Lecture 1

Requirements Management and Enterprise Architect

Enterprise Architect: is a powerful CASE tool for Specifying ,Documenting and Building software project.

Enterprise Architect (EA) is one of the few UML tools that integrate requirements management with other software development disciplines, by creating requirements directly in the model.

Enterprise Architect Features

- The ability to create and view requirements directly in the model.
- The ability to detail use cases directly in the model.
- The ability to enter attributes for each requirement such as difficulty, status, and type; you can also define your own attributes.
- The ability to trace requirements to business rules, test cases and analysis artifacts.
- A Relationship Matrix for traceability and viewing the impact of changes to requirements.
- The ability to create documents.

Requirements Management with UML

Requirements are essentially what the system, application or business process is required to do.

The management of requirements is one of the more difficult and problematic disciplines in the software development industry. The most significant reasons for this are the following:

 Diverse group input into the requirements

 Volatility of requirements

 Imprecision and ambiguities of natural languages

The Enterprise Architect can be used to reduce (and in many circumstances remove) these problems.

Glossary of Terms

The following is a list of terms, and how they relate to requirements management and Enterprise Architect:

 Element – A generic term referring to a singular object in a model. Some of the common elements you will come across include requirements, actors and systems.

 External requirement – A requirement that is modeled as an element.

 Internal requirement – A requirement that is modeled as the "responsibility" of an existing element.

 Model – A representation of a particular system, such as a business process or a database.

 Diagram – A common way of representing the way in which models and elements interact.

 Attributes – Data fields containing information within requirement elements.

Requirements Modeling

for model requirement, we must use the Requirements palette of the Toolbox that is located at the far left of the Enterprise Architect interface, in the Toolbox as shown in the figure 1.

The contents of the toolbox

1- **Package**: is a namespace as well as an element that can be contained in other Package's namespaces. A Package can own or merge with other Packages, and its elements can be imported into a Package's namespace, (Package imports or merges).

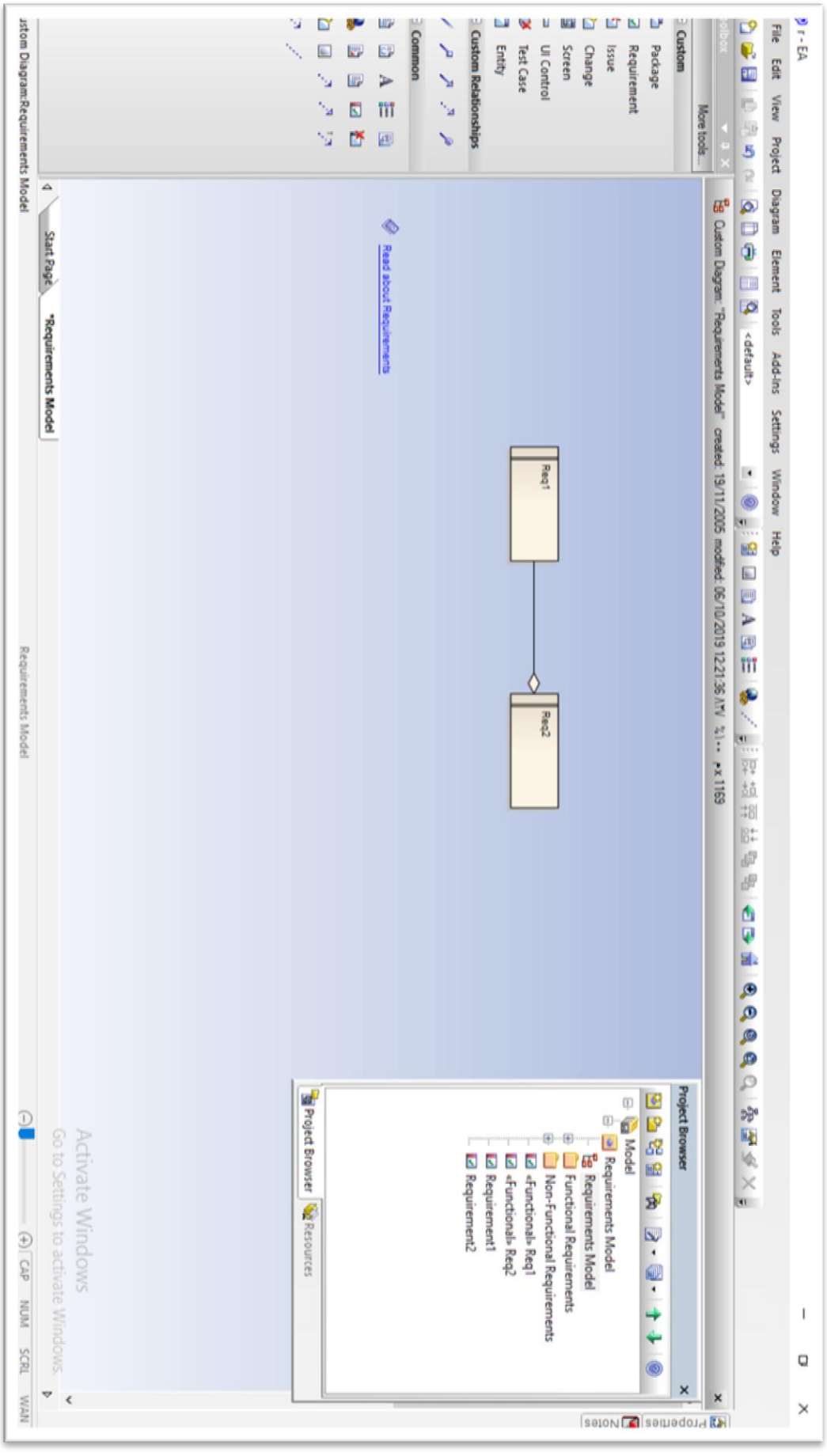

File Edit View Project Diagram Element Tools Add-Ins Settings Window Help
Custom
Package
Requirement
Issue
Change
Screen
UI Control
Test Case
Entity
Custom Relationships
Common
Req1
Req2
Read about Requirements
Start Page
*Requirements Model
Project Browser
Model
Requirements Model
Functional Requirements
Non-Functional Requirements
«Functional» Req1
«Functional» Req2
Requirement1
Requirement2
Activate Windows
Go to Settings to activate Windows.

Figure (1)

2- **Requirement**: is a custom element used to capture requirement. A requirement expresses required system behavior. You can connect requirement to other element using realize connector.

3- **Issue**: is a structured comment that contains information about defect and issues relating to the system.

4- **Change:** is a structured comment that contains information about changes requested to the system.

5- **Screen:** is used to prototype User Interface screen flow. you can build up a solid and detailed understanding of user interface behavior without having to use code. It is display a GUI.

6- **UI Control:** represents a user interface control element (such as an edit box). It is used for capturing the components of a screen layout and requirements in a Custom or User Interface diagram.

7- **Test Case:** Within the Test Case element properties you can define test requirements and constraints, and associate the test with test files. The Test Case element enables you to give greater visibility to tests.

8- **Entity:** simple element that represents any general thing.

Lecture 2

Relationship Types (Connector Types)

1- ***An Association***: implies two model elements have a relationship, this connector can include named roles at each end, multiplicity, direction and constraints. ***Association***: is the general relationship type between elements.

2- ***An Aggregation:*** Requirements linked by aggregation relationships form a composition hierarchy. is a type of association that shows that an element contains or is composed of other elements. It is used to show how more complex elements (aggregates) are built from a collection of simpler elements (component parts; for example, a car from wheels, tires, motor and so on).

3- ***A Generalization***: is used to indicate inheritance. Drawn from the specific classifier to a general classifier, the generalize implication is that the source inherits the target's characteristics.

4- ***A Realizes:*** a source object implements or Realizes its destination object. Realize connectors are used to express traceability and completeness in the model.

5- *A* ***Nesting***: is an alternative graphical notation for expressing containment or nesting of elements within other elements. It is most appropriately used for displaying Package nesting in a Package diagram.

Figure (2) shows the requirements model for Inventory Management.

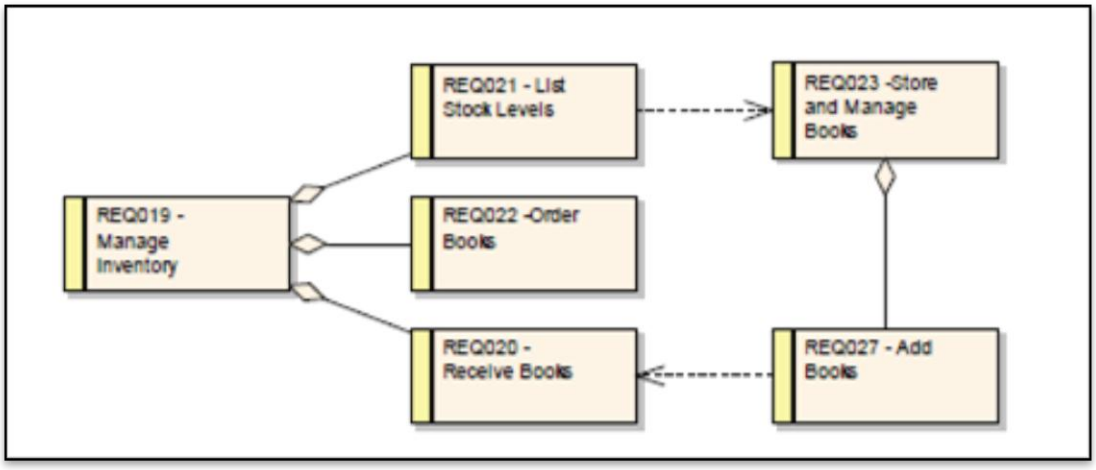


Figure (2)

Figure (3) shows the requirements model for User Account Management

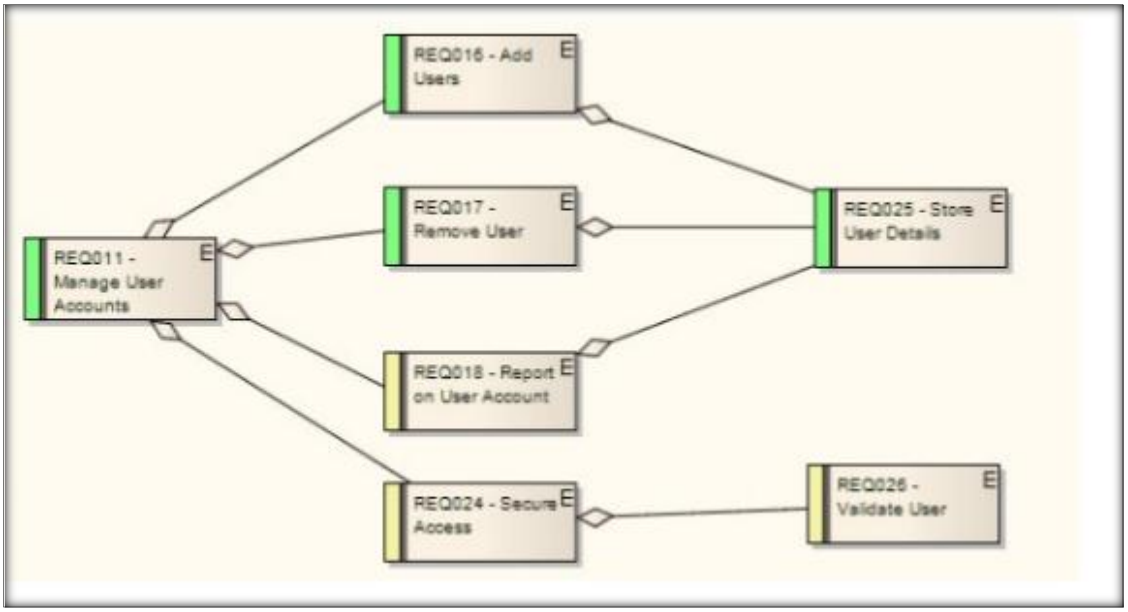


Figure (3)

Lecture 3

Requirement Attributes

a requirement element that is part of a model has properties or Attributes. In Enterprise Architect these are assigned in the properties sheet. (*Double-click on the Requirement*). Enterprise Architect has built-in requirements Attributes such as status, difficulty, priority, and type. Figure 4 shows an example of the properties for a requirement.

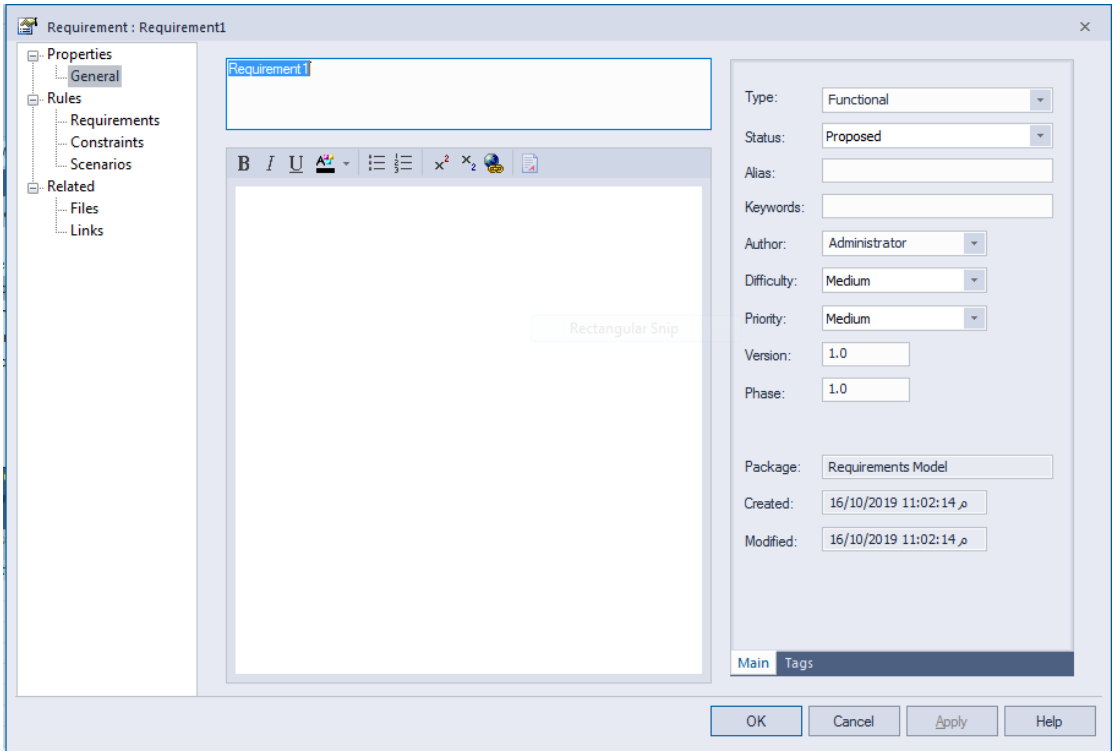


Figure4 : External requirements properties

Adding Custom Attributes to requirements

You can enter any number of additional Attributes such as stability, cost, and lateness penalty through the use of Tagged Values.

Tagged Values can be defined for a specific element, or predefined to be added to all new requirement elements.

Tagged Value data for an element is available on a separate window, which is accessed from the main menu Design| Tagged Values.

See Figure 5 for a diagram showing a one-off addition of a Tagged Value.

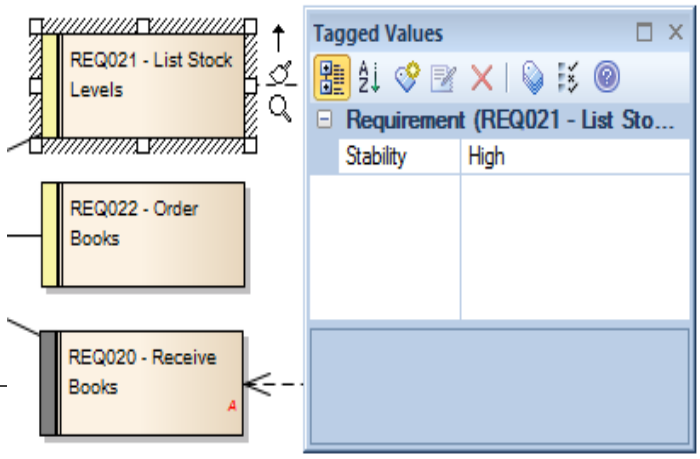


Figure 5: Requirements Tagged Value dialog allowing the assignment of Attributes

Predefining Tagged Value types for requirements

The predefined Tagged Value types can include a number of standard formats, such as date/time, calendar view and drop-down lists.

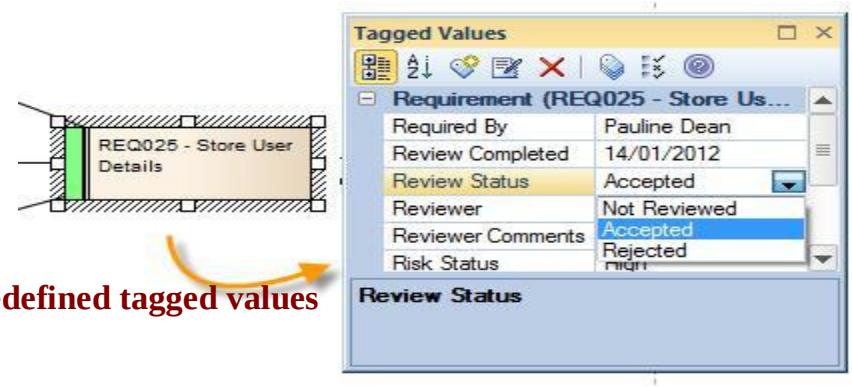


Figure 6: Using predefined tagged values

These extended Attributes can also be viewed directly on the element in the diagram. To set this mode for a specific diagram, right-click on the diagram, and in the context menu, select: Properties | Elements | Show Compartments | [✓] Tags. Below is the same element in Figure 7 viewed in this mode.

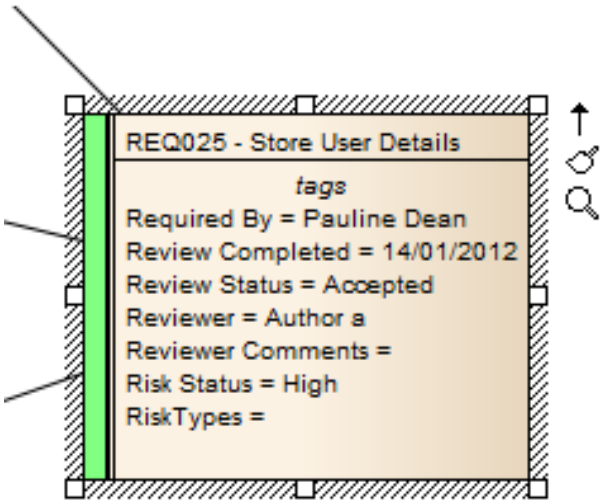


Figure 7: Tagged Values visible on elements

Lecture 4

Element Numbering

Enterprise Architect also supports creating a hierarchy of elements under a package. Element Numbering combined with this hierarchical structure, allows for elements within a Package to be numbered in a 1.1.1 format. This feature can be set on any package and applies to the Elements contained in the root of that Package (it does not apply to child packages).

Below is an example an Element hierarchy with the Element Numbering set on:

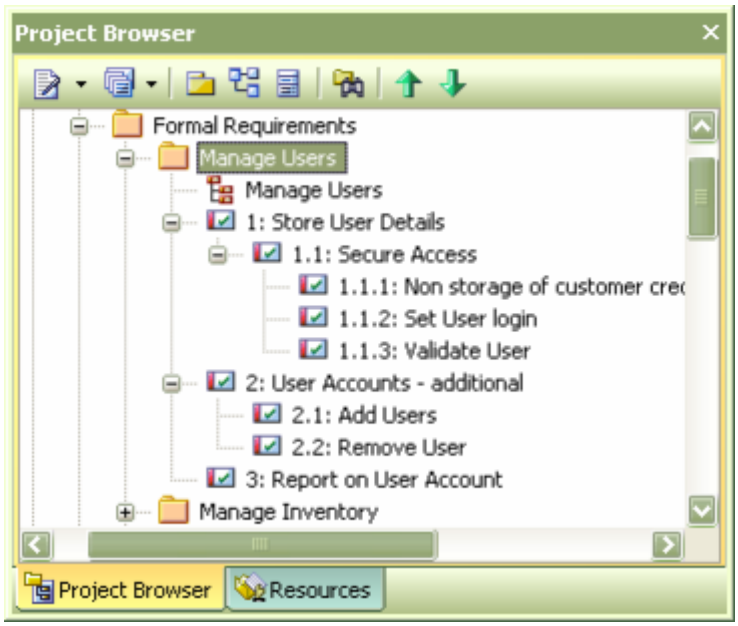


Figure 8: An Element Hierarchy with Level Numbering

To enable this option:

- Select a package in the Project Browser
- Right-click and from the context menu select: advanced -Turn on Level Numbering

Different Views of Requirements Using List View

Enterprise Architect supports a text-based view of the requirements, while maintaining a hierarchical structure in the Project View.

To enable this option:

- Select a package in the Project Browser
- Right-click and from the context menu select: view as list

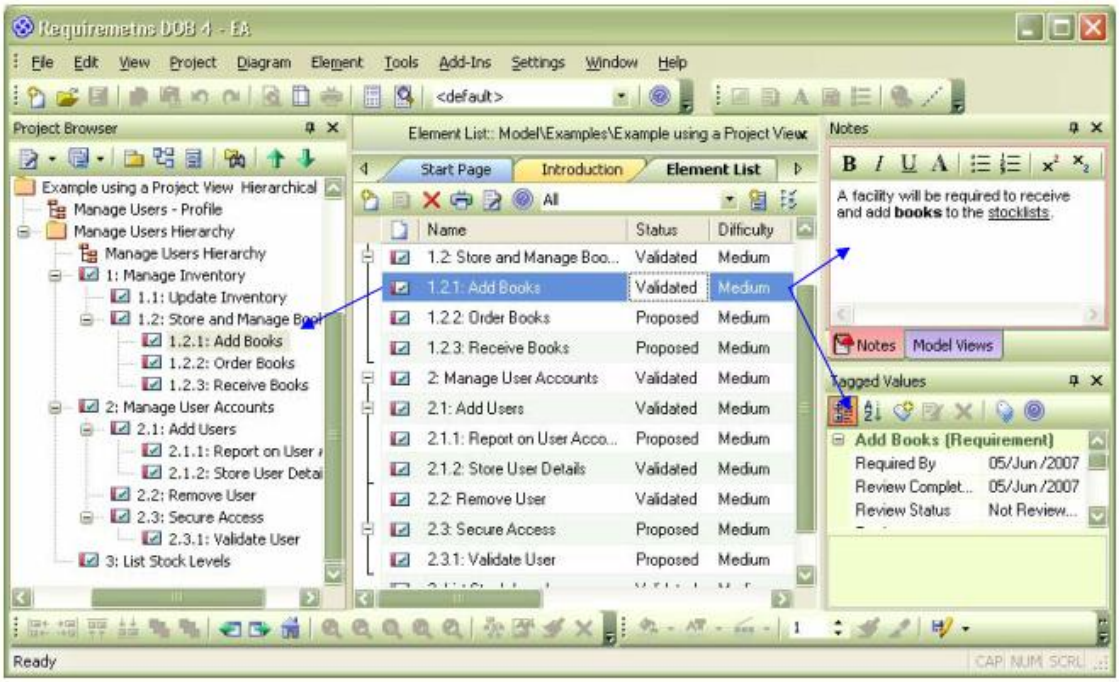


Figure 9: An alternate visual layout

Lecture 5

Traceability and relating requirements

When modeling using requirement elements there are numerous UML connector types that can be used, however there are two types of relationship that are commonly used with requirement management. One for setting relationships between peer requirements (Aggregation), and another for representing how they will be implemented (for example a Realization by a Use Case).

Creating and Viewing Relationships In Enterprise Architect

there are three key methods used for tracking requirements and forming relationships between the requirements and their related elements. These relationships define how those requirements are to be implemented within the system. The three key methods are as follows:

- **Creating and viewing relationships using diagrams** Relationships between elements are easily created in a diagram using standard relationships defined in the Toolbar.
- **Creating and viewing relationships using the Relationship Matrix** The Relationship Matrix provides a process for viewing or creating links between elements in different packages, independent of them being defined in a diagram.
- **Tracing relationships using the Traceability View** The Traceability window provides a feature for tracing all the relationships of a selected element.

Creating relationships using diagrams

Creating relationships between elements on a diagram is a simple process in Enterprise Architect.

Creating a common diagram

Creating links between objects in different packages can be a simple process, using a common diagram.

To do this, simply:

- Create a new diagram
- Drag onto the diagram, from the Project Browser, the elements in the different packages. Below is an example of a diagram with elements from different packages that were linked via the Relationship Matrix.

Lecture 6

The Relationship Matrix

The Relationship Matrix allows you to create and view relationships, regardless of what diagram or package the elements are placed in. It can be used with any UML element, but it is particularly useful in Requirements Management for two reasons:

1) With a large system definition it may be cumbersome using diagrams to define large sets of relationships between requirements and other elements. An alternative is to use the Relationship Matrix to quickly set relationships without the need to draw these in a diagram.

2) As the development phase progresses, each element that defines either an Aggregation or Realization of a requirement, such as another requirement or a Use Case, must be linked to its root requirement definition using a connector. It is this linking that is critical to backward traceability. This is where the Relationship matrix can be useful tool for verification of links.

Figure 1 is an example of two related requirements that are in separate packages.

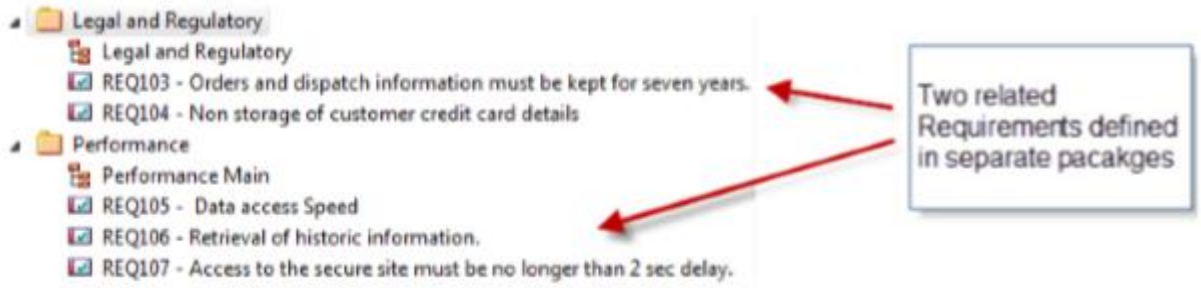


Figure 1: Requirements defined in separate packages

Figure 2 shows the Relationship Matrix view connection between the requirements in Figure 1.

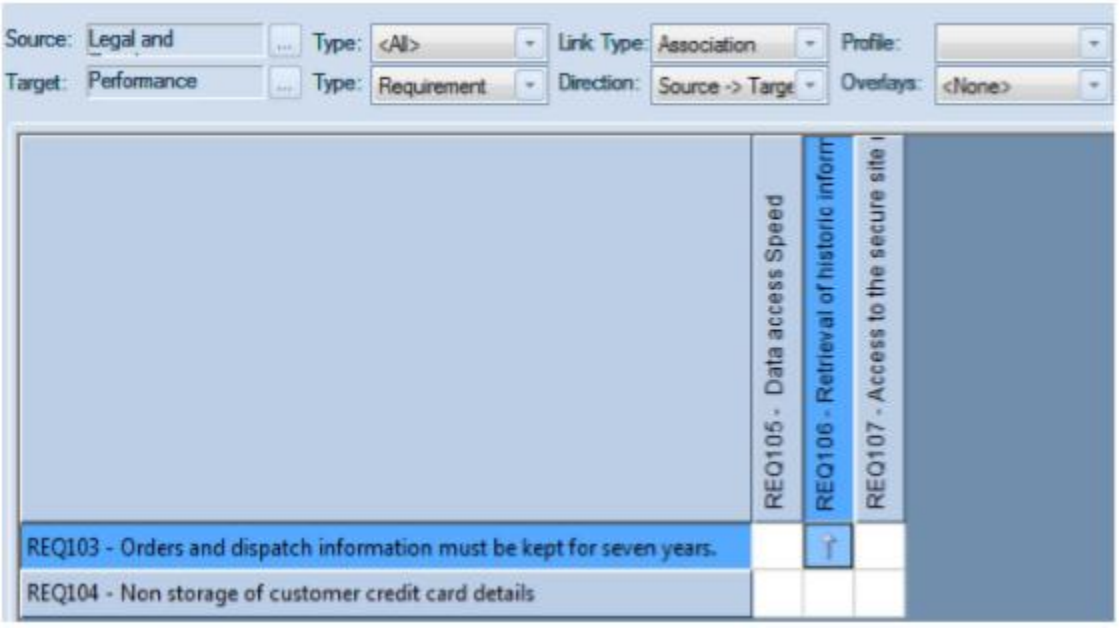


Figure 2: A Relationship Matrix view connections between Elements from different packages

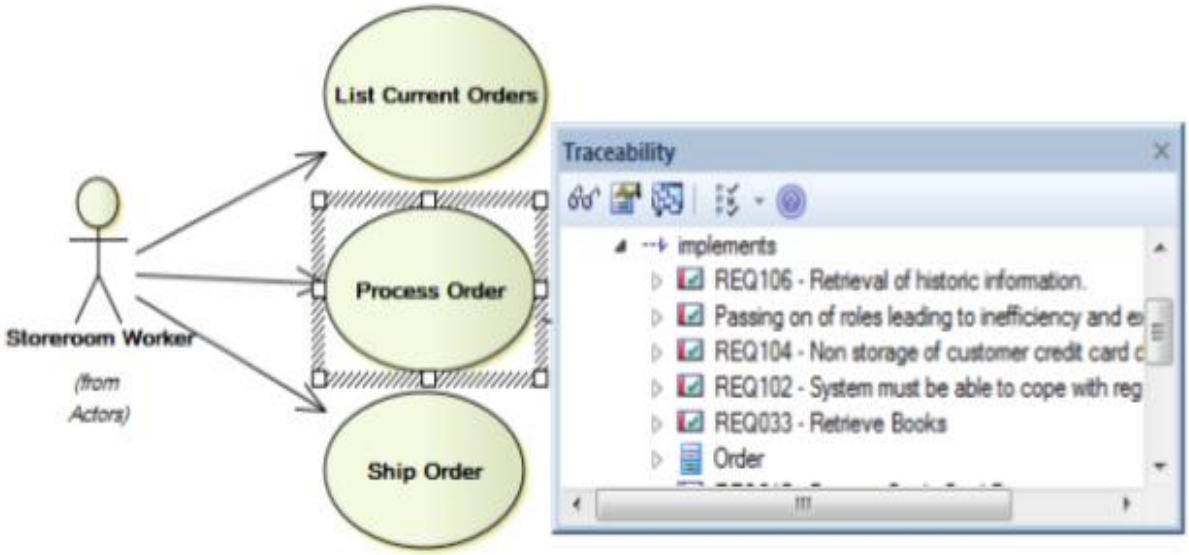
Tips and tricks

- Use the Relationships Matrix to create, edit and delete relationships, rather than doing this graphically in the model diagrams. This is most applicable when crossing different levels of abstraction e.g. from requirements to Use Cases.
- Use the automatic process of creating a relationship using drag-and-drop.

Using the Traceability window

The Traceability window allows you to view the relationships across a hierarchy of elements. It is particularly useful to see the relationships from Requirements to Use Cases, and down through the different levels of UML diagrams. Below is an example

of relationships between the Requirements, and the Use Case for ‘Processing an Order’.



To use the Traceability window for viewing relationships

- Open the Traceability window (Design > Impact > Traceability or Ctrl+Shift+4).
- Select the element for which you want to display relationships.

Lecture 7

Change Control

Enterprise Architect supports features for monitoring changes to requirement definitions. These include:

1- Auditing.

2- Managing Baselines.

3- Element Change requests and Issue logging.

Auditing

The Audit feature enables you to record model changes in Enterprise Architect. It records details of who changed an element, when and what was changed, and the prior state of the model. This can be particularly useful for recording a history of changes to requirements models.
Configure ⟶ model ⟶ Audit

Figure (1) is an example of viewing alterations to an element directly in the Audit View. This shows a number of alterations with the first selected to show the details on the right pane.

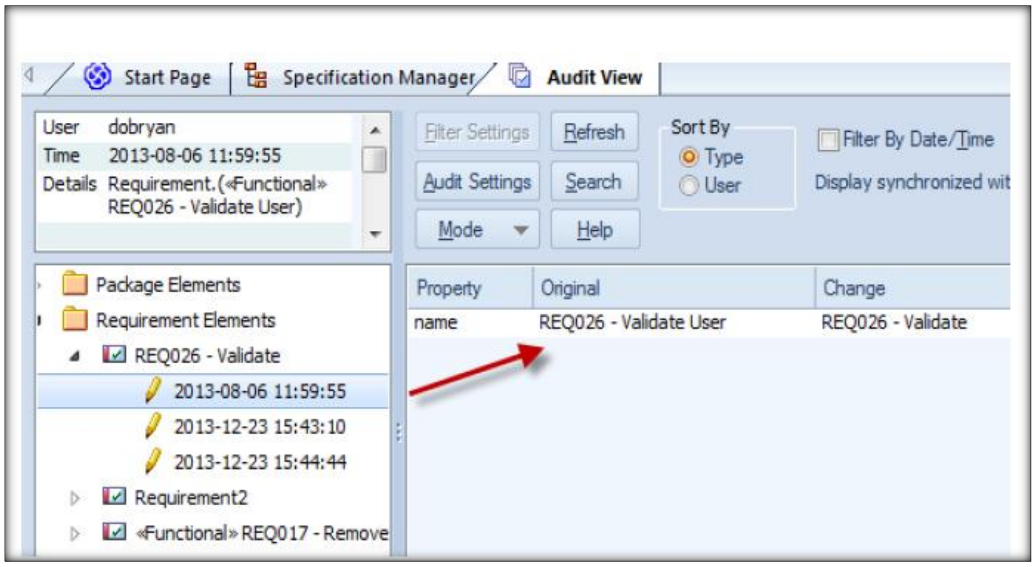


Figure (1) Audit view showing a list of alterations with the details of a Name change shown

Using Baselines

The auditing feature outlined above provides continuous tracking and logging of changes to requirements. The Baseline Management feature provides additional support for comparing and merging changes. It allows Baselines of a model to be created on a periodic basis (such as by month, phase, version or build). Baselines can then be compared to the current model and changes selectively rolled back.

Design ⟶ Baseline

Lecture 8

Change Control

Change Requests and Issues on External Requirements

Enterprise Architect supports logging of Change-requests against requirements. This can be defined using two different methods:

a) Using the Maintenance window to list Changes, Defects, Issues and Tasks against each element.

b) Using custom elements of type 'Issue' and 'Change' linked to the External Requirements being altered.

Each has their different uses which are outlined as follows:

Using the Maintenance window

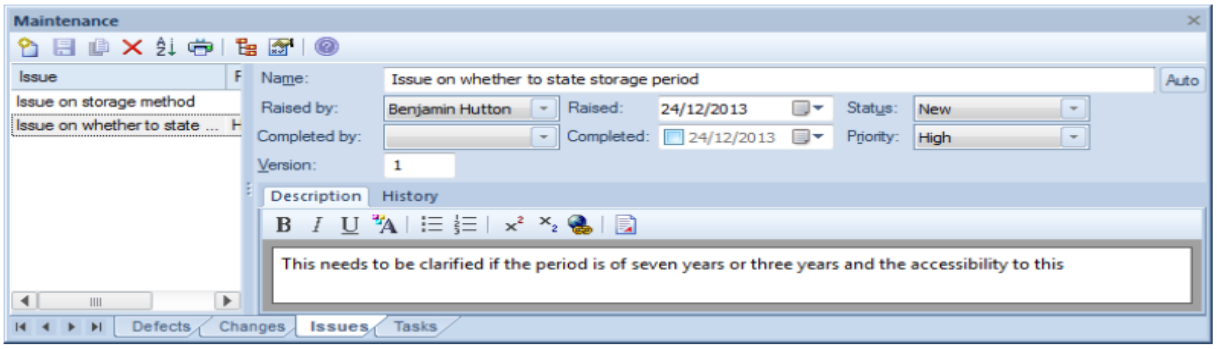
The Maintenance window can be used to log changes against any element or package. This provides listings for:
o Element Defects

o Element Changes

o Element Issues

o Element Tasks

These include fields for recording 'by whom' and 'when' the request was made and completed, as well as Status, Priority, Description and History. The Maintenance window can be accessed from the main menu using: (Alt+4). Figure 2 is an example of a set of changes listed for an element:



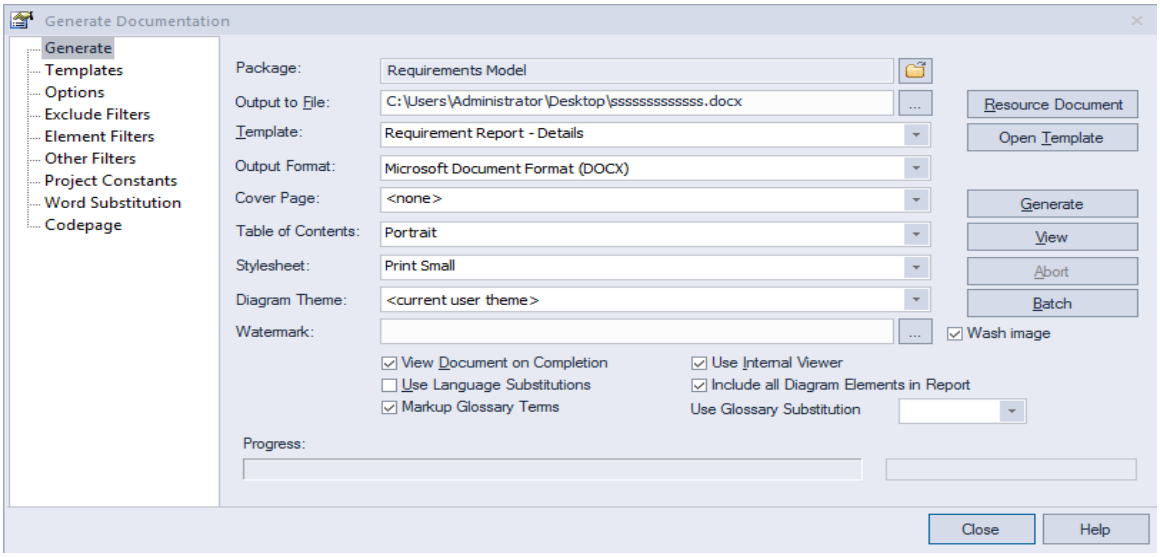
Lecture 9

Creating quality requirements documentation

A definition of a requirement is often used as a contract either between different departments within an organization or between organizations. Therefore, it is often required that high quality documentation of this definition can be generated.

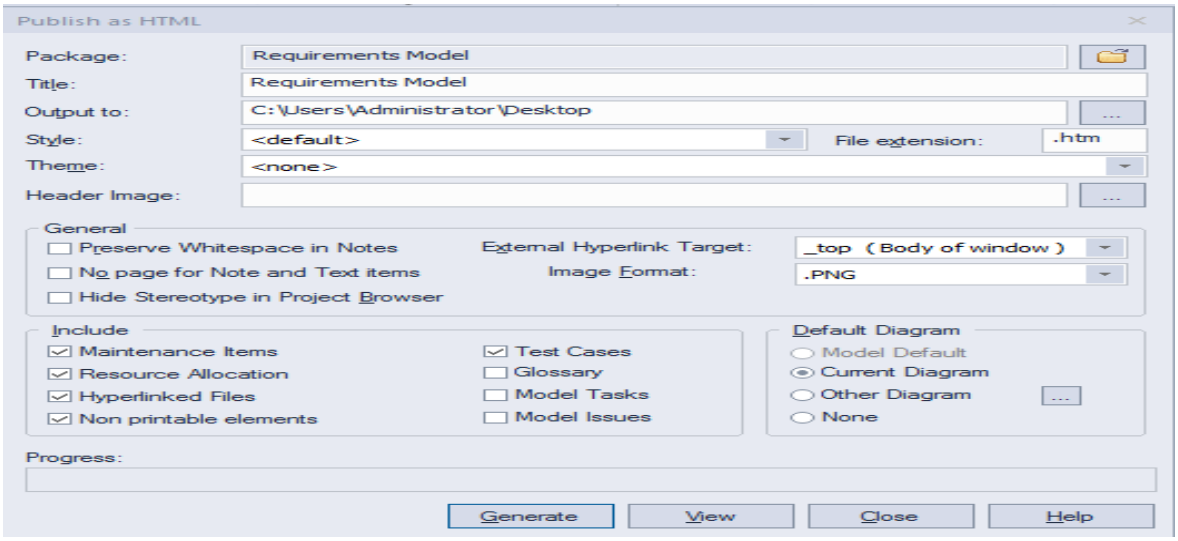
1- Create report

To create report RK on main Package →documentation → Generate documentation :



2- Create web page for Requirement diagram

To create web pages RK on main Package →documentation → Html report(Shift+F8):

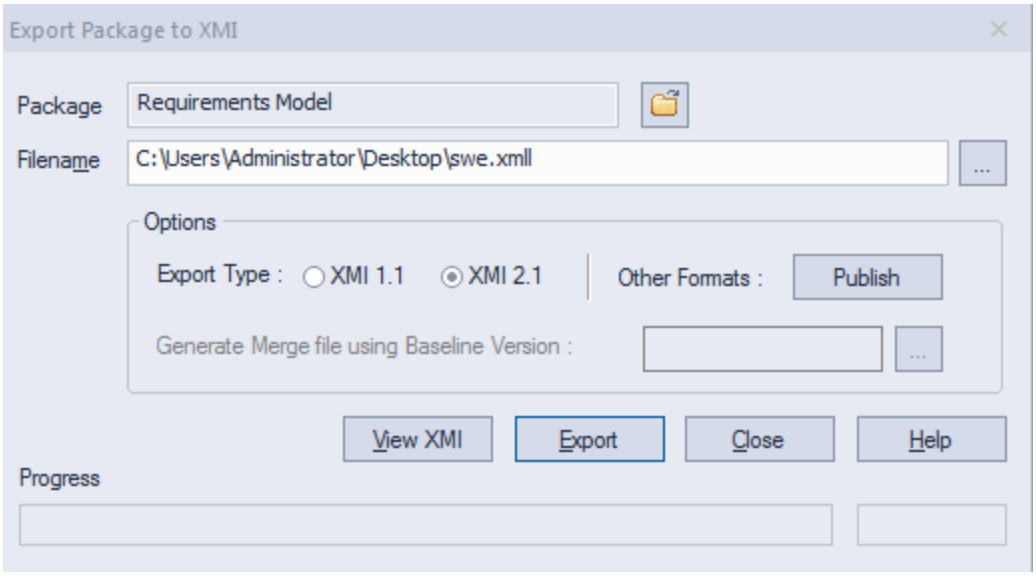


Importing /Export Requirements

1-Export in EA

To Export the Package from Enterprise Architect to XML file follow the step:

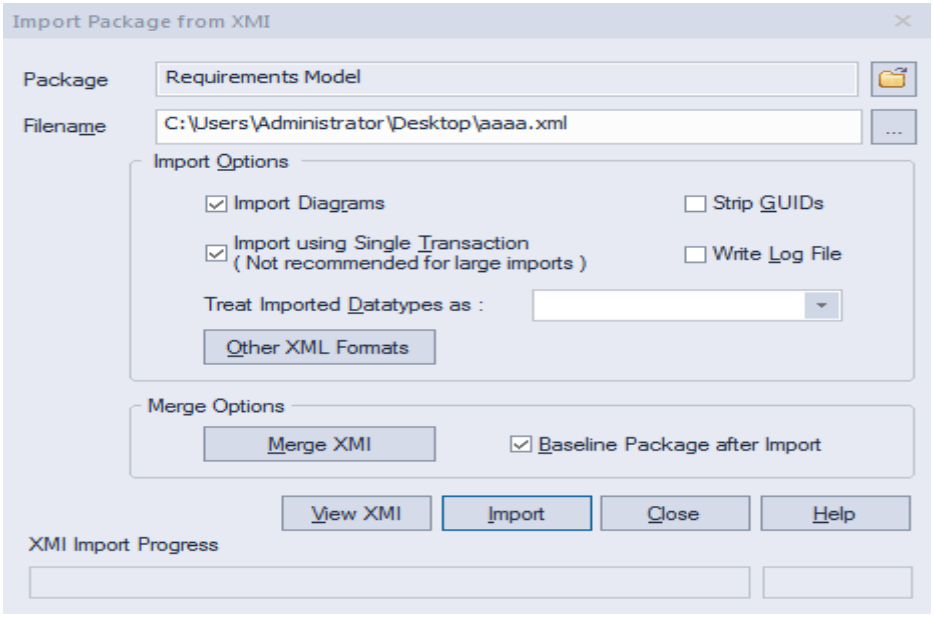
RK on main Package →import/export →export package to XMI



2-Import in EA

Where you need to import requirements from an external source(that must be export previously) follow the step

RK on main Package →import/export →import package from XMI



Lecture 10

Additional Requirements Management Features

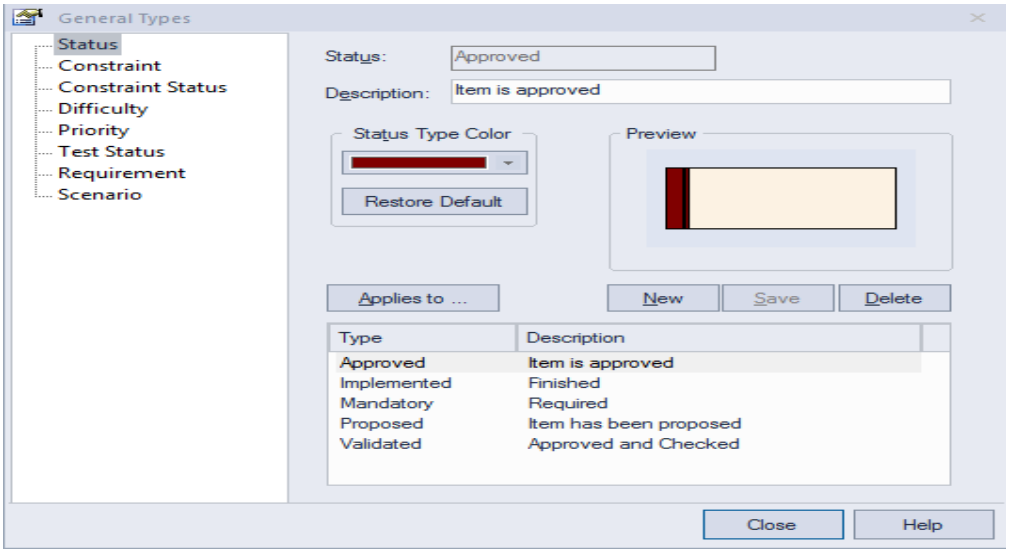
Enterprise Architect provides a number of other features for requirements management that will be covered in the following

1- Color Coding Requirements

External requirements may be color coded to enable quick visual cues indicating the status of a requirement. To enable color coded external requirements take the following steps:

From layout → appearance → configure statues color

Check the Show Status colors on diagrams checkbox.



must be select status of Requirements(approved, implemented, mandatory ,proposed, validated) to appear color

2-Auto Element Naming

This feature allows you to configure Enterprise Architect to automatically name and number

requirements as they are created. (It can be used with almost any UML element such as use case and classes). It is particularly useful with requirements, since

these are often required to have a unique reference for external checking ,To enable auto element naming, take the following steps:

1. Select Configure | Settings | Auto name and counter.

2. Change the Type to requirement.

3. Set the Prefix or Suffix. Both are optional (eg. “REQ”).

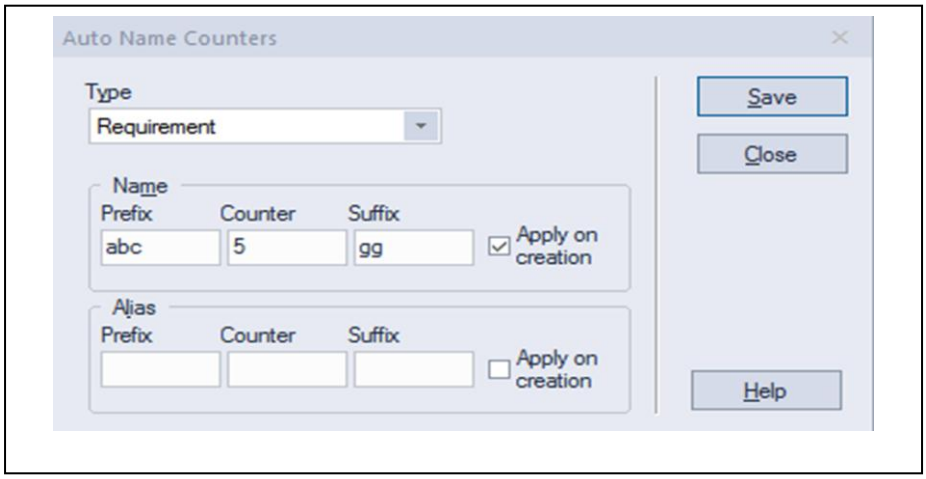
4. Type in the value for your first requirement in the Counter box (eg. “0002” or “99”).

5. Check the Active checkbox to enable the auto naming.

6. Click Save to save your changes.

The first requirement will take the value entered into the counter box and Enterprise Architect will then

increase the value by one for every new requirement (eg. “5”).



3-Drag and Drop Realizations

A fundamental aspect of the management of requirements is the ability to trace the parts of the system that implement, or realize, a particular requirement. A quick method of generating a realization link is to drag a requirement element from the project browser over an element in a diagram, which is to be the implementing element. EA will interpret this as a request to create the realization link and do so automatically.

Lecture 11

Additional Requirements Tools In addition to the Key Tools listed in Meet the Requirements there are a number of other tools that can be used for Requirements development and management and team collaboration in the requirements engineering discipline. These include the ability to assign sequential numbers to requirements, importing Requirements from a spreadsheet file, creating requirements documentation and team collaboration features such as Element Discussions.

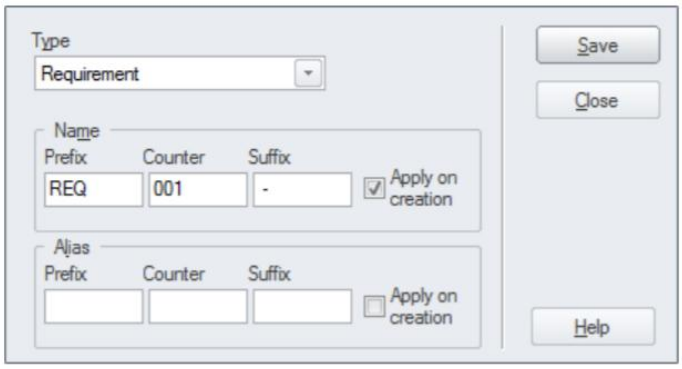
More Requirements Tools

Tool	Overview
Auto Names and Counters	Use to assign a sequential number to a Requirement including a prefix and suffix.
Requirements Checklist	Provides a graphical list of checks that can be applied to individual requirements.
Import and Export Spreadsheets	A tool to import and export Requirements from Spreadsheet files in the CSV format.
Documentation	A powerful engine to automatically create high quality documentation directly from the model, using built-in or user defined templates.
Glossary	A feature to create and maintain a lexicon of terms and their meaning that can grouped by type and styled when included in documentation.
Auditing	Used to keep a trail of what has changed in a repository, who it was changed by and when.
Element Discussions	A facility to allow modelers to create posts and replies to discuss model elements.
Maintenance Items	A series of Items that can be applied to requirements to define such things as Changes, Issues, Defects and more.
Team Reviews	Allows reviews to be created for user defined categories and topics with links to model elements such as Requirements and Scenarios that can be referenced in the review.

Auto Names and Counters

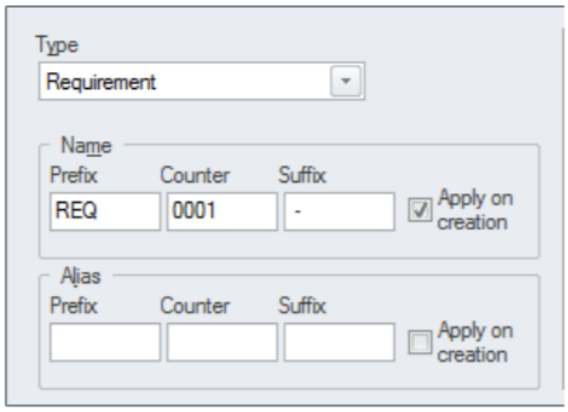
Getting to know Auto Names and Counters Introducing Auto Names and Counters To aid, regulate and enforce a naming standard, Enterprise Architect includes some

capabilities to configure the default names assigned to new elements of a specific type. This is a useful feature when dealing with complex and large sets of requirements, but is also relevant when dealing with smaller data sets. Auto Names and Counters can be used to assign a sequential number to any element type including Requirements. It includes a prefix definition, a counter and a suffix definition allowing numbers such as: 'REQ007 - Manage Inventory' to be created.



Where to find Auto Names and Counters Main Menu: Project | Settings | **Auto Names and Counters**

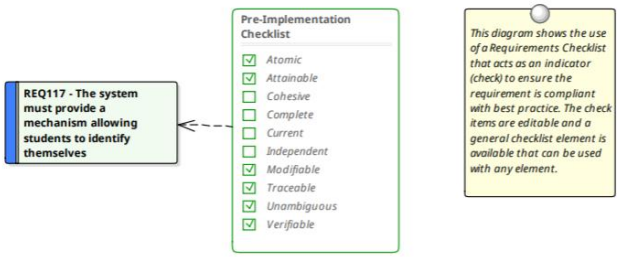
Usage of Auto Names and Counters Analysts and others can use the sequential number for communicating unambiguously about the requirements without having to use the often long requirement name. The 'Apply on Creation' option must be checked to start using the auto numbering feature, this can also be used to temporarily suspend auto naming for example if other types of requirements are being entered that don't need to have sequential numbers assigned. Options for Auto Names and Counters There are options to define the prefix, counter and suffix for a requirement.



lecture 12

Requirements Checklist

Getting to know the Requirements Checklist Introducing the Requirements Checklist The Requirements Checklist is a convenient element that acts as a tally to indicate whether a Requirement complies with a set of predefined measures such as whether the Requirement is Atomic, Cohesive, Traceable and Verifiable. It can be assigned to any Requirement and the measures can be updated directly in the diagram. When working with requirements it is sometimes very useful to refer to a common set of 'best practices' and qualities that help define the nature of a well formed specification. The Requirement Checklist element is designed to meet this need.



Where to find the Requirements Checklist	Toolbox | More Tools... | Requirements | Extended Requirements Page | Requirements Checklist
Usage of the Requirements Checklist	Analysts and Requirements Managers can use the checklist to annotate whether one or more requirements meet a set of predefined checks.
Options for the Requirements Checklist	The list of measures is completely configurable and items can be added or removed from the list for each individual checklist by using the Checklist **Tagged Value** notes.

```
<Checklist>
        <Item Text="Atomic" Checked="True"/>
        <Item Text="Attainable" Checked="True"/>
        <Item Text="Cohesive" Checked="False"/>
        <Item Text="Complete" Checked="False"/>
        <Item Text="Current" Checked="True"/>
        <Item Text="Independent" Checked="False"/>
        <Item Text="Modifiable" Checked="True"/>
        <Item Text="Traceable" Checked="True"/>
        <Item Text="Unambiguous" Checked="True"/>
        <Item Text="Verifiable" Checked="True"/>
</Checklist>
```