Four Key Improvements When Managing the Teamwork Server Repository

OVERVIEW OF THE PROJECT USAGE MAP

Saulius Pavalkis
System Analyst for the MagicDraw
saulius.pavalkis@nomagic.com
Contents

Introduction .......................................................................................................................................................... 3
Clear and Valid Teamwork Repository - Complexity Addressed ................................................................. 4
All Cyclic Usages Are Valid ............................................................................................................................ 5
Consistent Version, Branch Usages, and Mount Points ................................................................................ 7
All modules are Required and used .................................................................................................................. 9
Conclusion ......................................................................................................................................................... 9
Abstract: This document details some common undesirable situations users have encountered while working with a Teamwork Server repository containing multiple projects. All of these situations can easily lead to more serious problems such as: data loss, duplicated and inconsistent data, and lost time from cleaning up errors. This paper suggests an easy way to identify and remedy issues in the early stages using the new MagicDraw capability - Project Usage Map.

Keywords: Project Usage Map, Teamwork Server, Modules, Cyclic usages.

Introduction
No Magic’s Teamwork Server is software that allows more than one user to work with the same model. The model is stored centrally in the Teamwork Server repository and every modeler working with either MagicDraw, Cameo Business Modeler, or Cameo Enterprise Architecture may collaborate on the same project.
The Project Usage Map is a live visual graph that represents Teamwork Server project usages as well as identifies potential problem areas.
The Project Usage Map allows for representing projects and their dependencies in two views:

- *All Projects view* that shows all projects and all the dependencies among them.
- *Individual project view* that shows a particular project along with other directly and indirectly used modules.

Using the Project Usage Map you can easily do the following:

- Identify, analyze, and validate dependencies among projects (for example, you can find out easily all the projects, wherein a particular module is used).
- Identify cyclic dependencies among projects.
- Identify and fix inconsistent dependencies among projects.
Clear and Valid Teamwork Repository - Complexity Addressed

Problem
If you are developing a large model that has several dependent parts, it is advisable to split it into several modules (a module is a project which has shared packages which other projects can use). Partitioning enables reusability of components in different projects and may improve performance on very large projects when modules are loaded selectively. Also, different users can work on their own projects which are part of another bigger project. Users can be assigned with different access rights in each module and versions of each module are tracked separately in Teamwork Server. When the number of users grows, the number of projects and usages between them grows too. Large numbers of usages makes it difficult to understand and validate them from a single project perspective.

It becomes important to manage the Teamwork Server repository efficiently. You would like to identify, analyze and validate dependencies between projects. If you have multiple projects in the repository, this task becomes difficult or even impossible without dedicated tools.

Solution
MagicDraw provides the Project Usage Map to identify, analyze and validate dependencies between projects.

Note. You only need to be connected to your Teamwork Server repository from MagicDraw in order to invoke Project Usage Map.
The Project Usage Map shows project usages (i.e. directly and indirectly used projects which are visible from the selected project) in a graph. Graph nodes are individual projects and graph edges are usages. The Usage Map has two views:

- **All Projects view** that shows all projects and all the dependencies among them.
- **Individual project view** that shows a particular project along with other directly and indirectly used modules.

After reviewing the entire repository and identifying problems you can dive in to an individual project view to view its usages in detail.

Both views have the capability to filter information by projects and categories. Your filter settings will be saved so you can continue your analysis next time you launch Project Usage Map.

**All Cyclic Usages Are Valid**

**Problem**
When a project decomposition is used, the project is split into smaller projects. You can benefit from this decomposed project - you need to load only small part of the project instead of all of it. This reduces complexity and even improves performance.

In addition to this, you would not expect and most likely not care to load the remaining parts of the main project each time you open a single part of it. This can happen if you have cyclic
usages i.e. project parts are using the main project. Typically, such usages increase complexity and reduce performance. Figure 3 and Figure 4. Also, if the project takes part in a cycle, it can't be reused as a totally independent part in another project. Other projects taking part in the cycle will be automatically used as well. Cycles are often a symptom of unintentional usages, created unknowingly by the user. They can often cause further problems – such as inconsistent mounting points or module version usage inconsistencies.

Figure 3. Cyclic usages

Figure 4. Cycles through unshared usages
Sharing usage shows project parts which are exposed to other projects. In the above example (depicted in the Figure 4), usages are represented as dashed lines and sharing usages are represented as solid lines. Shared packages of Project C are visible for Project A because project B is using and sharing Project C. But Project B shared packages are not visible for Project D because Project A is only using (but not sharing) Project B. Project A depicted in Figure 4 is a project from whose perspective the cycle exists, i.e. the cycle is formed out of sharing and using and ordinary relations.

**Solution**
The Project Usage Map automatically identifies cycles and highlights them in the Repository View. You can then open a Usage Map for the suspected projects participating in a cycle to analyze it more closely and if necessary - break usages that cause cycles.

**Consistent Version, Branch Usages, and Mount Points**

**Problem**
It is not unusual for one project to be used by multiple other projects if it is a library or profile for example. It is also likely that a single project will use some other projects that already use that same library or profile.
You will therefore get multiple usages of the same project. This is normally not a problem. However several problematic cases can occur:

1. You are using different versions of the same project in your main project (see Figure 6).
2. You are using a version from the trunk and branch of the same project in your main project (see Figure 7).

![Figure 7. Inconsistent project branch usage](image)

3. You are mounting (mount - the other project usage in a particular package of the main project) the used project in different packages in your main project (see Figure 8).

![Figure 8. Inconsistent mount points](image)

All of the above cases are inconsistencies which may be difficult to understand without a specialized means. All of them could cause user confusion, loss of time and even lost data in some cases (careless editing in case of read/write modules).

**Solution**
The Project Usage Map highlights these inconsistencies. You can then open projects with inconsistent usages and fix them by unifying the used project version, branch, or mounted package information.
All modules are Required and used

Problem
When the number of projects in the repository grows it is common for some of the projects to become outdated or not used anymore. You would prefer removing them BUT you are not sure if they are not used by some other, still-active project.

Solution
The Project Usage Map highlights unused modules. Based on this information, you can move all of them into the deprecated category or remove them entirely from the repository.

Conclusion
1. We conducted an overview of potential repository problems which usually occur when the number of projects and users working with them grows large. Potential problems:
   - Unclear usages between the projects
   - Cyclic usages
   - Inconsistent version, branch usages, and mount points
   - Unused modules
   - Not confirmed usages
2. We introduced solutions to resolve these problems
3. We did a basic overview of the means to address these problems - Project Usage Map
4. We showed that the Project Usage Map provides the ability to identify, analyze and validate relations between projects quickly.

Let's imagine you have analyzed the Project Usage Map results. They are graphical and visual, however, you would like to save them not only as a picture, but also into model form - to be able to create your own validation rules to check the usages and their properties. Also, you may want to compare the usages you have now with the ones you will have in the NEXT two months. Or additionally, you may wish to compare an actual and a typical map to find inconsistencies. In order to accomplish this, use action to export the Project Usage Map directly to the model and everything is saved.

Figure 11. Export to model

References

[1] Project Usage Map online demo is available at http://www.nomagic.com/support/demos.html


Note: Project Usage Map is v17.0.3 capability.
About the Author

Saulius Pavalkis
System Analyst for the MagicDraw
saulius.pavalkis@nomagic.com

Keep in touch!

Contacts

No Magic Europe, UAB
Savanoriuave, 363, Kaunas
LT - 49425, Lithuania
Phone: +370-37-324032,
Fax: +370-37-320670
E-mail: sales@nomagic.com

No Magic, Inc.
One Allen Center, 700
Central Expressway South,
Suite 110 Allen, TX 75013
Phone: +1-214-291-9100,
Fax: +1-214-291-9099
E-mail: sales@nomagic.com

No Magic Asia
719 KPN Tower, 22nd floor,
Rama IX Road, Bangkapi, Huaykwang,
Bangkok 10310, Thailand
Phone: +66 2717 0250
Fax: +66 2717 0251
E-mail: nomagicth@nomagicasia.com