

Formale Methoden:
UML, OCL und vdm++

Prof. Dr. Hans-Jürgen Buhl

2004

Fachgruppe Mathematik / FB C
Institut für Angewandte Informatik
Bergische Universität Wuppertal

Praktische Informatik
PIBUW – SS0401

April 2004
1. Auflage, 2004

Praktische Informatik 02

Link-Sammlung

- UML-Dokumentation und Beispiele
 - <http://lci.cs.ubbcluj.ro/ocle/links.htm>
 - <http://www.agilemodeling.com/style/classDiagram.htm>
 - <http://www.agilemodeling.com/artifacts/classDiagram.htm>
- UML-Tools
 - <http://lci.cs.ubbcluj.ro/ocle/links.htm>
 - <http://uml.sourceforge.net/index.php>
 - <http://www.borland.com/together/>
 - <http://www.mid.de/de/>
 - <http://www-306.ibm.com/software/rational/offerings/design.html>
- OCL-Standard-Drafts
 - <http://neptune.irit.fr/Biblio/03-01-07.pdf>
 - <http://neptune.irit.fr/Biblio/01-09-77.pdf>
- OCL-Tools
 - <http://lci.cs.ubbcluj.ro/ocle/links.htm>
 - <http://www.klasse.nl/ocl/>
 - <http://www.eclipse.org/>
 - <http://www.eclipse.org/cdt/>
- OCL-Beispiele
 - <http://www.klasse.nl/ocl/ocl-reasons.html>
 - <http://www.klasse.nl/ocl/ocl-introduction.html>

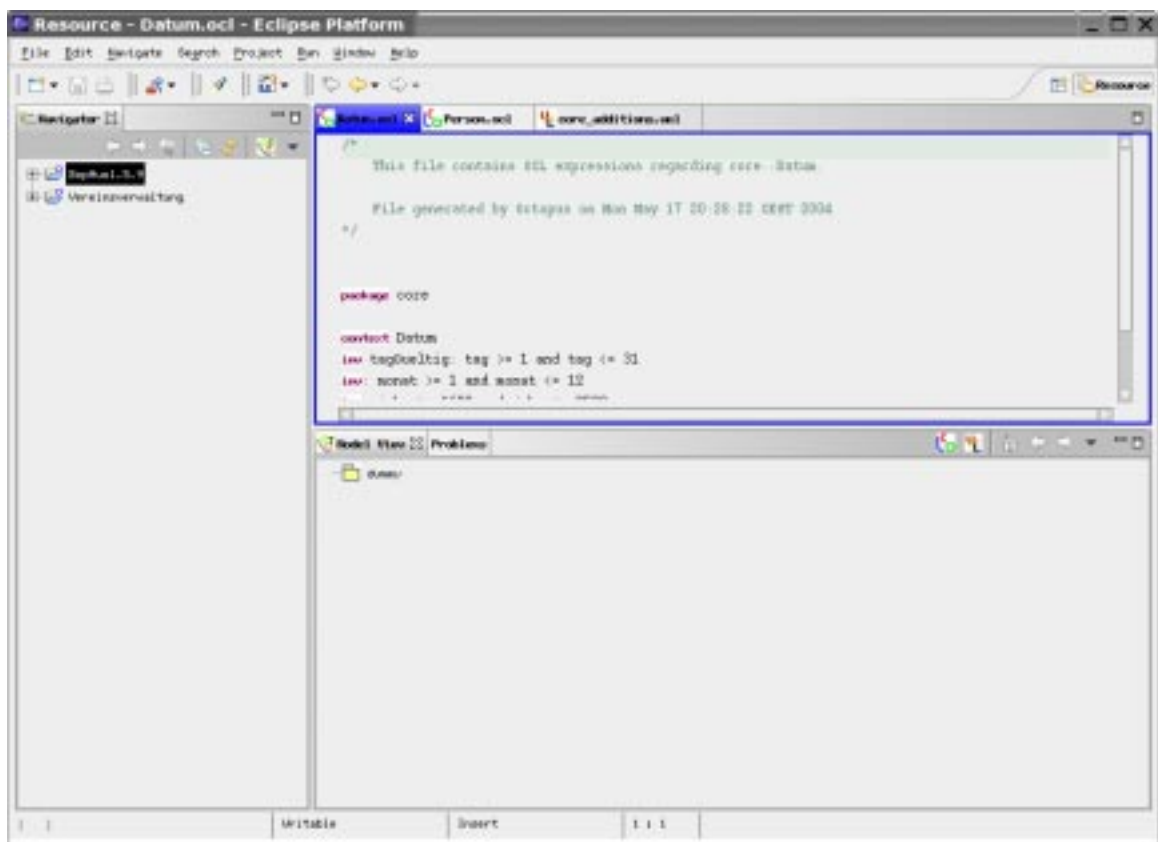
Kleine Einleitung in die Benutzung von Octopus

Stand: Eclipse 3.0.0 mit Octopus 1.0.1

Starten Sie Eclipse durch den Aufruf von

> Eclipse

Sie erhalten:



Wählen Sie

File

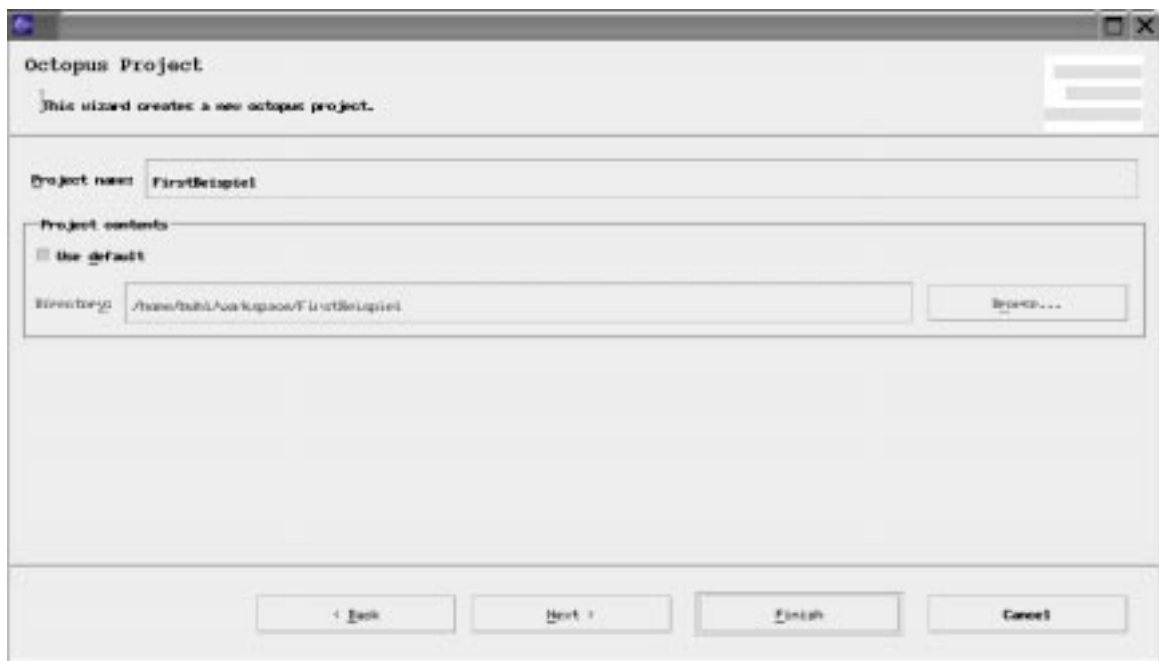
New

Project

aus:



Nach Anklicken der Option „Show all Wizards“ und Auswahl von „Octopus project“ erscheint nach Anwahl von Next:



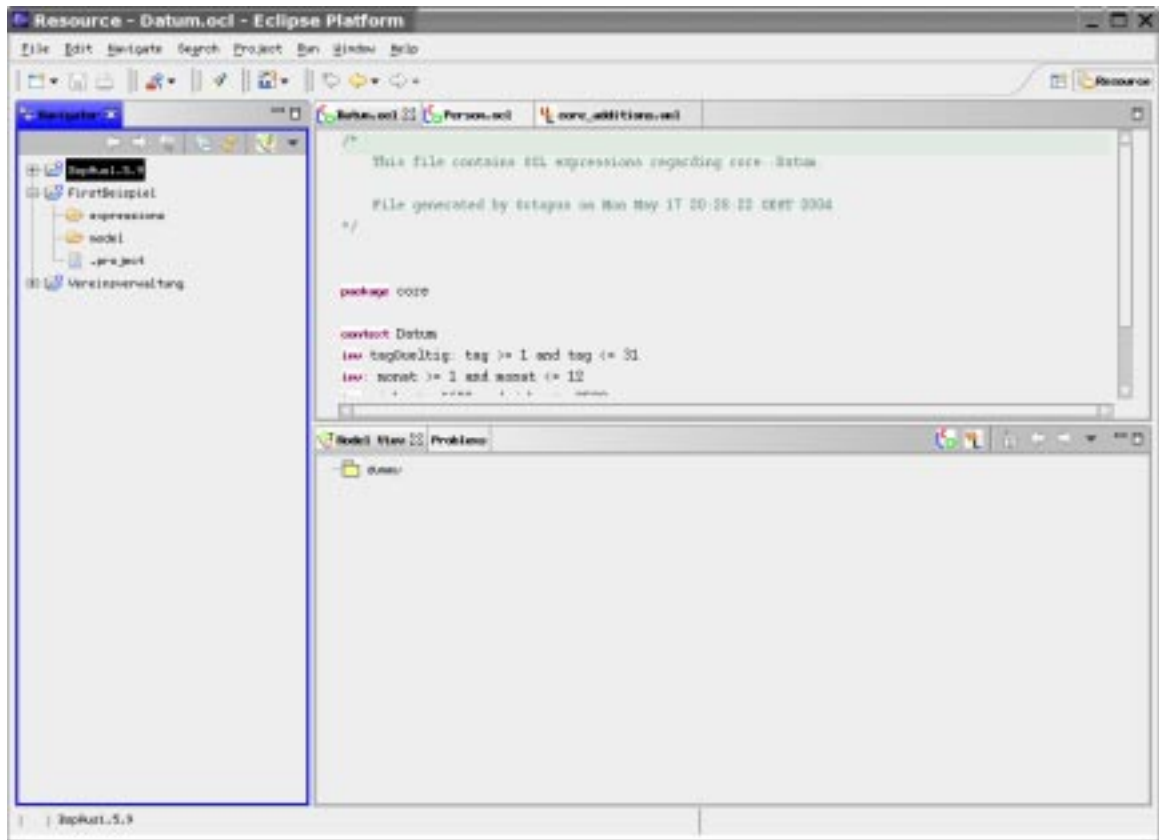
Vergeben Sie bitte einen Projektnamen (hier zum Beispiel FirstBeispiel)

und klicken Sie erneut auf **Next**:



Akzeptieren sie schließlich durch Klick auf **Finish**.

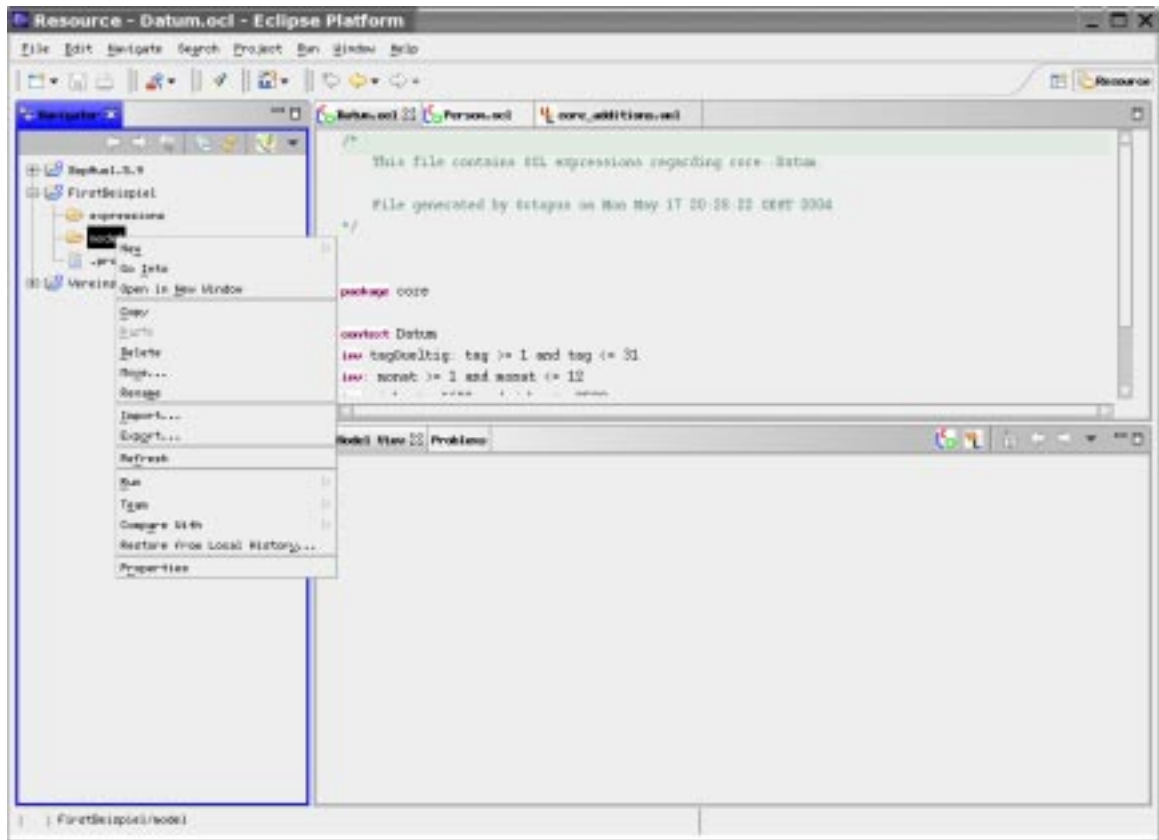
Im Navigator-Fenster erscheint das neue Projekt mit den (noch leeren) Unterordnern **expressions** sowie **model**:



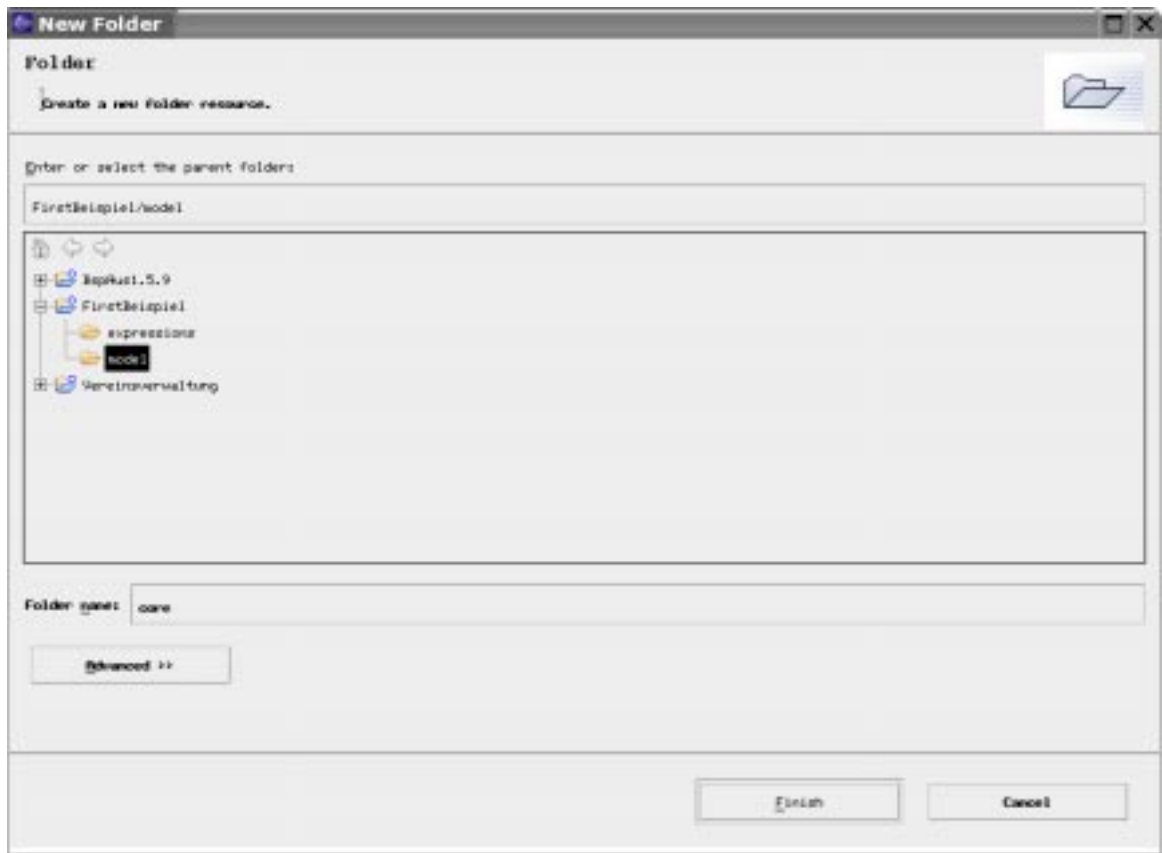
Wählen Sie `model` an und erzeugen sie darin mit Hilfe der rechten Maustaste über

New
Folder

einen weiteren (Unter-)Unterrodner, der einem UML/OCL-package entspricht:

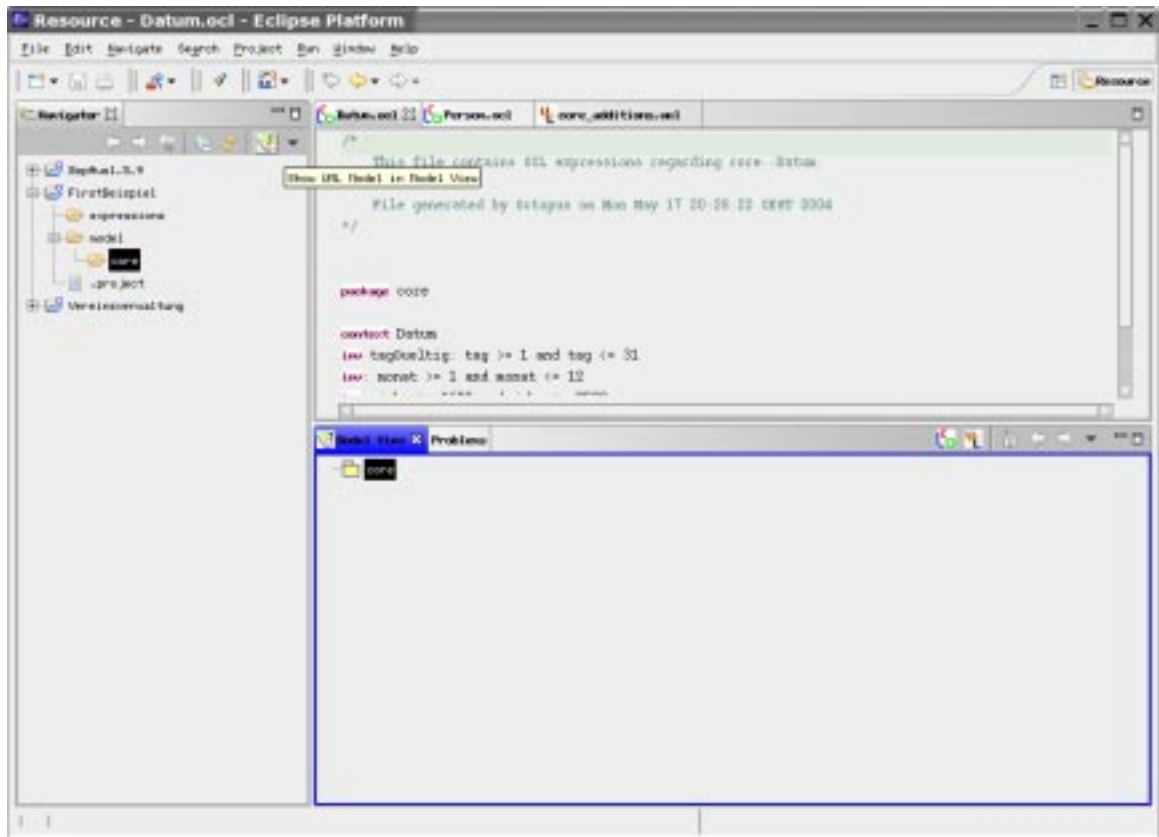


P roject...	
F older	
F ile	
O ther...	Ctrl+N

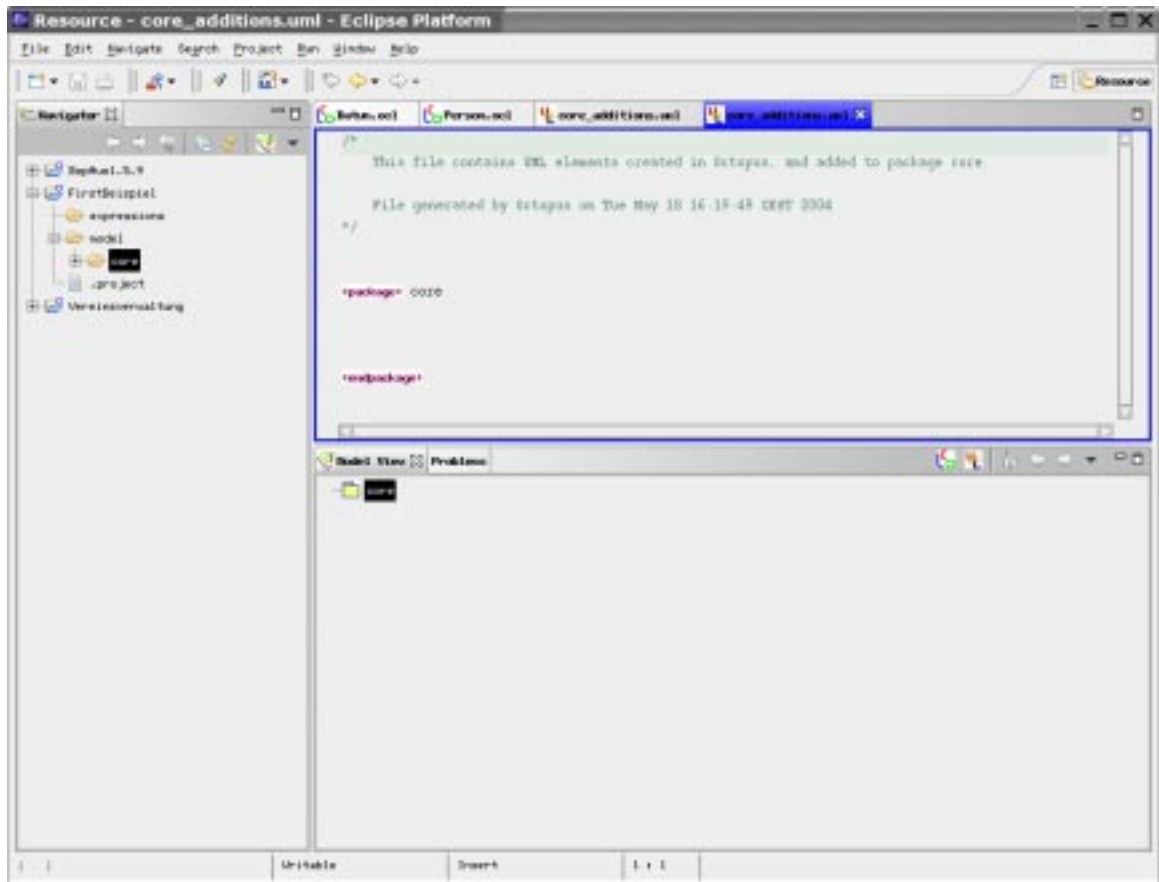


Im erschienenen Fenster „New Folder“ ist nun ein Folder-Name (sprich: Package-Name) zu spezifizieren — im diesem Beispiel: `core` — und danach **Finish** anzuklicken.

Klicken sie nun im Navigator-Fenster auf das OCT-Icon (Octopus), so wird der Octopus-UML-Model-View aktualisiert (statt `dummy` erscheint `core`):



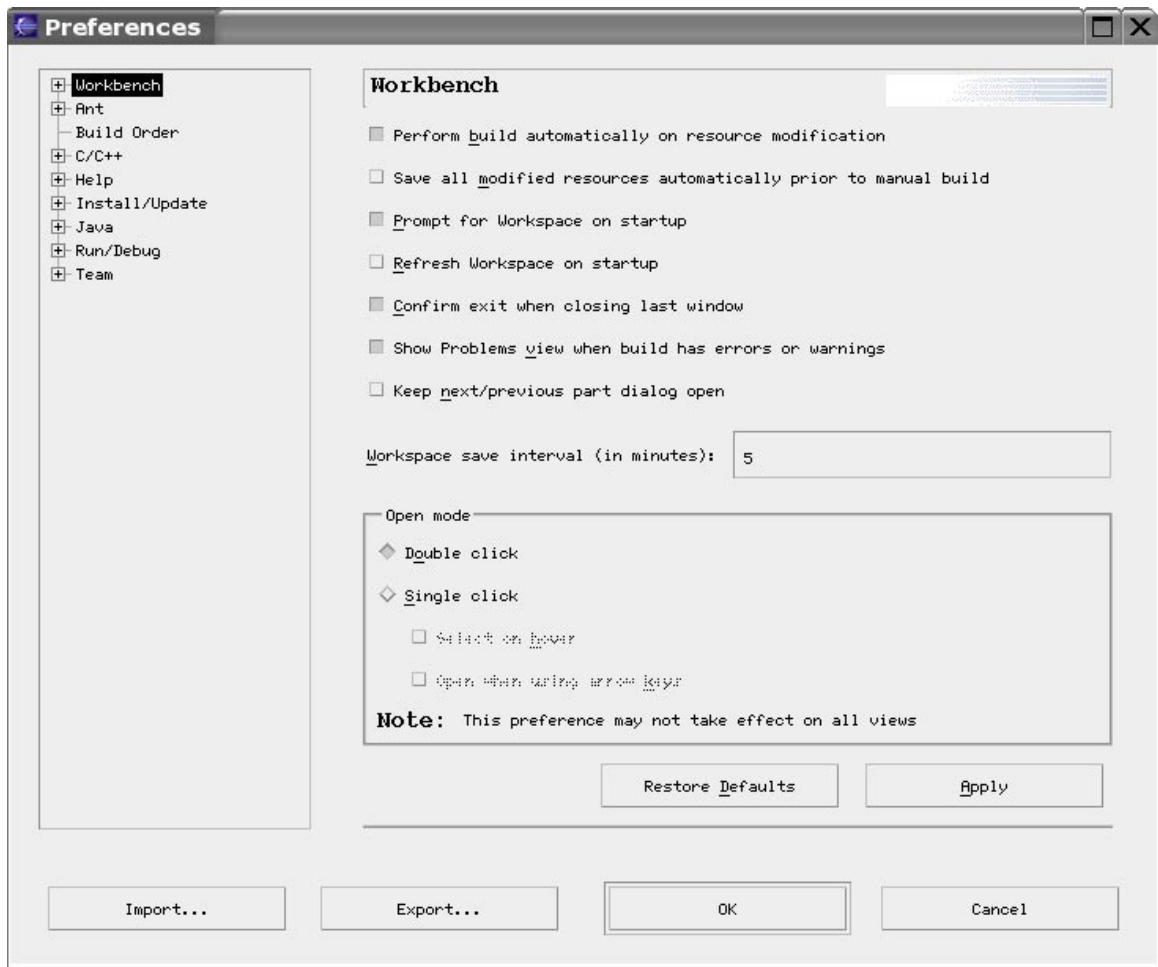
Wählen Sie im Model-View-Fenster `core` an und klicken Sie danach auf das UML-Icon des Model-View-Fensters: Es erscheint ein schon vorausgefüllter Dateinhalt für das Package `core` im UML-Editor oben rechts:



Für die weitere Arbeit ist es wichtig, dass Sie die Window-Preferences richtig eingestellt haben. Rufen Sie zur Überprüfung über

Window
Preferences

das zugehörige Fenster auf

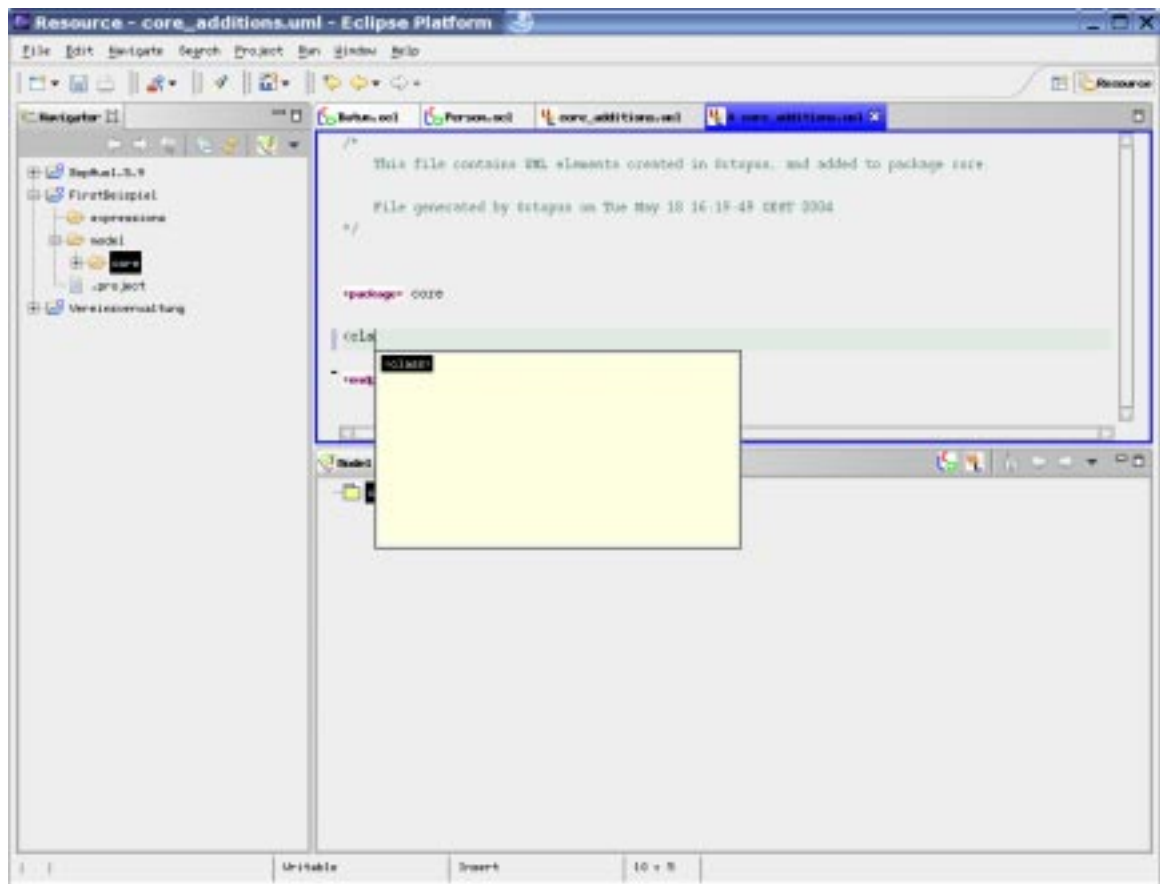


und vergewissern Sie sich, dass sowohl „Perform build automatically on resource modification“ als auch „Show Problems view when build has errors or warnings“ eingeschaltet sind.

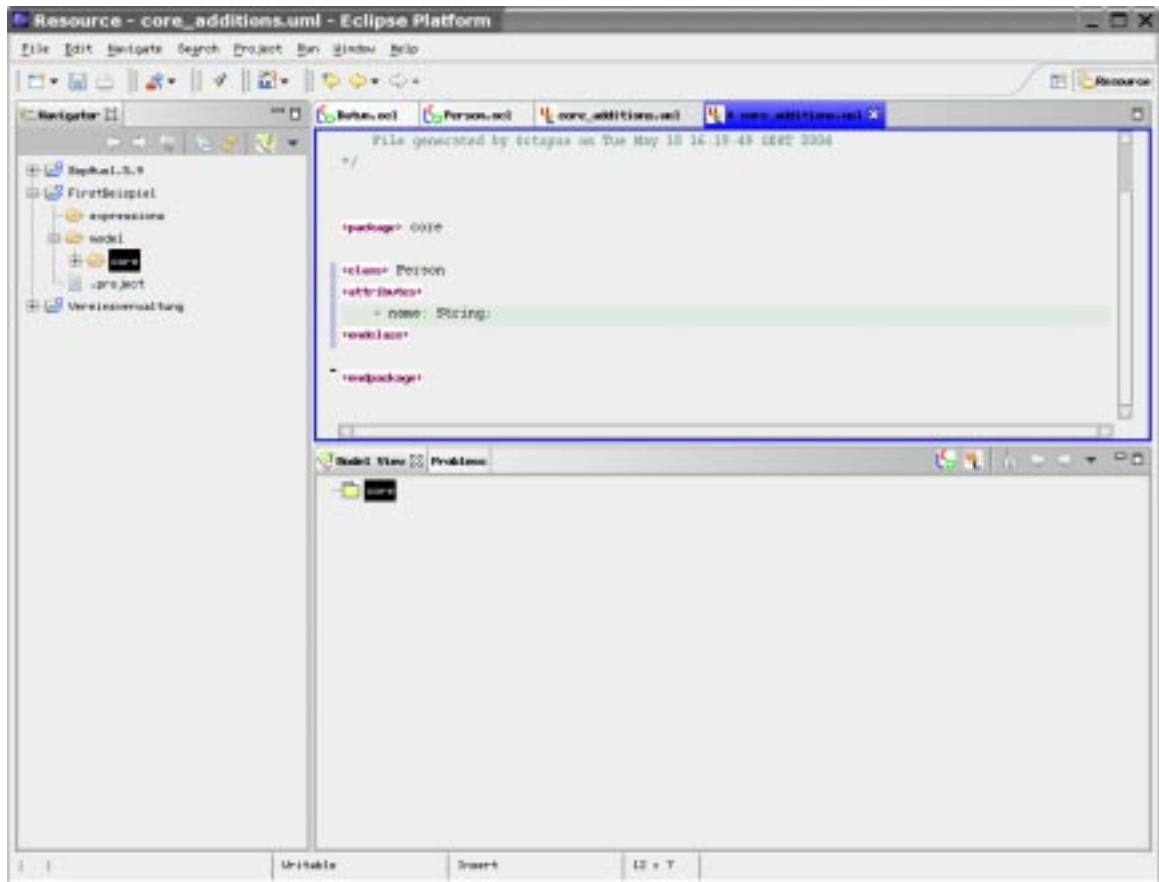
Nun sollten sie textuell eine UML-Klasse eingeben. Nach Eintippen von <cla wird mittels der Tastenkombination

<Ctrl Leertaste>

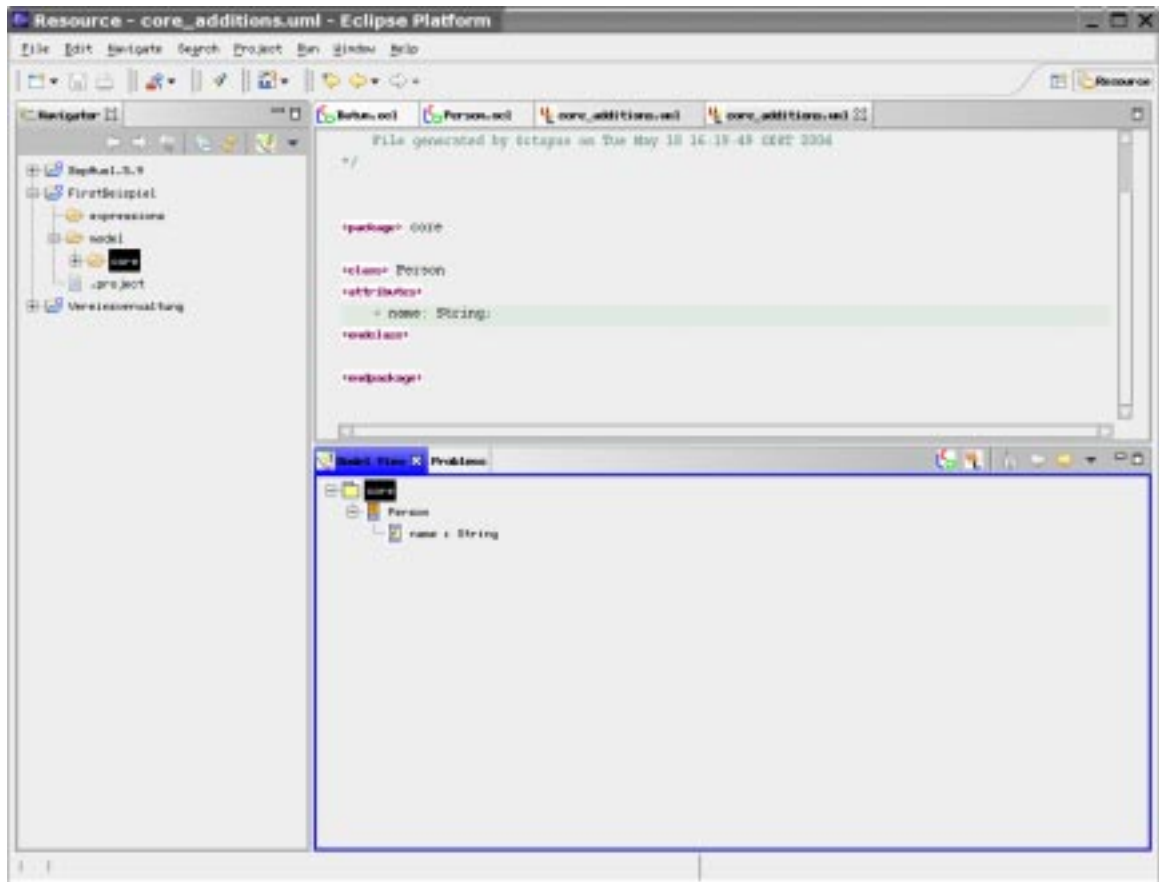
die automatische Schlüsselwortergänzungsfunktionalität angefordert:



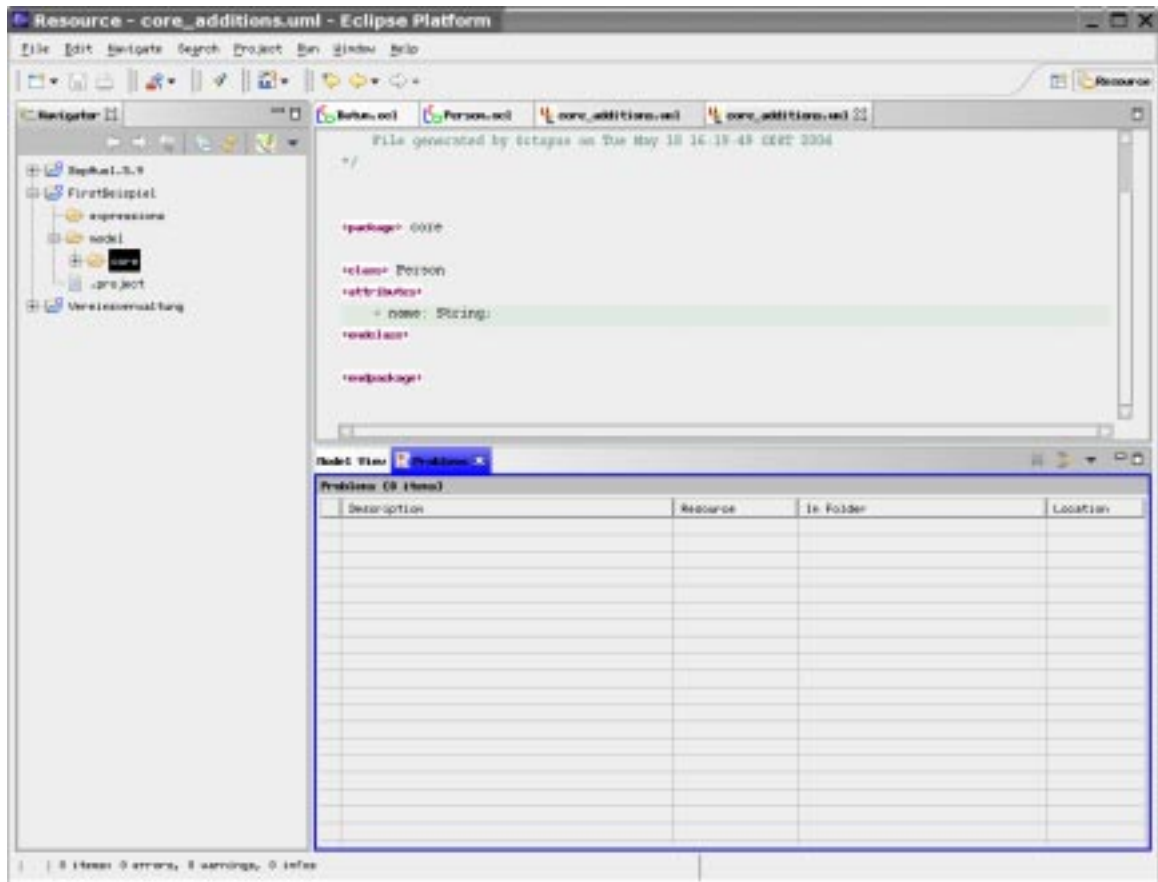
Ergänzen Sie die Klasse zur Klasse **Person** nach folgendem Muster



und klicken Sie auf das Disketten-Icon, um mit der manuellen Abspeicherung aller Projektänderungen auch (implizit) ein build mit einem Syntaxcheck auszulösen: Es erscheint der neue OCT-Model-View

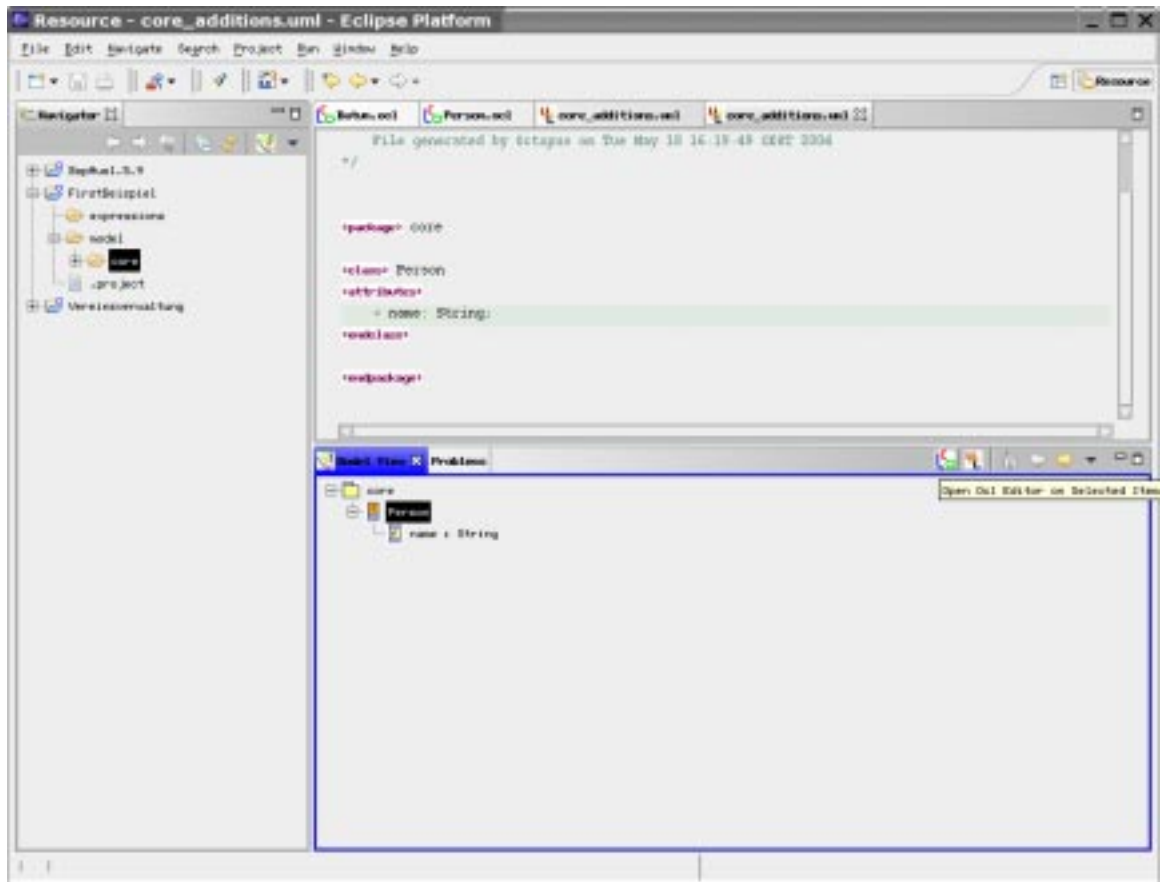


oder eine Fehlermeldung im Problems-Fenster.

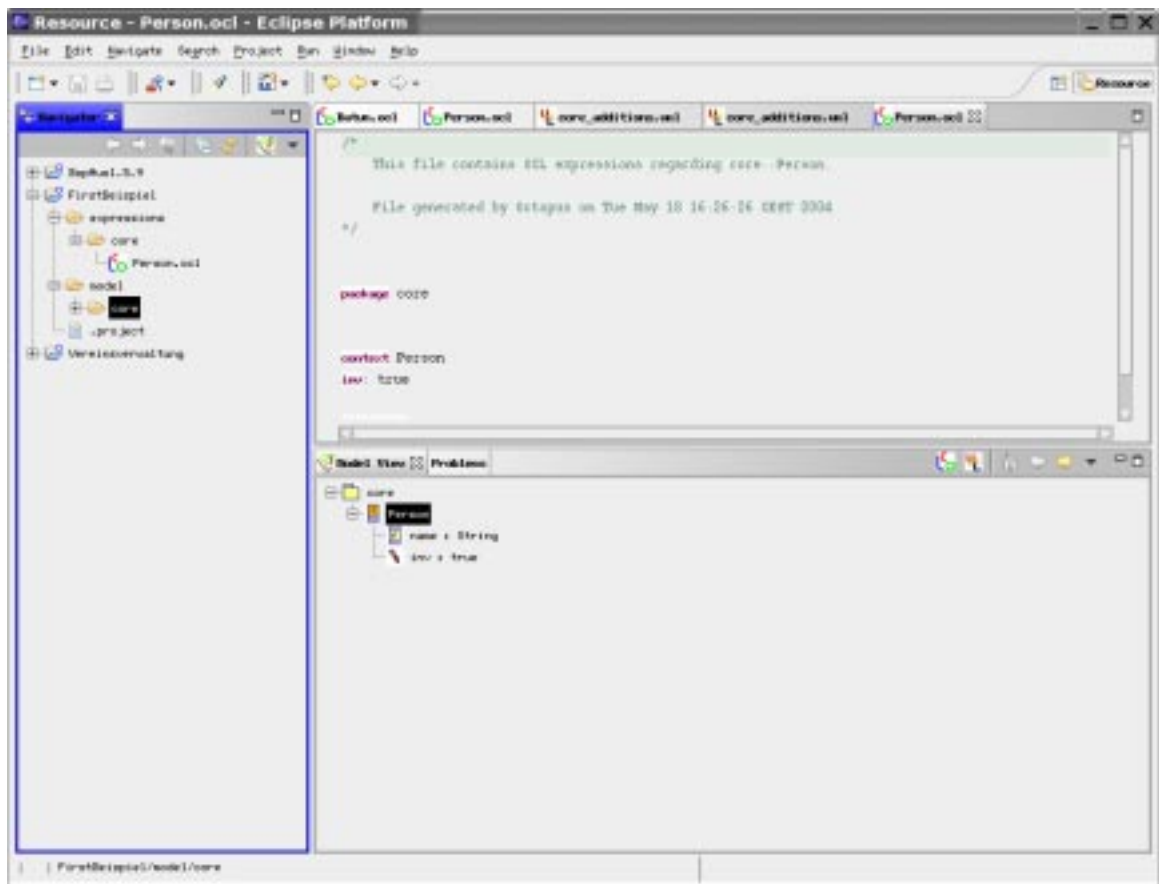


Ein leeres Problems-Fenster (wie hier) zeigt an, dass keine Syntaxfehler festgestellt wurden.

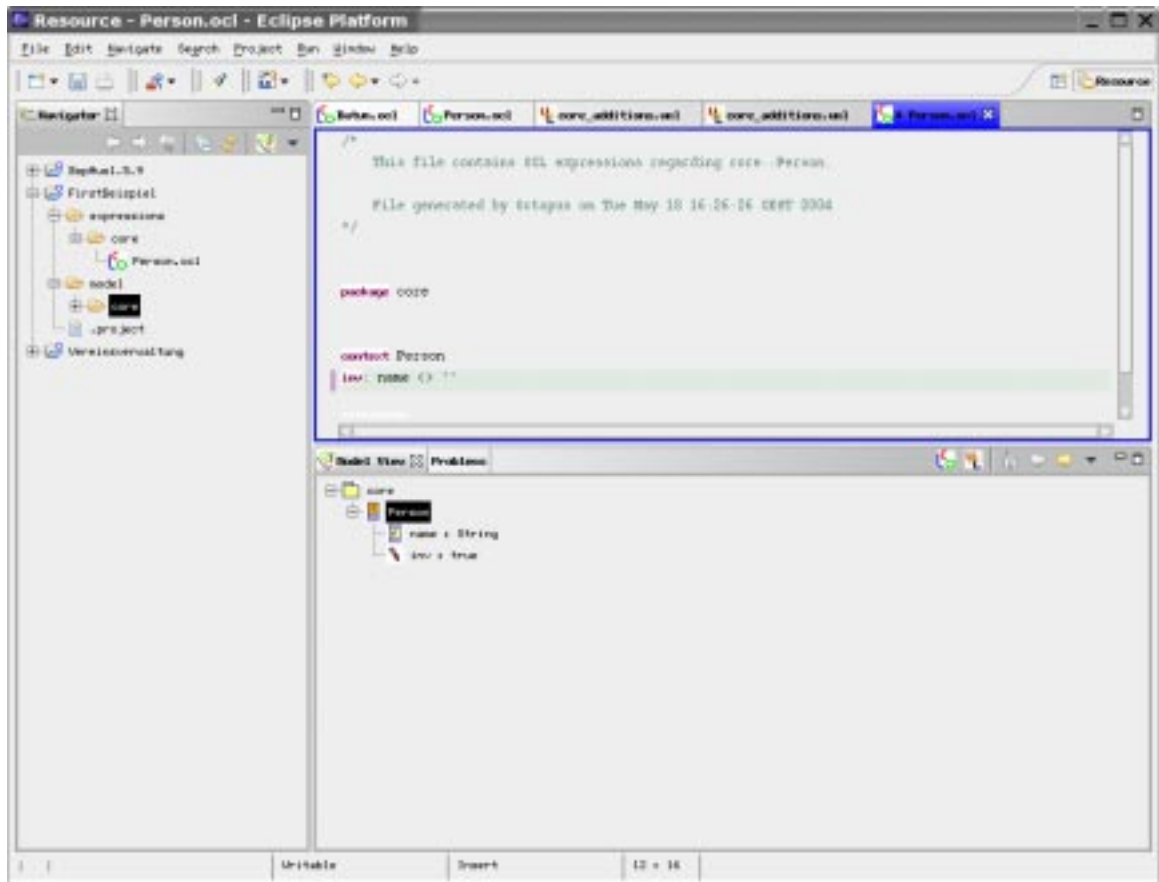
Klicken Sie nun im Model-View-Fenster auf die Klasse **Person** und danach auf das OCL-Icon in der Model-View/Problems- Kopfzeile, um OCL-Constraints für die Klasse **Person** einzugeben: Im OCL-Editor erscheint eine vorausgefüllte OCL-Datei



die Sie nach Aktualisierung des Model-View-Fensters



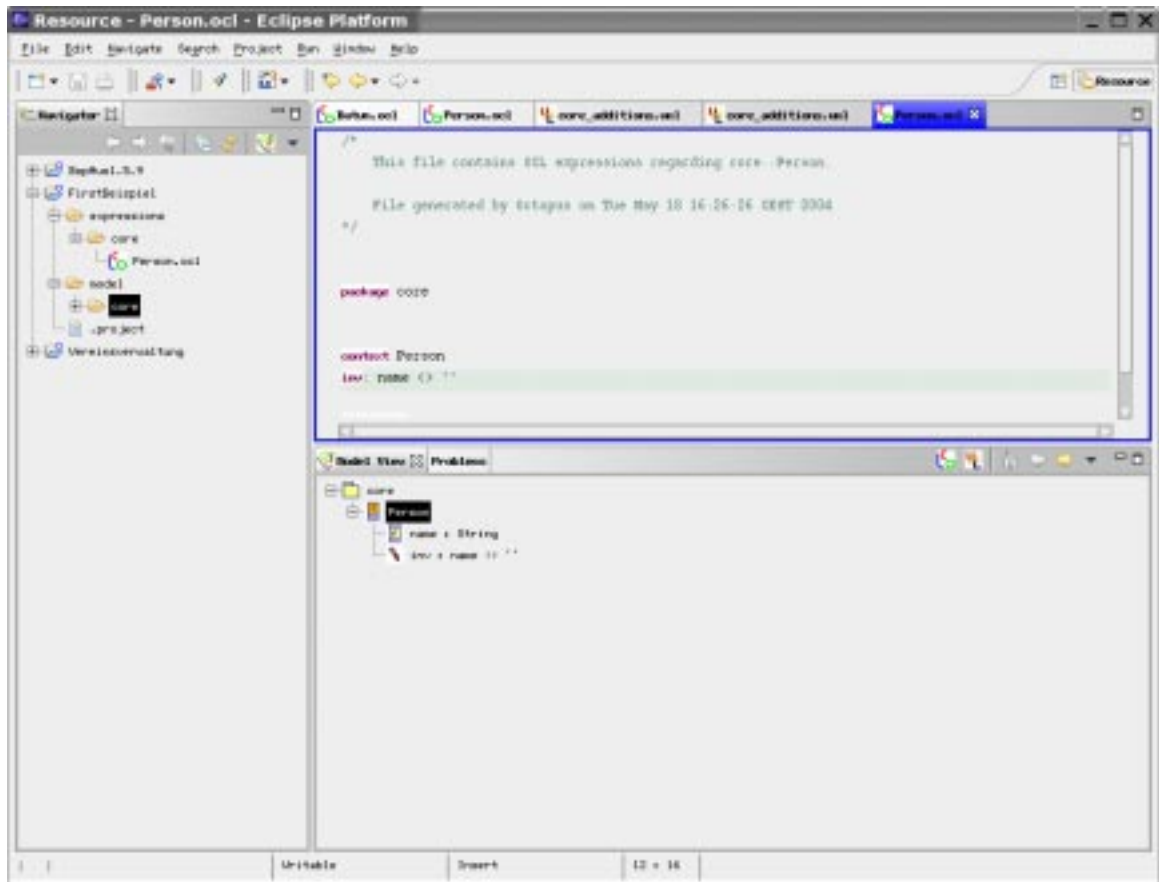
jetzt nur noch entsprechend zu modifizieren brauchen



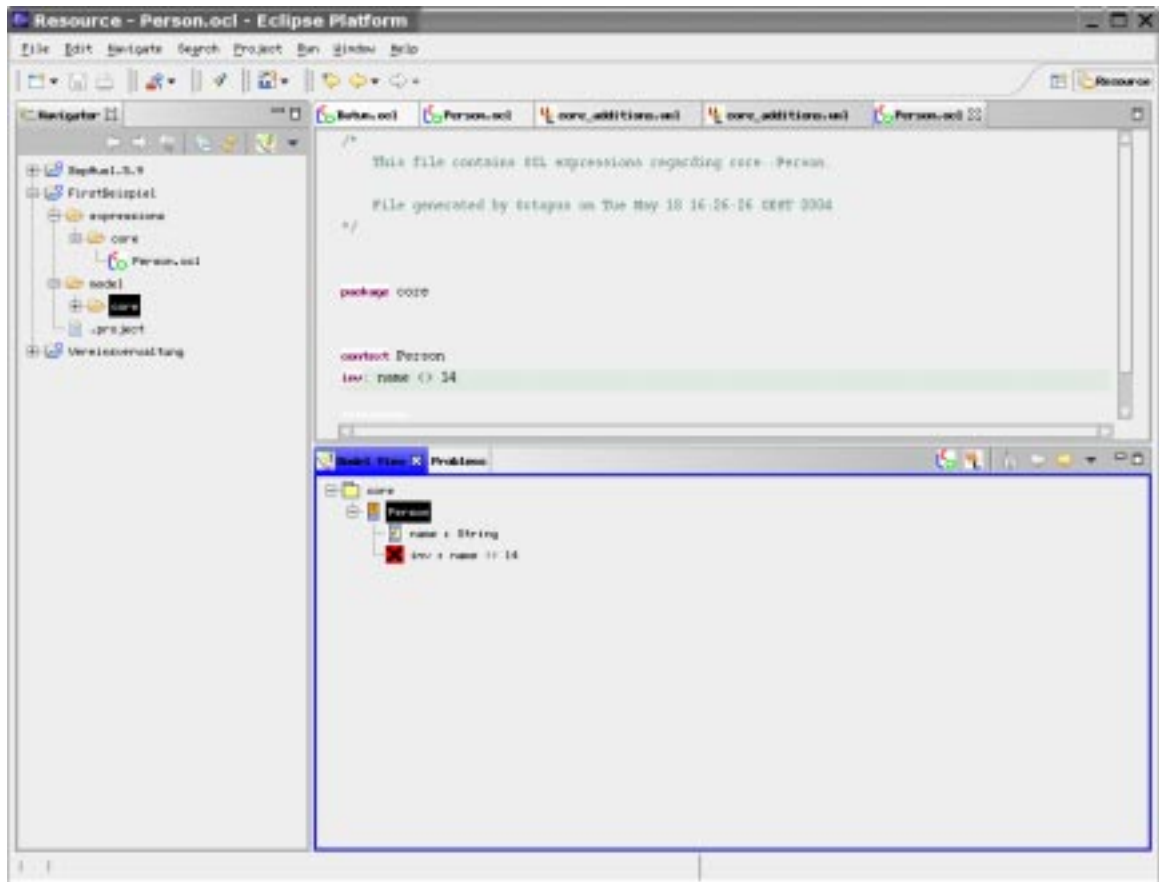
um dann nach Anklicken des Save-Icons von der syntaktischen Korrektheit der OCL-Klassen-Invarianten

```
name <> ''
```

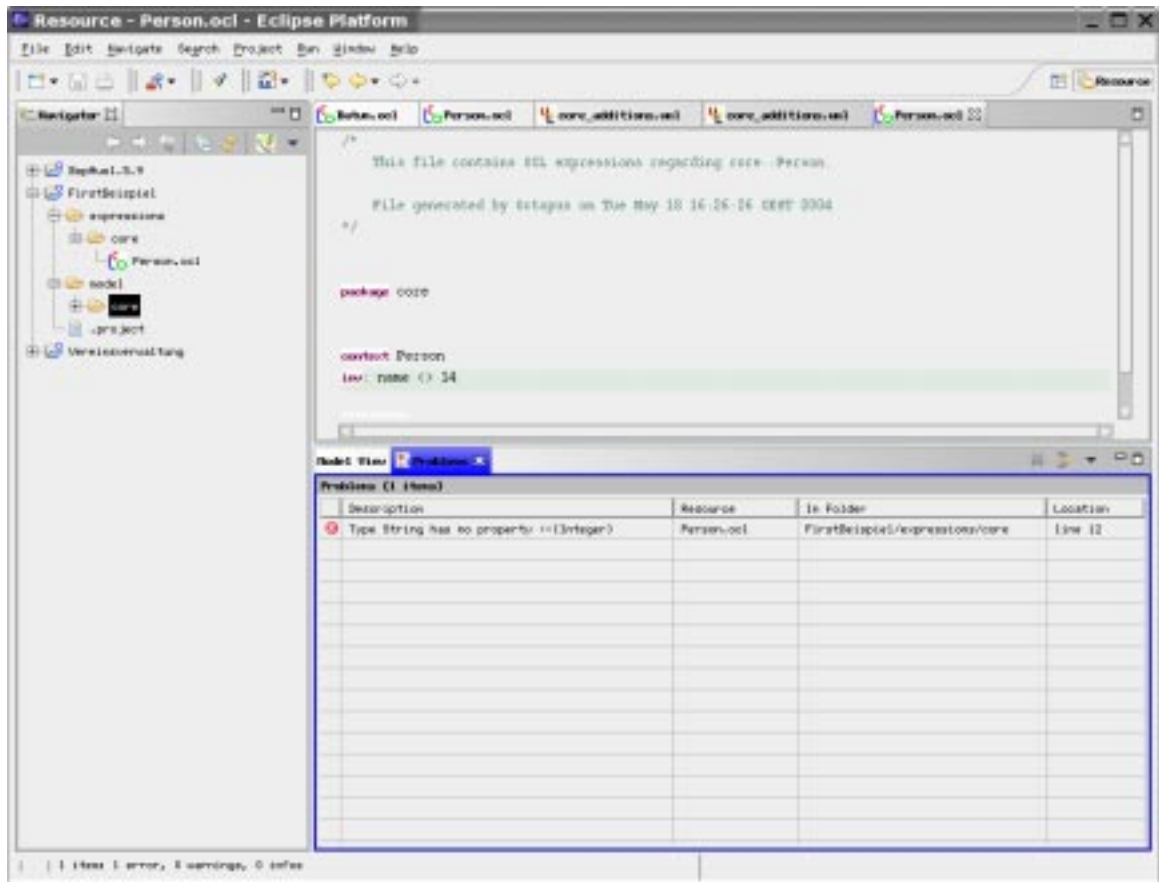
überzeugt zu sein



beziehungsweise einen Fehler sowohl im Model-View



als auch im Problems-Fenster feststellen zu können (rotes Problem-Kreuz):



Und nun viel Vergnügen bei der Planung von formal spezifizierten Software-Modellen...