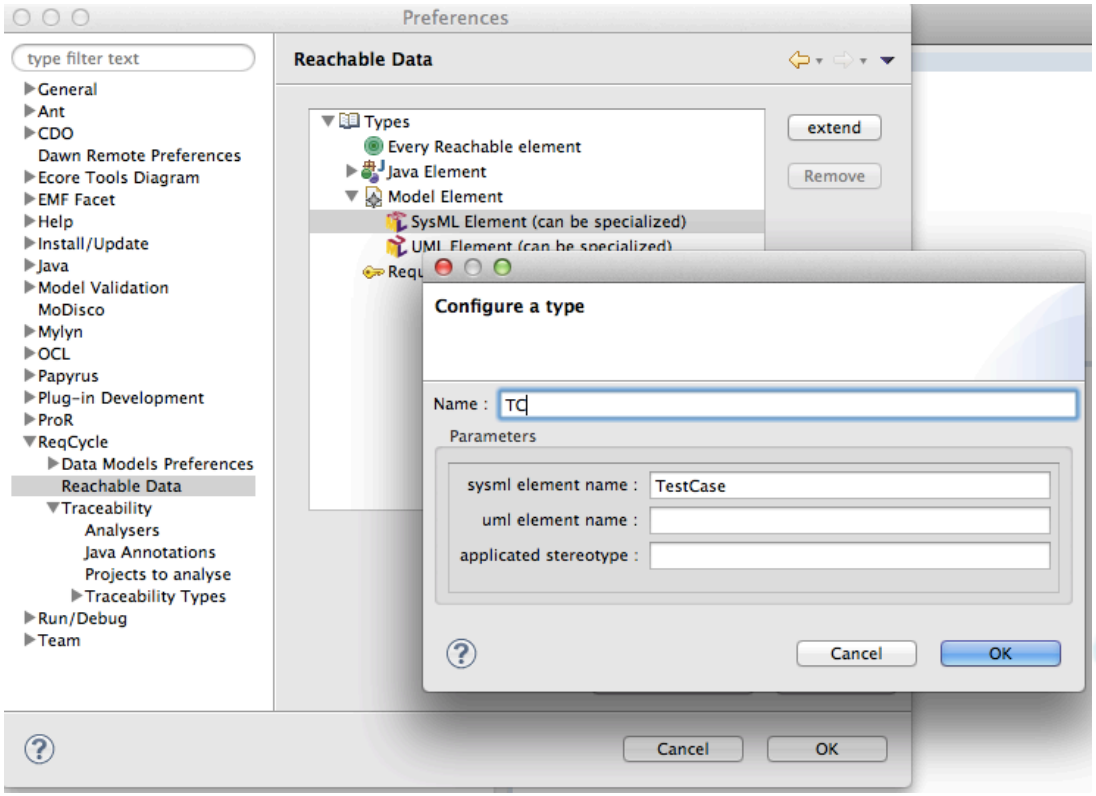
“ImplementedBy” traceability type



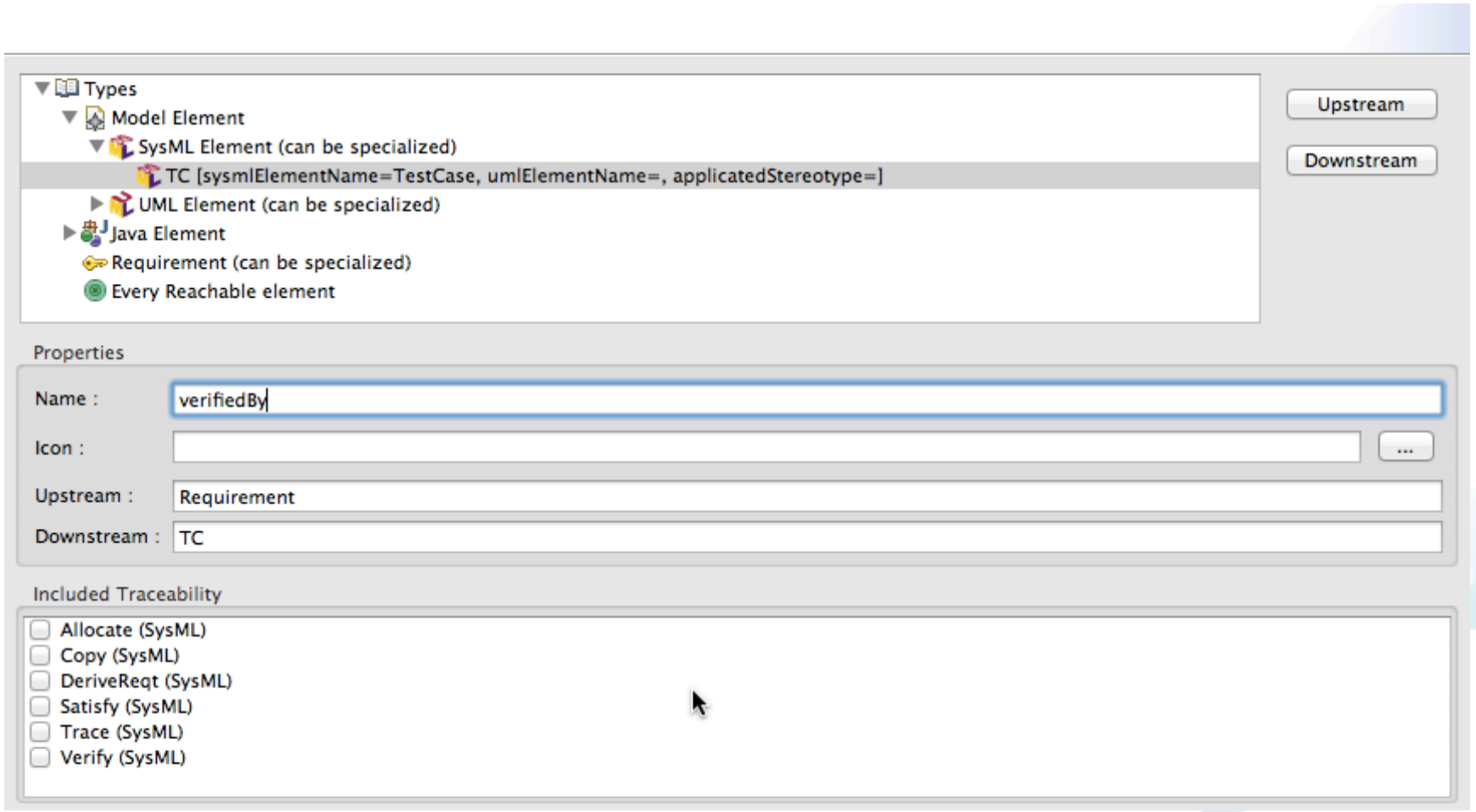
- OK and then Apply

Create “VerifiedBy” traceability type (1)

- “VerifiedBy” first requires defining “TC” reachable data
  - Go to preferences>ReqCycle>Reachable Data
  - Select SysML Element and click “extend” to define “TC”

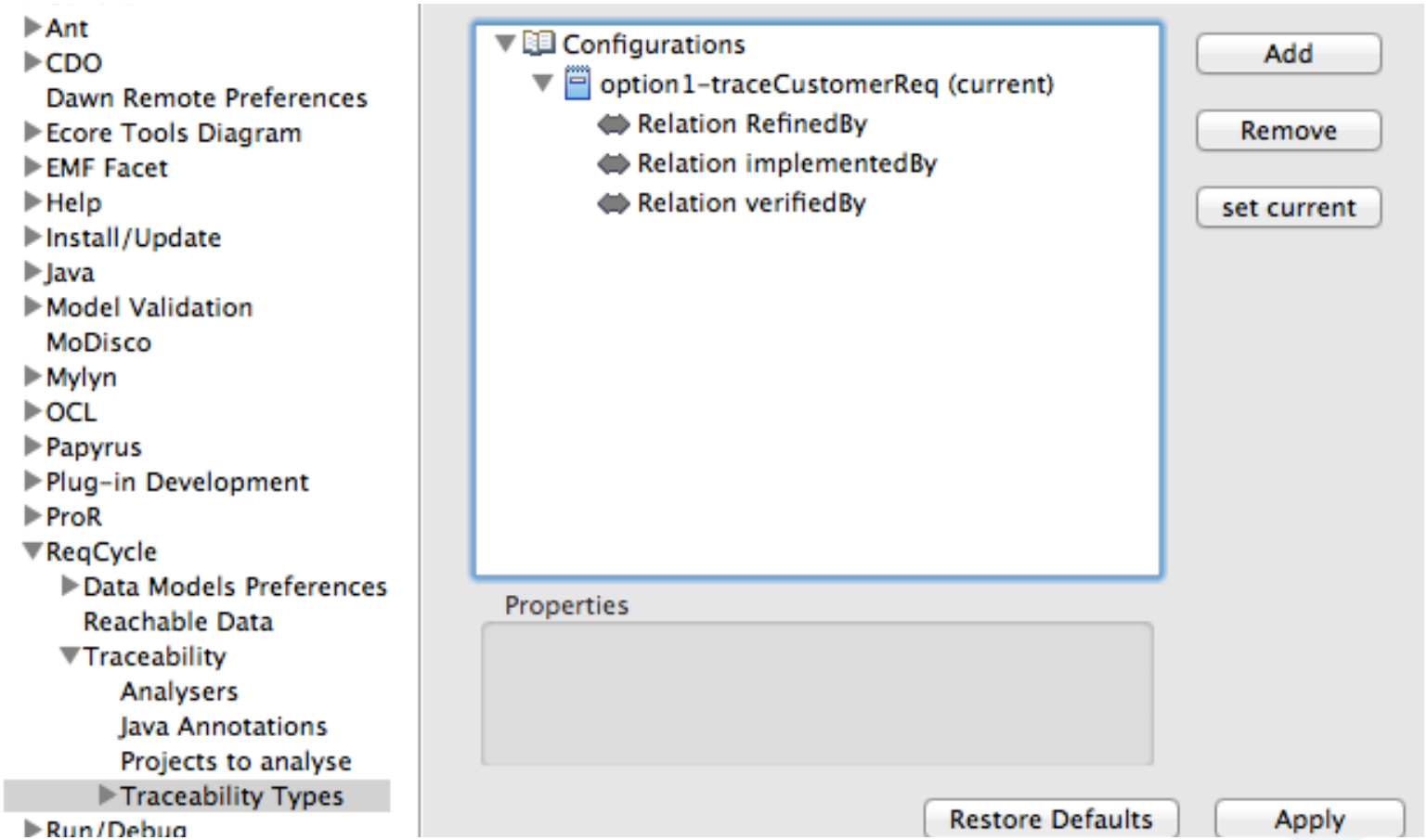


Create “VerifiedBy” traceability type (2)



- OK and then Apply

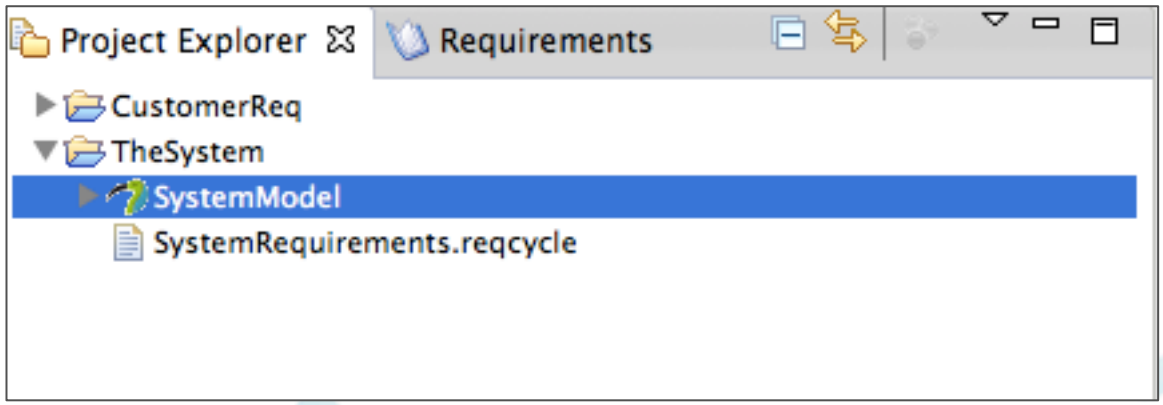
Summary of “traceCustomerReq”



- Now let us create traceability links that conform to this configuration

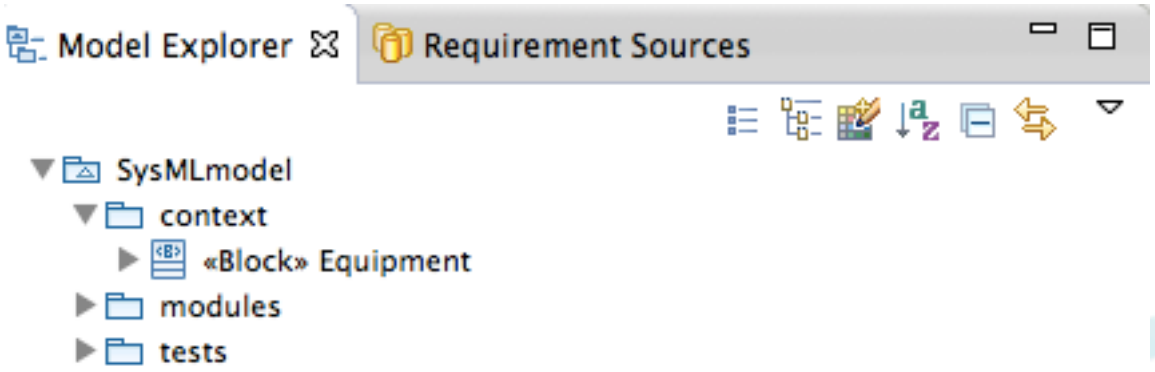
Create traceability links (1)

- Now that we have defined and applied a configuration and set it “current”, we can trace links
- Let us create a “System model” in an Eclipse project
  - Go to project explorer view and create “TheSystem” project
  - Select “TheSystem”, right click>new other… papyrus model



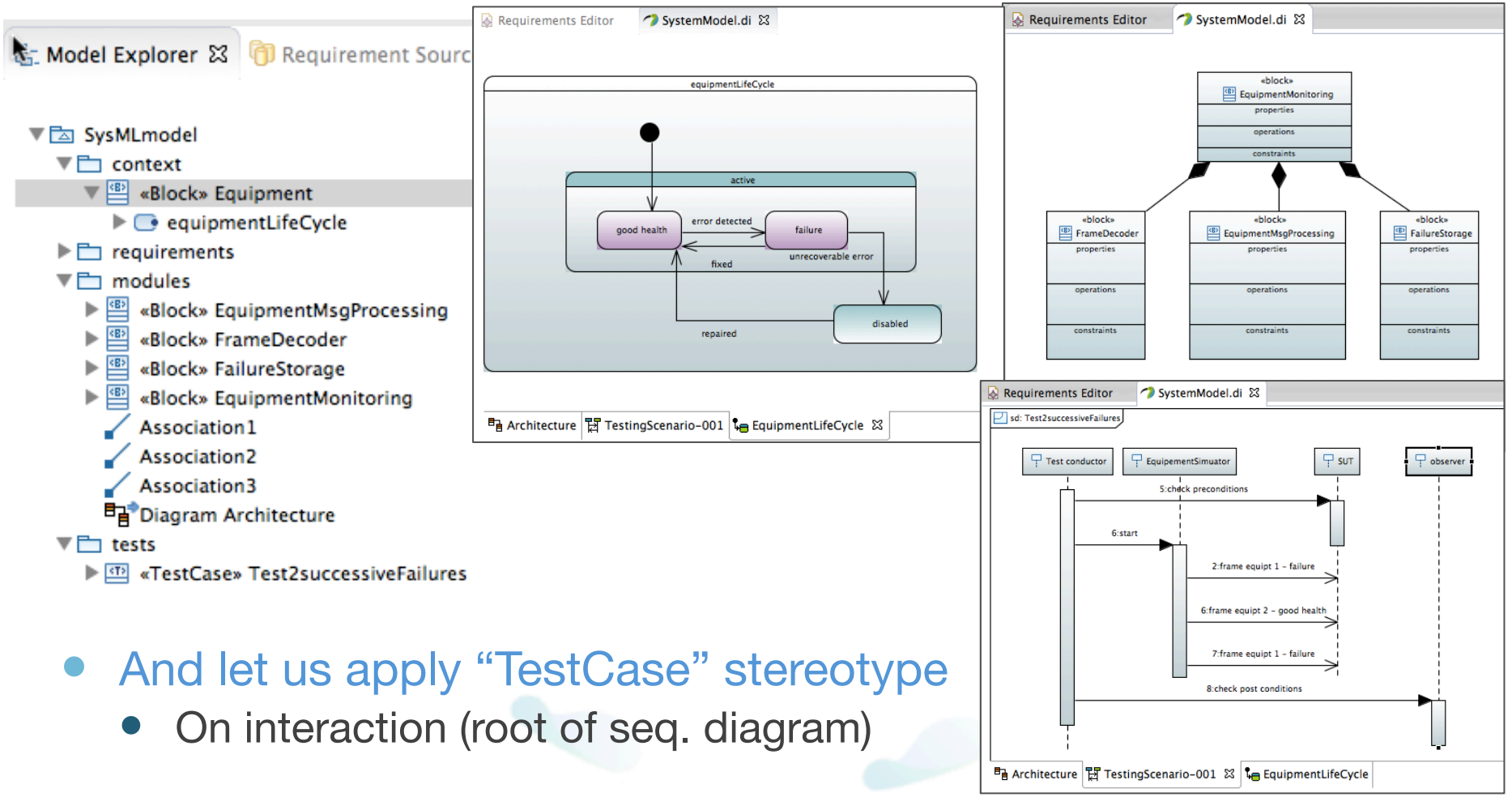
Create traceability links (2)

- Let us create 3 packages
  - “context” to refine equipment lifecycle
  - “modules” to formalize logical architecture
  - “tests” to formalize a few testing scenarios
- In context create “Equipment” block



Create traceability links (3)

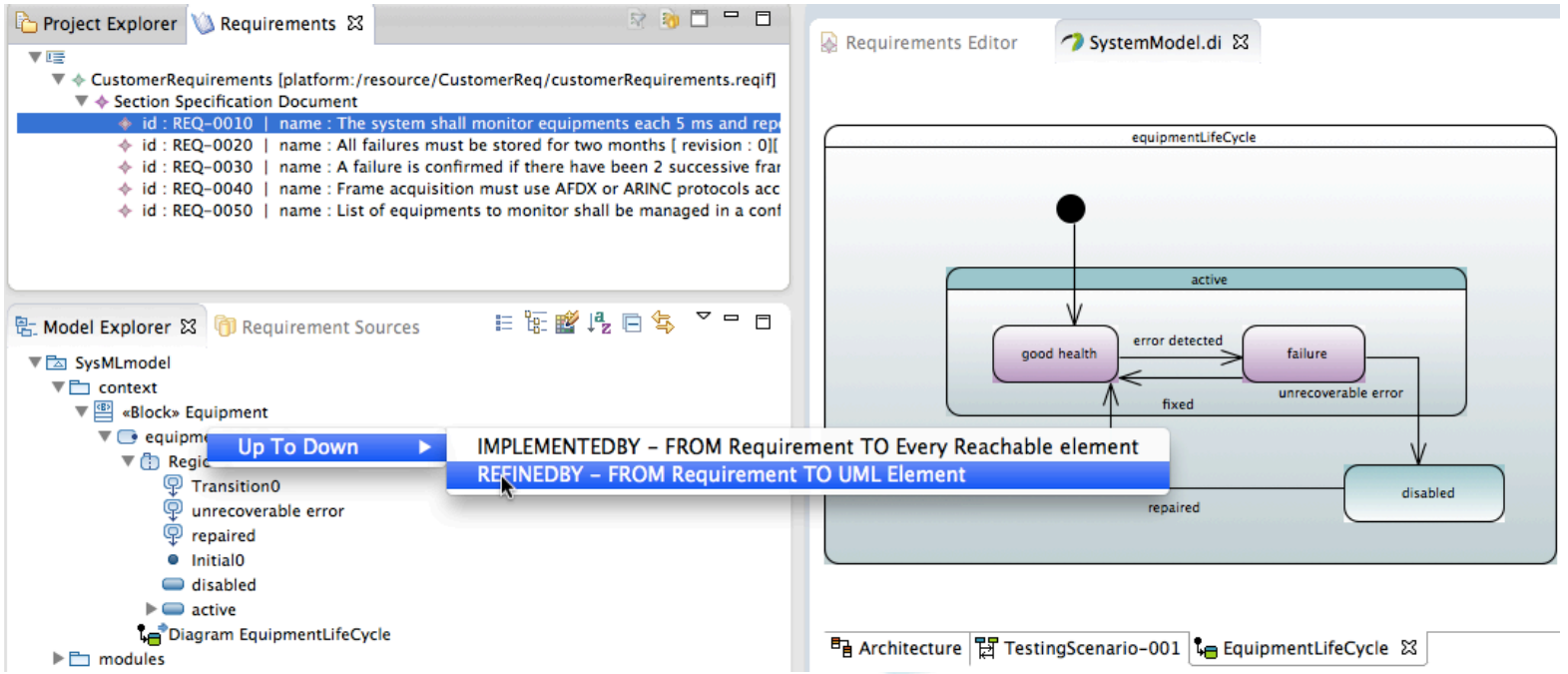
- Let us create following diagrams to get some contents



- And let us apply “TestCase” stereotype
  - On interaction (root of seq. diagram)

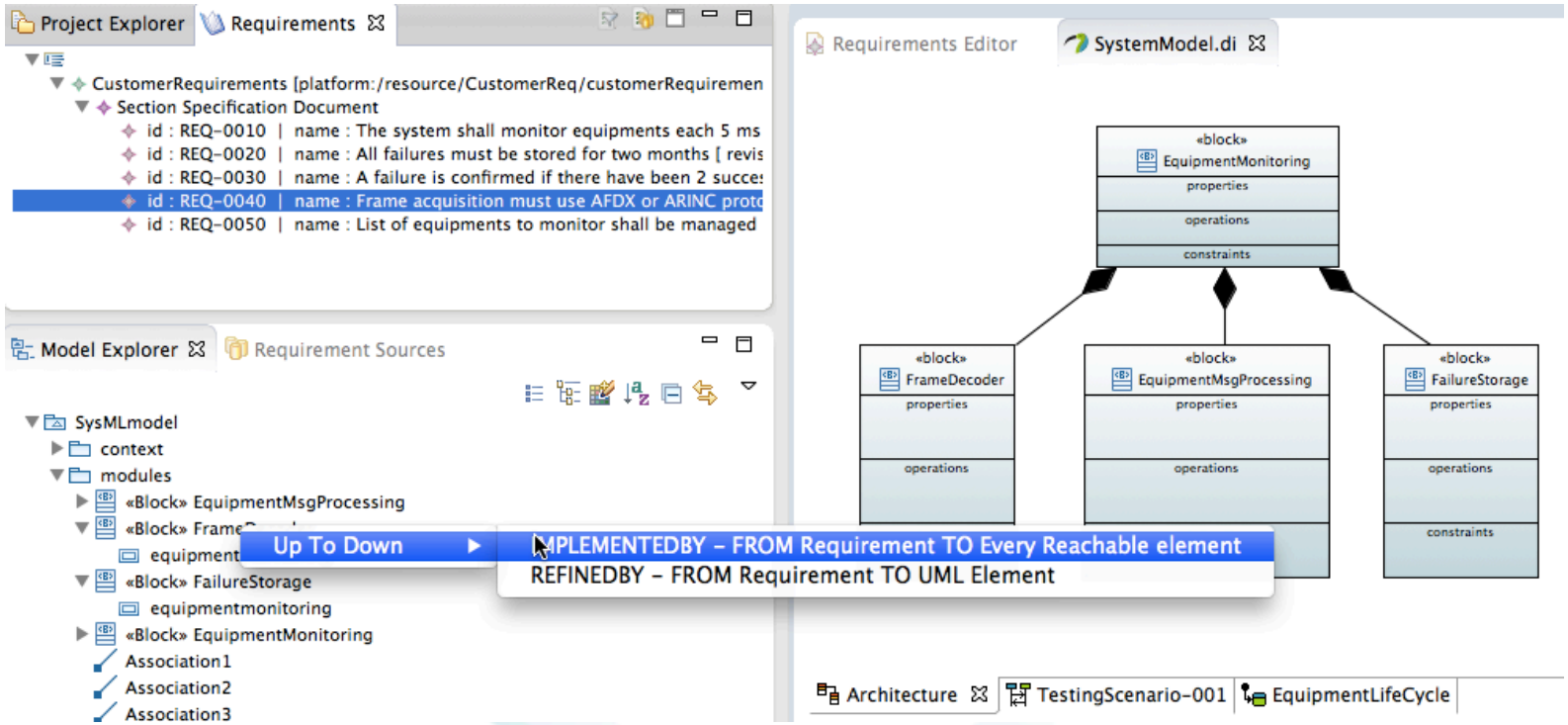
Create traceability links (4)

- D&D one customer requirement on model explorer
  - For instance “REQ-0010” on “equipmentLifeCycle”
  - As target element is reachable and UML, 2 links are available
  - Target deals with system definition, so we select “RefinedBy”



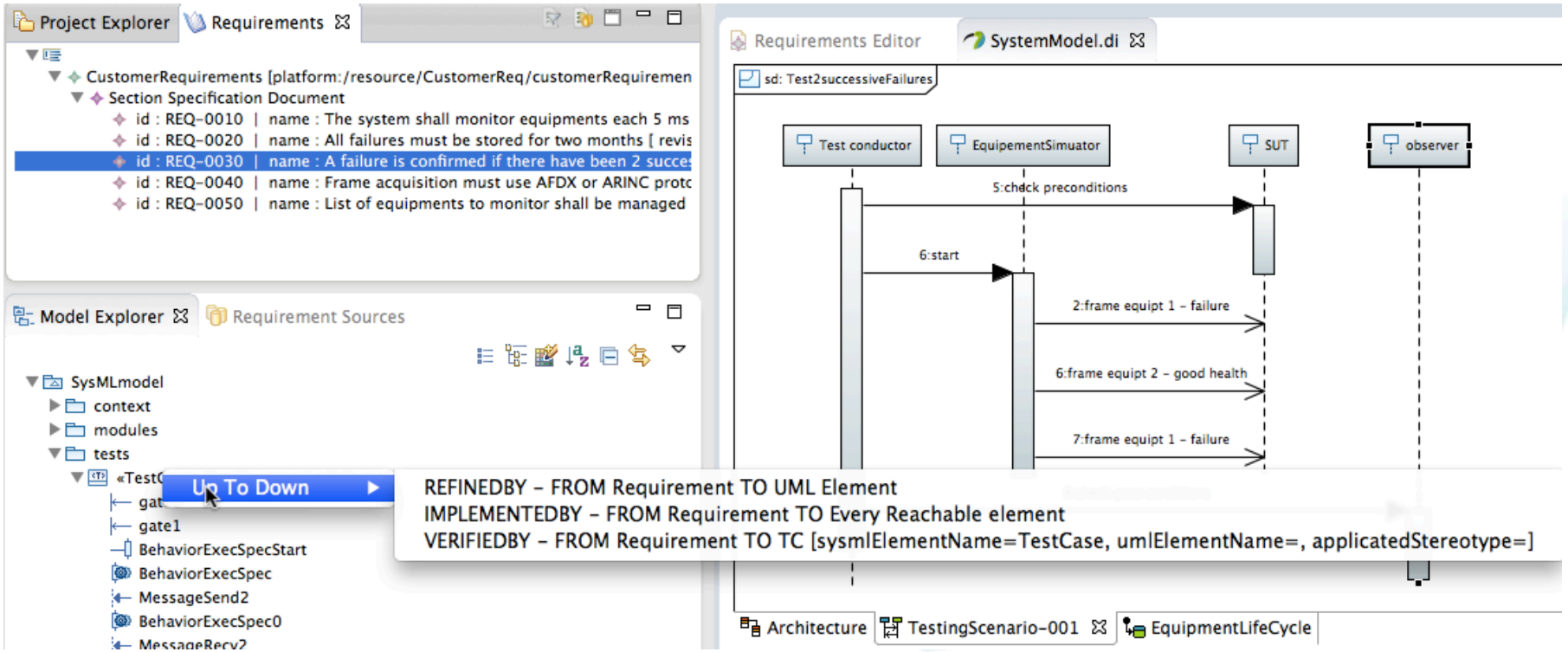
Create traceability links (5)

- Now let us create an “implementedBy” link
  - For instance “REQ-0040” on “FrameDecoder” block



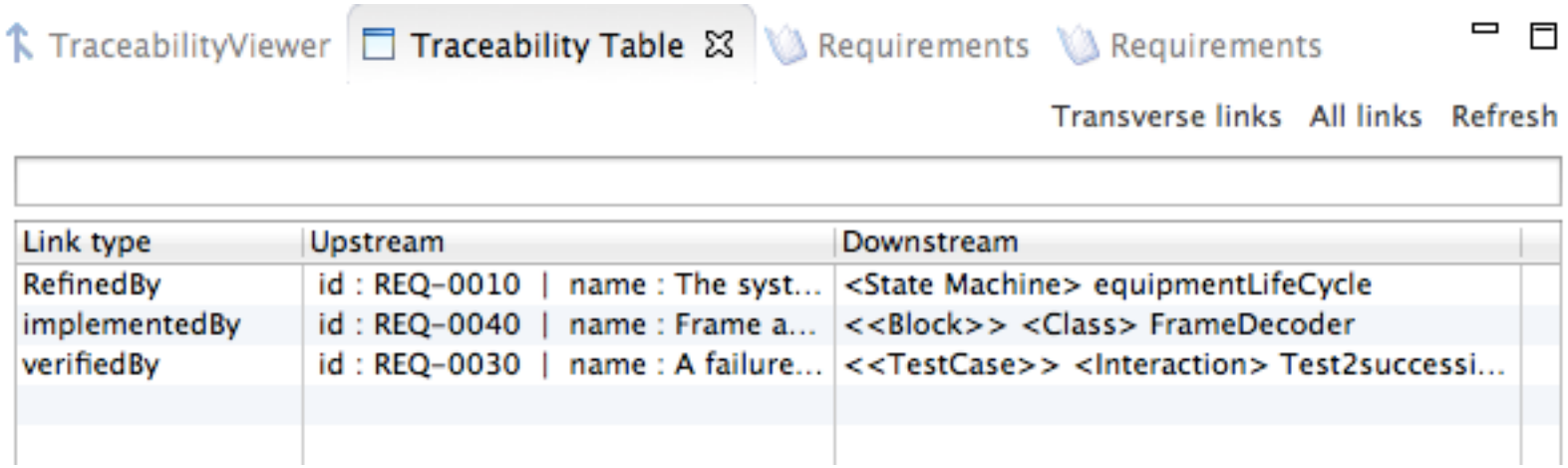
Create traceability links (6)

- Now let us create a “verifiedBy” link
  - For instance “REQ-0030” on “Test2successiveFailures” TC
  - You can notice a new “VerifiedBy” link available…



See and manage traceability links

- So far we have traced 3 customer requirements

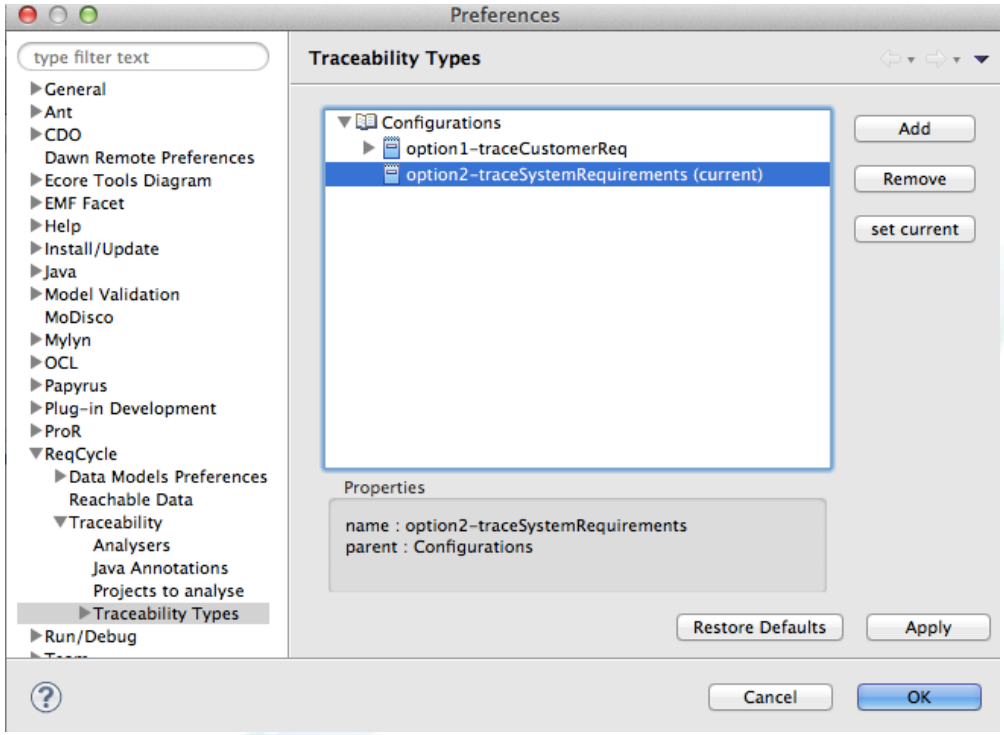


- You can jump to step 4 to see how to visualize traceability links created or captured, in a table or a treeview…

… or you can continue step 3 to see how to refine customer requirements into textual system requirements (option 2)

Configuration to trace System req.

- In preferences>ReqCycle>Traceability>Traceability types
  - Select “Configurations” root node and click “add”
  - Give name (for instance “option2-traceSystemReq”) - OK
  - Click “set current”
    - To activate this conf.
  - Click “apply”
    - To register data



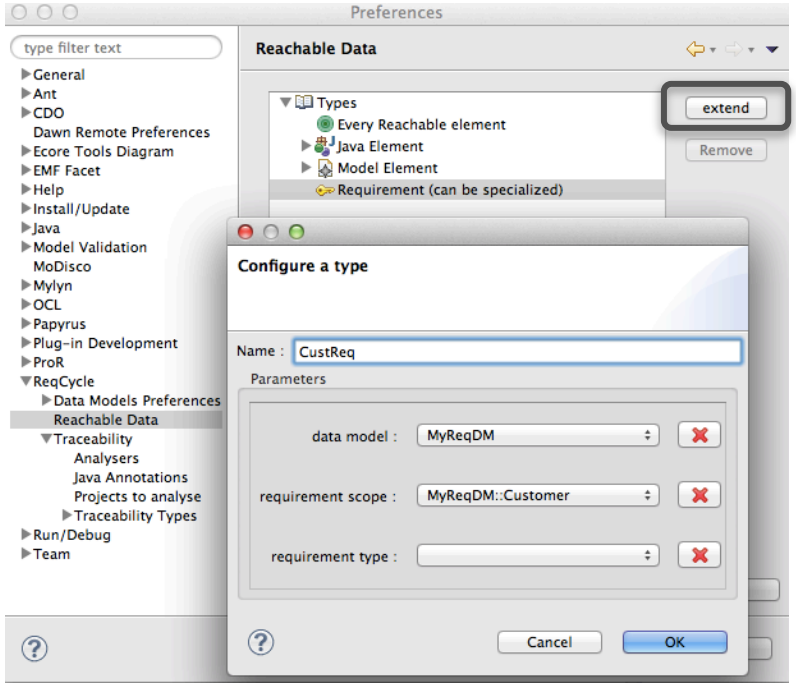
Traceability links for System Req.

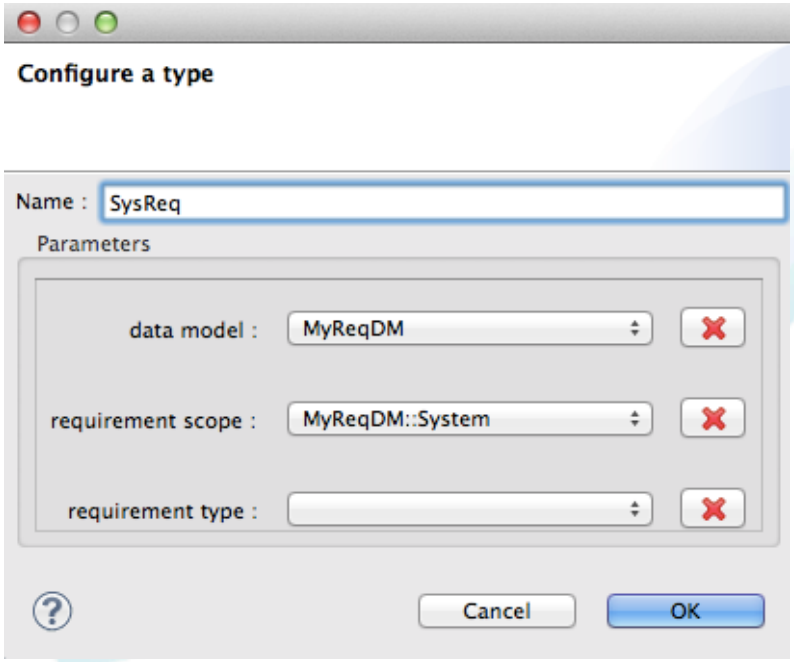
- To trace our system requirements we want to define several kinds of links. For instance:
  - “Refines” that will concern Customer requirements
  - “ImplementedBy” that can concern any reachable data
  - “VerifiedBy” that only concerns SysML test cases

- Next slides show how to do “Refines” link type
  - For other relationships, do same as in previous slides

Refine a customer req into system req

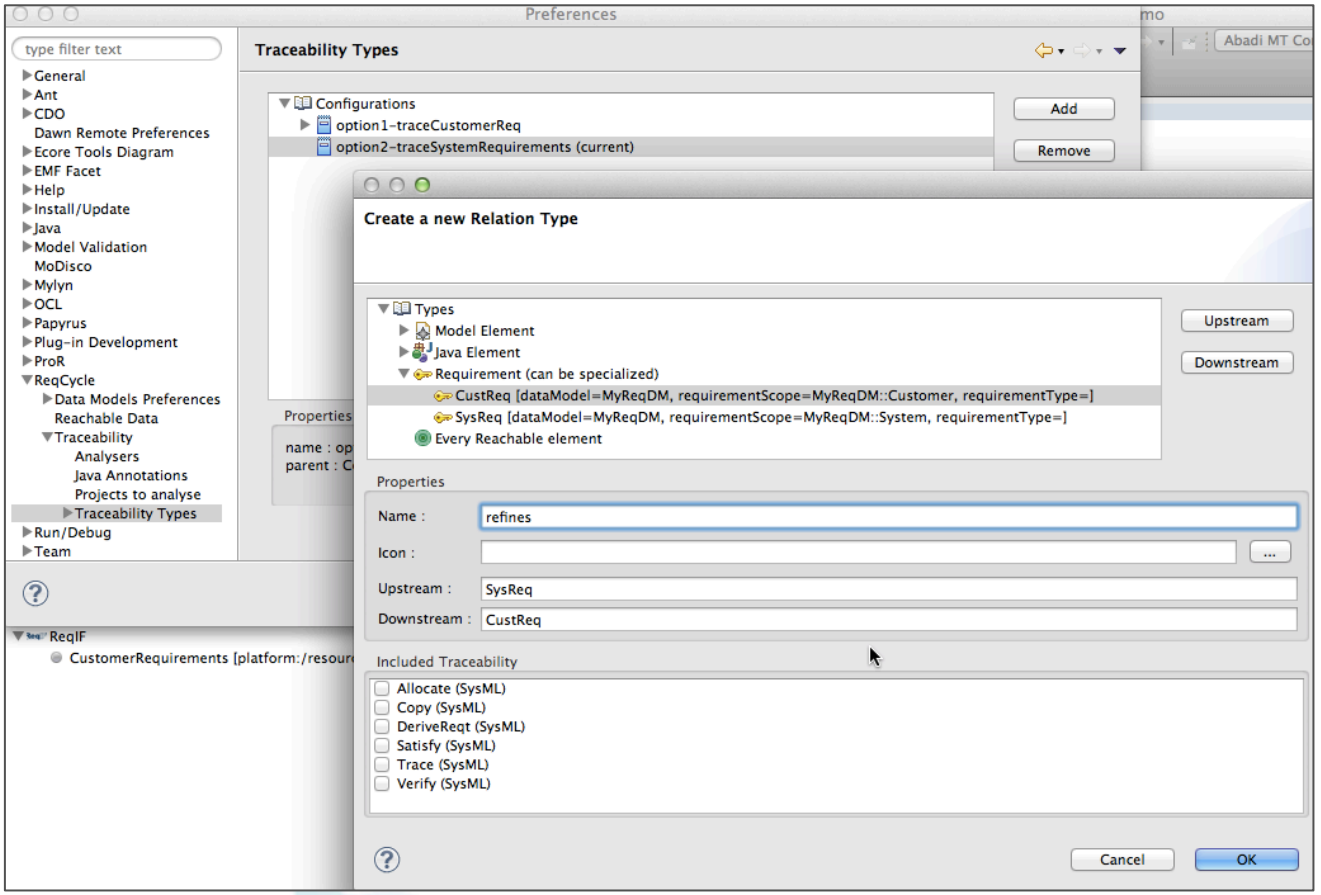
- At first sight we could simply define a Req2Req relation
  - Upstream = Requirement, downstream = Requirement
  - But it would allow creating relations between customer req…
- Let us rather define CustReq. And SysReq reachable types





Define Refine link type

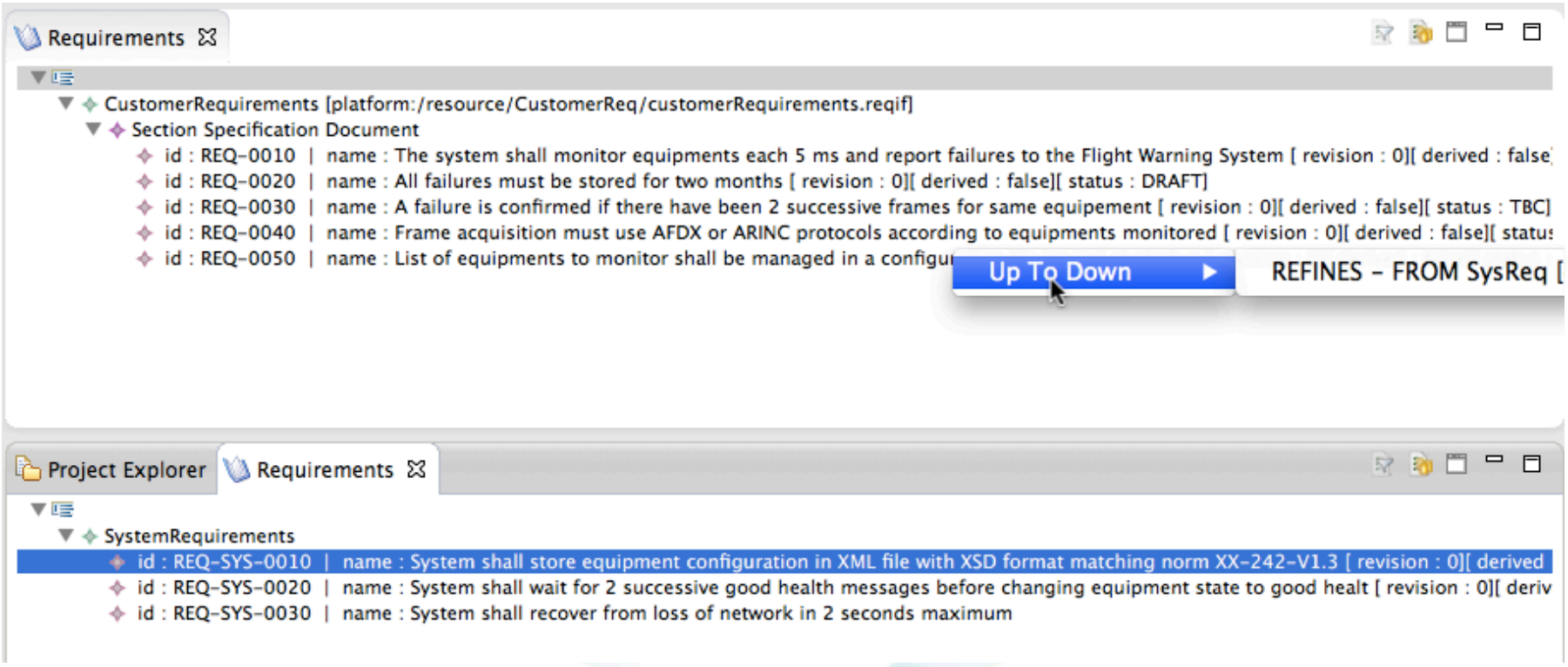
- Now we can define precisely our “refines” link type



- OK and then do not forget to apply to register the link

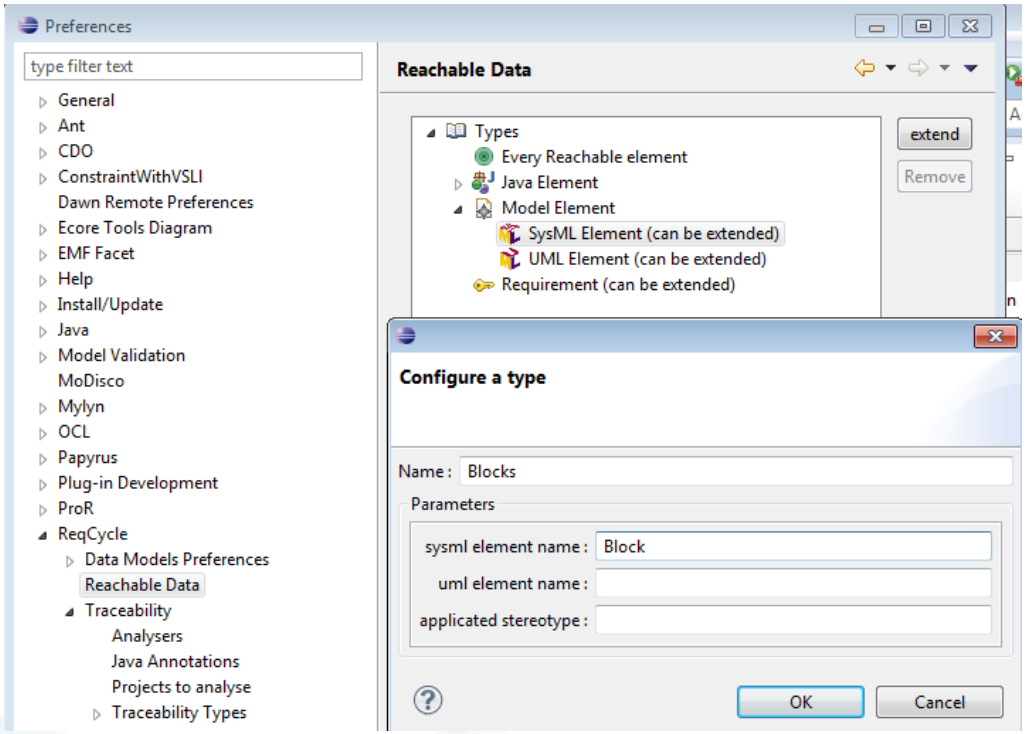
Refine a customer req with a system req

- You can now drag and drop a system requirement to a customer requirement
  - Note: no possible link between customer reqs



Restrict link target

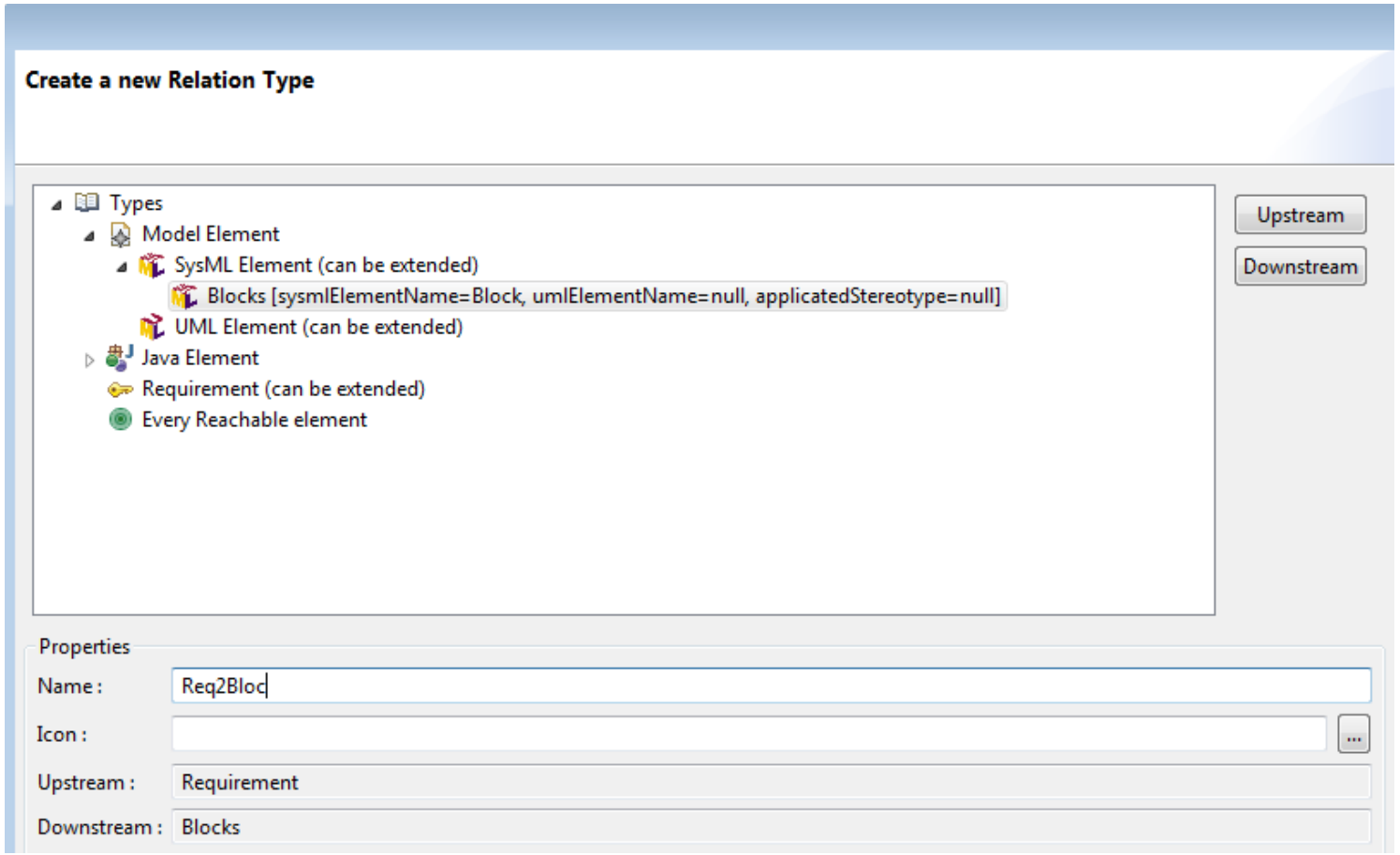
- Can restrict target reachable data
  - Go to ReqCycle preferences>Reachable data
  - Extend UML or SysML model element
- Example
  - Blocs



- Note
  - Use “apply” on new type and then “OK”

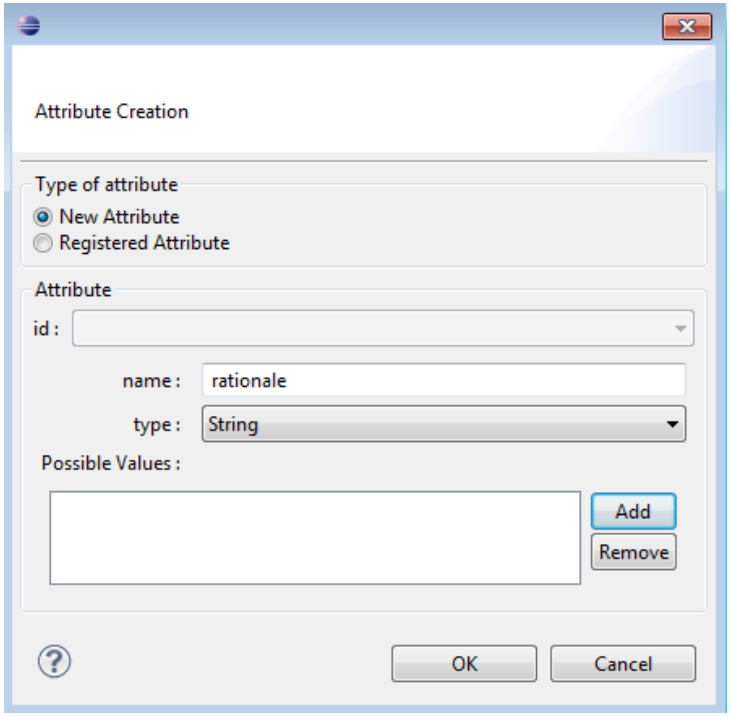
Restrict links to blocks - definition

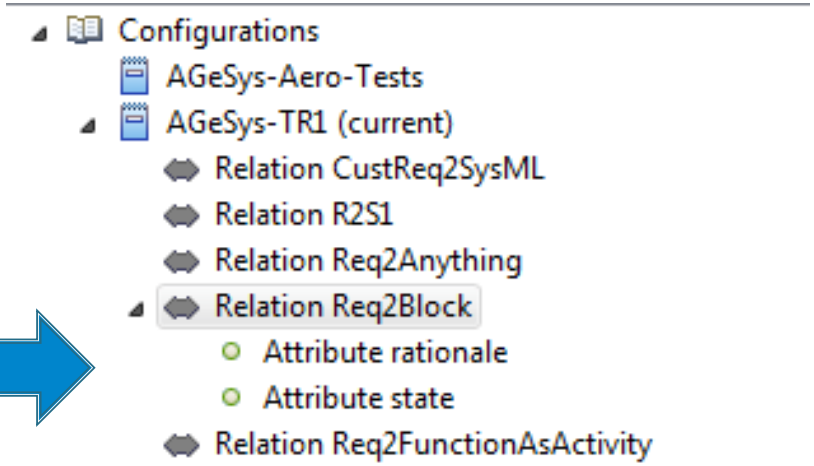
- Can then create a link type from any requirement to any SysML block



In the future…

- Possibility to add link attributes





- Note : development partially done but not enough tested

Summary of 3rd step

- We have demonstrated we could create some links between requirements and other reachable elements
  - For instance between ReqIF requirements and UML elements
  - Even reachable data belong to different languages/formats
- We have shown we could drive the creation of links through the configuration of link types and requirement data model
  - Limit links available according to the traceability link types
  - Limit link source/target to some requirement scopes or types
- Now we are going to see in step 4 how to capture and display traceability links whatever the way its was created

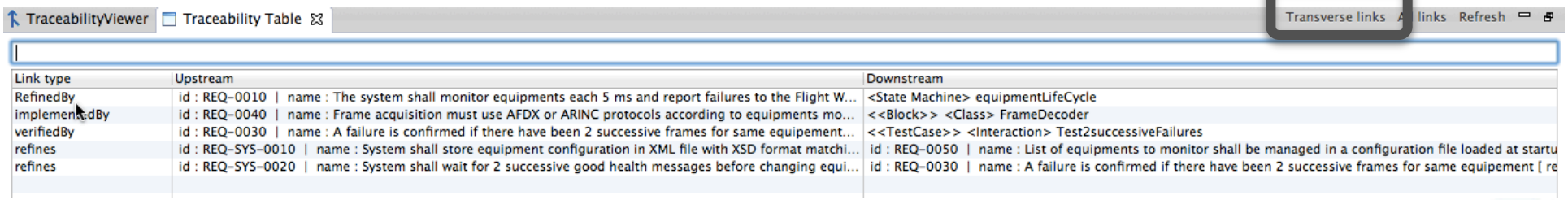
SAMARES
ENGINEERING
Accelerate Systems Design

4th step – Capture / display traceability

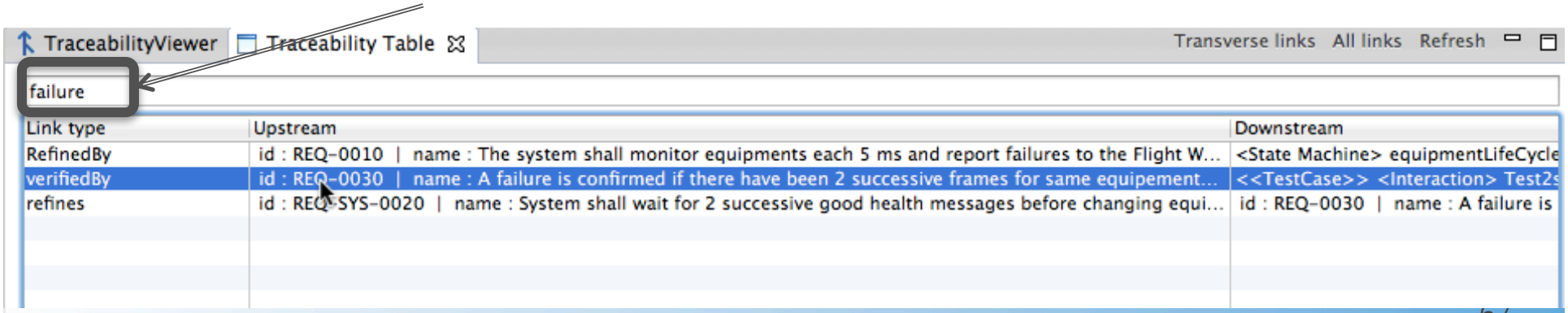
Concepts and screenshots

Traceability table

- Apply “Transverse link” button
  - Requires selecting an Eclipse project

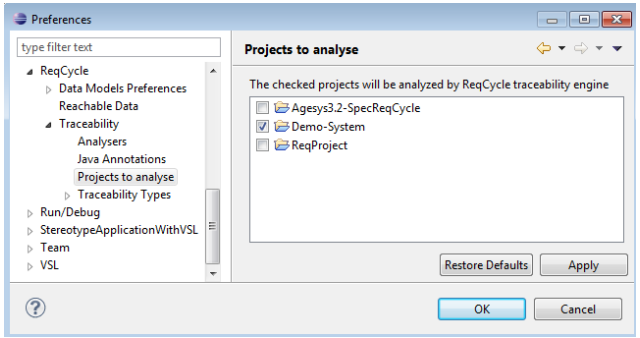


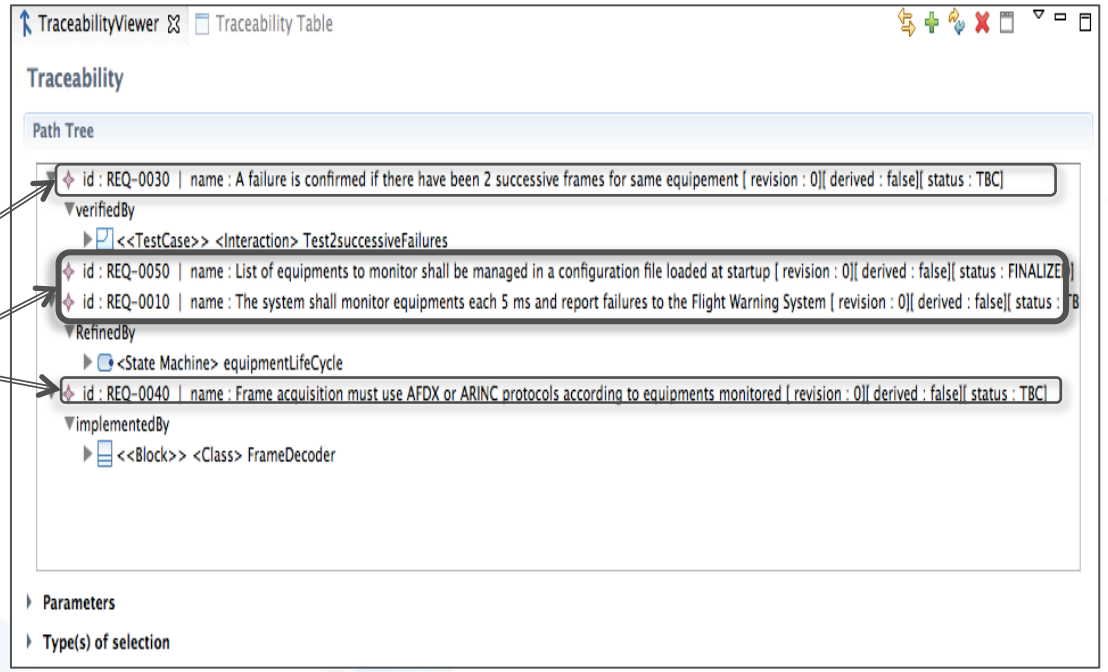
  - Can delete a link (contextual menu)
  - Can search on any part of text



Traceability viewer

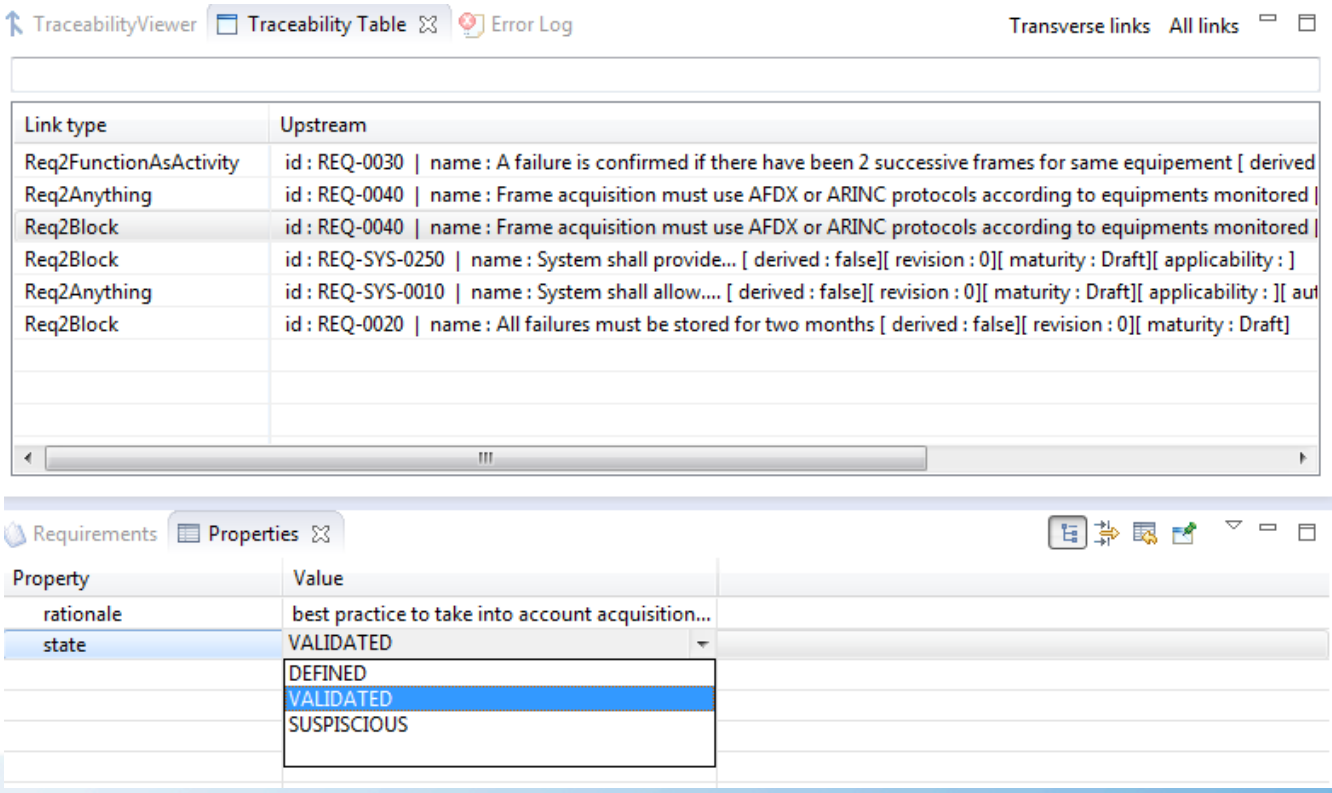
- In ReqCycle preferences, select projects to analyse
- Then open traceability Viewer view
- DnD requirements in Path Tree area
  - Traceability is captured in real time and displayed





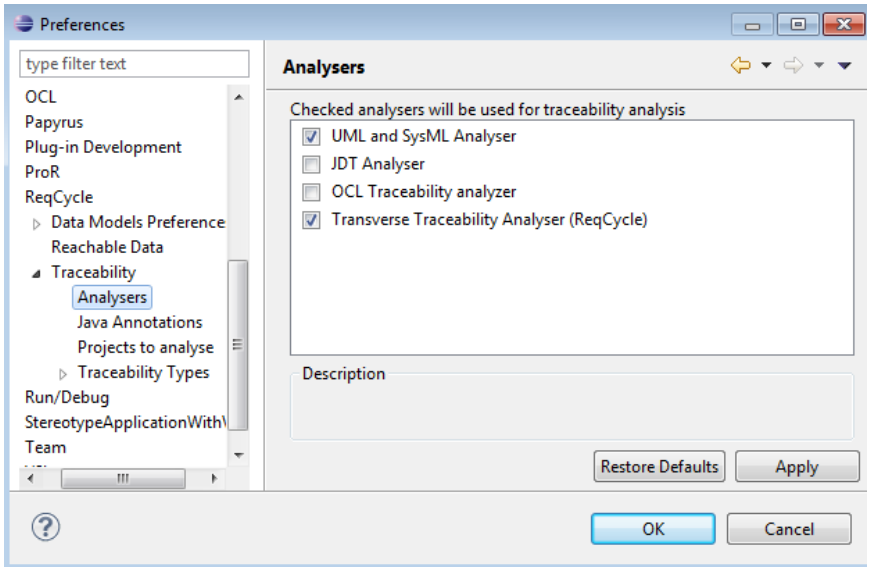
In the future…

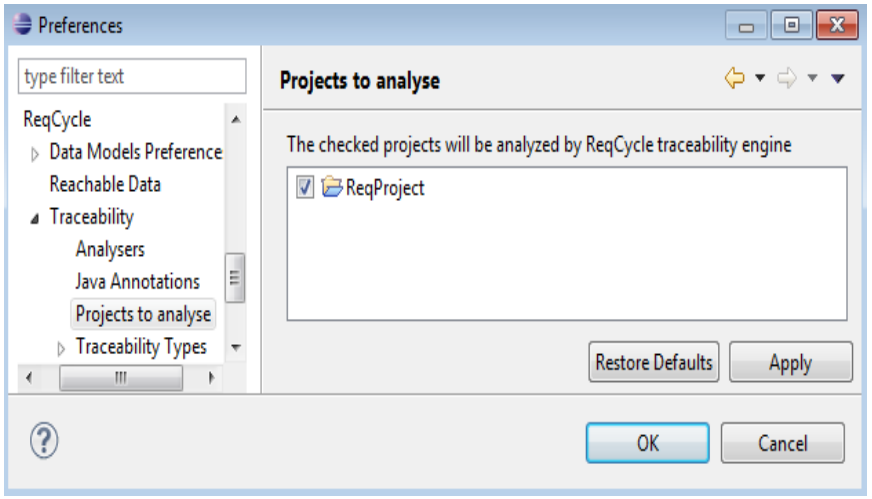
- Ability to edit link attributes
  - A first step toward link validation and management of link changes



Capture existing links

- ReqCycle provides analysis engines
  - Can select some engines

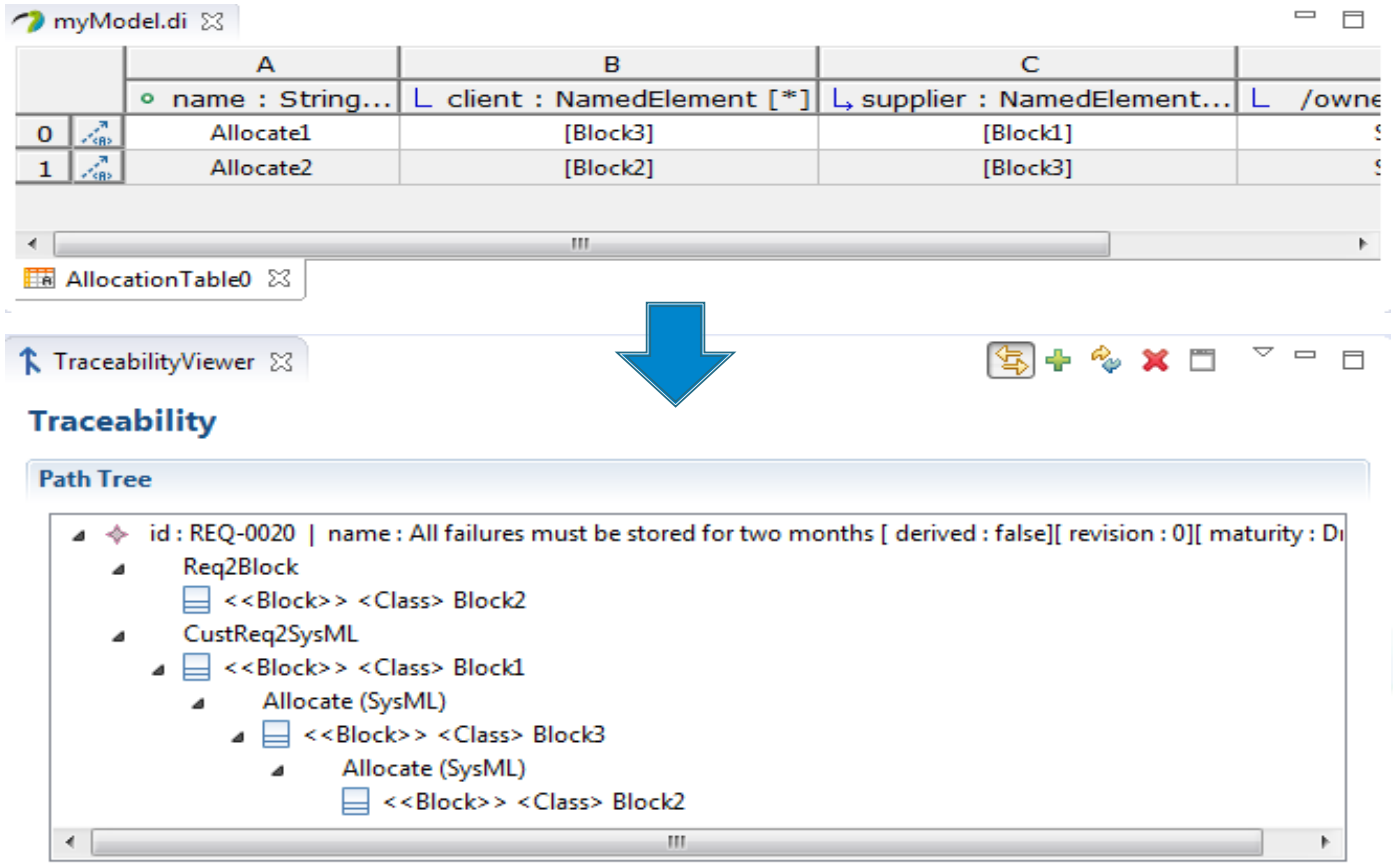




  - And can choose projects to analyse

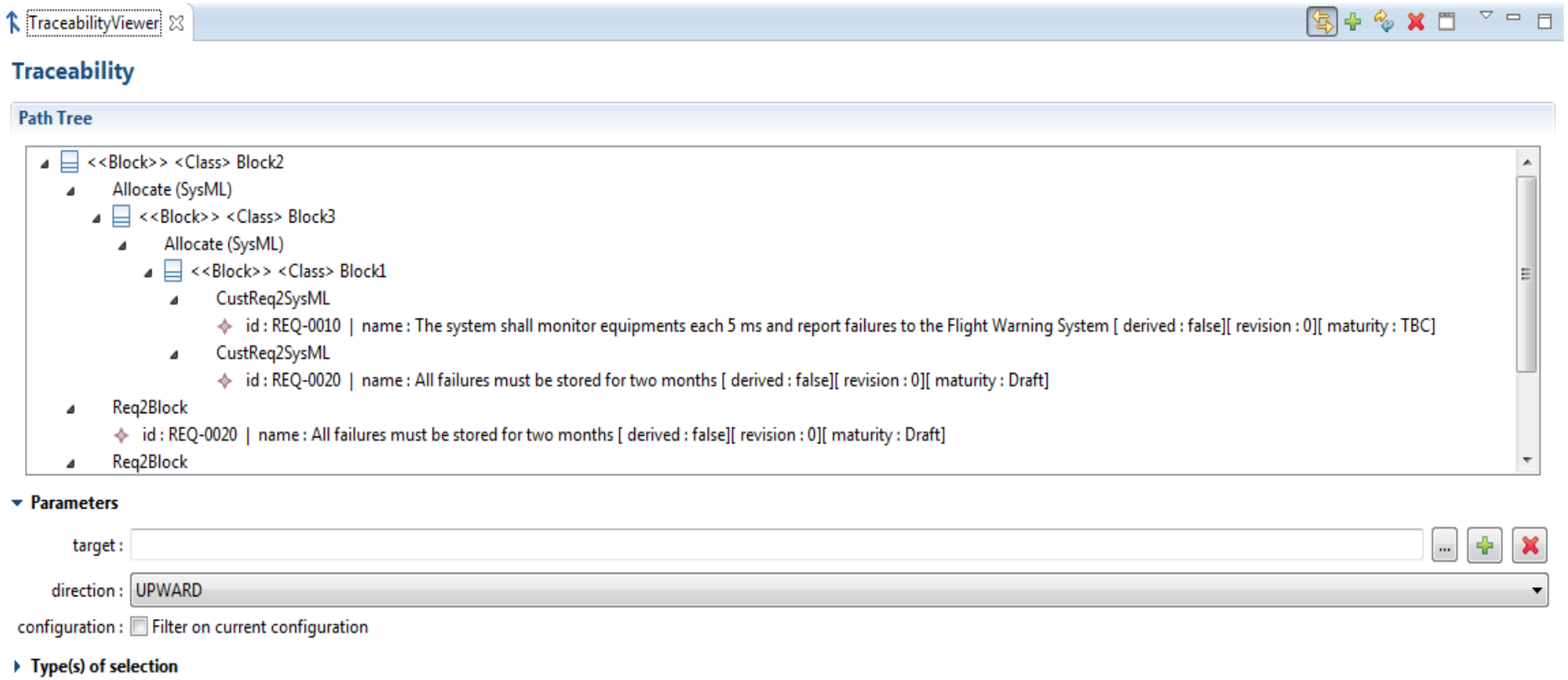
Extended traceability

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



Upward traceability

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



SAMARES
ENGINEERING
Accelerate Systems Design

5th step – export requirement and traceability links

Concepts and screenshots

Export requirements

- Not yet supported
- Planned for mid 2014

Export of traceability links

- Not yet supported
- Planned for mid 2014

Teamwork support

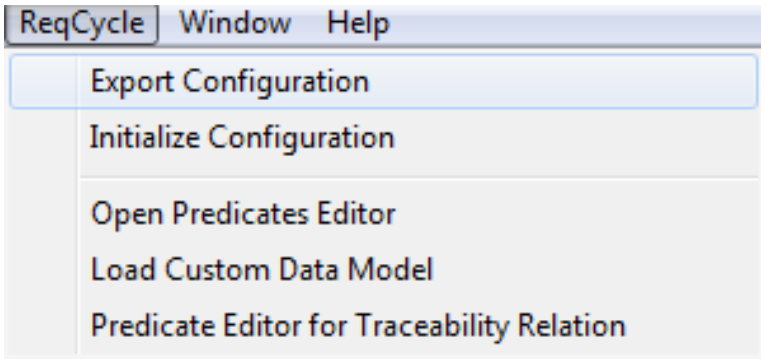
Concepts and screenshots

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

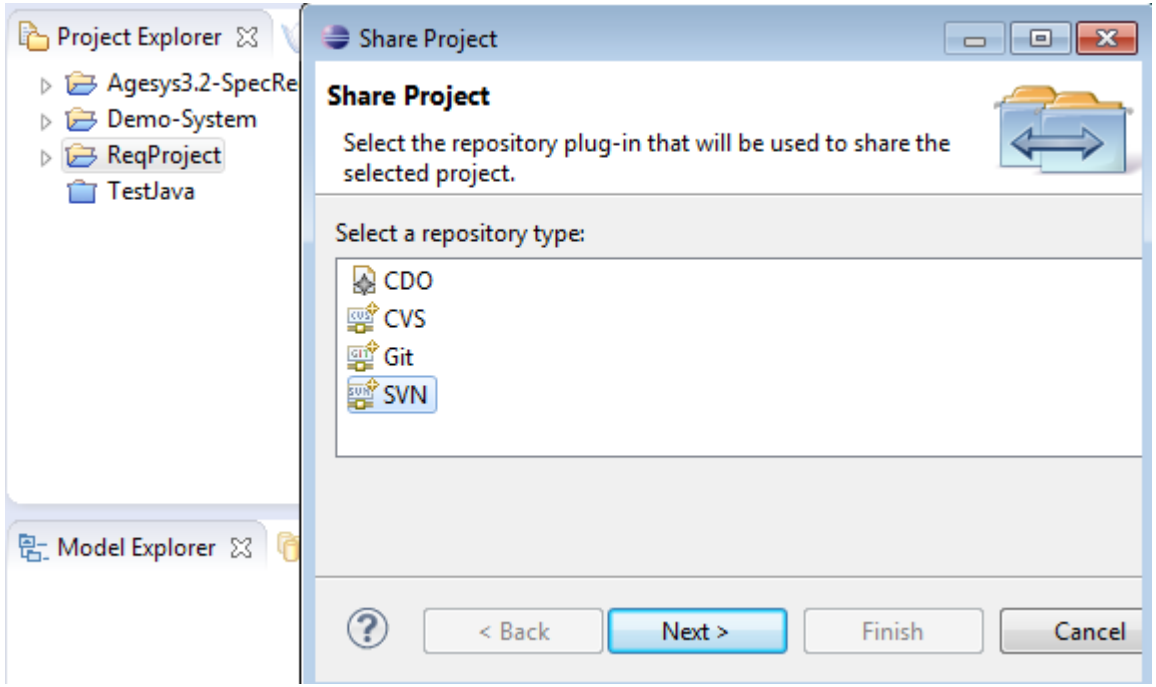
- Can export ReqCycle full configuration as a file
  - Data saved = Data models + reachable types + traceability configurations + requirement sources
  - Menu “ReqCycle”>export configuration
  - Save it as a file – can then be distributed to others



- Can then initialize configuration in an other workspace

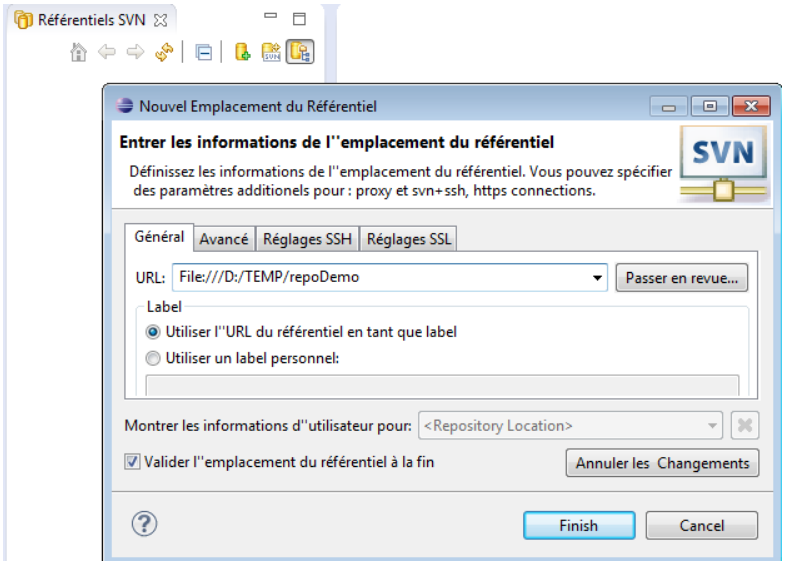
Put requirements in SVN

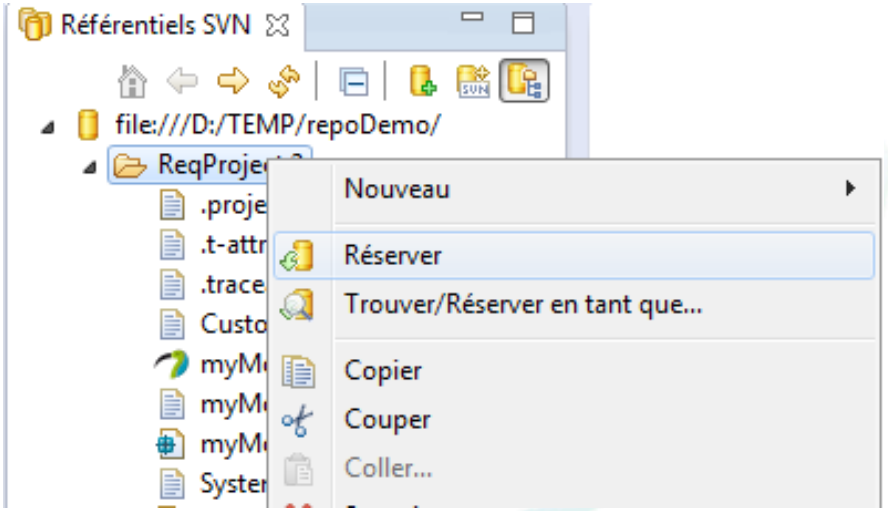
- Just share your Eclipse project (containing requirements) in SVN



Get requirements from SVN

- In other workspace, retrieve requirement projet put previously in SVN
  - SVN perspective: repository access
  - Then checkout





- Restart : req and links are there

More information…

- You can go to ReqCycle github before the project joins Eclipse PolarSys
  - https://github.com/raphaelfaudou/ReqCycle
- Contacts for roadmap and support
  - Raphael.faudou@samares-engineering.com
- Contacts to contribute on development
  - Raphael.faudou@samares-engineering.com
  - Anass.radouani@atos.net
  - Mathieu.velten@atos.net
  - Tristan.faure@atos.net