


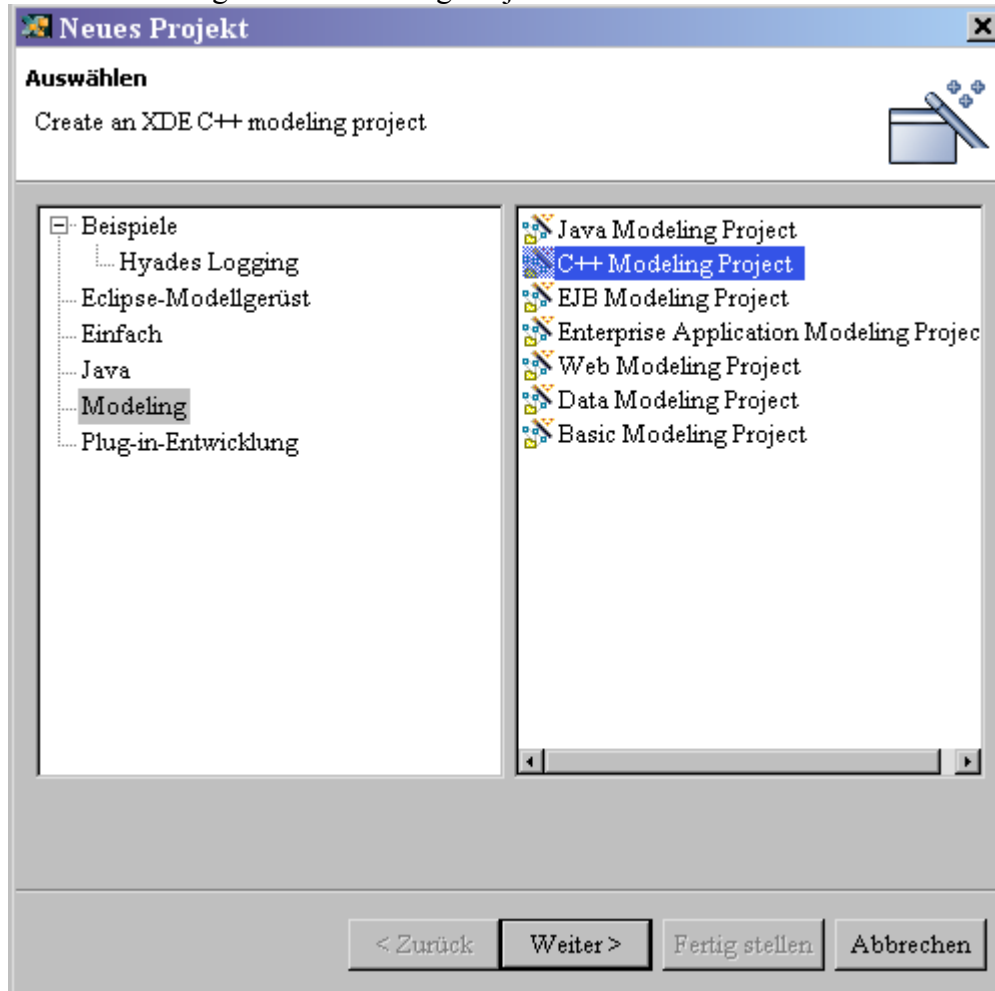
Getting Started with Rational Rose

Rational Rose starten:

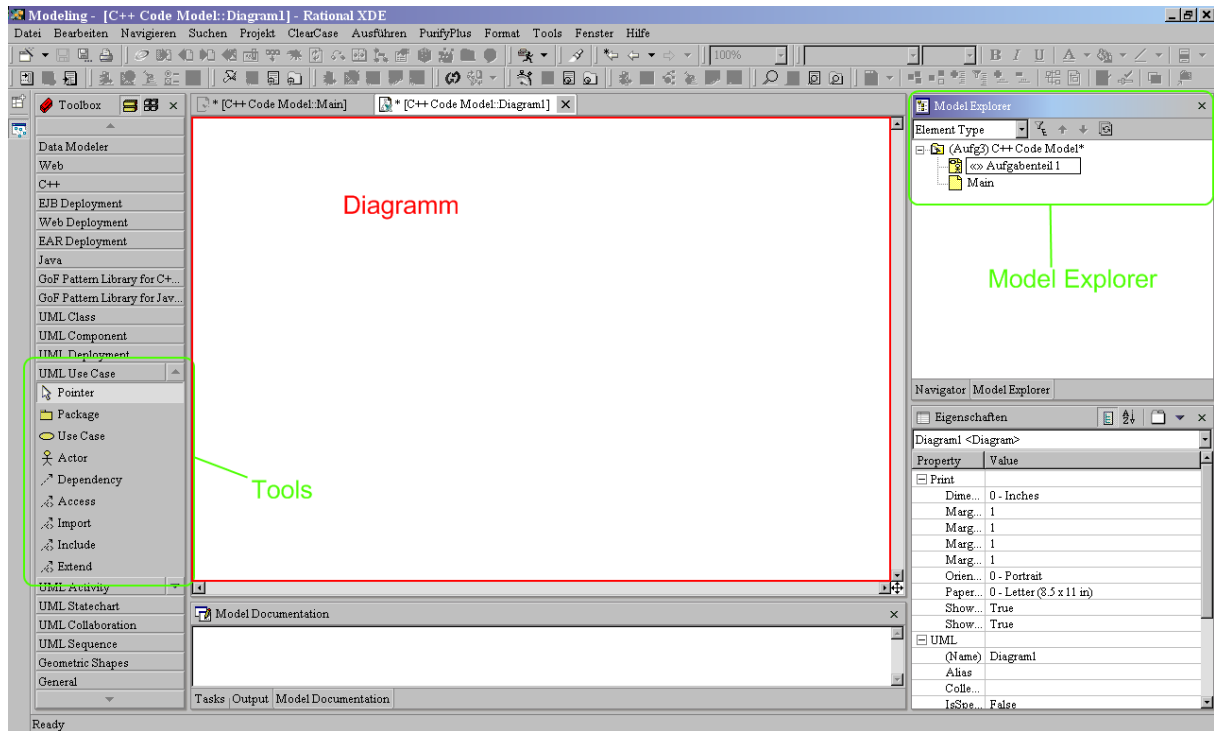
-  Rational XDE in ClearCase View

Projekt anlegen:

- Datei > Neu > Projekt
- Modeling > C++ Modeling Projekt

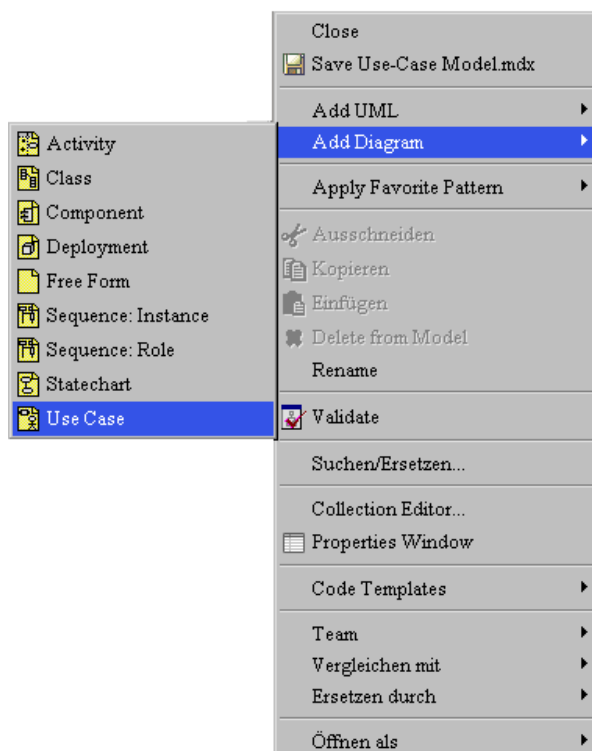


- Projektname und Zielverzeichnis eingeben
- Nun befindet Ihr Euch in der Hauptansicht. Links ist eine Leiste „Toolbox“ in der die zu den Modellen zugehörigen Objekte für die Modellierung zu finden sind. Diese Objekte könnt ihr einfach in Diagrammfenster ziehen.
- Rechts vom Diagrammfenster ist der „Model Explorer“. Dort seht ihr alle angelegten Diagramme und deren Inhalte.



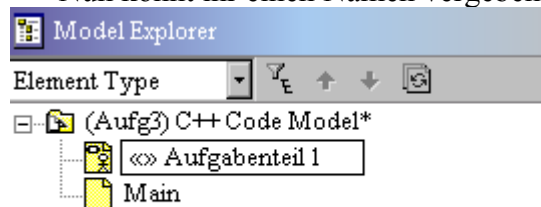
Neues Diagramm anlegen

- Als erstes müsst ihr ein Diagramm anlegen.
Hinweis: Legt für jedes Model ein neues Diagramm an wie im Folgenden beschrieben!!! Im Main hat ein Model nichts zu suchen!!!
- Im Model Explorer: Rechtsklick > Add Diagram > z.B. Use Case



- Das angelegte Diagramm erscheint daraufhin im ModelExplorer.

- Nun könnt ihr einen Namen vergeben und mit der Modellierung beginnen.



Hinweis: Für Sequenzdiagramme solltet ihr „Sequence: Role“ verwenden.

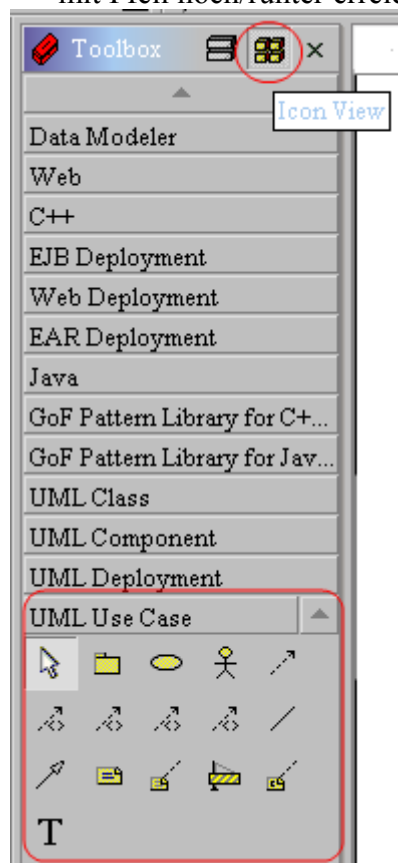
Vorsicht!

Wenn ihr etwas in der Hauptdiagrammansicht löscht, ist es trotzdem noch im Model Explorer vorhanden, also darauf achten, dass ihr solche Änderungen parallel in beiden Fenstern macht, oder am Besten direkt im Model Explorer, dann wird gleichzeitig auch das Objekt aus der Ansicht entfernt!

Weitere Hinweise zu Rational Rose:

Generelles:

- Icon View in der Toolbox einstellen, sonst sind nicht alle Symbole sichtbar und nur mit Pfeil hoch/runter erreichbar, Icon View ist übersichtlicher.



- Beim Entfernen/Ändern von Objekten am Besten: Rechtsklick > Select in Model Explorer verwenden

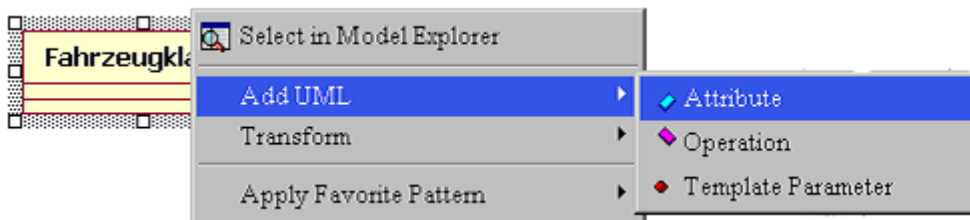
- Diagramme immer im zugehörigen Kontext modellieren!
Beispiel: Das Aktivitätsdiagramm eines UseCases kommt auch in den UseCase.
Ein Zustandsdiagramm gehört zur jeweiligen Klasse.
Wichtig: In den Diagrammen dann nicht nochmals Klassen, Methoden, etc. erstellen, sondern die benötigten Objekte aus dem Model Explorer in das Diagramm ziehen!

Klassendiagramm:

- Wenn Ihr an der Größe einer Klasse rumgespielt hat, macht das Programm die Skalierung nicht mehr automatisch. Um dies wieder einzuschalten, macht ggf. Ansicht > Perspektive zurücksetzen und drückt dann den Button: Enable/Disable auto resizing.



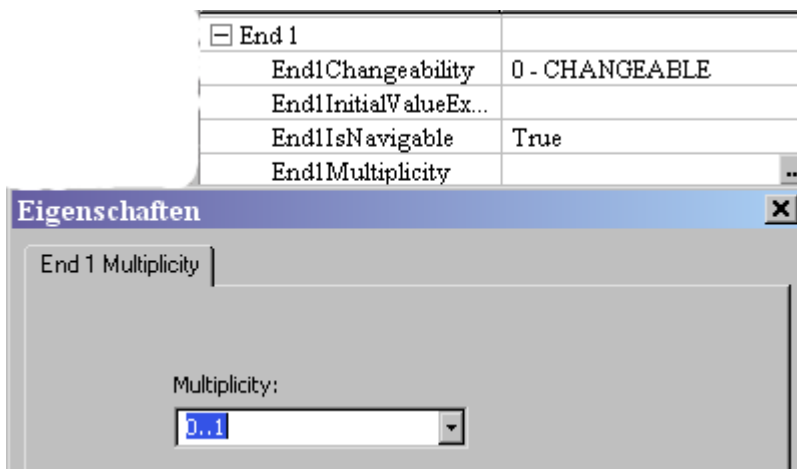
- Attribute und Operationen zu Klassen hinzufügen: Rechtsklick auf die Klasse > Add UML > Attribute



- Datentyp eines Attributes festlegen: Rechtsklick auf Klasse > Collection Editor > Register Attributes > Type Expression *oder* mit : Datentyp an den Namen in der Klasse selbst anfügen.



- Multiplizität einstellen: Verbindung anwählen, im Eigenschaften Fenster > End1Multiplicity Multiplizität im folgenden Fenster auswählen.



- public/private: markiert das Attribut oder die Operation, Eigenschaften > Visibility
Visibility 2 - PRIVATE
- bei private getter und setter- Methoden nicht vergessen!

- Mit folgenden Buttons könnt ihr die Sichtbarkeit von Attributen und Operationen, Ein- und Ausgabeparametern bei Operationen, und public und private(+ und -) einstellen:



Show/Hide Attribute compartment > Operation Compartment, Attribute Compartment

Show/Hide Signal signature > Operation Signature

Show Stereotype as Label for compartment >Visibility Style: Text

- Mit Rechtsklick auf eine Klasse > Generate Code könnt ihr C++ Code generieren. Mit Browse Source könnt ihr ihn euch ansehen.

Zustandsdiagramm

- Im Zustandsdiagramm können die Methoden aus dem Klassendiagramm im ModelExplorer einfach auf eine Assoziation gezogen werden.
- Eine Bedingung gibt man im Eigenschaften-Fenster einer Assoziation unter GuardCondition an.

`GuardCondition` uhrzeit=alarmzeit

Sequenzdiagramm

- Hier können die benötigten Klassen aus dem Klassendiagramm im Model Explorer einfach in das Diagramm gezogen werden.
- Wenn man zwischen 2 Klassen eine Message sendet, werden die verfügbaren Methoden angezeigt, keine neuen Methoden hinzufügen, die es nicht/bereits im Klassendiagramm gibt!