

**DT ALM**

APPLICATION LIFECYCLE MANAGEMENT SOLUTIONS



# DT ALM Newsletter

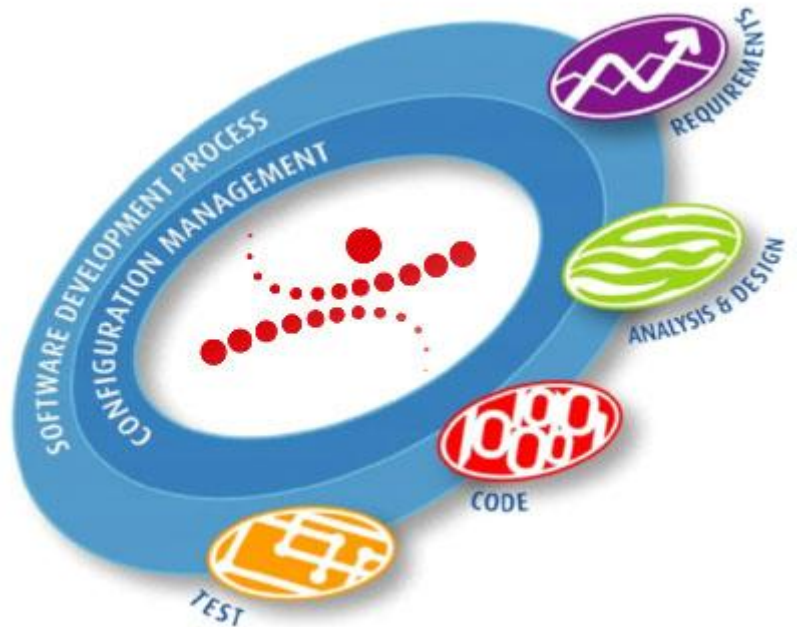
## October 20087

### Issue 4

---

### Introduction

Following on from issue 3's version control, we now look at how you can use Enterprise Architect to manage changes within your organisation. Because of Enterprise Architect's flexibility, there is more than one way of achieving this. Here we will look at three methodologies and their relative merits.



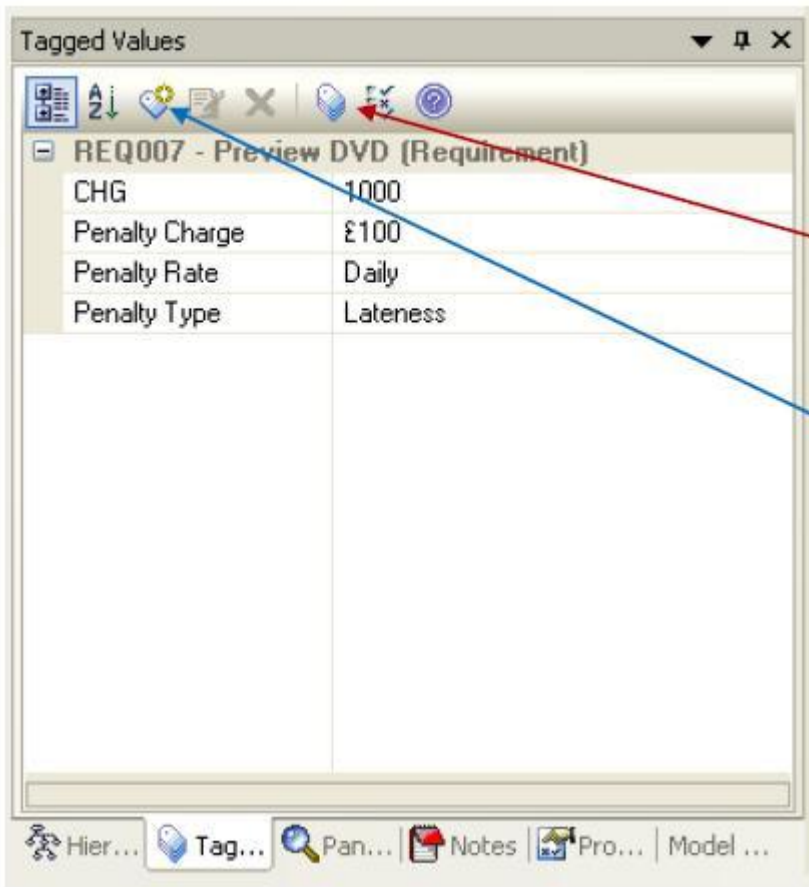
---

## Change Management in Enterprise Architect

### Tags

If your organisation already has a change management system, rather than migrating it to Enterprise Architect you might wish to simply reference the information within the relevant entities.

If your model already has full traceability then the most obvious place to store the tags would be in the requirement entities. With the exception of bug-fixes, any changes to your system should involve a change to a requirement somewhere. Therefore the amount of additional work required to reference your change record should be negligible as the associated requirements would normally need to be modified anyway.

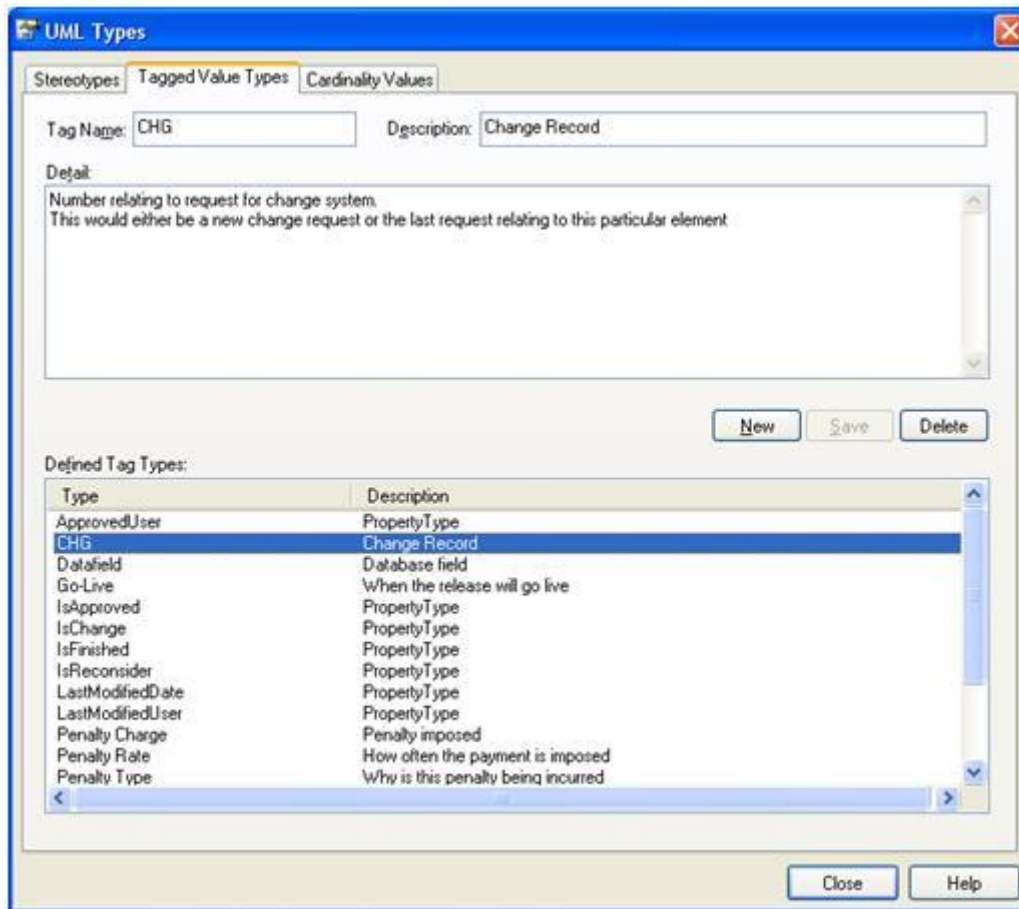


Edit Tagged Value Types

Add a tag to this element

The fictional requirement shown above has several tags. The “CHG” tag refers to its associated change record. Tags can be added to any entity by clicking on the <New Tag> button highlighted with the blue arrow.

You can create your own custom tags. Simply open up the Tagged Value Types tab either by selecting <Settings|UML> from the drop-down menu or by clicking the button highlighted with the red arrow.



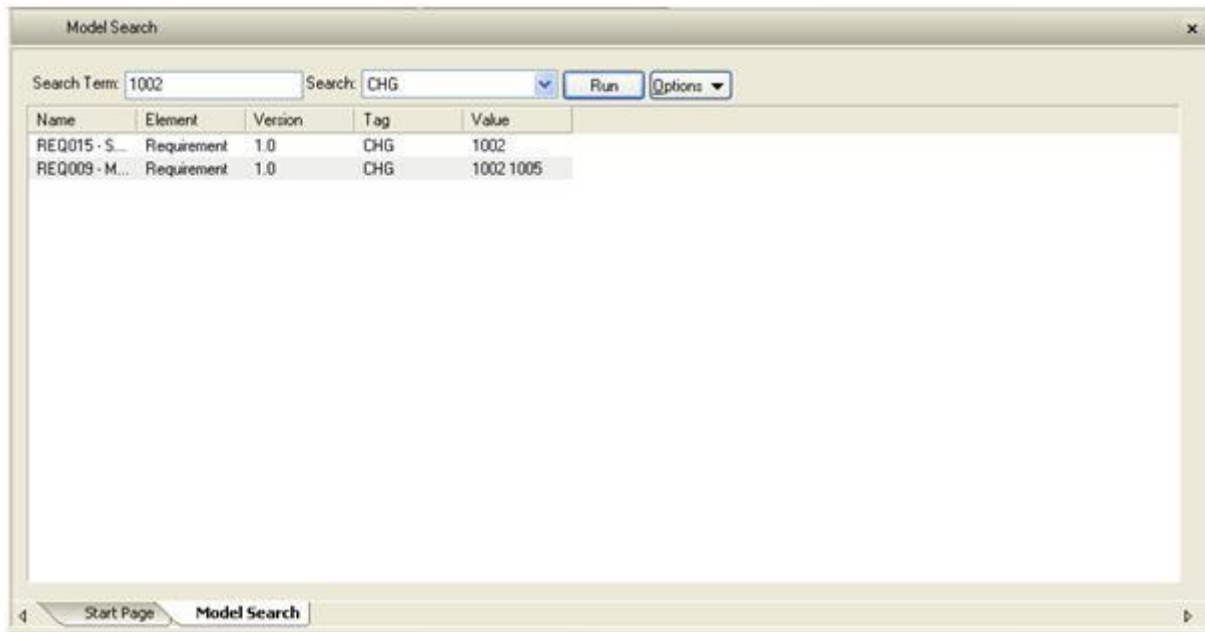
Then create a new tag by entering the appropriate details and clicking <Save>. Ensure you have clicked <New> beforehand else you will overwrite an existing tag.

Unfortunately the trade-off for the simplicity of adding tags is the difficulty of finding them within the model. In order to find them you have to write a custom SQL search.

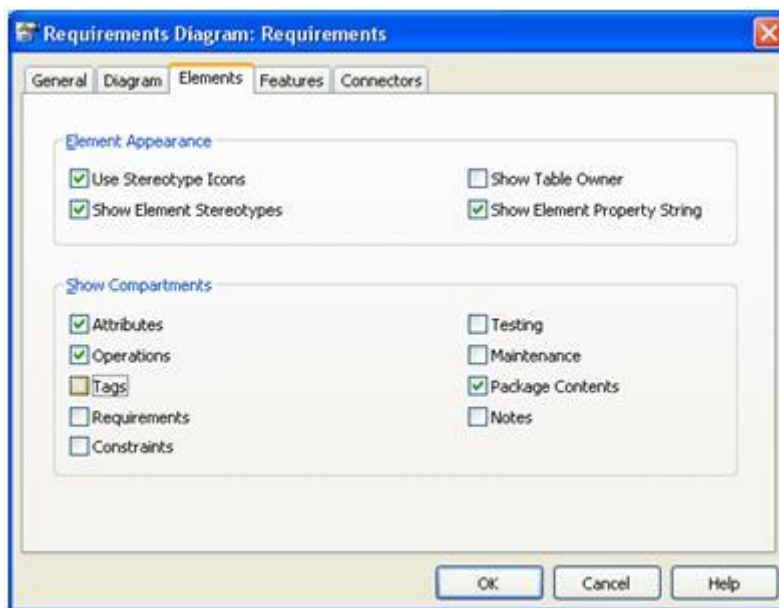
To find specific CHG tags in my model, I entered the following SQL into a custom search:

```
select a.Name, a.Object_Type as Element, a.Version, b.Property as Tag, b.Value from t_object a, t_objectproperties b where a.object_id = b.object_id and b.property = 'CHG' and b.Value like '%<Search Term>%'
```

When 1002 is entered as the search term, I get the following results



Note that REQ009 has more than one change record in its tag but the search has still found it. You can show tags in your diagrams by checking the corresponding option in the diagram's properties.

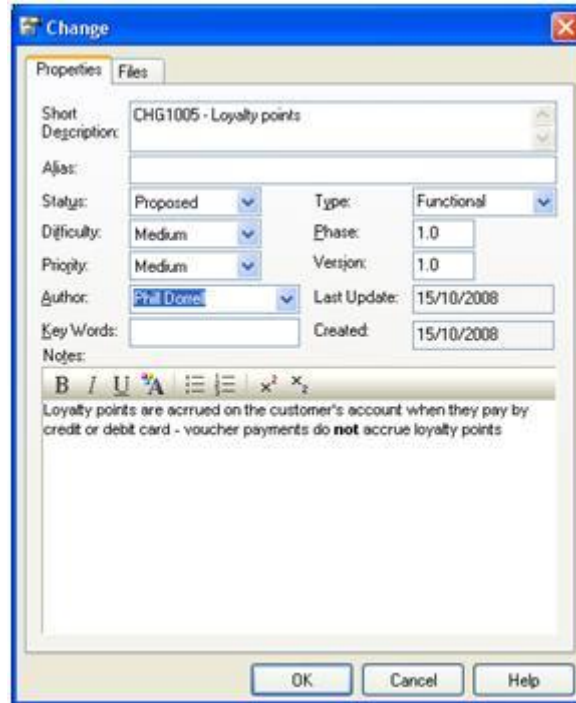


If you already have a change management system and merely want to reference it in your model then tags will satisfy your needs with little effort required. However if you don't have a change management system or you wish to integrate your change management into your model then you would want to use something that enables you to add more detail.

## Change Elements

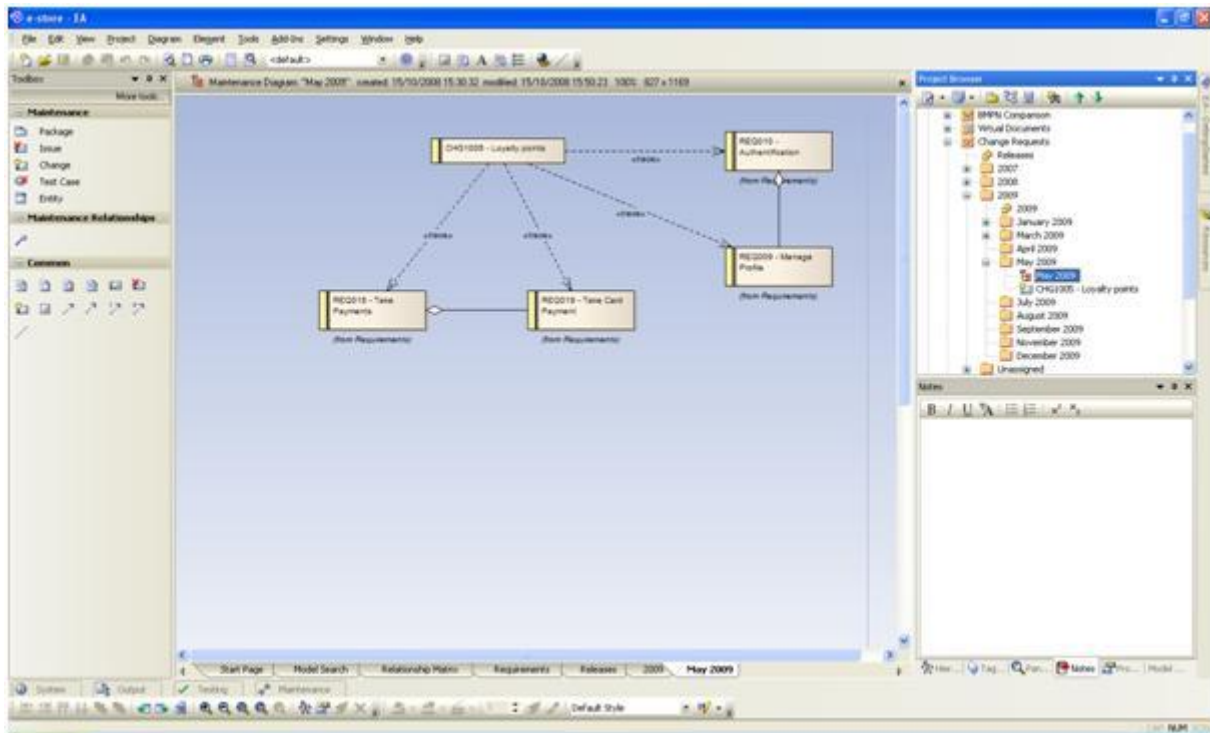
Change elements can offer far more detail than tags. They are similar to requirement elements in terms of their appearance and the

information they can contain.



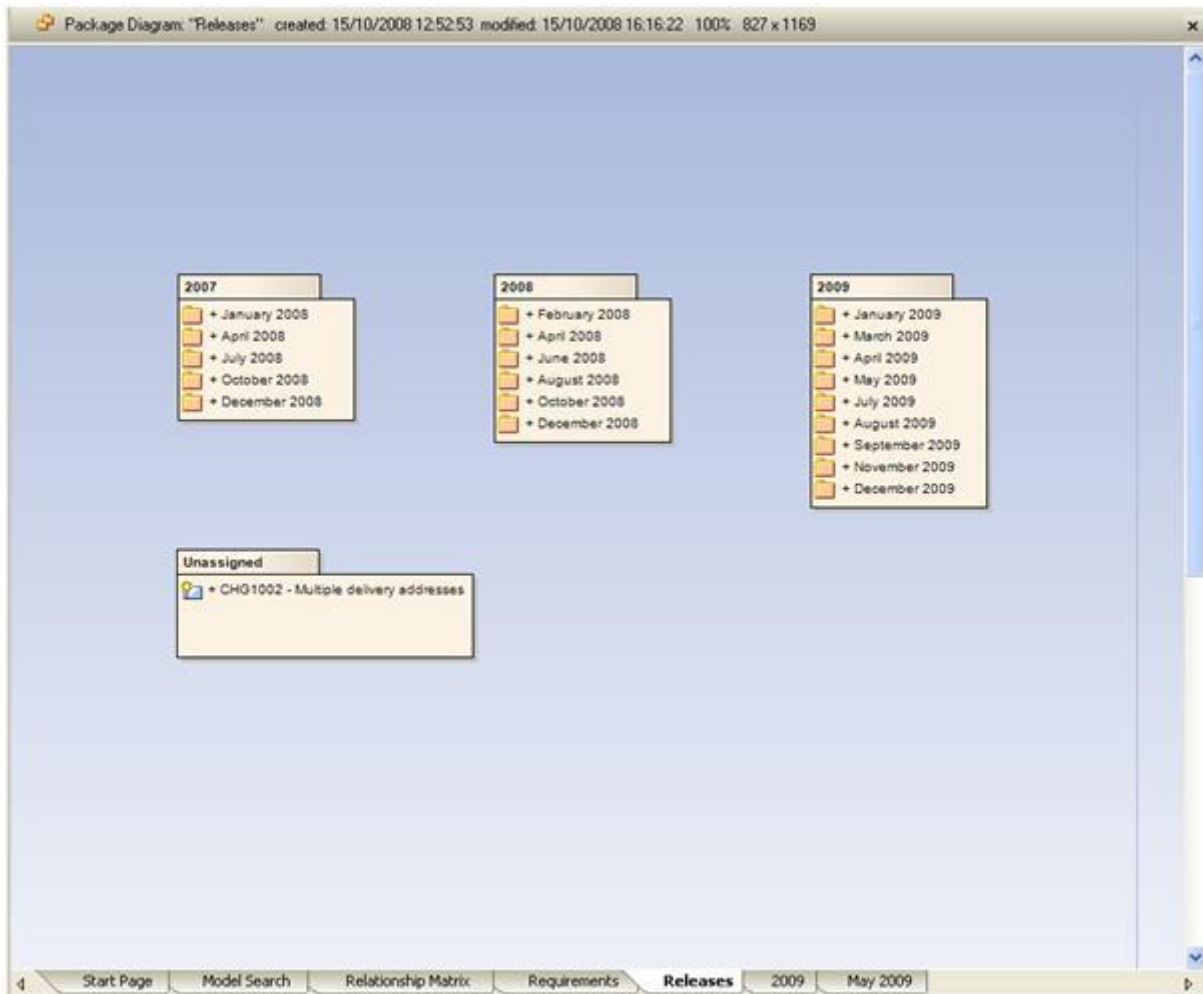
To create change elements, simply right-click on the package you want to add them to then select <Add | Add Element> and specify Change as the element type. If desired, you can set up automatic numbering just as you would for any other element.

Alternatively create a Maintenance Diagram and drag new change elements across from the palette.

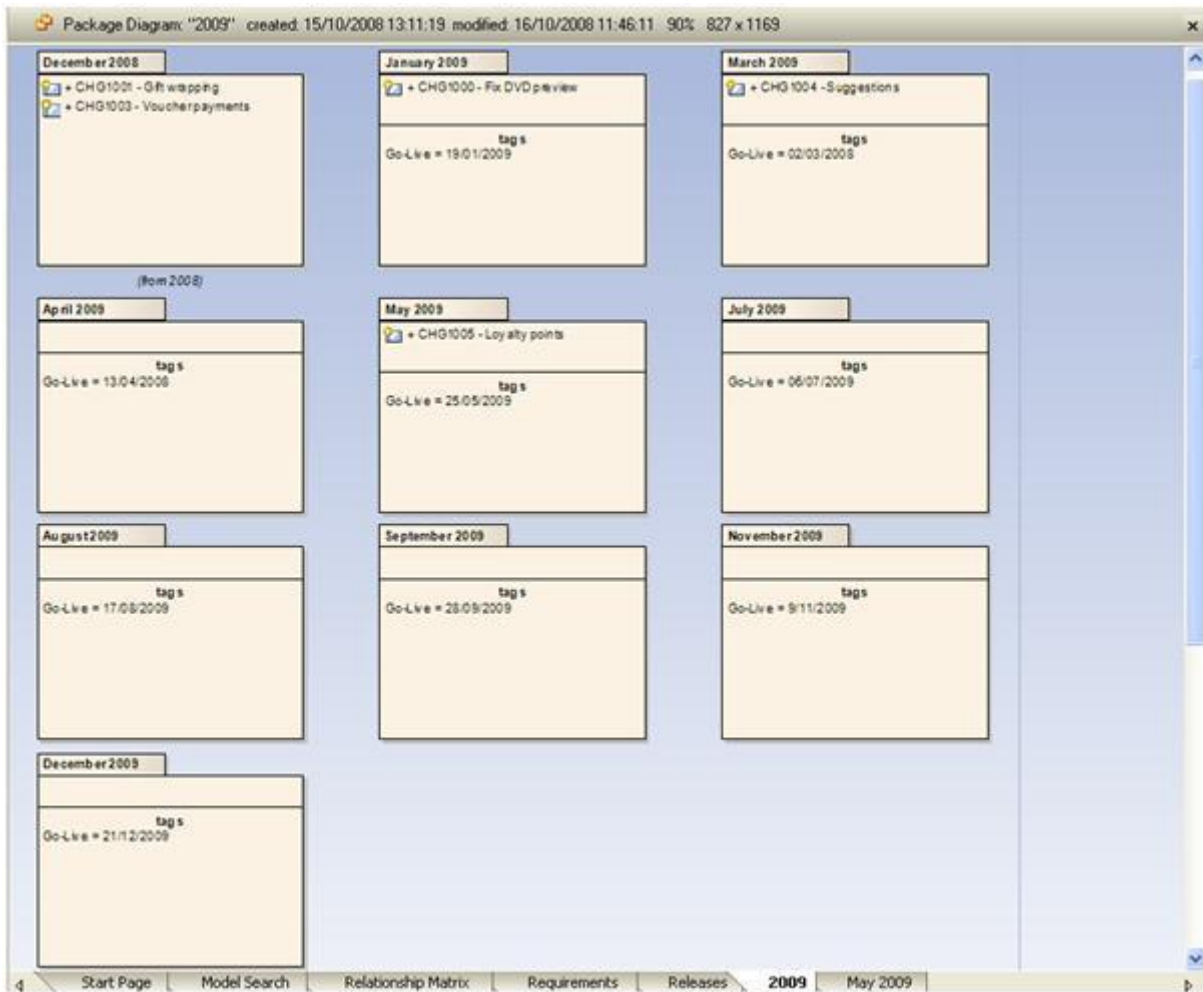


The above diagram represents a fictional online shopping system. The selected diagram shows all the changes for May 2009 and all the requirements the changes are linked to.

If you look at the project browser to the right of the diagram you will see that I have structured the change requests into release packages that are themselves within year packages.



This is the release overview diagram showing all the releases for each year as well as a package containing unassigned changes.



Drilling down into the 2009 package we get an overview of all the changes contained in each release package. Note that I have included the last release from the previous year as a reference and that all 2009 releases have a 'Go-live' tag.

Unlike tags change elements can be profiled in the relationship matrix. By structuring the changes in release packages that in turn are structured into years, you can easily configure the relationship matrix for all the changes in a given release; a particular year or the entire model. You can then save the profile so that it can be referred to again with ease.

Relationship Matrix: All Changes vs Reqs

Source: May 2009    Type: Change    Link Type: Dependency    Profile: May 2009 Changes vs Reqs

Target: Requirements    Type: Requirement    Direction: Source -> Target    Refresh    Options

	REQ001 - Internal authentication	REQ002 - Create profile	REQ003 - Preview items	REQ004 - Modify shopping cart	REQ005 - Preview CD	REQ006 - Preview book	REQ007 - Browse catalogue	REQ007 - Preview DVD	REQ008 - Search catalogue	REQ009 - Manage Profile	REQ010 - Authentication	REQ011 - Add item	REQ012 - Remove item	REQ013 - Change quantity	REQ014 - System quantity over	REQ015 - Ship order	REQ016 - Maintain catalogue	REQ017 - Offer gift wrapping	REQ018 - Take Payments	REQ019 - Take Card Payment	REQ020 - Take Voucher Payment	REQ021 - Automatic product s
CHG1005 - Loyalty points									X	X									X	X		

Changes in May 2009

Relationship Matrix: 2009 Changes vs Requirements

Source: 2009    Type: Change    Link Type: Dependency    Profile: 2009 Changes vs Requirements

Target: Requirements    Type: Requirement    Direction: Source -> Target    Refresh    Options

	REQ001 - Internal authentication	REQ002 - Create profile	REQ003 - Preview items	REQ004 - Modify shopping cart	REQ005 - Preview CD	REQ006 - Preview book	REQ007 - Browse catalogue	REQ007 - Preview DVD	REQ008 - Search catalogue	REQ009 - Manage Profile	REQ010 - Authentication	REQ011 - Add item	REQ012 - Remove item	REQ013 - Change quantity	REQ014 - System quantity over	REQ015 - Ship order	REQ016 - Maintain catalogue	REQ017 - Offer gift wrapping	REQ018 - Take Payments	REQ019 - Take Card Payment	REQ020 - Take Voucher Payment	REQ021 - Automatic product s
CHG1000 - Fix DVD preview								X														
CHG1004 - Suggestions							X															X
CHG1005 - Loyalty points									X	X									X	X		

Changes in 2009

Relationship Matrix: All Changes vs Reqs

Source: Change Requests    Type: Change    Link Type: Dependency    Profile: All Changes vs Reqs

Target: Requirements    Type: Requirement    Direction: Source → Target    Refresh    Options

	REQ001 - Internal authentication	REQ002 - Create profile	REQ003 - Preview items	REQ004 - Modify shopping cart	REQ005 - Preview CD	REQ006 - Preview book	REQ007 - Browse catalogue	REQ007 - Preview DVD	REQ008 - Search catalogue	REQ009 - Manage Profile	REQ010 - Authentication	REQ011 - Add item	REQ012 - Remove item	REQ013 - Change quantity	REQ014 - System quantity over	REQ015 - Ship order	REQ016 - Maintain catalogue	REQ017 - Offer gift wrapping	REQ018 - Take Payments	REQ019 - Take Card Payment	REQ020 - Take Voucher Payment	REQ021 - Automatic product s
CHG1000 - Fix DVD preview								X														
CHG1001 - Gift wrapping																		X				
CHG1002 - Multiple delivery ...									X						X							
CHG1003 - Voucher payments																		X		X		
CHG1004 - Suggestions						X																X
CHG1005 - Loyalty points									X	X								X	X			

### All changes

The benefits of using Change elements instead of tags are clear as you can specify more information, visually illustrate the change and easily modify which model elements they link to.

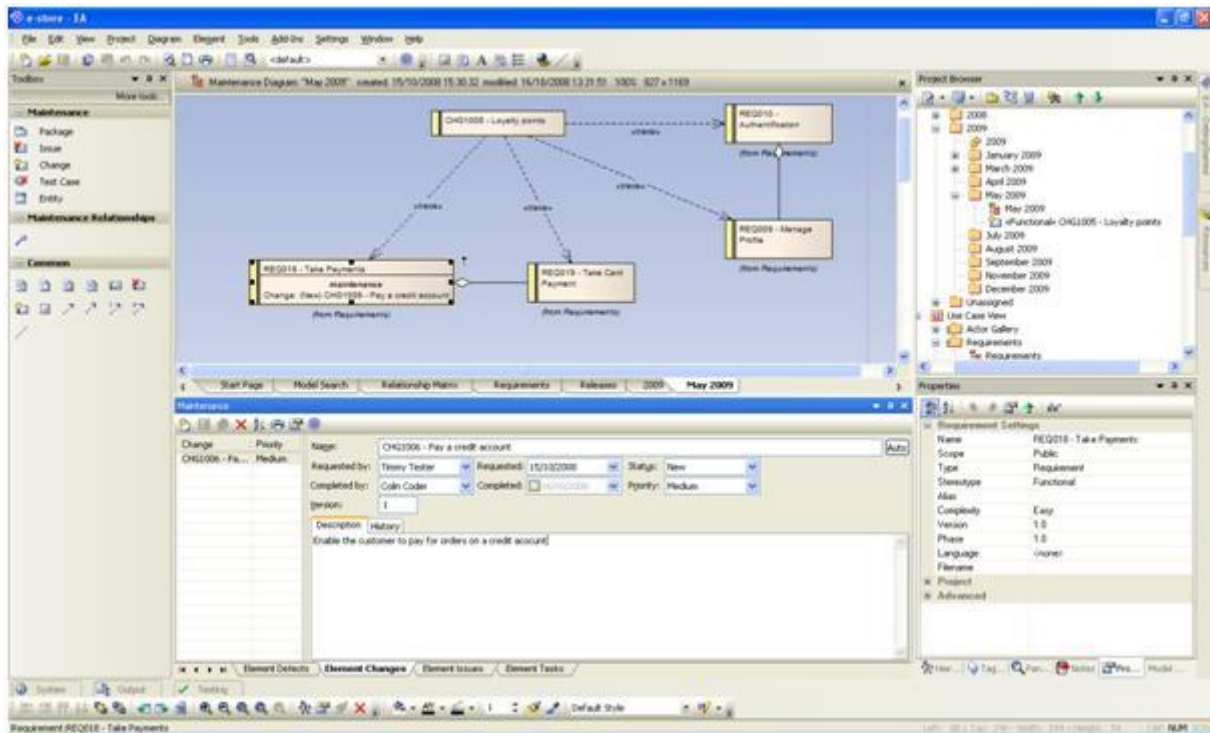
However they do require a bit more effort than simply adding a tag to a requirement.

## Element Maintenance

Every element in Enterprise Architect can have maintenance records recorded against them. There are four categories within the maintenance view:

- Element Defects
- Element Changes
- Element Issues
- Element Tasks

We will be looking at element changes. The maintenance view defaults to a pop up window at the bottom of the screen but it can be moved if desired.

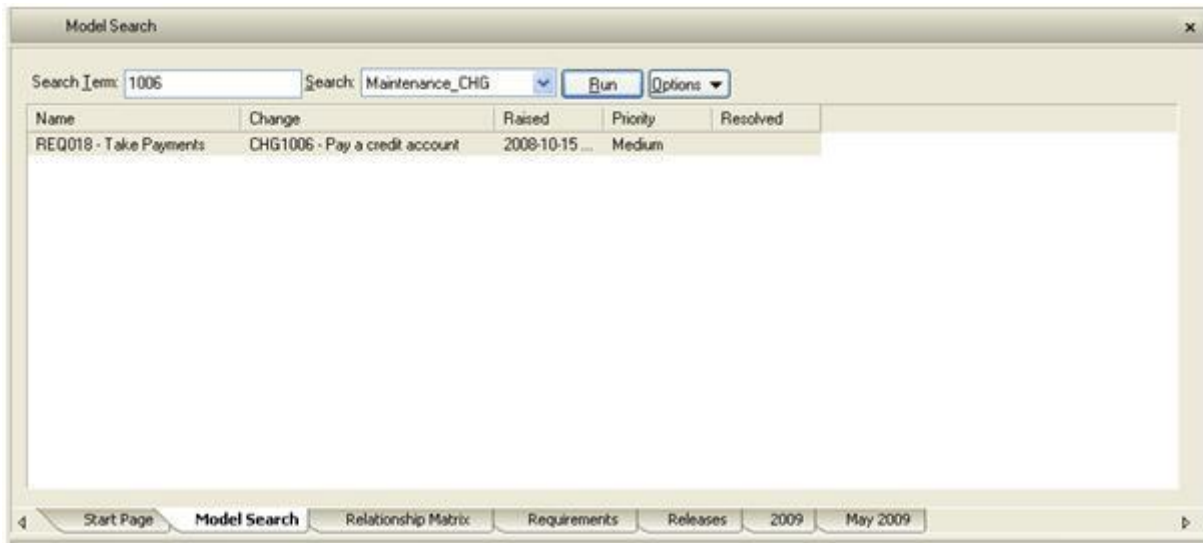


In the above diagram we can see that an Element Change maintenance record has been created for “REQ018 – Take Payments”. Note that I have set the diagram to show maintenance details in the same way that you can show tags.

Whilst the amount information we can specify for a maintenance change is useful (Who requested it; who completed it; when it was requested; when it was completed etc.) the record is exclusive to the element it was recorded in. If your change affected more than one element, you would need to perform the same task several times. It is more like an extended version of a tag than the independent change record you would get using change entities. Like tags, finding the information would require a custom SQL search:

```
select a.Name, b.problem as Change, b.dateresolved as Resolved from t_object a, t_objectproblems b
where a.object_id = b.object_id and b.ProblemType = 'Change' and b.problem like '%<Search Term>%'
```

The above SQL gives the following results in a model search:



## Summary

All three methodologies have their relative merits.

Feature	Tags	Change Elements	Maintenance Changes
Low Effort	Yes	No	No
Independent	No	Yes	No
Documented	Yes	Yes	Yes
1 Change Many Reqs	No	Yes	No
1 Req Many Changes	Yes	Yes	Yes
Relationship Matrix	No	Yes	No
Audit History Level	Low	Medium	High

Tags would only be suitable to complement a separate change management system but the other two options could be used as a standalone change management solution. Change elements would be more useful in this scenario as they are independent to the elements they relate to.

As an optimum solution you could combine change elements with either a series of custom tags or a maintenance change record. The tags would be more useful as you could add the information relevant to your business.

If all your change elements need to have the same tags then you could set up a template package. Simply create a package in your model, create a change element within that package and add the tags you want to it. Then go to the drop down menu and select <Settings | Template Package> and select the package containing your templates.

Note that this will then be the template for all elements of the types within that package, so if the package contains a class element, all new classes in your model will take that form.

# DT ALM

APPLICATION LIFECYCLE MANAGEMENT SOLUTIONS

