



Interaction between ARCSeeker and Enterprise Architect

by SparxSystems Japan



1. Introduction

ARCSeeker is a tool which supports reuse of software asset by collecting scattered information in one place and improving retrieval performance. Not only source code but specification, test item and test result are also considered as a 'component (parts)'. This improves the efficiency of information management and usage in a meaningful unit.

This document, intended for those who are already using the UML modeling tool "Enterprise Architect," introduces how the ARCSeeker divide and store the existing UML models as "assets." Also, it mentions how to search and use the stored information. Rather storing the created UML models as-is, this makes it possible to divide the model into small units to be referred and used conveniently, and build the structure to combine the models to reuse as needed.

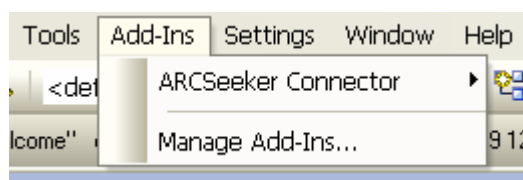
ARCSeeker version 2.3 build 046 is used in this document.

2. Enterprise Architect • Overview of ARCSeeker Interaction

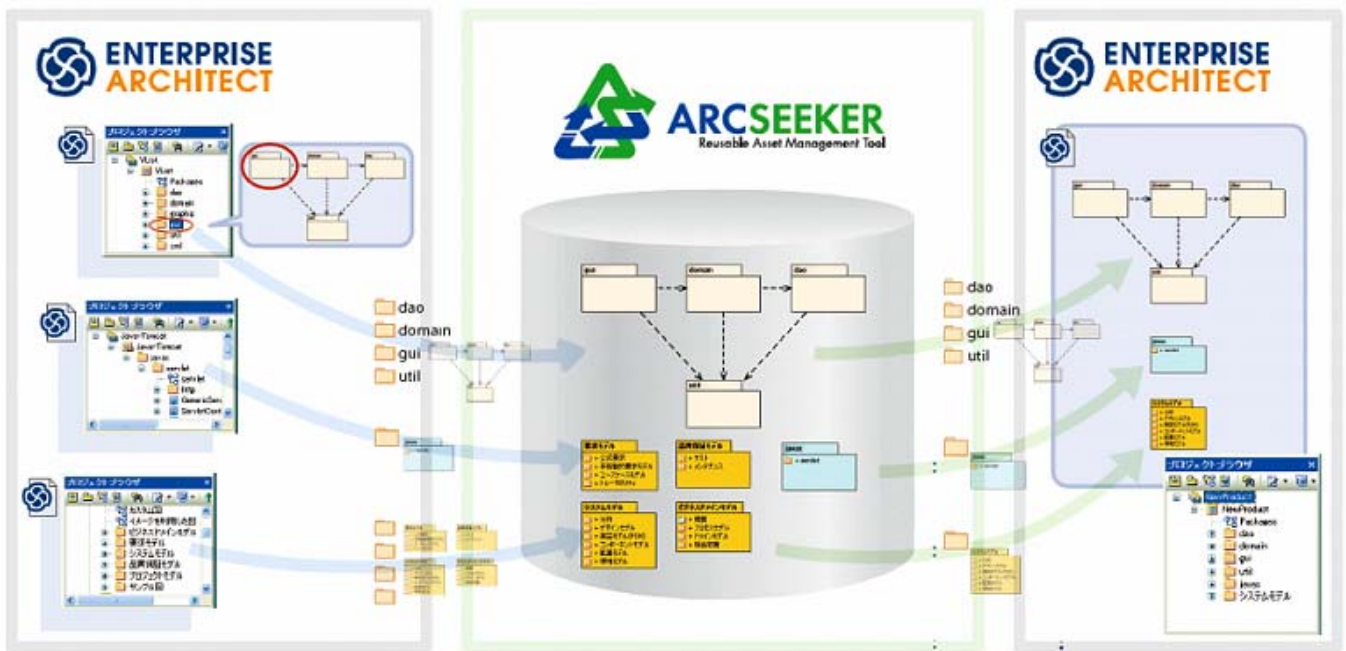
ARCSeeker can import packages of Enterprise Architect as components, and divide them into small parts. When importing the packages, its related packages can also be imported, or the supplementary information can be added into the component at this point.

And from Enterprise Architect, importing the components of ARCSeeker into the project is easily handled.

Interaction function to Enterprise Architect is automatically installed at the ARCSeeker installation. When ARCSeeker is installed, "ARCSeeker Connector" is appeared under "Add-Ins" of Enterprise Architect main menu.



Following is an image of interaction between ARCSeeker and Enterprise Architect.

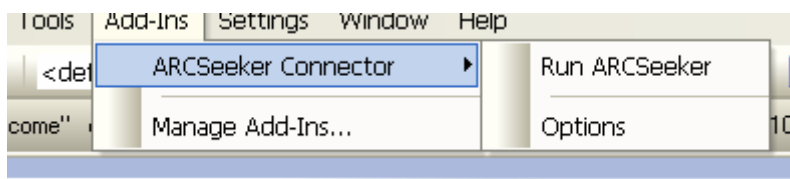


ARCSeeker collects and stores the UML models, created by Enterprise Architect, as package units. Stored data is tagged and associated. Based on that information, ARCSeeker searches and finds the needed packages.

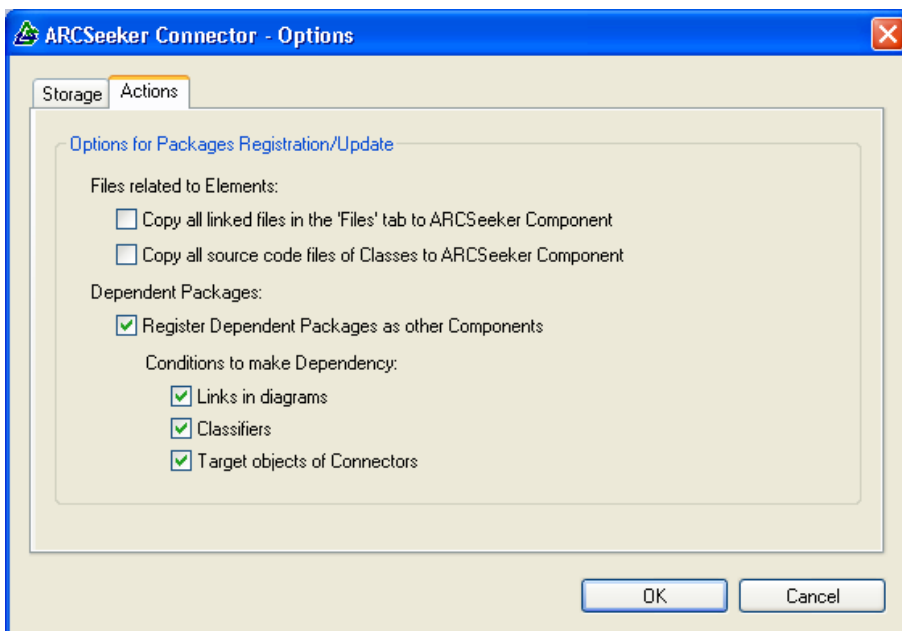
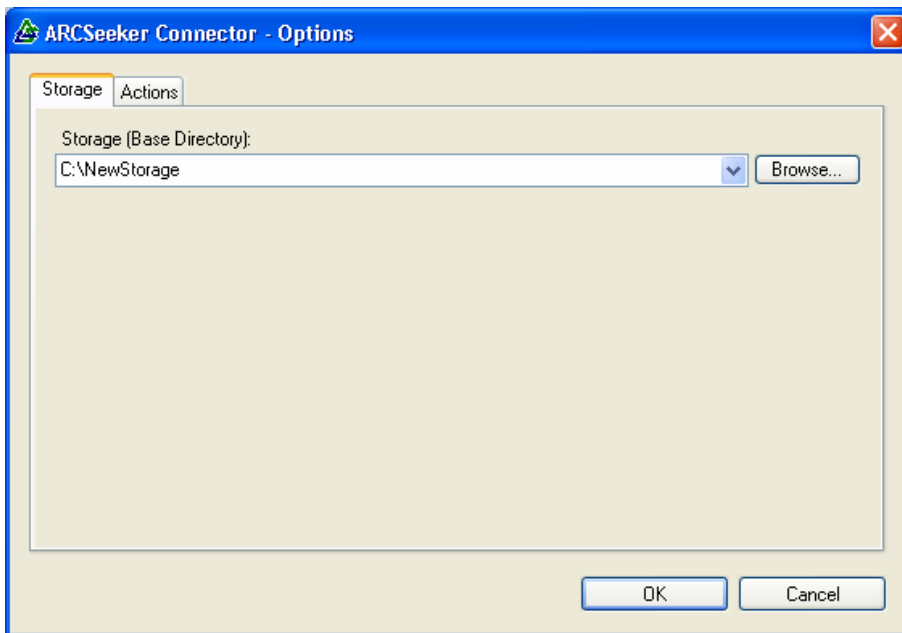
3. Preparation

Now, prepare for the interaction between Enterprise Architect and ARCSeeker.

When Enterprise Architect runs, “ARCSeeker Connector” is shown under “Add-Ins” of main menu. As its sub menu, “Run ARCSeeker” and “Options” are shown.



Select “Options” to show the following dialog.



Specify a storage path of ARCSeeker in the “Storage (Base Directory)” box. If the storage has yet to be created, you need to create one in advance. Refer to the information in chapter 8, run ARCSeeker and create storage.

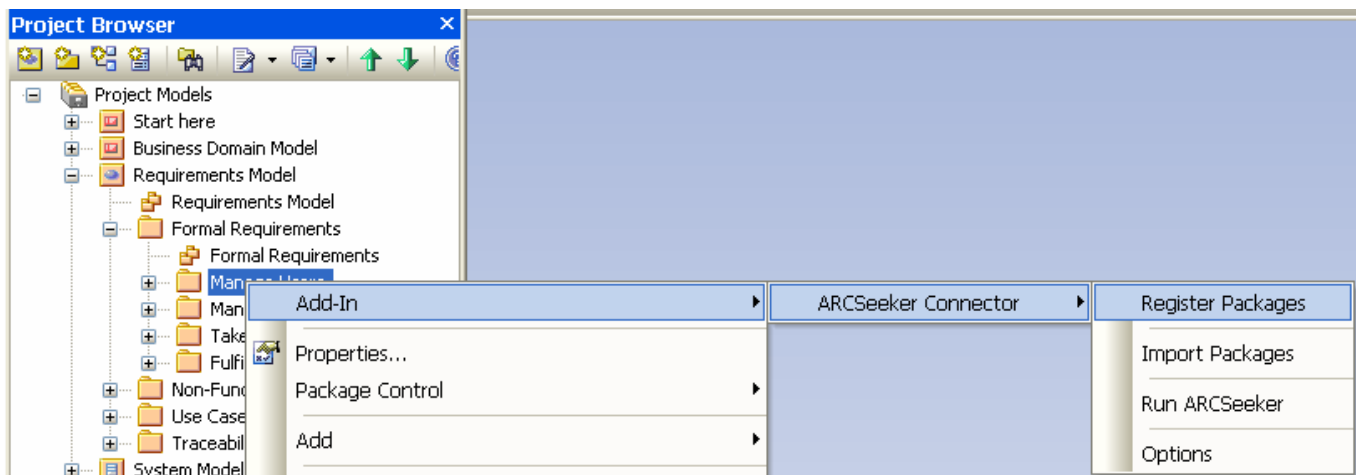
Three features of “Options for Package Registration/Update” in the Actions tab are an option applied when creating new components from the Enterprise Architect packages. Set them as needed.

Now the preparation is all set. Open the Enterprise Architect project file to be imported as components.

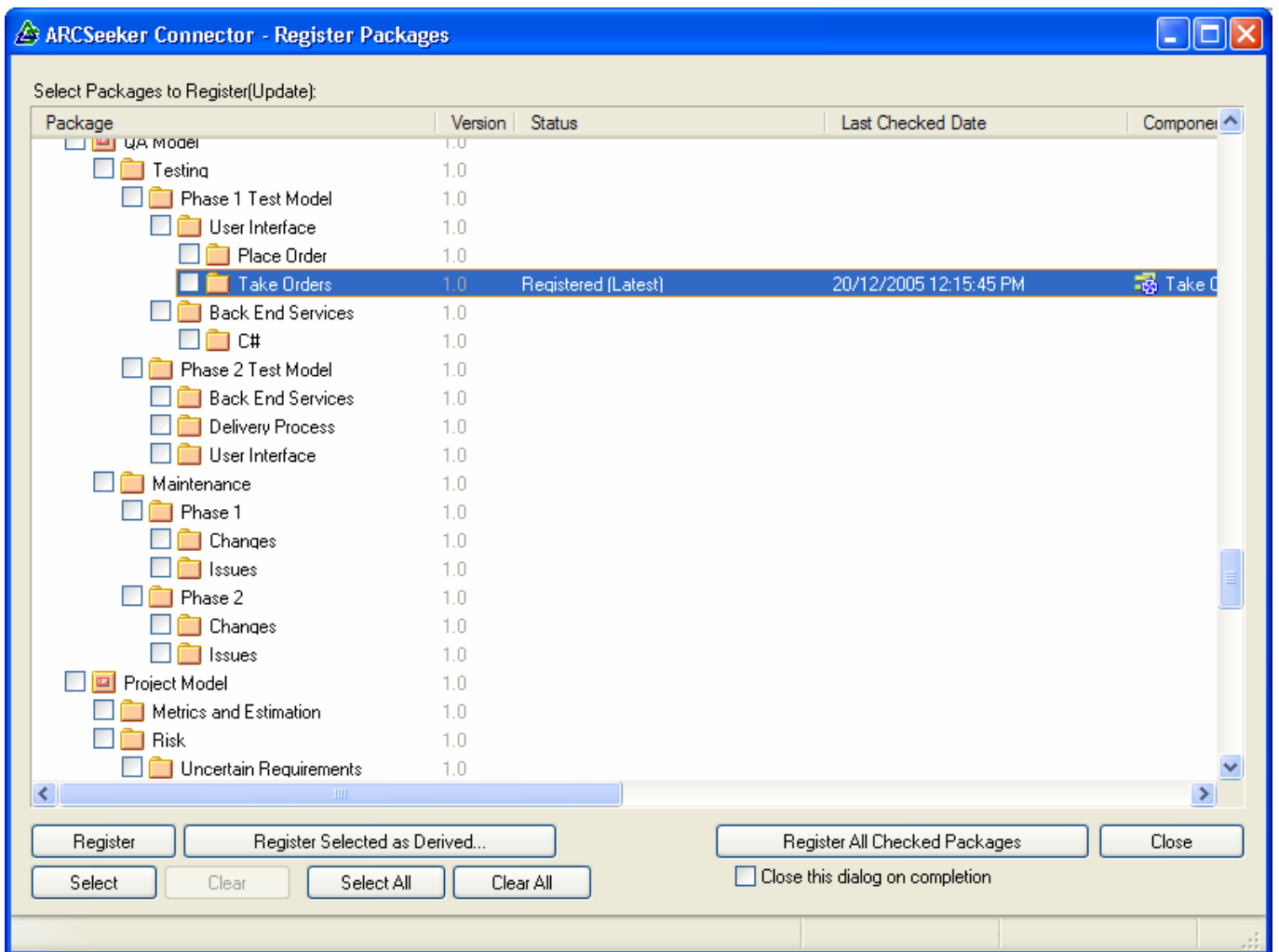
4. Registration

Register the Enterprise Architect package as components. In here, the sample file of Enterprise Architect, “EAExample.eap,” is used.

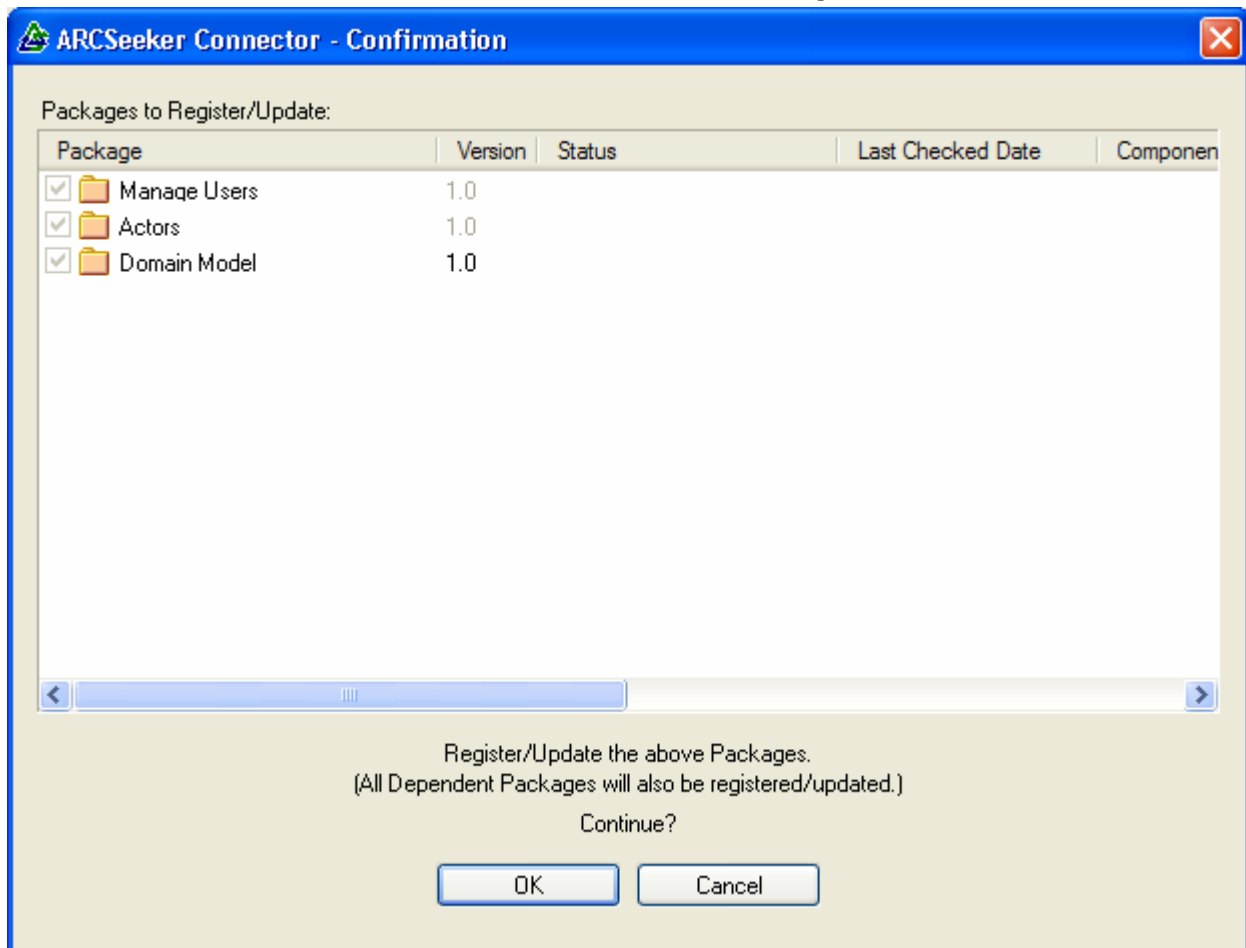
At the project browser, right click the package to be registered, and point Add-In, ARCSeeker Connector, and Register Packages.



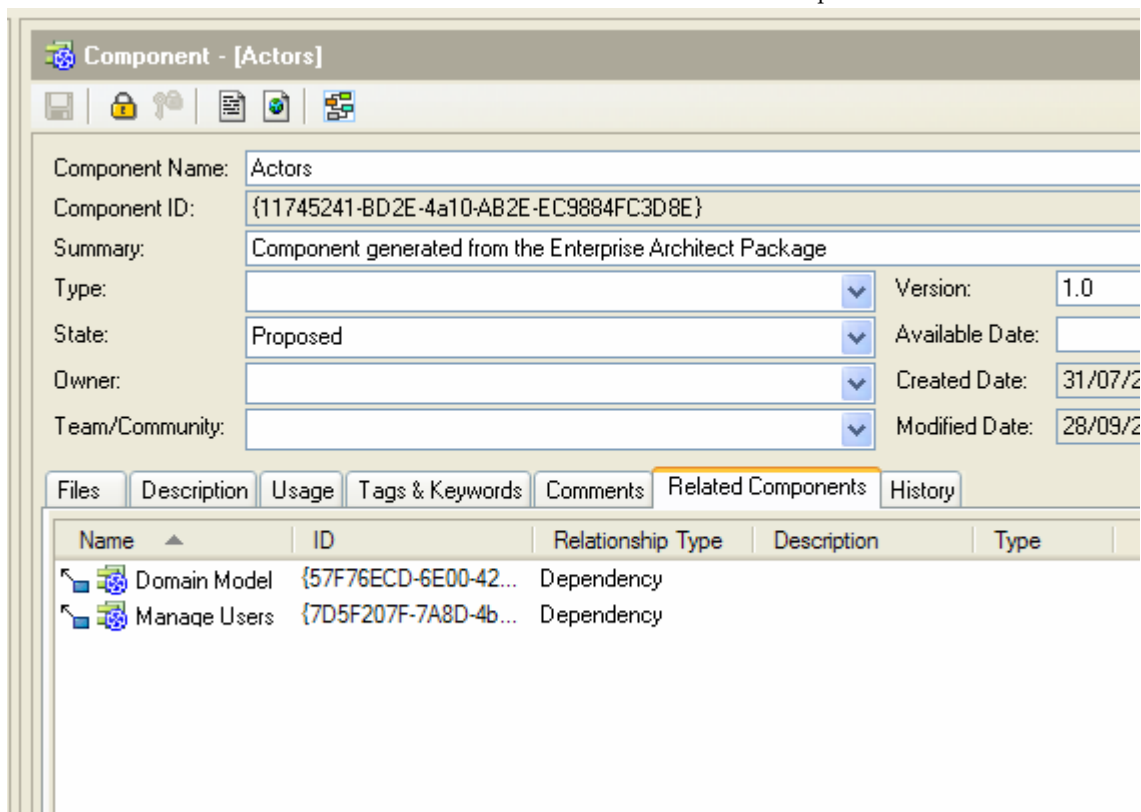
Below dialog is shown to select packages. Check the packages to be imported as components.



Depending on the package, unspecified packages are also shown as importing packages. For example, in the EAExample model, only the “Manage Users” package is selected, but in the check dialog before execution shows the packages that haven’t been selected.

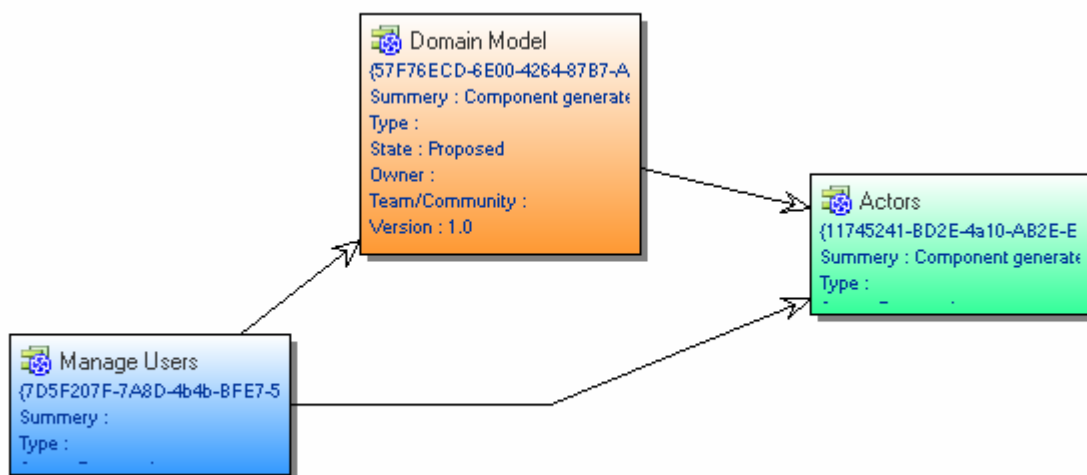


In the diagram of the selected package, ARCSeeker automatically checks the dependency relationships between those and shows the required packages in this dialog. This happens when the selected package uses the elements of those packages. The related packages are automatically transferred as components. You can easily find out a dependency between the packages in the “Related Components” tab of ARCSeeker, as below.

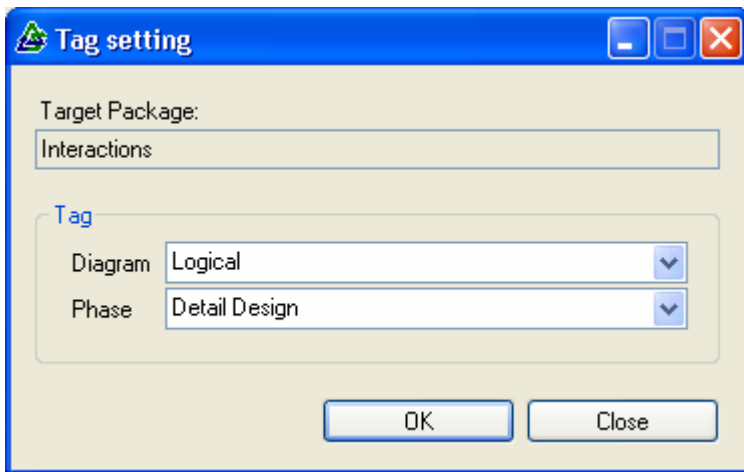


Double click the component in this “Related Components” tab to check the property of its component.

Relationships diagram between the components are shown as below. To see this diagram, right click the component and click “Open Relationships diagram.”



“Tag” value is uniquely stated at the registration of the components. Please see chapter 8 for the detail.



Enter the information for the selected package in this dialog, and click OK. If the default is predetermined, it shows the default information. If options are defined, items can be selected from the list in the combo box. Direct entry of the value is also possible.

After this setting, the packages are registered as components in ARCSeeker. Some properties like states and notes of UML model packages are registered as-is. Also, the diagrams in the packages are saved as images. The option settings enable ARCSeeker to import the files specified in the element's "File" tab, or source files of class elements into the components.

Registering the components into ARCSeeker is pretty easy as described above; just right click the package in the project browser and call the importing function.

5. Classification, search, reference

In this way, register the other packages. Following components are registered as below.

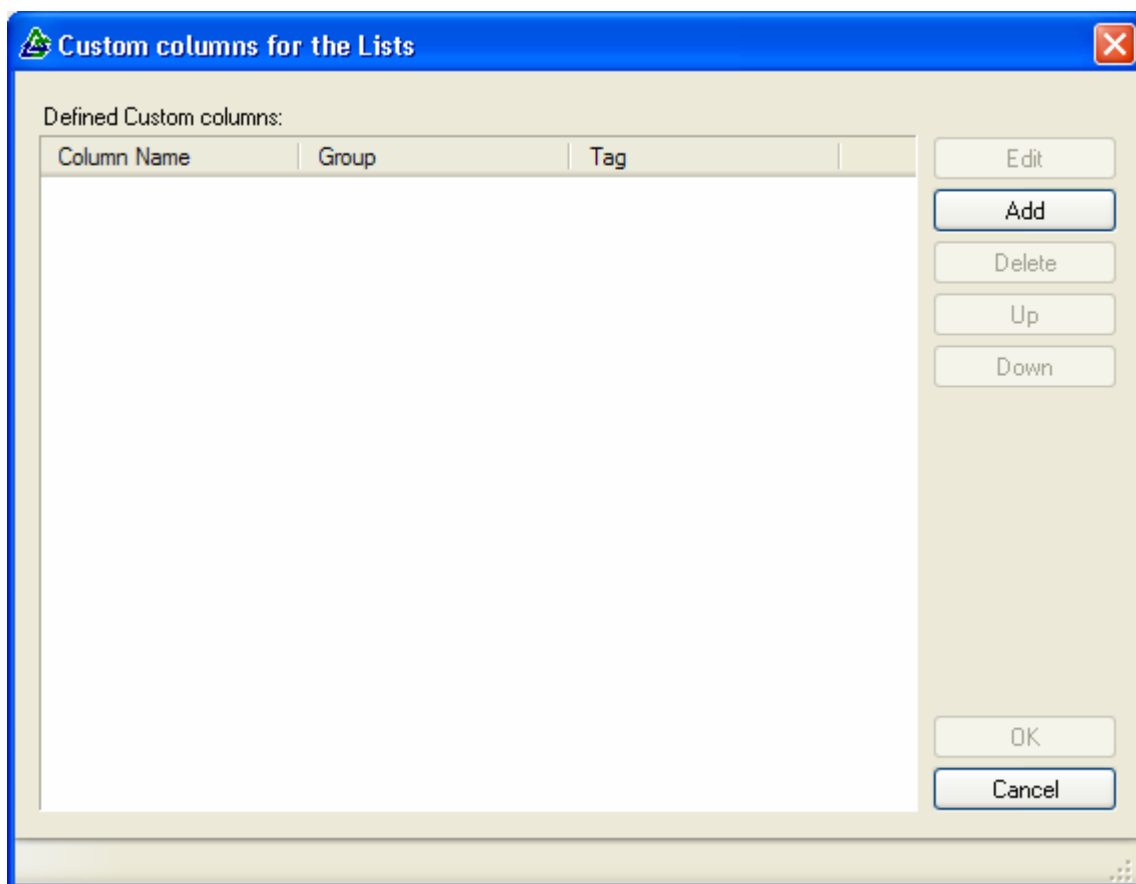
Name ▲	Version	Modified Date	Created Date
Storage (C:\Storage)		28/09/2009 9:29:2...	30/07/2008
Actors	1.0	28/09/2009 9:35:1...	31/07/2008
Domain Model	1.0	28/09/2009 9:25:0...	31/07/2008
Manage Users	1.0	28/09/2009 9:25:1...	28/09/2009
NewPackage	1.0	28/09/2009 3:43:5...	28/09/2009
Manage Users	1.0	28/09/2009 3:45:1...	28/09/2009
Manage Inventory	1.0	28/09/2009 3:45:1...	28/09/2009
Take Orders	1.0	28/09/2009 3:45:1...	28/09/2009
Fulfill Orders	1.0	28/09/2009 3:45:1...	28/09/2009
Login	1.0	28/09/2009 3:46:2...	28/09/2009
Search	1.0	28/09/2009 3:46:2...	28/09/2009
View Orders	1.0	28/09/2009 3:46:2...	28/09/2009
View Basket	1.0	28/09/2009 3:46:2...	28/09/2009
PlaceOrder	1.0	28/09/2009 3:46:2...	28/09/2009
MakePayment	1.0	28/09/2009 3:46:2...	28/09/2009
Login Screen	1.0	28/09/2009 3:46:2...	28/09/2009
Register User	1.0	28/09/2009 3:46:2...	28/09/2009
Browse Catalogue	1.0	28/09/2009 3:46:2...	28/09/2009
Account	1.0	28/09/2009 3:49:3...	28/09/2009
LineItem	1.0	28/09/2009 3:49:3...	28/09/2009
Order	1.0	28/09/2009 3:49:3...	28/09/2009
OrderStatus	1.0	28/09/2009 3:49:3...	28/09/2009
ShoppingBasket	1.0	28/09/2009 3:49:3...	28/09/2009
StockItem	1.0	28/09/2009 3:49:3...	28/09/2009
Transaction	1.0	28/09/2009 3:49:3...	28/09/2009
LoanApproval_Process	1.0	28/09/2009 3:49:4...	28/09/2009
LoanServicesOrg	1.0	28/09/2009 3:49:4...	28/09/2009
Customer Order	1.0	28/09/2009 3:51:0...	28/09/2009
Fulfill Orders	1.0	28/09/2009 3:51:0...	28/09/2009

To recycle/reuse the past assets, needed components have to be found effectively from the so many components as above. Not be able to find required information causes a waste like recreating the existing models or source codes.

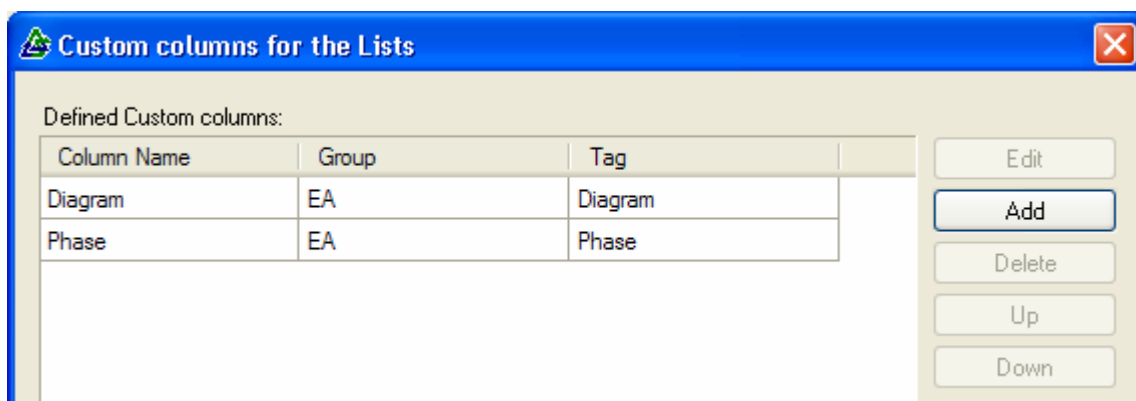
If ARCSeeker is not implemented, it is general to create folders, directories, in the network drive and classify them. This kind of static classification method takes a lot of time and effort. When changing the structure, it needs to be rearranged as well.

With ARCSeeker, it is possible that this kind of classification and hierarchy structure is dynamically constructed depending on the situation and goals of that moment. This function is called “Group List.”

To use “Group List” function, advance setting is required. Go to main menu “Settings,” and select “Custom columns for Lists.”

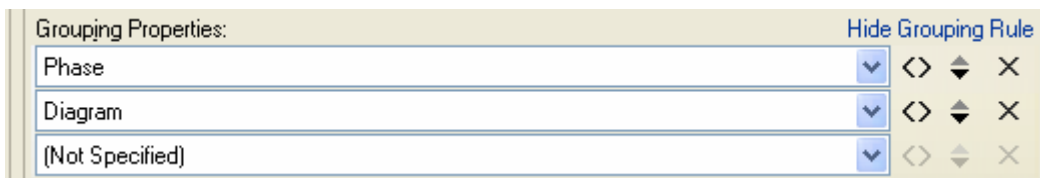


Click “Add” button in this dialog. It adds new items. Set each item as below.



Now, “Diagram” and “Phase” tag information are applied in ARCSeeker for utilization. Setting is completed. Click OK and close the dialog. Storage is reloaded and changes have been made.

Using the “Group List” function enables to easily classify and display the large amount of stored components. Classify the groups by “Phase” and “Diagram” as in the example, and it looks like this.



Name	Version	State	Available ...
BasicDesign			
Activity			
Fulfill Orders	1.0	Proposed	
Legal and Regula...	1.0	Proposed	
Logical			
Domain Model	1.0	Proposed	
Extensibility	1.0	Proposed	
Sequence			
Take Orders	1.0	Proposed	
View Basket	1.0	Proposed	
UseCase			
Fulfill Orders	1.0	Proposed	
DB Design			
Logical			
Manage Inventory	1.0	Proposed	
Manage Users	1.0	Proposed	
Manage Users	1.0	Proposed	
Performance	1.0	Proposed	
Detail Design			
Logical			
PlaceOrder	1.0	Proposed	
Positioning	1.0	Proposed	
Register User	1.0	Proposed	
Stakeholders	1.0	Proposed	
Sequence			
Stakeholders Inte...	1.0	Proposed	
Take Orders	1.0	Proposed	

By classifying those as above, it is easy to find components related to the possible type of diagram which is useful in the possible situation. It is differed from the limit search, so the list also shows supplementary components. You may unexpectedly find a useful component from the list.

For example, when searching for the useful or relevant components at “Activity diagram” of “Overview construction” phase, classification like above shows the components including other diagrams of “Overview construction” phase near-by. By this, the components that might be able to use in the future will be on your sight.

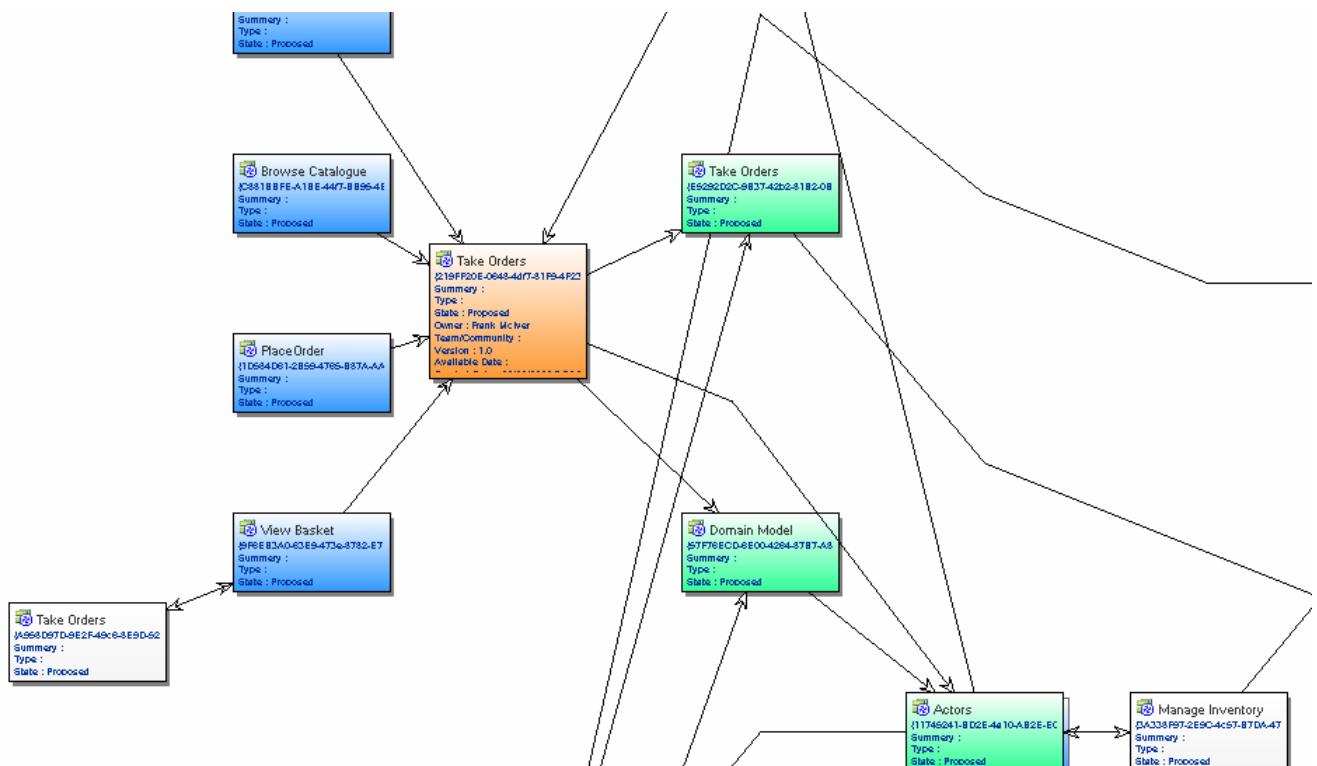
When condition is reversed as “Diagram” to “Phase,” the components including the “Activity diagram” used in various phase, are appeared near-by. This might help you how to write the diagram at the other phases.

In this manner, “Group List” enhances the possibility to find the components, frequently overlooked by the simple search.

Do not skip checking the “Related Component” tab content of the found component. It might offer the other useful components, unexpectedly. When you consider the component has some relation, you can add relation manually to encourage other person to reuse the components.

(By dragging other component into this “Related Components” tab, you can add relation easily. Or “Relationship diagram,” showed in earlier, can also add relation.)

Using relationship diagram is a good way to see a picture of complex relations between components.

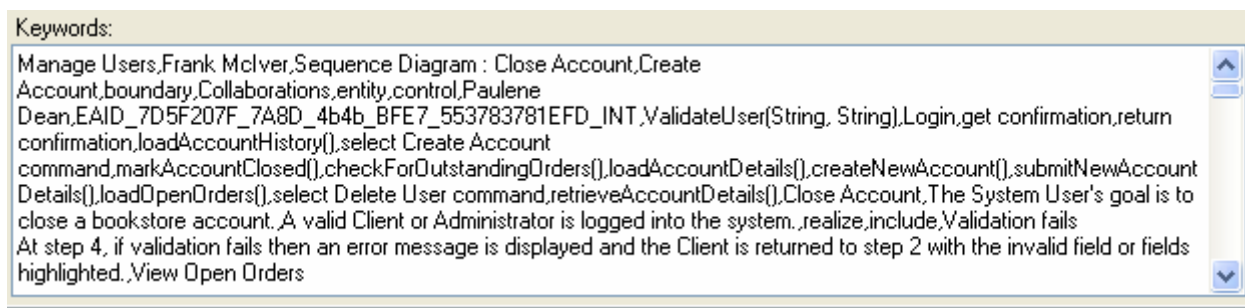


Like this, even if the components are not classified in the storage, ARCSeeker’s component list enables to classify and search the components in various ways.

To briefly check the contents of component, use an image of the diagram stored in “File” tab of component. In the “Images” folder, the diagram stored in the package is shown as an image. Double click the file to check the content of the image. This allows the brief check of component, even before the files are not imported into Enterprise Architect.

Name	Path	Version	File Type	Modified Date
Images	Images	1.0	File Folder	
Close Account	Images/EAID_A7...	1.0	Bitmap Image	28/09/2009 9:24
Close Account	Images/EAID_2C...	1.0	Bitmap Image	28/09/2009 9:24
Create Account	Images/EAID_E3...	1.0	Bitmap Image	28/09/2009 9:24
Create Account	Images/EAID_C7...	1.0	Bitmap Image	28/09/2009 9:24
Delete User	Images/EAID_A6...	1.0	Bitmap Image	28/09/2009 9:24
Delete User	Images/EAID_97...	1.0	Bitmap Image	28/09/2009 9:24
Login	Images/EAID_AB...	1.0	Bitmap Image	28/09/2009 9:24
Login	Images/EAID_74...	1.0	Bitmap Image	28/09/2009 9:24
Manage Users	Images/EAID_D3...	1.0	Bitmap Image	28/09/2009 9:24
View Account de...	Images/EAID_84...	1.0	Bitmap Image	28/09/2009 9:24
View Account de...	Images/EAID_81...	1.0	Bitmap Image	28/09/2009 9:24
View History	Images/EAID_40...	1.0	Bitmap Image	28/09/2009 9:24
View History	Images/EAID_FF...	1.0	Bitmap Image	28/09/2009 9:24
View Open Orders	Images/EAID_B5...	1.0	Bitmap Image	28/09/2009 9:24
View Open Orders	Images/EAID_BE...	1.0	Bitmap Image	28/09/2009 9:24
Manage Users.xml	Manage Users.xml	1.0	xmi File	28/09/2009 9:24

Elements' names, notes, and all that in the objective packages of Enterprise Architect are searchable by ARCSeeker. Once the packages are imported by add-in, it creates index as "Keywords" in the "Tag and Keywords" for the automatic search.

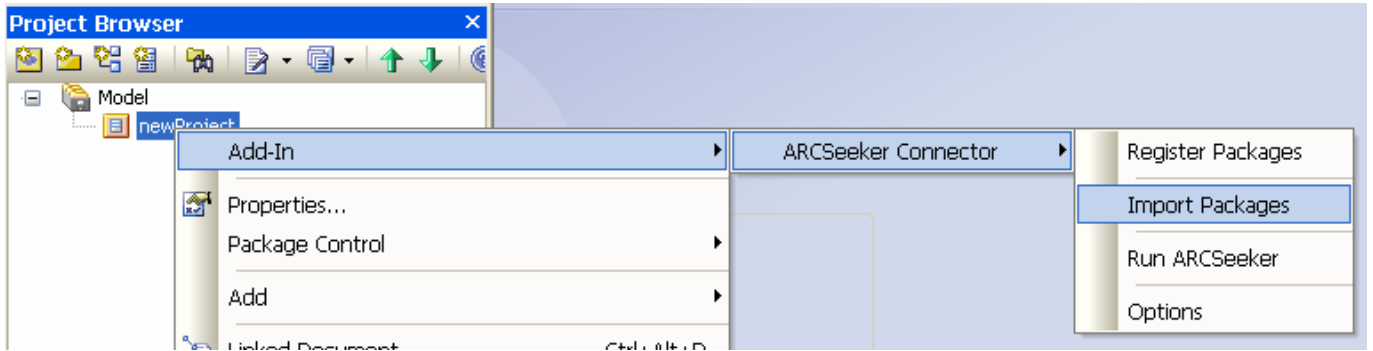


When executing the search in ARCSeeker, above function allows searching the information including element's name of UML models, notes, and so on. This is something that Enterprise Architect cannot do on its own, but the unique advantage of ARCSeeker.

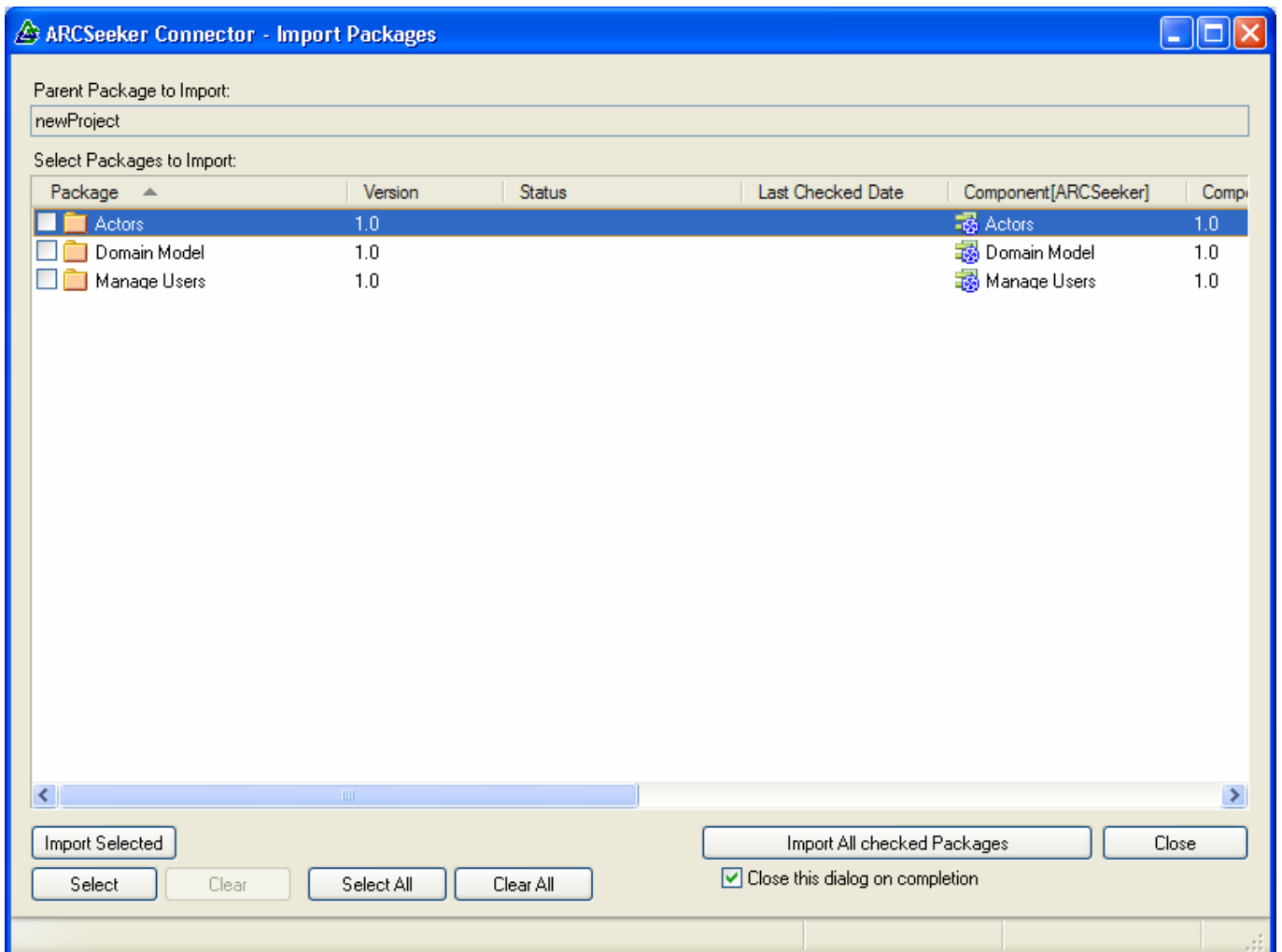
6. Usage

Using the stored or classified components is very easy.

In the project browser of Enterprise Architect, right click the parent package, and select "Add-In," "ARCSeeker Connector," and "Import Packages."



The importable components are shown by list. Select a component from the list. When there are components imported previously, it shows if the components are updated since the import.



Select all the objective packages, and execute importing. If there is dependency to other components, the information of those components are shown in the confirmation screen before importing.

By importing the related components at the same time, it shows the contents of UML model, reliably.

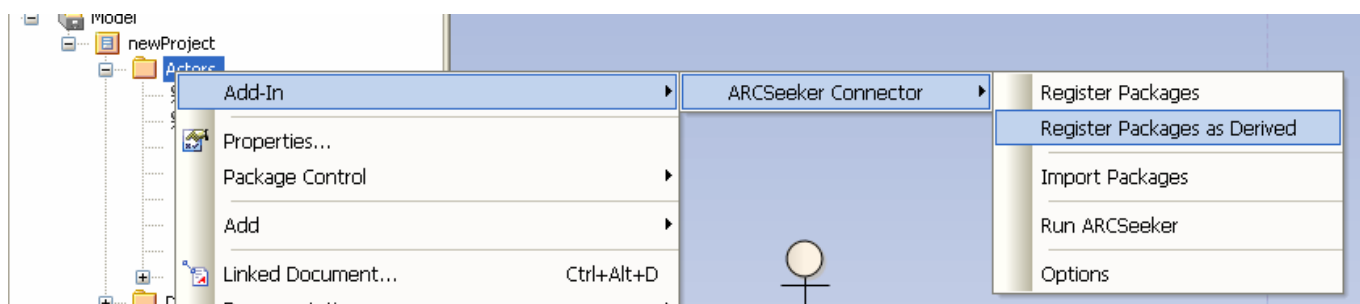
Even if the original package of Enterprise Architect is deleted, the component stored in ARCSeeker remains as-is. If the imported content is unnecessary, delete the package from Enterprise Architect. When you need to refer it later days, you can import the package from ARCSeeker again.

7. Registration of update and derivation

When the component imported from ARCSeeker is used by Enterprise Architect and even the small changes have been made, it needs to be re-registered. There are two ways to register; “normal registration” which overwrites the existing component, and “derived registration” which registers the component as new.

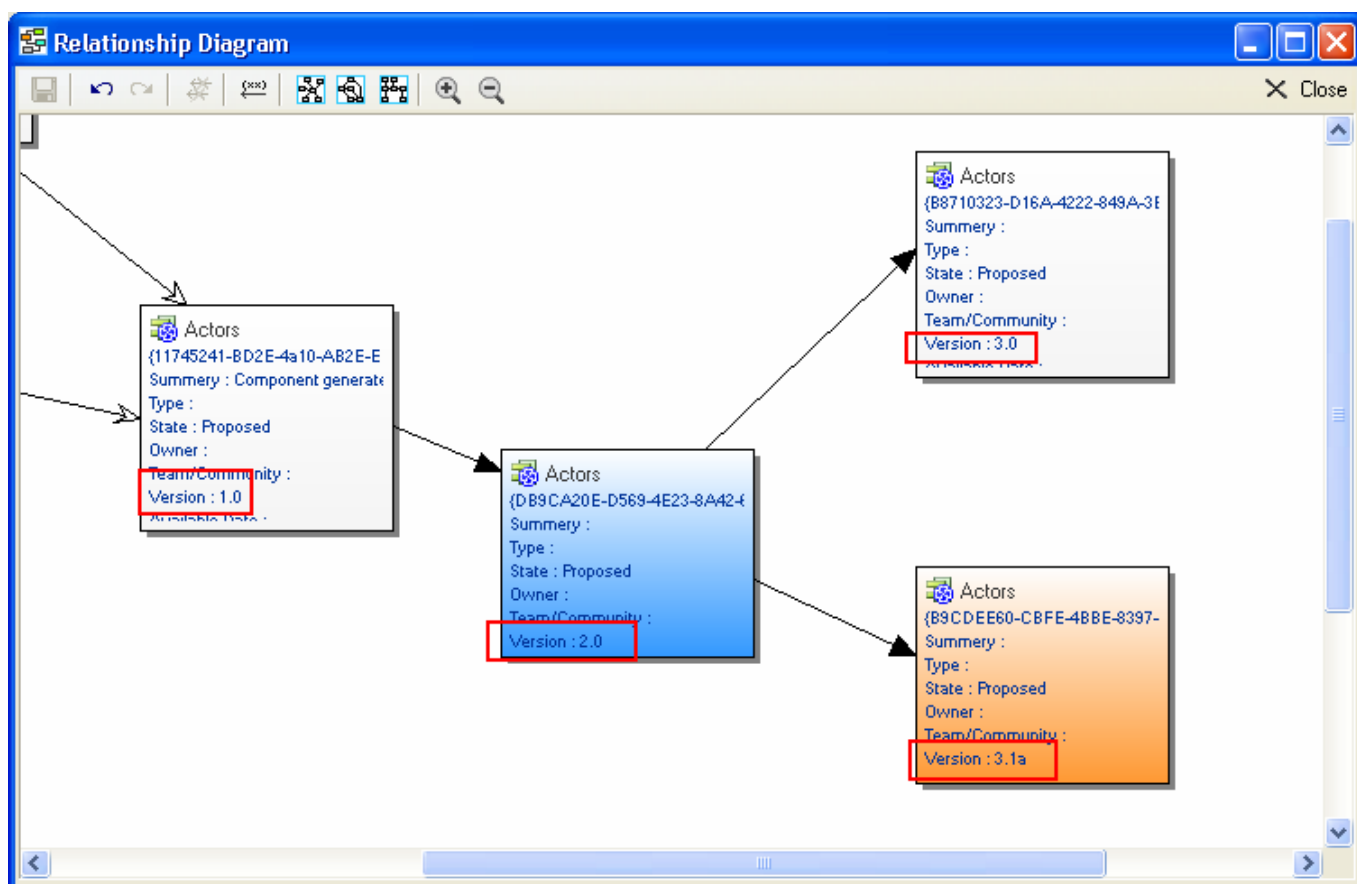
For “normal registration,” see chapter 4. The process is exactly the same.

For “derived registration,” select “Register Packages as Derived” in the sub menu appeared by the right click of objective packages.



When derived registration is executed, ARCSeeker’s component is newly added so that the multiple components exist with the same names on those. We recommend to distinct those by version number.

When the components are created by derivation, the relationship diagram looks like the below image. It figures out the derivation relationship. This also enables to track down the problems of the past components and its influences.



(Version information is displayed in red boxes.)

8. Summary

The UML models created by Enterprise Architect are not simply applied into the designs of other systems, and also, it does not seem realistic. However, a closer look at the UML models in small units shows that there is appropriable information or reference information of model creation. Realizing the day-to-day small increase of efficiency is an idea of “information recycle/reuse.” (Please see a free brochure for the information recycle/reuse.)

When starting to implement the information recycle/reuse, it takes a lot of effort to check the contents of Enterprise Architect project files if the data is simply saved in Enterprise Architect. Storing the project files as parts and adding a lot of information for search, ARCSeeker helps the efficient consultation.

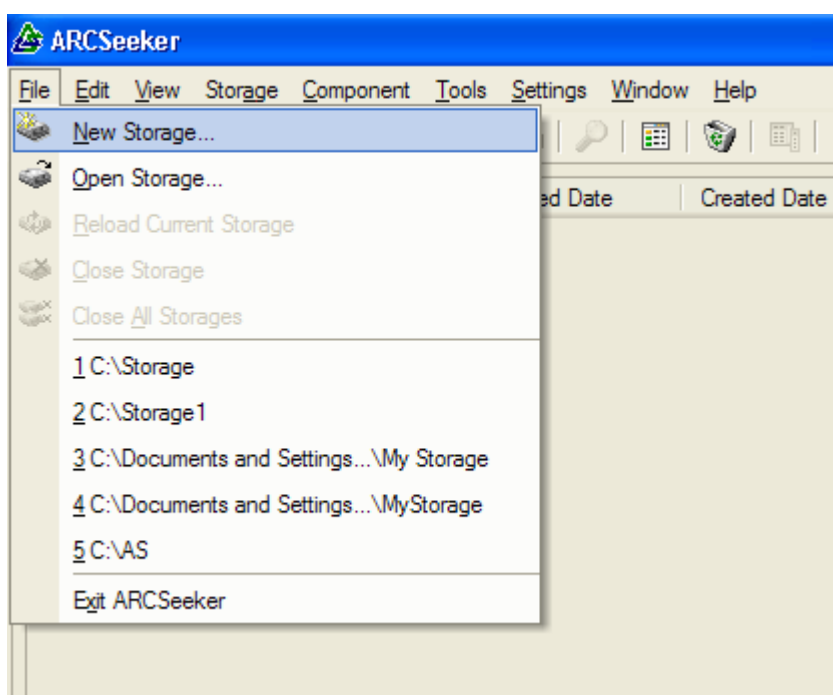
We recommend the immediate implementation of ARCSeeker, right after the first completion of design by Enterprise Architect. Many people await implementing ARCSeeker until the applicable models are collected on some level. It is one of the points of success that storing the information, components, right away and continuing to seek the best way of storing those information. At the beginning, only a few people use the system, so it is a good chance of defining the storage structure, tag contents, and so on.

9. Settings of Interaction between ARCSeeker and Enterprise Architect

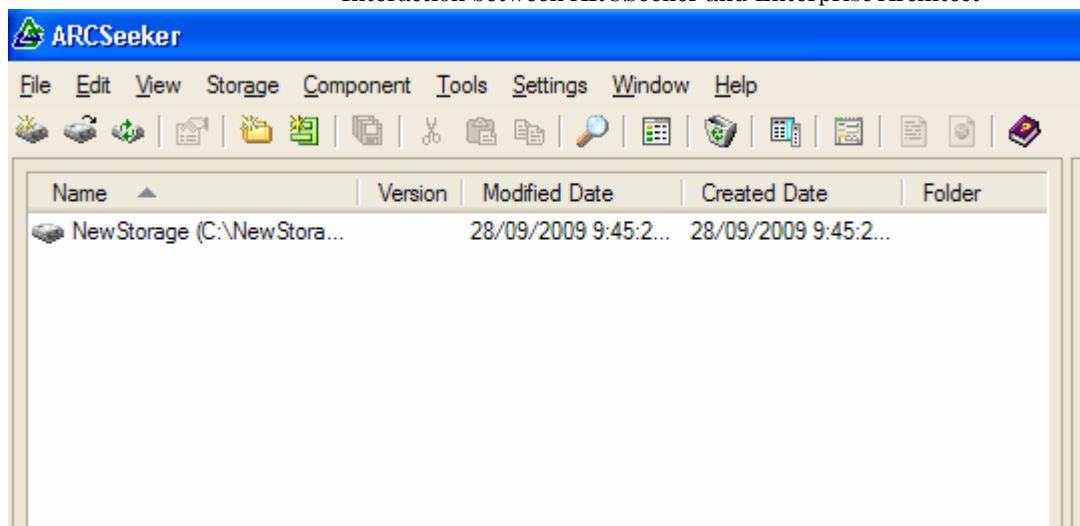
In this chapter, a storage setting is described. Before importing the existing Enterprise Architect information into ARCSeeker by the connector add-in, it is convenient to set the storage of ARCSeeker for leverage the interaction.

(Please refer to the ARCSeeker operation guide or help file for the operation of ARCSeeker.)

Run ARCSeeker. Select “File” from the main menu, and “New Storage” to create storage. When you use the existing storage, open the storage.



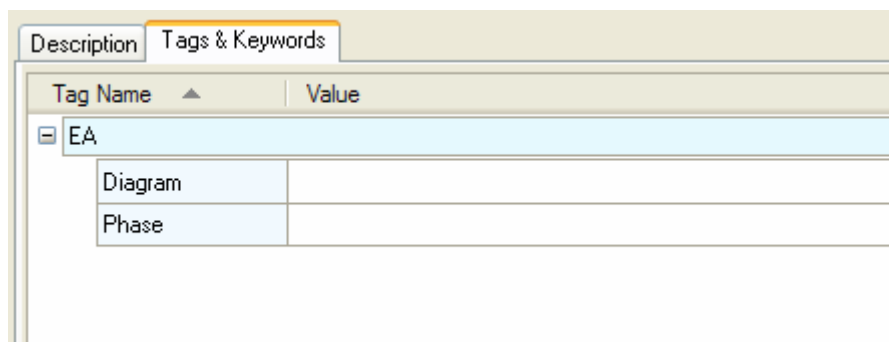
Double click the “Storage” icon from the “Component List” in the left to show the property in the right.



In this storage's property, set the items in "Tags & Keywords" to be able to specify the unique additional information when importing the Enterprise Architect packages.

Edit the storage group name as "EA" for this setting. At the creation of new component, the tag information defined in this "EA" group become available for users to set the value of the tag.

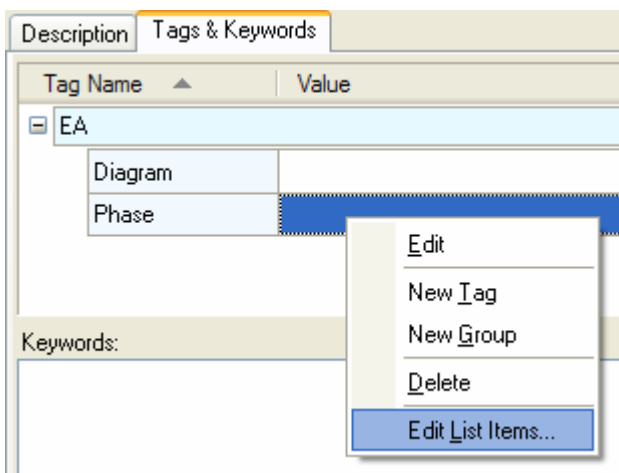
To create a tag group in ARCSeeker, right click the blank field of the "Tags & Keywords" and select "New Group." To add a tag, right click the group name field and select "New Tag."



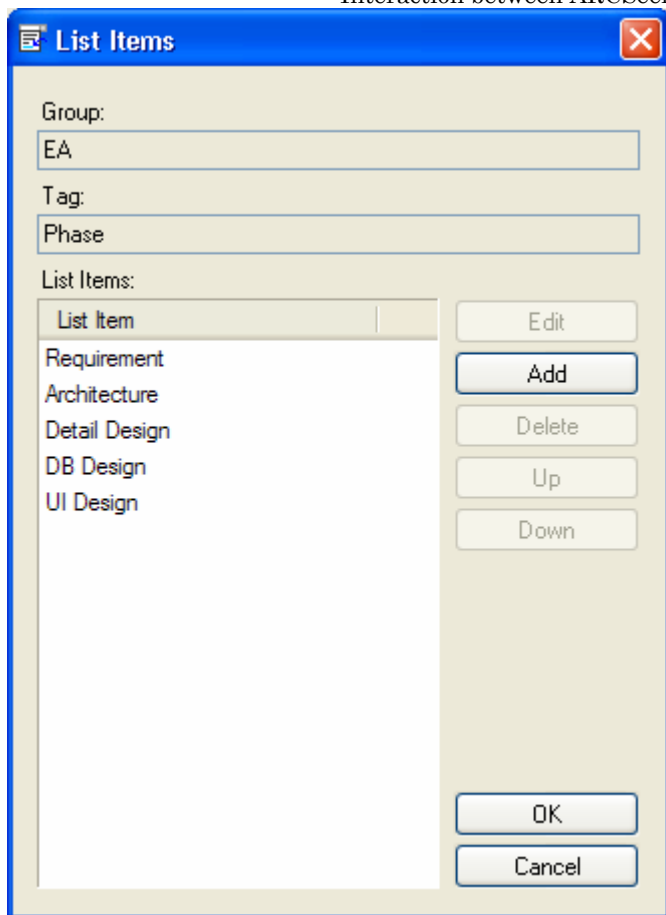
For each item, you can define default and alternative (enumerated value). Enter the value in the above tag to specify the default. Defined value of storage tag is used as "default value."

Tag Name	Value
EA	
Diagram	
Phase	
TagWithDefault	DefaultValue

“List Items” enables to set choices of value to be used at the component creation. Right click the tag to show the context menu, and select “Edit List Items.”



Then, following dialog is appeared. The items added here will be displayed as alternatives when the component is created from the package. As an example, following items and alternatives are created.



In this example, “Phase” shows the designing level of deliverable, and “Diagram” shows the mainly included diagrams. Defining the “useful items for a later reference” simplifies the distinction, classification, and search of components.

ARCSeeker uniquely define and establish the information for easy sorting when importing the Enterprise Architect packages. It depends on the contents of the importing Enterprise Architect project for what kind of items should be added, so it is difficult to define exclusively. According to the circumstances, “Programming Language” or “Customer Name” might be the adequate items.

Fundamental settings of ARCSeeker are completed now. With these settings, tag-added value can be defined by the following dialog at the Enterprise Architect UML model importing.

