

# Тема 3: Использование Rose в команде

# Where Are We?

- ★◆ Team-based modeling
  - ◆ Controlled Units
  - ◆ Virtual Path Maps
  - ◆ Reuse
  - ◆ Version Control Add-Ins
  - ◆ Model Integrator

# Team-Based Modeling

- ◆ Rational Rose supports
  - **Controlled evolution** of the model.
  - **Partitioning of models** into architecturally significant units.
  - **Reuse** of architecturally significant model elements.

# Controlled Evolution

- ◆ Rose supports architecture-based modeling through the use of UML packages and subsystems.
- ◆ Rose helps users work on low-level design details without affecting the work of others.
  - Specification of architecture-level functionality (interfaces) can be separated from specification of implementation.
- ◆ Rose helps users avoid creating inappropriate dependencies between architectural units.
  - Show Access Violations report supports this.

# Partitioning

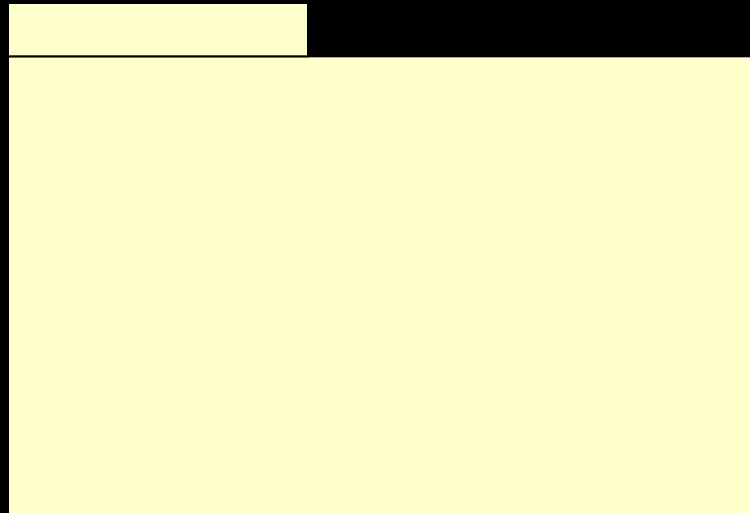
- ◆ In Rose, UML packages/subsystems can be maintained as separate files called controlled units.

# Where Are We?

- ◆ Team-based modeling
- ★◆ **Controlled Units**
- ◆ Virtual Path Maps
- ◆ Reuse
- ◆ Version Control Add-Ins
- ◆ Model Integrator

# Controlled Units

- ◆ A package is the smallest element that can be a controlled unit.
- ◆ In the UML, a package is represented by a file folder.



**Package**

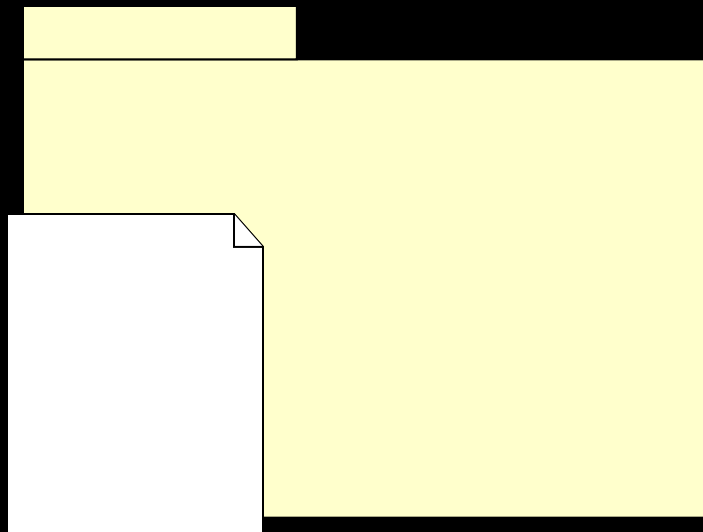
# Controlled Units

- ◆ A controlled unit is a model element that can be placed under version control.
- ◆ The following model elements can be controlled units
  - Model file itself (.mdl file)
  - Logical View and Use-Case View packages (.cat file)
  - Component View packages (.sub file)
  - Deployment View diagram (.prc file)
  - Model properties (.prp file)

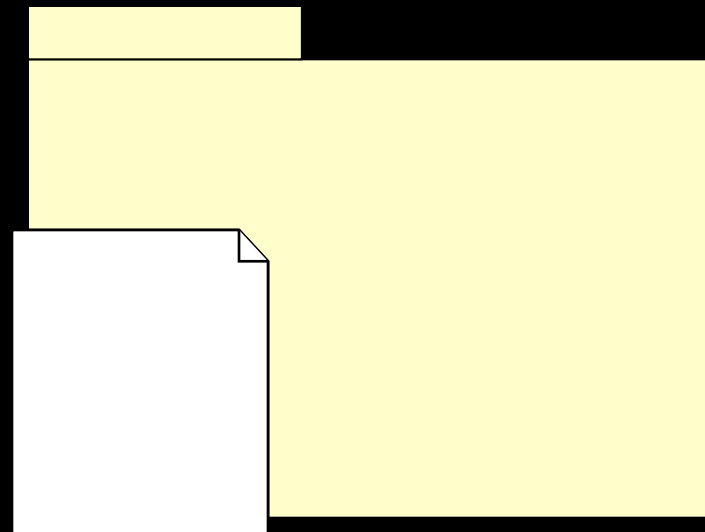


# Controlled Units

- ◆ A controlled unit can be loaded or unloaded. In Rose, a controlled unit is represented in the browser as follows



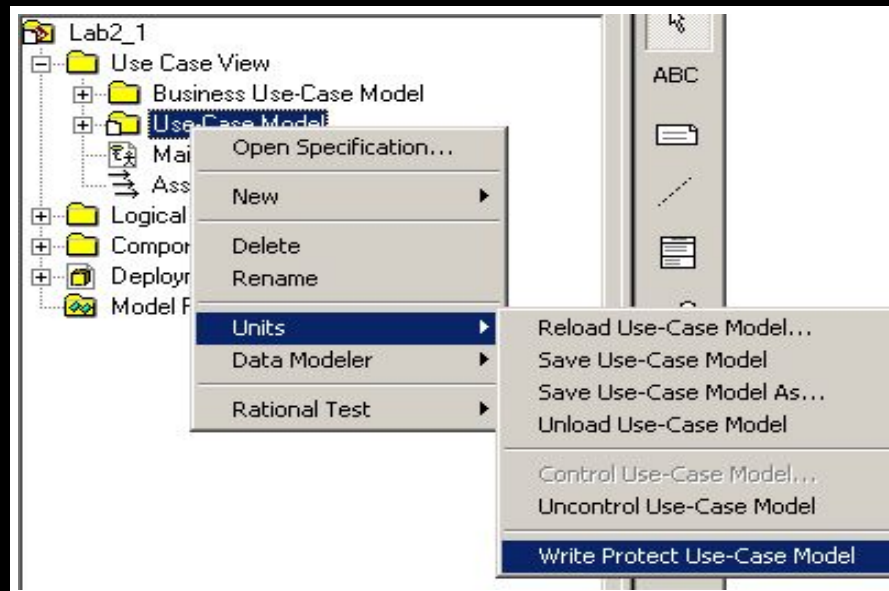
**Loaded Controlled Unit**



**Unloaded Controlled Unit**

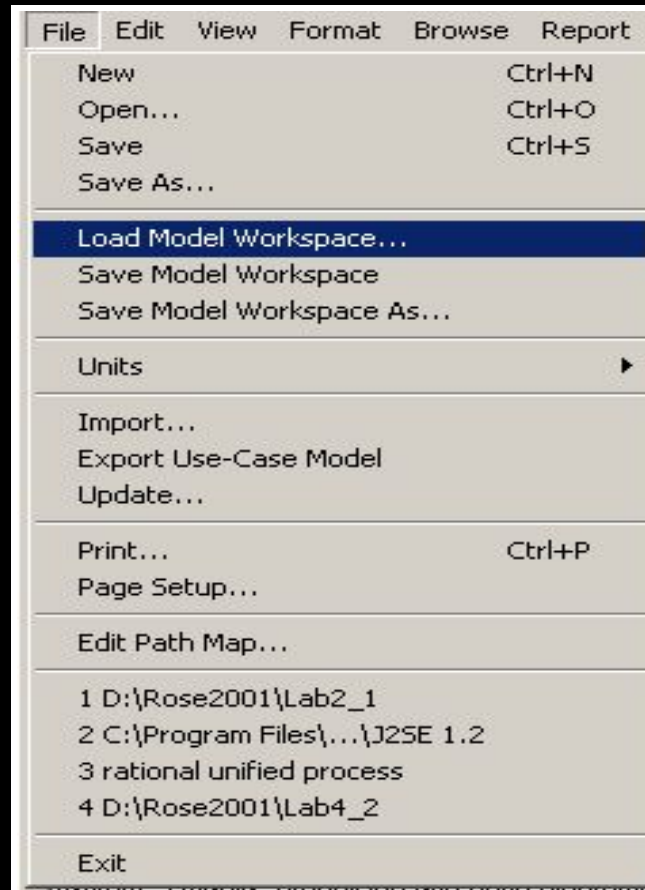
# Controlled Units

- ◆ A controlled unit may be write-protected or write-enabled depending on the file's status in the file system.
- ◆ A controlled unit can also be write-protected or write-enabled manually.



# Controlled Units

- ◆ A model workspace is a snapshot of all currently loaded controlled units and open diagrams.

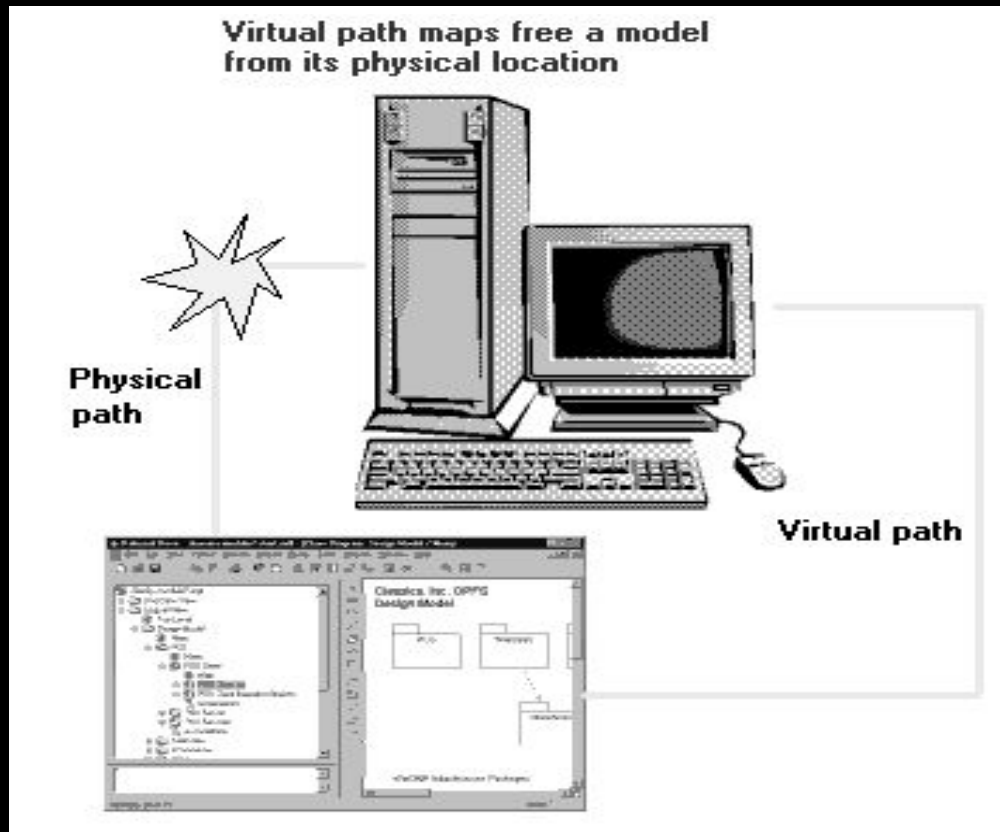


# Where Are We?

- ◆ Team-based modeling
- ◆ Controlled Units
- ★◆ Virtual Path Maps
- ◆ Reuse
- ◆ Version Control Add-Ins
- ◆ Model Integrator

# Virtual Path Maps

- ◆ A virtual path map allows models to be moved between different folder structures and to be updated from different workspaces.



# How do virtual paths work?

- ◆ For example, if a user has defined a virtual path,  
**\$MYPATH=Z:\ordersystem**  
and saves a package as

**Z:\ordersystem\user\_services.cat**

the model file will refer to the package as

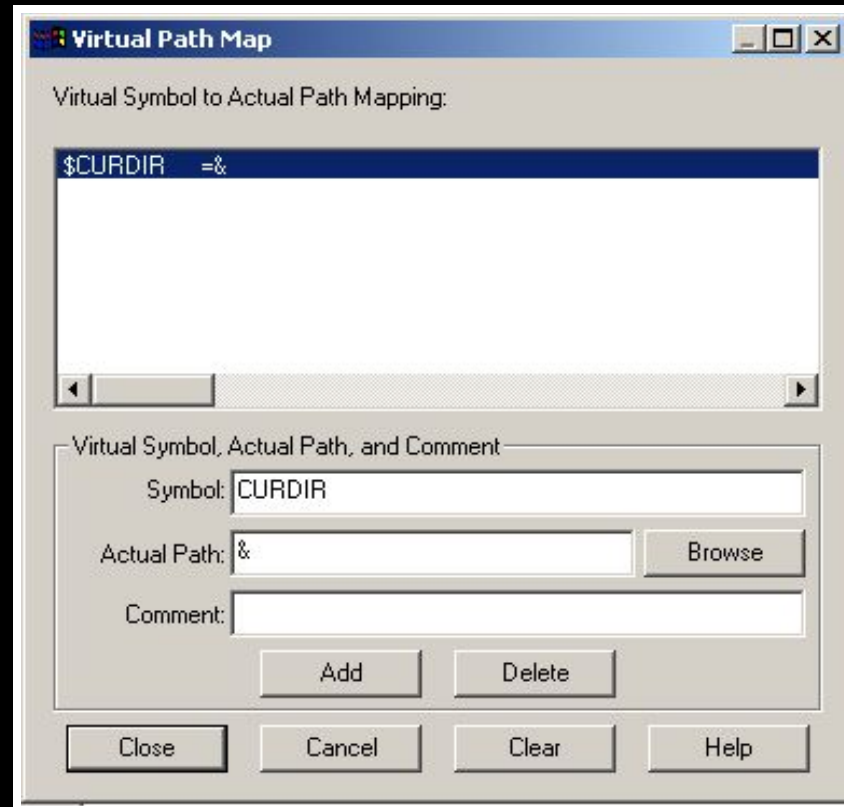
**\$MYPATH\user\_services.cat**

- ◆ When another user, who has defined **\$MYPATH** as  
**\$MYPATH=X:\ordersystem**  
Rational Rose loads the following file:

**X:\ordersystem\user\_services.cat**

# Virtual Path Maps

- ◆ In Rose, the path map reference “&” equals the path to the directory where the current .mdl file or controlled unit is located.



# Review

1. What is a package?
2. What is a controlled unit?
3. Name two model elements that can be controlled units.
4. Identify a loaded and unloaded controlled unit.
5. What is a virtual path map?





# Where Are We?

- ◆ Team-based modeling
- ◆ Controlled Units
- ◆ Virtual Path Maps
- ★◆ Reuse
- ◆ Version Control Add-Ins
- ◆ Model Integrator

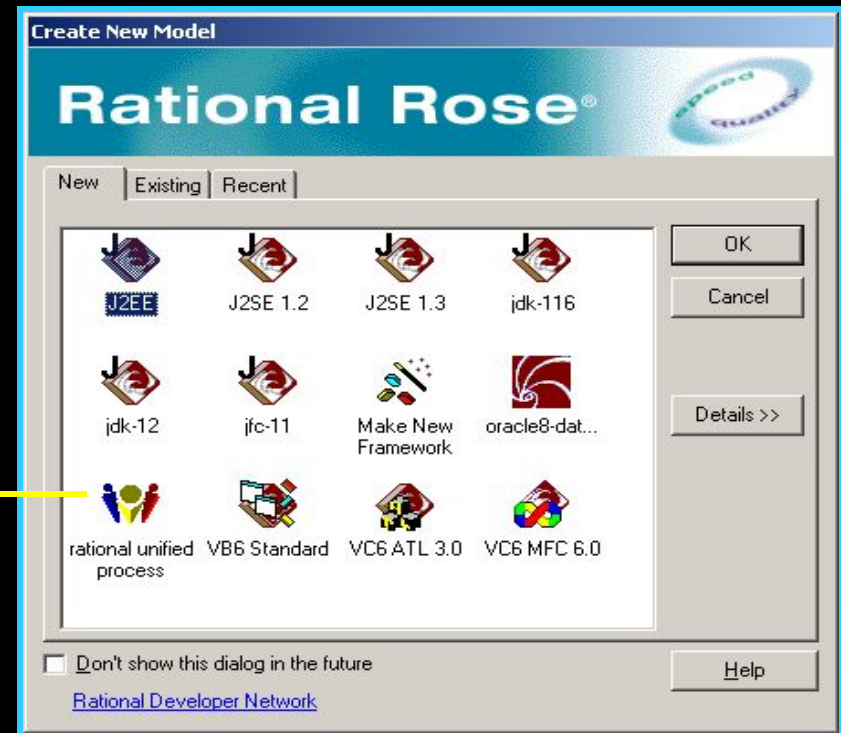
# Reuse

- ◆ Reuse refers to archiving and maintaining artifacts for future projects.
- ◆ Organizations gain significant benefits from reusing large-scale design elements like
  - Frameworks
  - Architecturally-significant packages
  - Subsystems
  - Mechanisms

# Frameworks

- ◆ A framework in Rational Rose is a set of predefined model elements that are used to model a certain kind of system and to provide a set of reusable components.

Framework Wizard



# Where Are We?

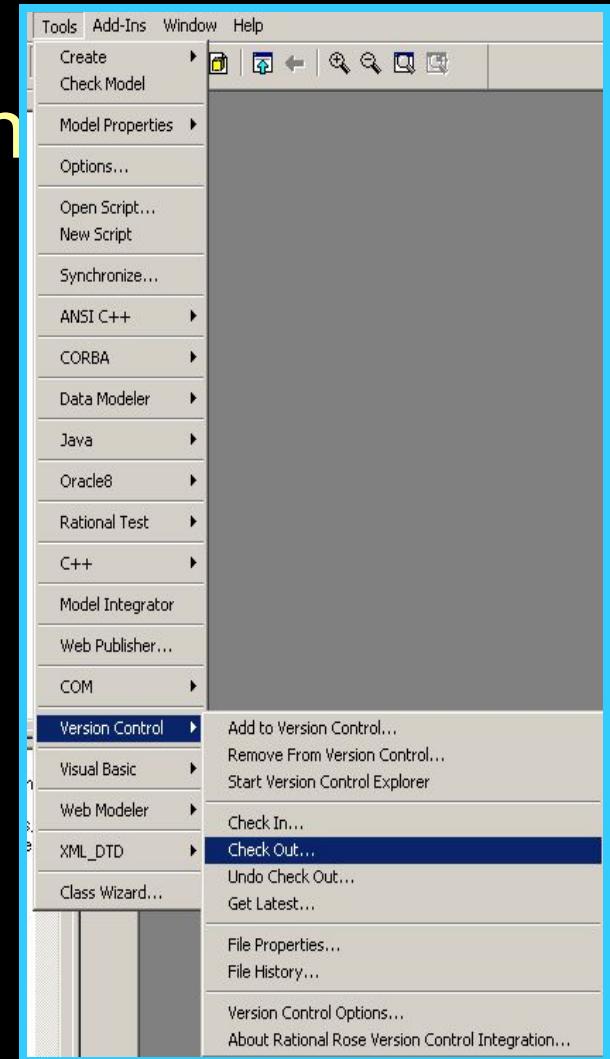
- ◆ Team-based modeling
- ◆ Controlled Units
- ◆ Virtual Path Maps
- ◆ Reuse
- ★◆ Version Control Add-Ins
- ◆ Model Integrator

# Version Control Add-Ins

- ◆ Rose provides two add-ins for version control
  - Version Control Add-In
  - ClearCase Add-In

# Version Control Add-In

- ◆ The Version Control Add-In provides integration between Rational Rose and any SCC-compliant version control system.



# ClearCase Add-In

- ◆ The ClearCase Add-In provides a tight integration between Rational Rose and Rational ClearCase.

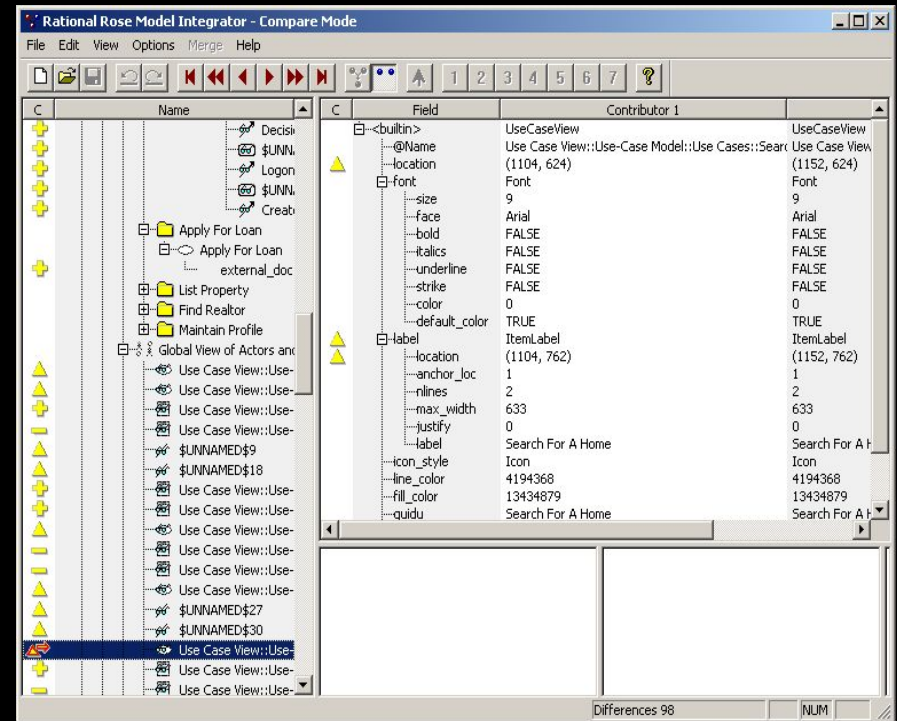
# Where Are We?

- ◆ Team-based modeling
- ◆ Controlled Units
- ◆ Virtual Path Maps
- ◆ Reuse
- ◆ Version Control Add-ins
- ★◆ Model Integrator



# Model Integrator

- ◆ The Model Integrator is a stand-alone tool that can be used to
  - Compare the differences between Rose models.
  - Merge different Rose models into a resultant model.



# Review

1. What capabilities in Rose support reuse?
2. Name the two version control add-ins supported by Rose.
3. What is the Model Integrator?

