

HOW VISIO 2003 CAN INFLUENCE INFORMATION TECHNOLOGY – CREATING UML MODELS

Adrian LUPAȘC

*Department of Accounting and Economic Informatics
University "Dunărea de Jos" of Galați
e-mail: alupasc@ugal.ro*

Developing an overall architecture and design before you begin to write code helps you produce a software system that meets requirements and is easier to develop and maintain in the long run. As you model a software system, you progressively develop the detail of that system, alternately decomposing high-level objectives and broad requirements into manageable pieces, and then assembling software components into packages and eventually a complete run-time system. Models and diagrams make it easier to visualize both high-level architecture and low-level components, so you can make the most of design opportunities or spot potential problems before you write code.

Keywords: *UML, Visio, diagram, class, object.*

Introduction

The Unified Modeling Language (*UML*) defines diagrams and the semantics behind them for modeling software systems through each phase of the software development life cycle, as specified by the Object Management Group, which is the not-for-profit consortium that produces and maintains specifications for interoperable enterprise applications. In *Visio* Professional, you can build object-oriented models using the *UML* Model Diagram template, which contains stencils and shapes that use the *UML* notation and support the different types of diagrams used with *UML*. One *Visio* drawing file can contain all the models and diagrams for one software system. You can build and work on models in the hierarchical view provided by the *UML* Model Explorer window or on diagram drawing pages. In addition, the *UML* Model Explorer window is a handy tool for navigating to the diagram or component you want.

In addition to constructing models for new software systems, the *UML* Model Diagram template can help you model existing systems. You can reverse engineer projects created in several Microsoft programming languages and generate *UML* static structure models for those projects.

In this work, you are introduced to the templates and stencils that *Visio* offers for *UML*, and you

learn how to use the modeling tools that *Visio* provides for building *UML* models. In addition, you will learn how to work with each type of *UML* diagram that *Visio* offers, and examine the differences between *Visio*'s *UML* shapes and terminology and those used in the current *UML* specification.

Exploring the *UML* Model Diagram Template and the *UML* Menu

Large or complex software systems require a coordinated development approach in which teams can collaborate on analysis and design for each phase of a system life cycle. The *UML* methodology and *Visio* Professional's *UML* Model Diagram template provide tools to model and document each phase of system design and development.

The *UML* Model Diagram template includes stencils for several types of *UML* diagrams you create during a development project. Although this chapter describes only the basic approach for specifying properties, you can specify dozens of properties for most of the shapes on the *UML* stencils to represent the different attributes and conditions that *UML* supports. You can double-click any shape on a drawing page or an element in the Model Explorer window to open its *UML* Properties dialog box. If you're not sure what a

property does, click the Help button in the lower-left corner of the dialog box.

Visio 2003 is far from a perfect solution for producing *UML* models, Its *UML* templates do not fully support the entire *UML* notation and in some cases, its support for *UML* notation does not match the current *UML* specification. *Visio* 2003 still uses nomenclature from earlier *UML* specifications for *UML* diagrams and doesn't support Composite Structure diagrams, Interaction Overview diagrams, Object diagrams, and Timing diagrams for the *UML* 2 specification. However, in many cases, these discrepancies won't prevent you from performing your work.

When you use the *UML* Model Diagram template to create a new *Visio* drawing file, the *UML* menu appears on the *Visio* menu bar. Unlike the Database menu, which appears when you create a database model diagram using the *Visio* Database Model Diagram template, the *UML* menu doesn't include any wizards. However, you can use it to create, view, or modify system elements such as models, packages, stereotypes, and events. You can also display or hide the *UML* windows and specify options for the appearance and behavior of elements in your models and diagrams. Choose *UML* on the menu bar and then choose the following commands on the *UML* menu:

⇒ *Models* — Opens the *UML* Models dialog box, in which you can create or modify models for your system.

⇒ *Stereotypes* — Opens the *UML* Stereotypes dialog box, in which you can create or modify stereotypes to extend *UML* functionality.

⇒ *Packages* — Opens the *UML* Packages dialog box, in which you can create or modify packages for your system. This dialog box lists the packages for the different data types within your system, not packages that you create within models.

⇒ *Events* — Opens the *UML* Events dialog box, in which you can create or modify events within any of the packages in your system. This dialog box lists all packages in your system in the Packages list box. To add an event, select the package to which you want to add the event and then click New.

⇒ *View* — Choose Model Explorer, Properties, or Documentation from the View submenu to toggle between hiding and displaying those windows.

⇒ *Options* — Opens the Options dialog box, in which you can specify options for default behavior for shapes, packages, object names, and more in the *UML* Model Diagram template.

If *Visio* encounters an unexpected error from either an internal or external source, your template-specific menus, such as the *UML* menu, can disappear. If *Visio* shuts down unexpectedly, restart it and then choose *Tools* → *Options*. Select the Advanced tab, check the Enable Automation Events check box, and click OK. Save any open drawings, exit and restart *Visio*, and then reopen your *UML* model drawing file.

Working with *UML* Models and the Model Explorer

In *Visio* Professional, you can build *UML* models that act like blueprints for the system you want to develop. Whether you work in the Model Explorer window or with diagrams on drawing pages, you can use *UML* models and diagrams to develop your system from requirements through deployment.

You can represent an entire software system within one *Visio* drawing file, although that file typically contains multiple models and diagrams. The Model Explorer window displays the elements of a system as a hierarchy, including models and packages that you use to organize the system as well as diagrams that present different views of the system. In addition to viewing and navigating an entire system, you can also use the *UML* Model Explorer to create diagrams, add elements, or apply properties to elements.

When you create a *UML* drawing file, *Visio* adds several elements by default. As you work, you can create additional elements or organize them within the model hierarchy.

The hierarchy in the Model Explorer window includes the following elements:

⇒ *System* — the top node of the hierarchy represents the system you're designing or documenting.

⇒ Static Model — by default, *Visio* creates a static model, which contains all the packages, elements, and diagrams for a software system model.

⇒ Top Package — by default, for every static model in the system, *Visio* creates a top package, which contains all the elements and diagrams for that model. By *right-clicking* the top package, you can display a shortcut menu with commands for adding new elements to the package, renaming the package, or specifying its properties.

⇒ Package — you can create additional packages within a model to organize your model elements, including sub-packages nested underneath higher-level packages. The shortcut menu for packages includes commands for adding new elements to the package, duplicating, renaming, or deleting the package, or specifying its properties.

⇒ Static Structure-**n** — by default, for each model in the system, *Visio* creates a Static Structure diagram within the top package. This diagram corresponds to a drawing page of the same name in the *Visio* drawing file.

⇒ Elements — within a package, you can create additional elements, including subsystems, classes, interfaces, data types, actors, and use cases.

⇒ Data Types — by default, *Visio* includes packages for common data types, including *C#*, *C++*, *IDL*, and *Visual Basic*. You can also create packages for your own data types.

Using the *UML* Model Windows

The *UML* Model Diagram template includes several windows that make it easy to view and modify *UML* model elements and to navigate within the models and diagrams for your system:

➔ The Model Explorer window — shows all the elements in a software system in a hierarchical tree. You can add elements or access their properties from this window.

➔ The Properties window — shows the main properties associated with the selected element. The properties are for reference only. To edit element properties, *right-click* the element in the Model Explorer window and choose Properties from the shortcut menu. Although this window does not appear by default, you can open it by choosing *UML* → *View* → *Properties*.

➔ The Documentation window — displays the documentation tagged value of the element you select on the drawing page or in the Model Explorer window. You can add documentation to the selected element by typing text in this window. Although this window does not appear by default, you can open it by choosing *UML* → *View* → *Documentation*.

When you open a new *UML* drawing file, the Model Explorer window appears by default and docks on the left side of the screen. You can hide or show each of the *UML* windows by choosing *UML* → *View* and then choosing the name of the window you want to hide or display. Although the Properties and Documentation windows don't appear by default, *Visio* anchors them within the Model Explorer window when you display them. When more than one window is open at a time, *Visio* adds tabs for each window so you can switch among them. You can also float each window individually by *right-clicking* its tab and choosing Float Window from the shortcut menu. If you want to float the entire window with all the *UML* windows in it, drag the window title bar to a new location.

Organizing Models with Packages

For large or complex software models, you can subdivide your models and diagrams into more manageable pieces by using packages in the *Visio* Model Explorer. Each model you create includes a top package, which contains all the elements, packages, and diagrams you create within that model. The top package in a model is like a big suitcase you pack for a trip. You can create additional packages within the top package to organize elements into smaller groups, as you might pack your travel toiletries in a smaller container within your suitcase.

Each element in a model belongs to only one package. However, you can add any kind of model element to a package, including other packages, so you can partition the elements in your model any way you want. You can even use packages to partition diagrams when they become unwieldy. Use the following methods to package the contents of your model:

● Create a package in a model — *right-click* a package in the Model Explorer window and

choose **New** → **Package** from the shortcut menu. *Visio* adds a package icon to the model.

- Add a package to a diagram — display the diagram drawing page you want to package. Drag the Package shape from the *UML* Static Structure, *UML* Use Case, *UML* Component, or *UML* Deployment stencil onto the drawing page. *Visio* adds a package icon to the model, with the diagram within the package.
- Create a diagram from a package — to automatically create a new diagram when you add a package to your model, choose *UML* → *Options* and select the *UML* Add-on tab. Make sure the Create a Diagram Page When a Package or Subsystem Shape Is Added to a Document check box is checked.
- Show package contents on a diagram — in the Model Explorer window, right click the package whose contents you want to show on a diagram and choose the type of diagram you want from the shortcut menu. *Visio* displays the stencil for that type of diagram and opens a blank drawing page. Drag shapes for the elements you want to include in the package onto the drawing page.
- Partition a diagram — in the Model Explorer window, create multiple diagrams within one package. You can drag the elements you want to include onto as many of the diagrams in the package as you want. Each instance refers to the same element in the *UML* model. To view all the references to an element on diagrams, *right-click* the element in the Model Explorer window and choose **Views** from the shortcut menu.

Specifying *UML* Options

You can specify options to control the behavior of the *UML* Model Diagram template. For example, you can choose whether to use the lollipop or class like version of an Interface shape, or whether to prompt before deleting a model element when you delete a shape on a drawing page. To specify the options for the *UML* Model Diagram template, choose *UML* → *Options* to open the *UML* Options dialog box. Select the *UML* Add-on tab and then specify one or more of the following options:

- Shape Ctrl-Drag Behavior — to duplicate the *UML* element the shape represents, select **Copy Object** in the drop-down list. To create a view of the element the shape represents so that you can

drag the shape to another package or diagram, choose **Copy Object View** in the drop-down list.

- Create a Diagram Page When a Package or Subsystem Shape Is Added to a Document — check this check box to create a diagram page for every package or subsystem shape you add to a model.
- Create Watermark on Drawing Page — check this check box to display a watermark that identifies the type of diagram the drawing page represents.
- Prompt for Model Element(s) Delete on Delete of Shape(s) — check this check box if you want *Visio* to ask you if you want to delete the model element that a shape represents when you delete the shape on a drawing page.
- Delete Connectors When Deleting Shapes — check this check box to delete the connectors glued to a shape when you delete the shape.
- On Drop of an Interface from the Model Explorer — select an option to specify the default style you want to use for Interface shapes. If you want to show attributes on an Interface shape, select the **Class-Like Interface Shape** option.
- Auto Assign Name to Newly Created *UML* Model Element — check this check box if you want *Visio* to generate names for elements you create. You can rename the elements after you create them.

Creating *UML* Models

When you create a new *Visio* drawing file with the *UML* Model Diagram template, *Visio* opens the stencils for each type of *UML* diagram, docks the Model Explorer window on the left side of the screen, and adds the *UML* menu to the *Visio* menu bar. In addition, the *UML* Model Diagram template creates several default model elements, including the Static Structure-1 diagram and a corresponding drawing page by the same name. *UML* diagrams use a letter-size drawing page with portrait orientation and no drawing scale. To create a new *UML* drawing file, choose **File** → **New** → **Software** → **UML Model Diagram**.

You can create more than one model for a software system. To create a new model within the *Visio* drawing file, follow these steps:

1. Choose *UML* → *Models*.

2. In the *UML* Models dialog box, click a blank cell in the model column and type the name of the model.

3. To specify the properties for the model, click Properties in the *UML* Models dialog box. Select a category and specify the properties within that category. Click OK when you're done.

4. To add a diagram to the model, *right-click* a package in the Model Explorer window, choose New from the shortcut menu, and then choose the type of *UML* diagram you want to add. You can also *right-click* a drawing page and choose Insert *UML* diagram from the shortcut menu. *Visio* performs the following actions:

➔ creates a blank page named *<diagram>-n*, where *<diagram>* is the name of the type of diagram you are creating and *n* is the next number in the sequence of that type of diagram in the drawing file

➔ displays the new page in the drawing window

➔ brings the *UML* stencil for that type of diagram to the front in the Shapes window

➔ adds an icon for the diagram to the element to which you added it in the Model Explorer window

If you *right-click* a class or use case in the Model Explorer window and choose New from the shortcut menu, you can create diagrams appropriate for the selected element. You can choose from Static Structure Diagram, Activity Diagram, and *Statechart* Diagram for classes; and Activity Diagram and *Statechart* Diagram for use cases.

5. To add an element to the model, in the Model Explorer window, *right-click* a package, choose New from the shortcut menu, and then choose the element you want to add. You can also add subclasses to an existing class by *right-clicking* the class and choosing *New* ➔ *Class* from the shortcut menu.

Working with Static Structure Diagrams

Early in the development life cycle, you create conceptual class diagrams to show the real-world objects represented by your system and the relationships between them. These diagrams help

clarify the terminology used within the context of the system, and the classes of a system, including their operations, attributes, and interrelationships. You can use class diagrams to explore the domain concepts, to analyze system requirements, and to present the detailed design of object-oriented software. As you progress in the development cycle, class diagrams show the software classes that the system implements and how they relate to each other, as shown in *figure 1*. Class diagrams specify attributes, associations, operations, methods, interfaces, and dependencies for the classes in a system.

Creating Static Structure Diagrams

To create a static structure diagram in an existing model, follow these steps:

1. In the Model Explorer window, *right-click* the package to which you want to add the static structure diagram and choose *New* ➔ *Static Structure Diagram* from the shortcut menu.

2. Drag Object or Class shapes onto the drawing page to represent real-world objects or the software classes that implement them in your system.

3. To specify properties for an object or class, double-click the object or class in the Model Explorer window or its corresponding shape on the drawing page. In the *UML* Object Properties dialog box, you can specify attributes, constraints, and tagged values for objects; in the *UML* Class Properties dialog box, you can specify attributes, operations, receptions, template parameters, components, constraints, and tagged values for a class. Click OK when you're done.

When you select the Tagged Values category in the *UML* Object Properties or *UML* Class Properties dialog box, *Visio* displays default tags for the selected element. To add a value to a tag, select the tag you want to modify in the Tags list and type text in the Tag *<tag>* Value box. You can also add additional tags by clicking New.

4. To specify the information you want to display or suppress on shapes, *right-click* each object or class shape and choose Shape Display Options from the shortcut menu.

5. To show relationships between objects and classes, drag Association, Link, Dependency,

Generalization, or Composition shapes onto the

drawing page and glue them to the related objects or classes.

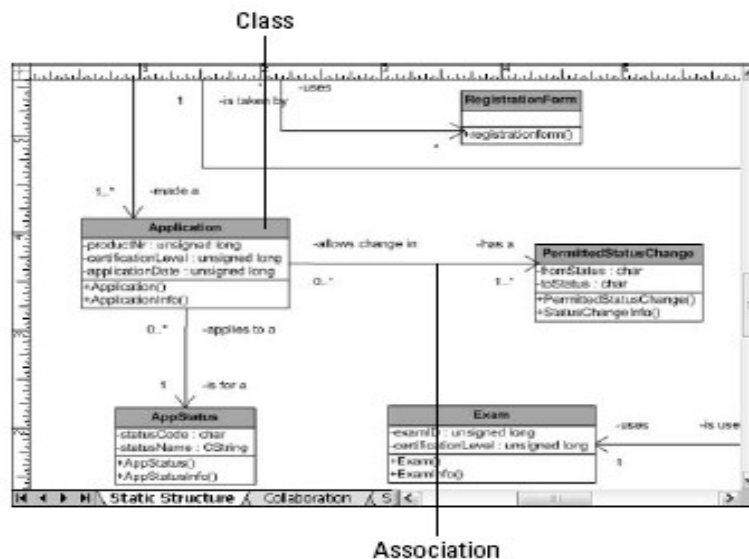


Figure 1: Class static structure diagrams show information about a system's software classes.

6. To specify properties for a relationship, double-click its relationship shape on the drawing page to open the *UML* <relationship> Properties dialog box, specify the properties you want, and click OK.

Conclusion

The *UML* Model Diagram template is available only in *Visio* Professional and *Visio* Studio .NET Enterprise Architect. To model a software system, you can create a single *Visio* drawing file and create the multiple models and diagrams of your system within that file. You can work on the model of your system in the Model Explorer window or by modifying shapes on drawing pages. However, the shapes on drawing pages represent views of the elements in your model. You can add elements to a model without displaying them on diagrams or you can add views of elements to more than one diagram. You

can also reverse engineer existing source code into *Visio* models.

References:

1. Craig Larman – *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process* (2nd Edition), Upper Saddle River, N.J.: Prentice-Hall, 2002;
2. Fowler Boston – *UML Distilled, Third Edition*, Addison Wesley Longman, 2003;
3. Scott Ambler – *The Object Primer, 3rd Edition: Agile Model Driven Development with UML 2.0* New York: Cambridge University Press, 2001;
4. <http://msdn.microsoft.com>;
5. www.agilemodeling.com/artifacts/packageDiagram.htm;
6. www.agilemodeling.com/essays/UMLDiagrams.htm;
7. www.omg.org;