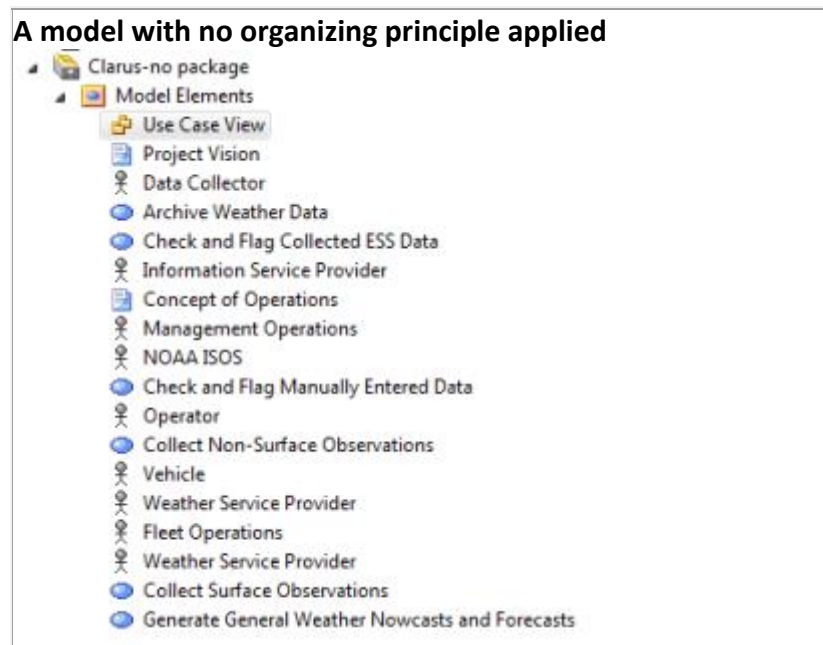


# UML Model Organization with Packages and the Package Diagram

J.D. Baker, 

It is no coincidence that the package element in UML is represented by a folder icon, similar to directories in a file system Graphical User Interface (GUI). Packages are used to organize the elements of a model just as folders are used to organize files. The contents of a package are any kind of element that is part of a model, including other packages. Beyond that description there's not much more information on how to use packages in the UML specification. Since they are a general grouping thing, it is up to usage scenarios to suggest what the best practices are for using packages. There are some practices that apply in most situations, and more that depend upon the modeling methodology or the purpose of the model.

First and foremost, use packages to organize your model, regardless of the model purpose. Even the most trivial model quickly becomes unmanageable without some kind of organization. There is no single correct way to organize a model. A modeler may choose to organize around process or product or some combination of the two. Second, use the package diagram to provide visualizations of that organization. The model browser view below illustrates how even a simple model can begin to be confusing with no organization of the model elements.



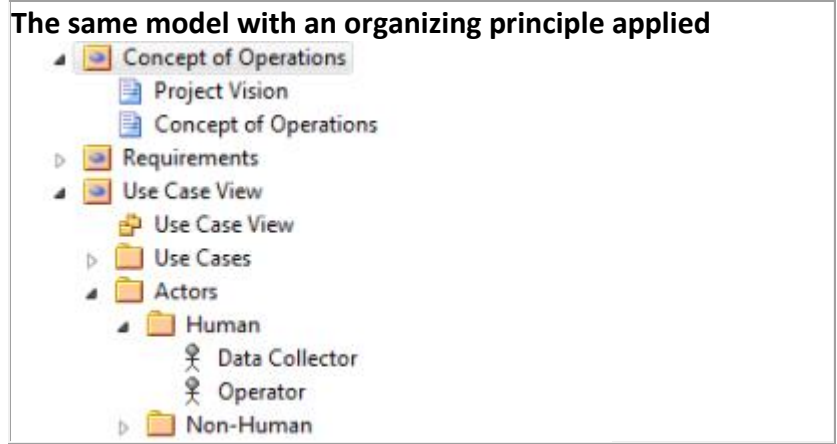


Figure 1. Models without and with organizing packages

In addition to being able to gather like items together to allow modelers to focus their attention on relevant elements and make those elements easier to find in the browser, packages provide a namespace for the elements they contain. Namespaces allow modelers to create unambiguous named references to each of the elements in a model. This is useful in situations such as when a modeler is evolving a system and creating an as-is and a to-be system model. It is natural to have elements named the same in each model. In order to disambiguate elements with the same name, they can be placed in different packages so that the fully-qualified names are different.

Packages can be placed on a UML package diagram. The modeler might choose to do this to present a high-level view of a model or to show relationships among the packages of elements. Three different package diagrams representing the model organization shown above are presented here. Each of these diagrams uses the optional diagram frame, showing the diagram type (pkg) and diagram name (Use Case View). Note that although the diagram and the package have the same name (Use Case View) this is not required. In the first diagram (Figure 2), the two high-level packages are shown. As with many other modeling situations, only the details we need are presented, and the details of the package contents are elided.

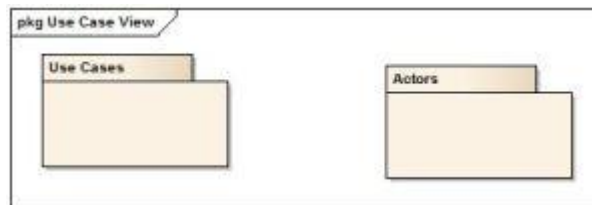


Figure 2. Example package diagram

In the second diagram (Figure 3), the packages contained in the Actors package are shown. The labels on the Human and Non-Human package describe the namespace of the containing package. The containment relationship also serves to describe the namespace. In this

example, the fully qualified name of the Actor Operator is *Use Case View::Actors::Human::Operator*.

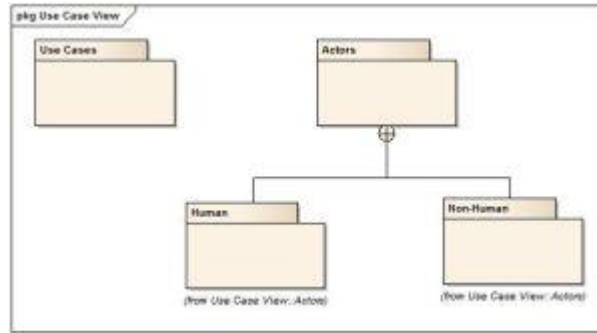


Figure 3. A package diagram showing package contents with a containment relationship

In the third diagram (Figure 4), the contents of the packages are shown embedded in the package elements. This representation is the third presentation option found in the UML specification.

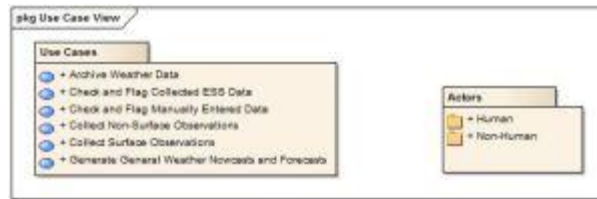


Figure 4. A package diagram showing package contents embedded

Your package diagrams will in all likelihood be some combination of each of these styles, as will your choice of organizing principles. Choose the one that is appropriate for its intended use.

If you want to print this article there is a PDF attachment on the Sparx Community site –You can also provide comments there.

### [Model Organization with Packages](#)