Werkzeugunterstützung für UML Profiles

Zwischenbericht Großer Beleg

Andreas Pleuß

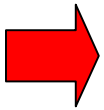
Gliederung

- Aufgabenstellung
- Einführung in UML Profiles
- Anforderungen an UML Profiles
- UML Profiles in UML 1.4
- UML Profiles im CASE-Tool „Objecteering“ von Softteam
- Zusammenfassung und Ausblick

Aufgabenstellung

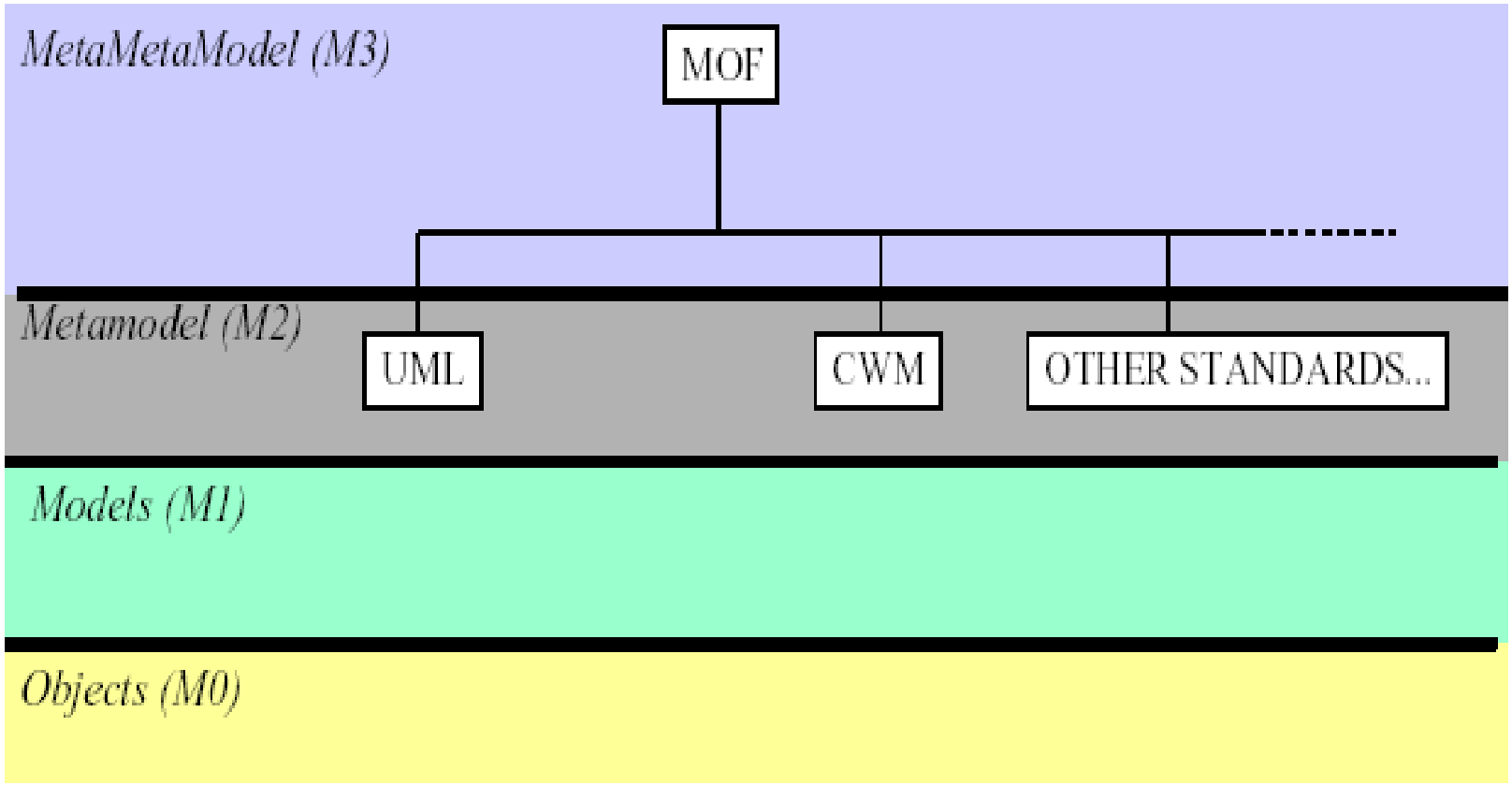
- Sammlung der Anforderungen an UML Profiles
- Untersuchung bestehender UML-CASE-Tool Unterstützung
- Untersuchung der Erweiterungsmöglichkeiten eines ausgesuchten UML-CASE-Tools
- Ausarbeitung von Empfehlungen für die Erweiterung des ausgewählten CASE-Tools und Nachweis der Machbarkeit mittels eines Prototypen

Gliederung



- Aufgabenstellung
- Einführung in UML Profiles
- Anforderungen an UML Profiles
- UML Profiles in UML 1.4
- UML Profiles im CASE-Tool „Objecteering“ von Softeam
- Zusammenfassung und Ausblick

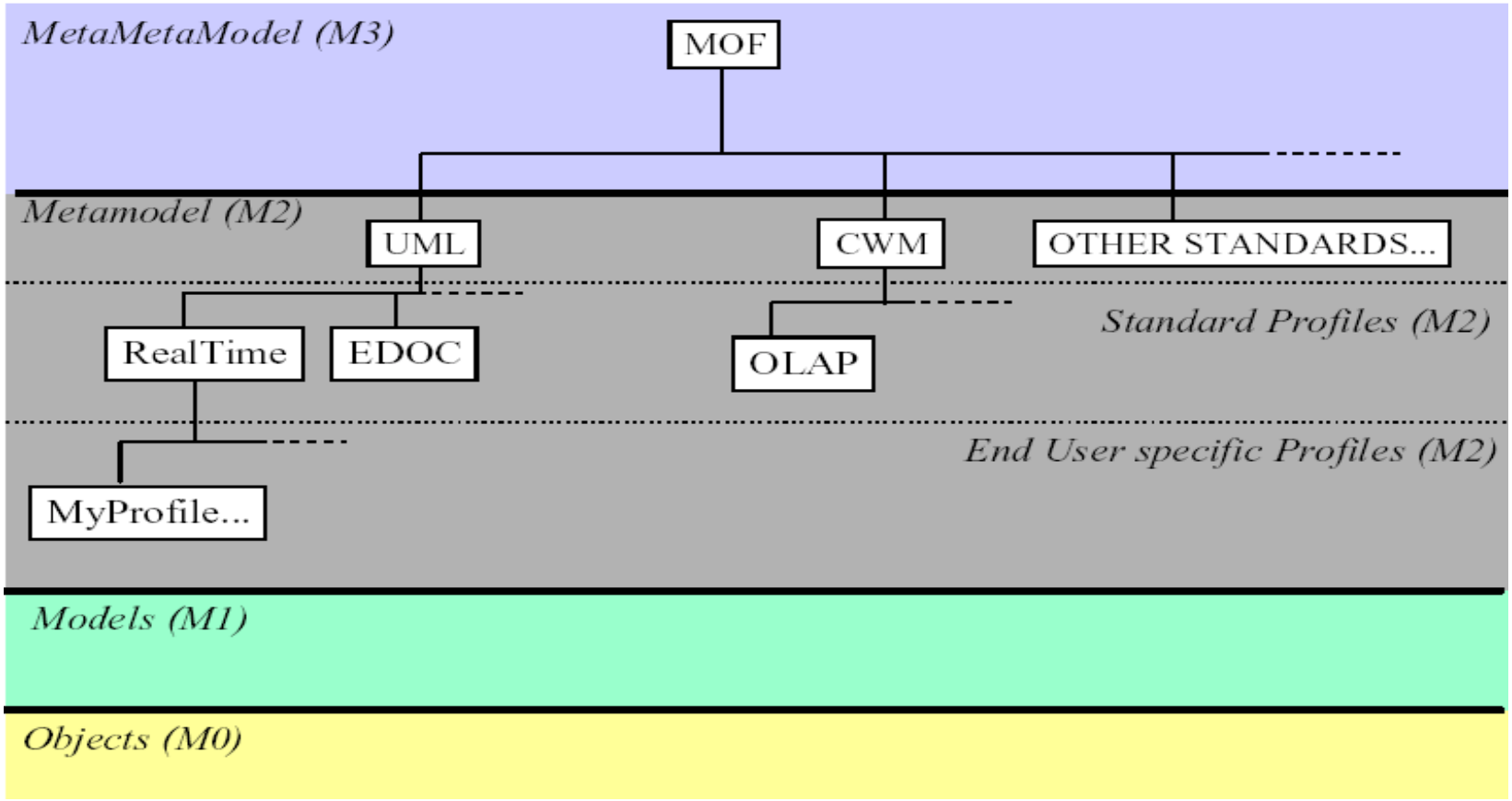
4-Schichten-Metamodellarchitektur



UML Profiles

- Motivation: Erweiterung
- Prinzipiell:
  - „heavyweight extensions“: Verändern eines Metamodells
  - „lightweight extensions“: Spezialisierung eines Metamodells

Profiles in der Metamodellarchitektur



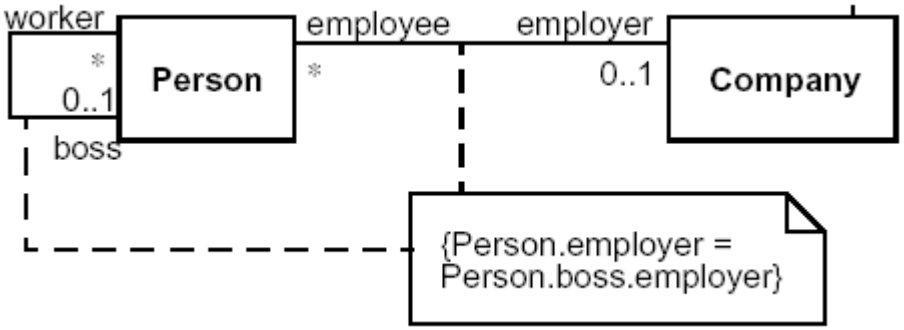
UML Profiles: Verwendung von UML lightweight extensions

- Nur Spezialisierung der Standardsemantik, ohne Veränderung des Metamodells
- Spezialisierung der UML zur Anpassung an:
  - Technische Implementierungen (z.B. Java)
  - Generelle technische Bereiche (z.B. RealTime)
  - Spezielle Anwendungsgebiete (z.B. Finanzsektor)
  - Softwareentwicklungsprozesse
- UML enthält lightweight extensions:
  - Stereotypes
  - Constraints
  - Tagged Values
- Profile: Fasst lightweight extensions zusammen

Stereotype

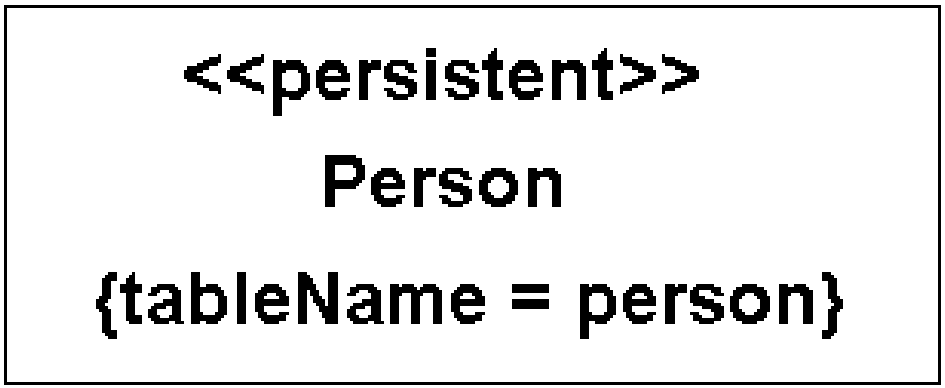
- Zur Definition neuer „virtueller Metaklassen“
- BaseClass: Metamodellklasse, die durch das Stereotype erweitert werden kann
- Kann zusätzliche Constraints und Tagged Values, sowie eine graphische Repräsentation einführen
- Constraints und Tagged Values werden auf Modellelemente angewandt, die mit dem Stereotype versehen sind

Constraint



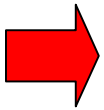
- Formuliert Regeln und Bedingungen
- Zuweisbar:
  - Einem oder mehreren Modellelementen direkt
  - Einem Stereotype => Bezug auf UML Metamodell

Tagged Value



- Paar aus Schlüssel und Wert
- Virtuelles Metaattribut

Gliederung



- Aufgabenstellung
- Einführung in UML Profiles
- Anforderungen an UML Profiles
- UML Profiles in UML 1.4
- UML Profiles im CASE-Tool „Objecteering“ von Softeam
- Zusammenfassung und Ausblick

Anforderungen (1)

- Allgemein:
  - Ermöglichen jeglicher Spezialisierung ohne Verletzung des Metamodells
- Inhalt:
  - Verwendung nur von „lightweight extensions“
  - Spezifikation Untermenge des Metamodells, auf die die Erweiterungen anwendbar sind
- Operationen auf Profiles:
  - Spezialisierung
  - Komposition

Anforderungen (2)

- Anwendung:
  - Angabe gewählter Profiles für Packages (bzw. Models und Subsystems) ermöglichen
- UML Model Libraries:
  - Profile referenziert UML Model Libraries
  - Zusammenbinden von Profiles und Model Libraries in eine Einheit ermöglichen
- Austausch:
  - Verwendung existierender Mechanismen (XMI)
  - Zusammen mit Modellen
  - Beziehung zu Modell „by reference“

Optionale Anforderungen

- Generelle Spezialisierung einer Metaklasse im Profile
- Konvention für graphische Definition von Stereotypes

Gliederung

- Aufgabenstellung
- Einführung in UML Profiles
- Anforderungen an UML Profiles
- UML Profiles in UML 1.4
- UML Profiles im CASE-Tool „Objecteering“ von Softeam
- Zusammenfassung und Ausblick

UML 1.4: Neuerungen bezüglich Profiles

- Profile: Mit Stereotype <<profile>> versehenes Package
  - Festlegungen zur Generalisierung
  - Beziehungen zu anderen Profiles analog Package
- <<modelLibrary>> für Package:
  - Modellbibliothek; enthält Modellelemente zur Wiederverwendung in anderen Packages
- <<modelLibrary>> für Dependency:
  - Zwischen <<modelLibrary>>-Package und <<profile>>-Package

UML 1.4: Neuerungen (2)

- <<appliedProfile>> für Dependency:
  - Zwischen Package (bzw. Subsystem oder Model) und <<profile>>-Package
  - Profile ist dann auf Package anwendbar
- {applicableSubset} für <<profile>>:
  - Auflistung der vom Profile verwendeten Metaelemente
- Überarbeitung von Stereotypes und Tagged Values

Offene Probleme

- Komposition von Profiles: Keine Behandlung möglicher Inkompatibilität
- Austauschformat: Graphische Repräsentation des Stereotypes
- Graphische Notation für Definition von Stereotypes: Nicht mit Metamodell vereinbar
- Generelle Spezialisierung einer Metaklasse im Profile: Nicht direkt möglich

Gliederung

- Aufgabenstellung
- Einführung in UML Profiles
- Anforderungen an UML Profiles
- UML Profiles in UML 1.4
- UML Profiles im CASE-Tool „Objecteering“ von Softeam
- Zusammenfassung und Ausblick

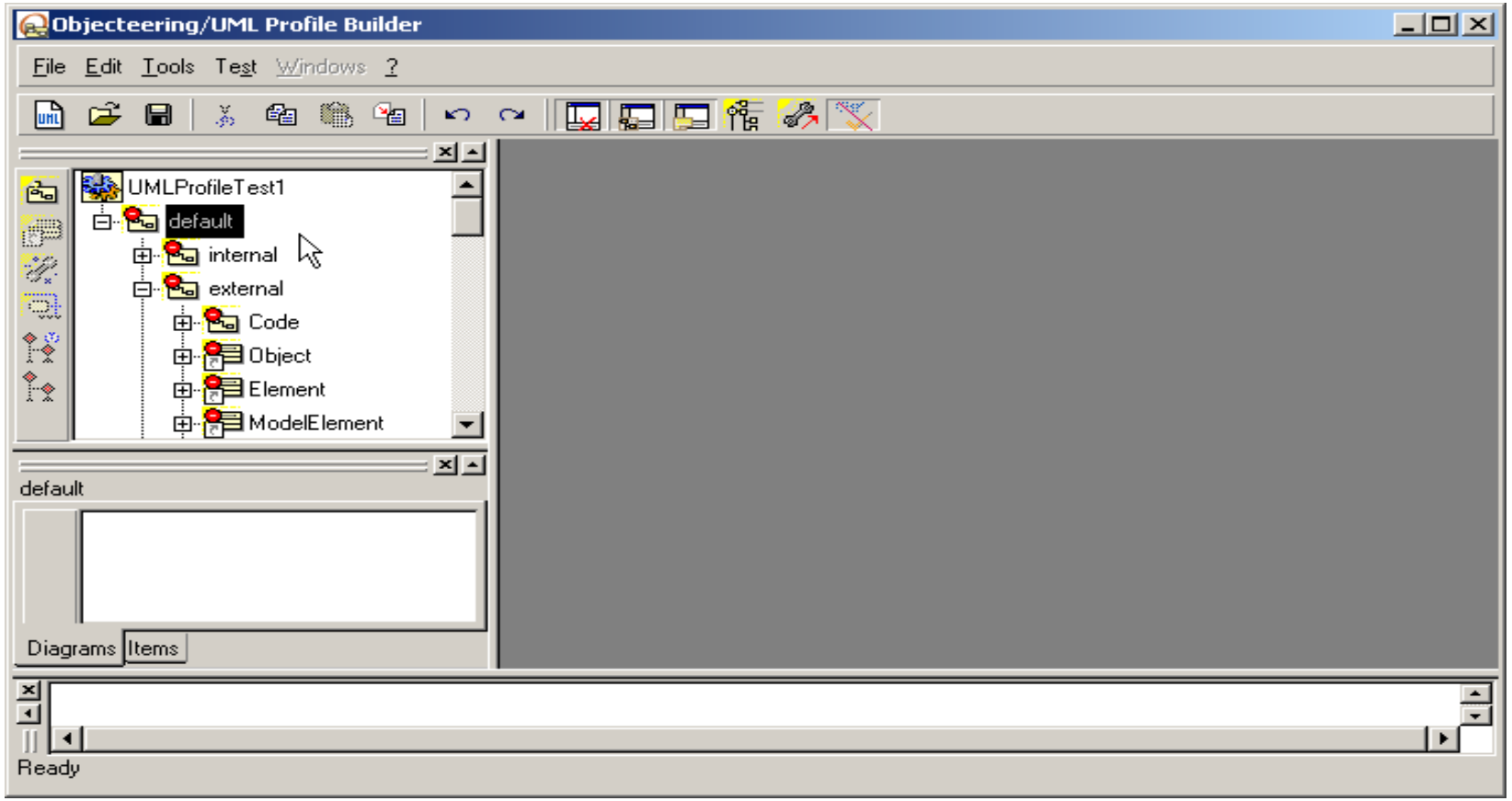
Objecteering

- Wichtigstes CASE-Tool bezüglich Profile-Unterstützung
- Hersteller: Softeam (Frankreich)
- Version: 5.10
- Objecteering UML Profile Builder:

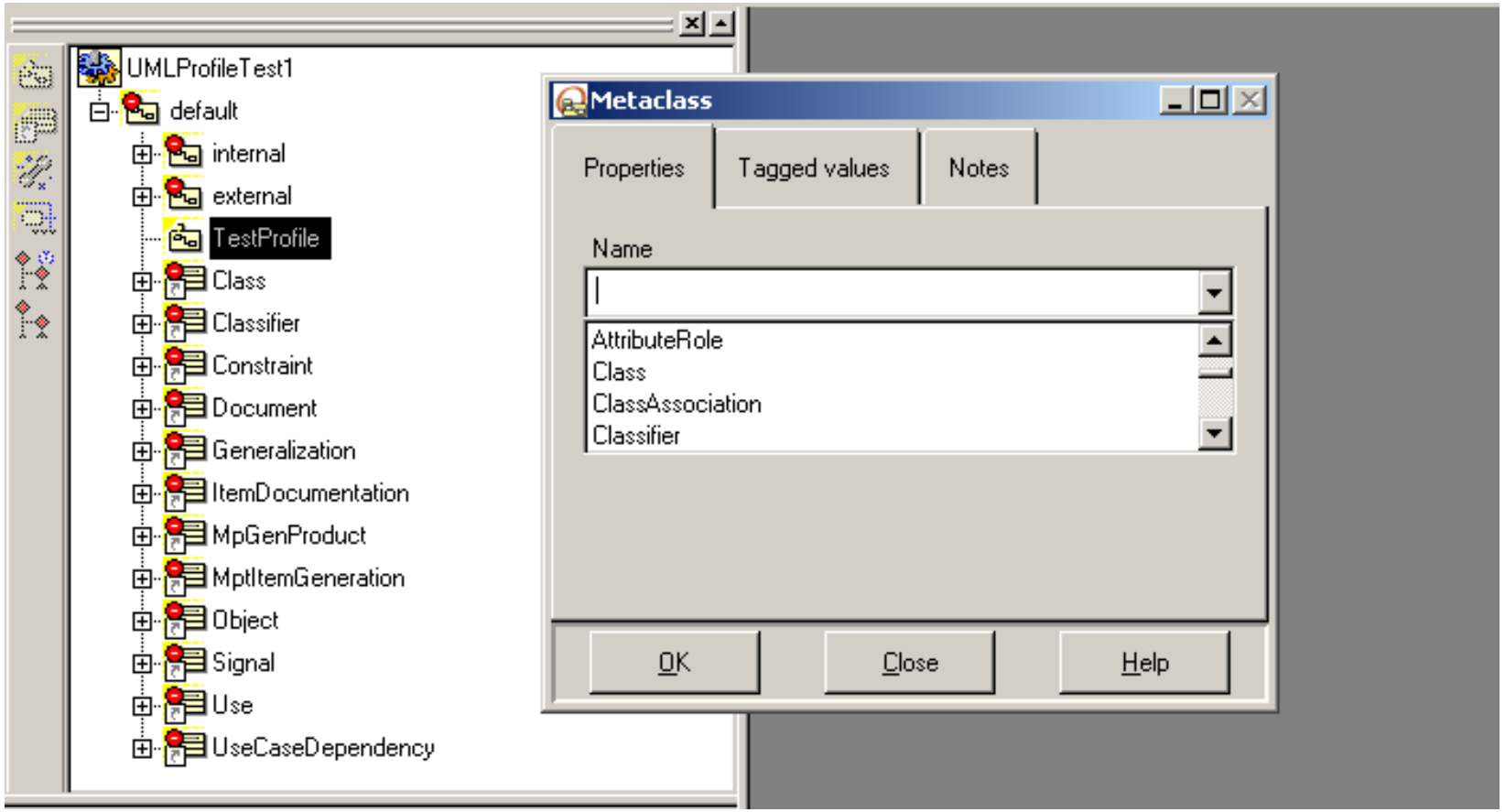
  Erstellung von UML Profiles
- Objecteering UML:

  Erstellung von Modellen und Generierung von Dokumentation und Code unter Verwendung von UML Profiles

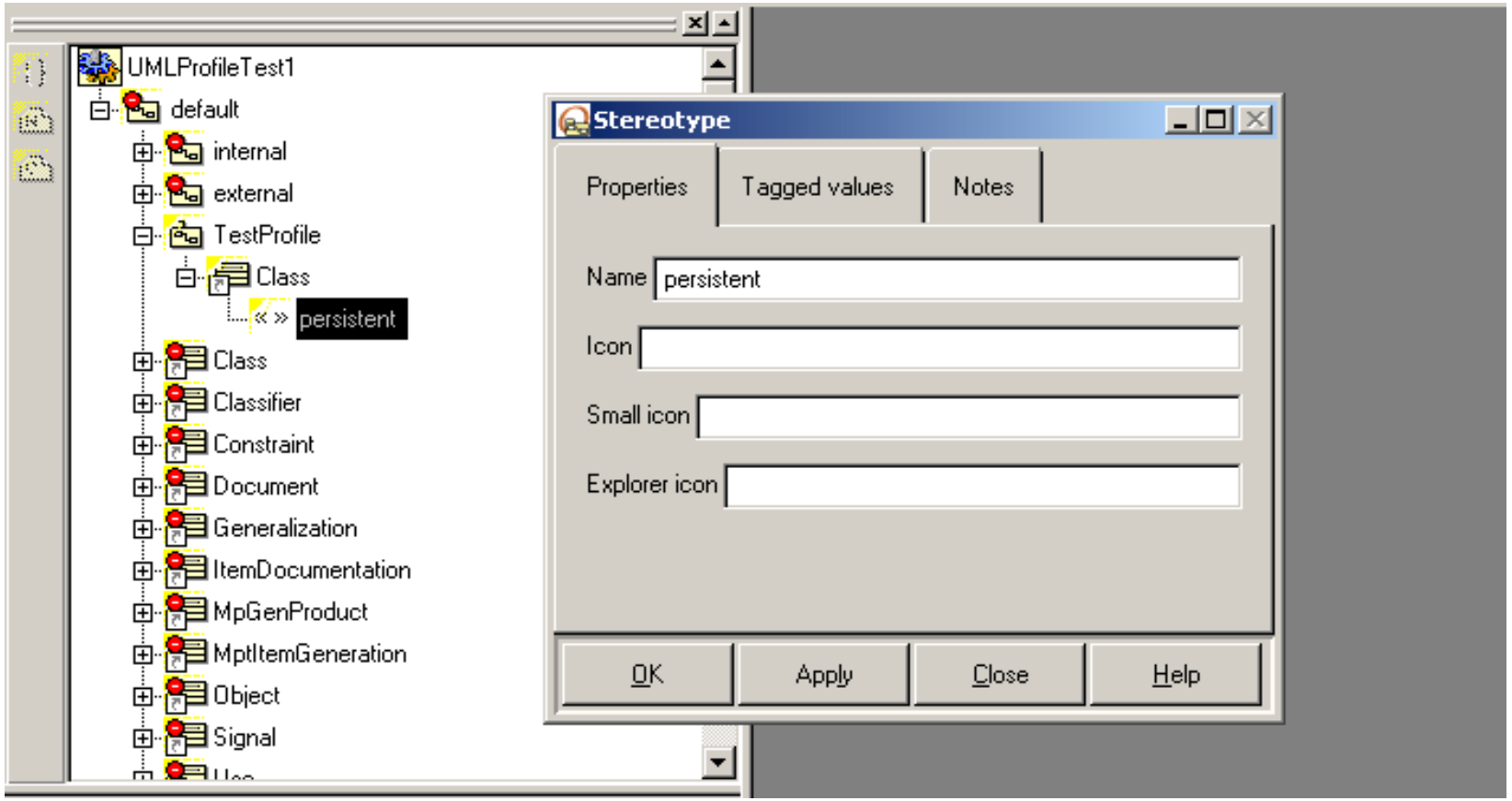
Erstellung eines Profiles



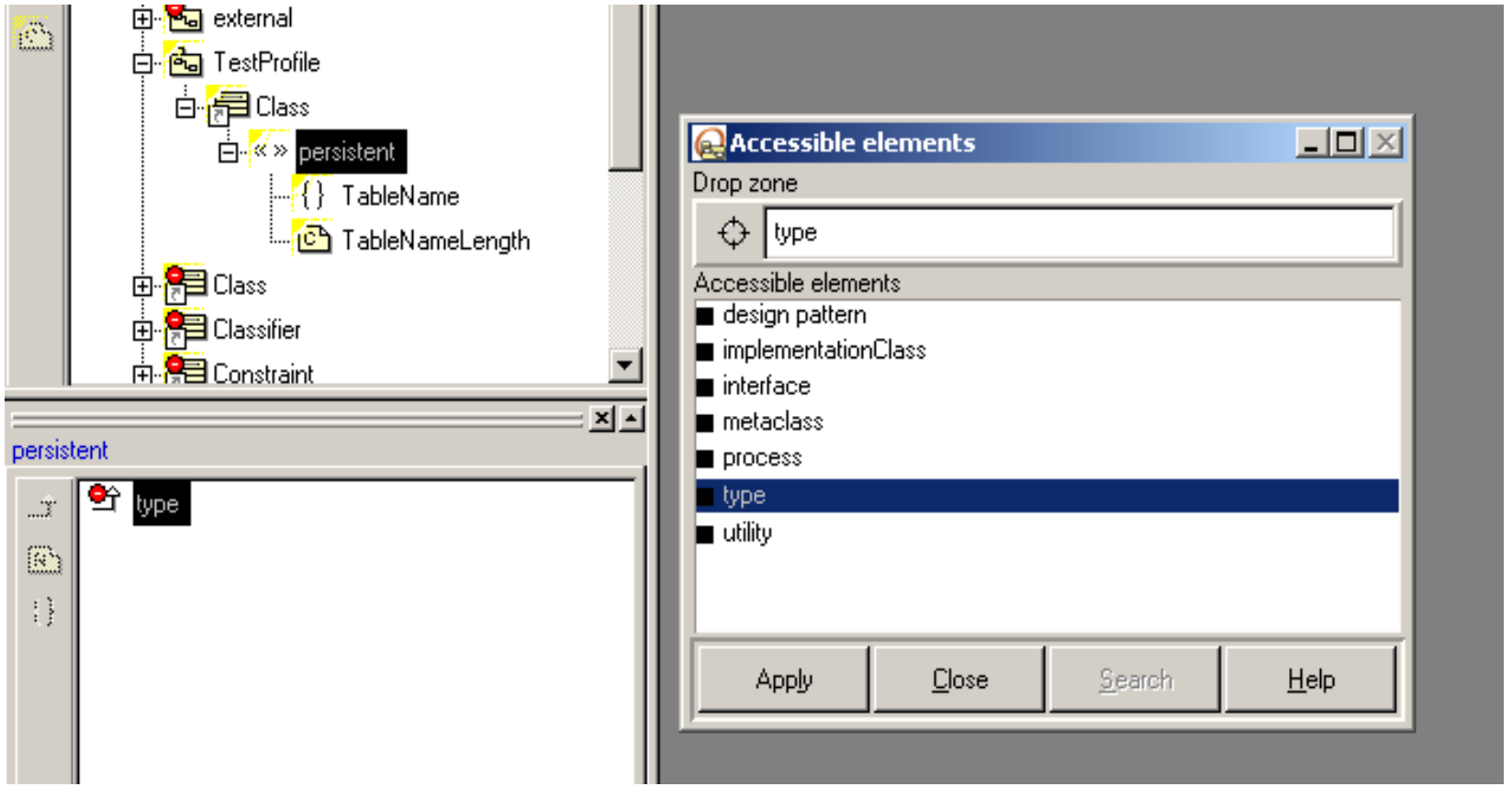
Referenzen auf Metaklassen



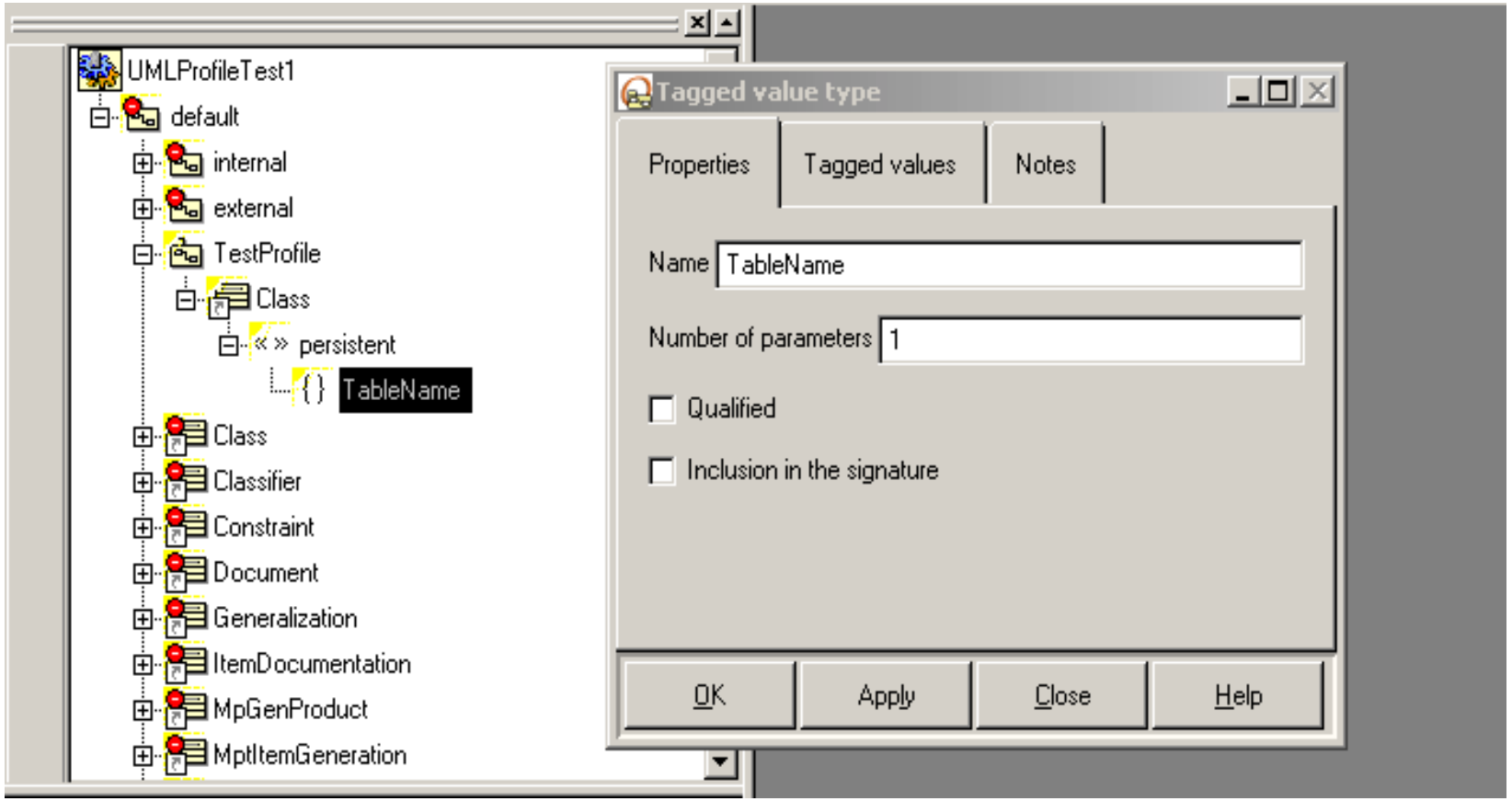
Definition eines Stereotypes



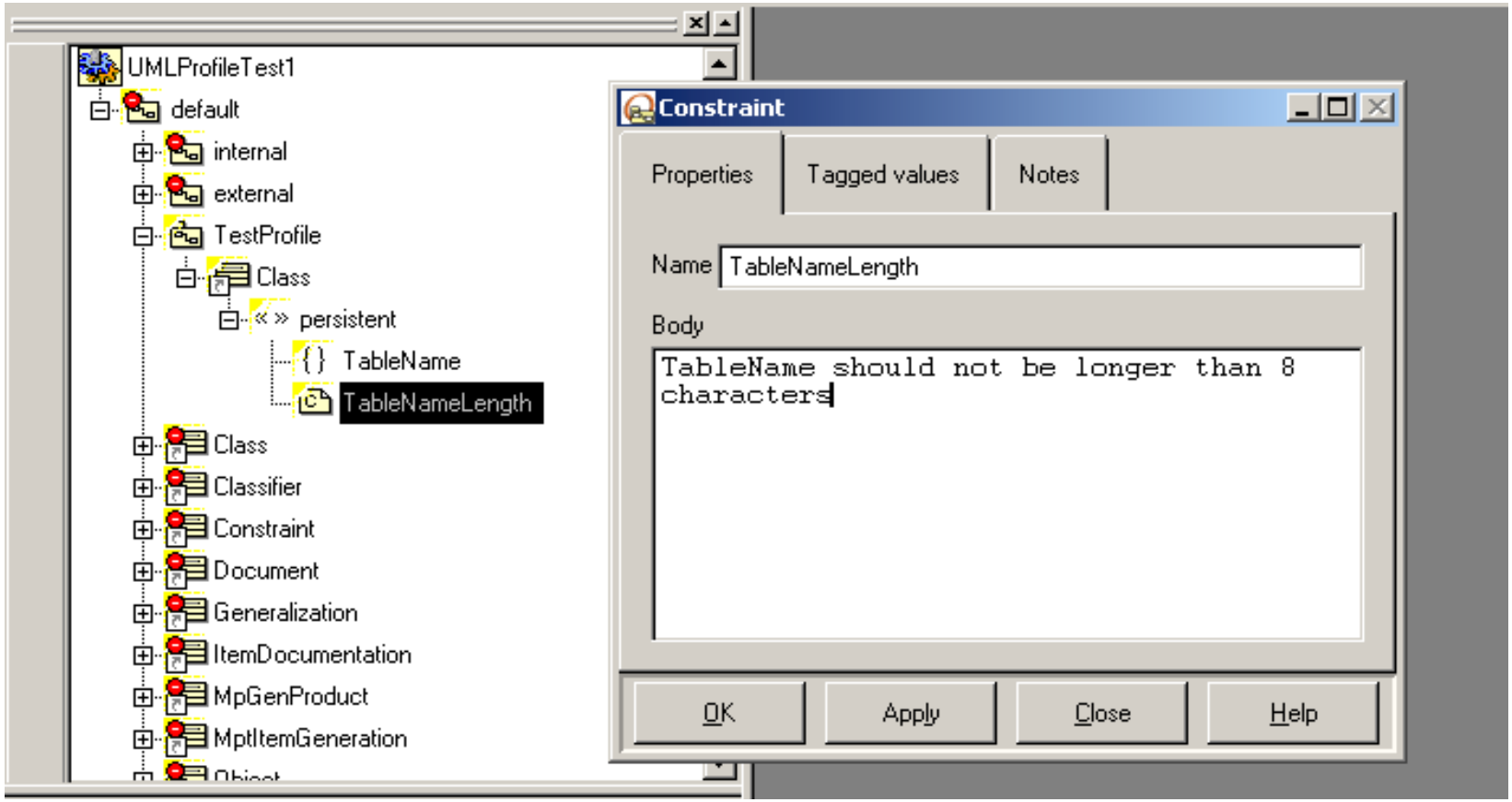
Spezialisierung von Stereotypes



Definition eines Tagged Value Types



Definition einer Constraint



Zusätzliche Erweiterungsmöglichkeiten

- Motivation: Anpassen des CASE-Tools
  - Funktionalität während Modellierung
  - Bei Genierung von Code und Dokumentation

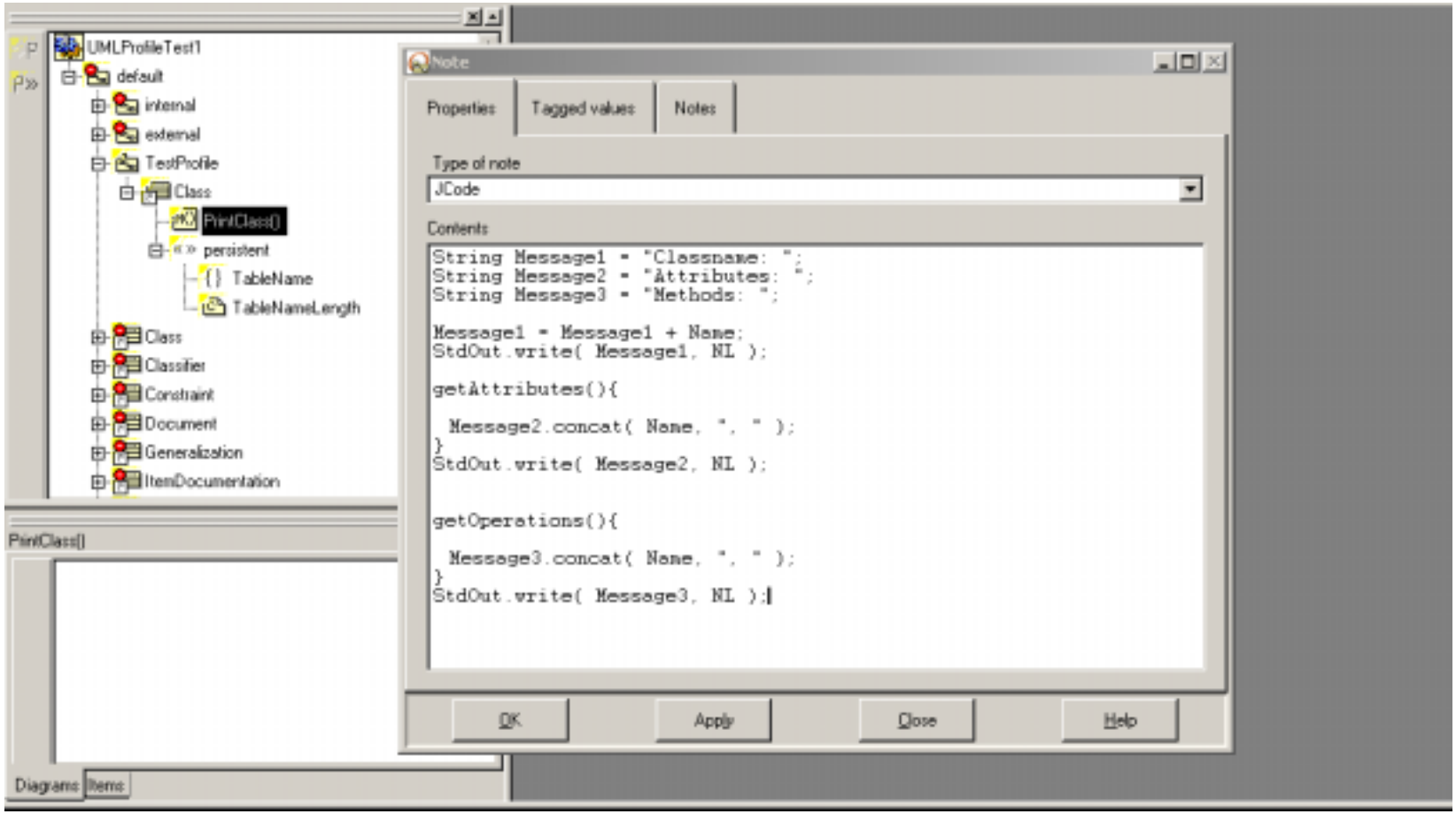
UML Profile Builder: J Language

- Einfache Java-ähnliche Sprache zur Spezifikation von J Methods
- Bietet Zugriff auf alle Metamodellelemente
- J Methods: Methoden, die Metaklassen zugeordnet werden
- J Attributes:
  - Metaattribute für Metaklassen
  - Zur Verwendung von J Methods

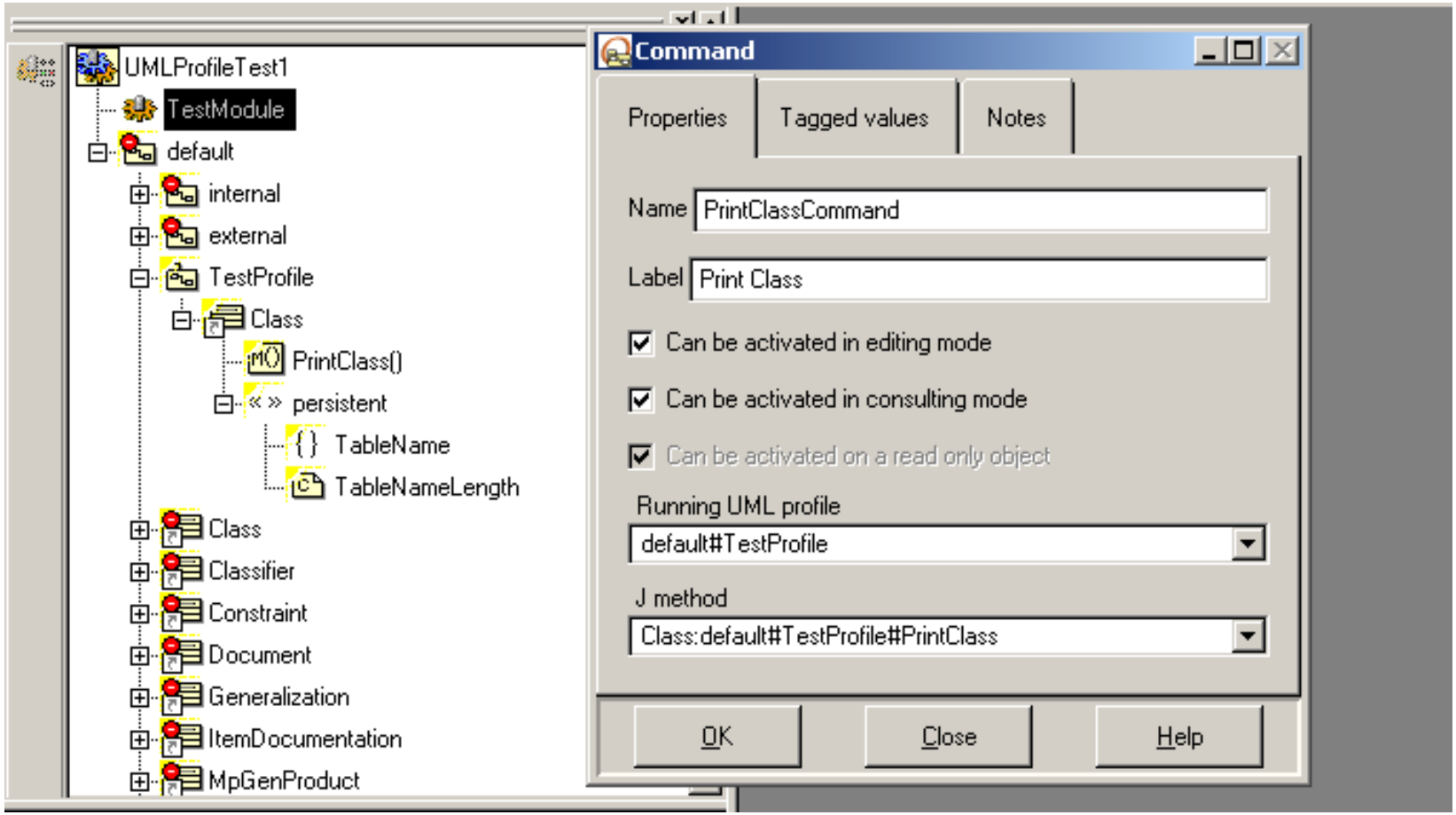
UML Profile Builder: Module

- Enthält ein oder mehrere Profiles
- Anwendung und Austausch nur von Modules
- Festlegung von zusätzlichen Details für die Anwendung des Profiles
- Für Modellierung Auswahl beliebig vieler Modules möglich

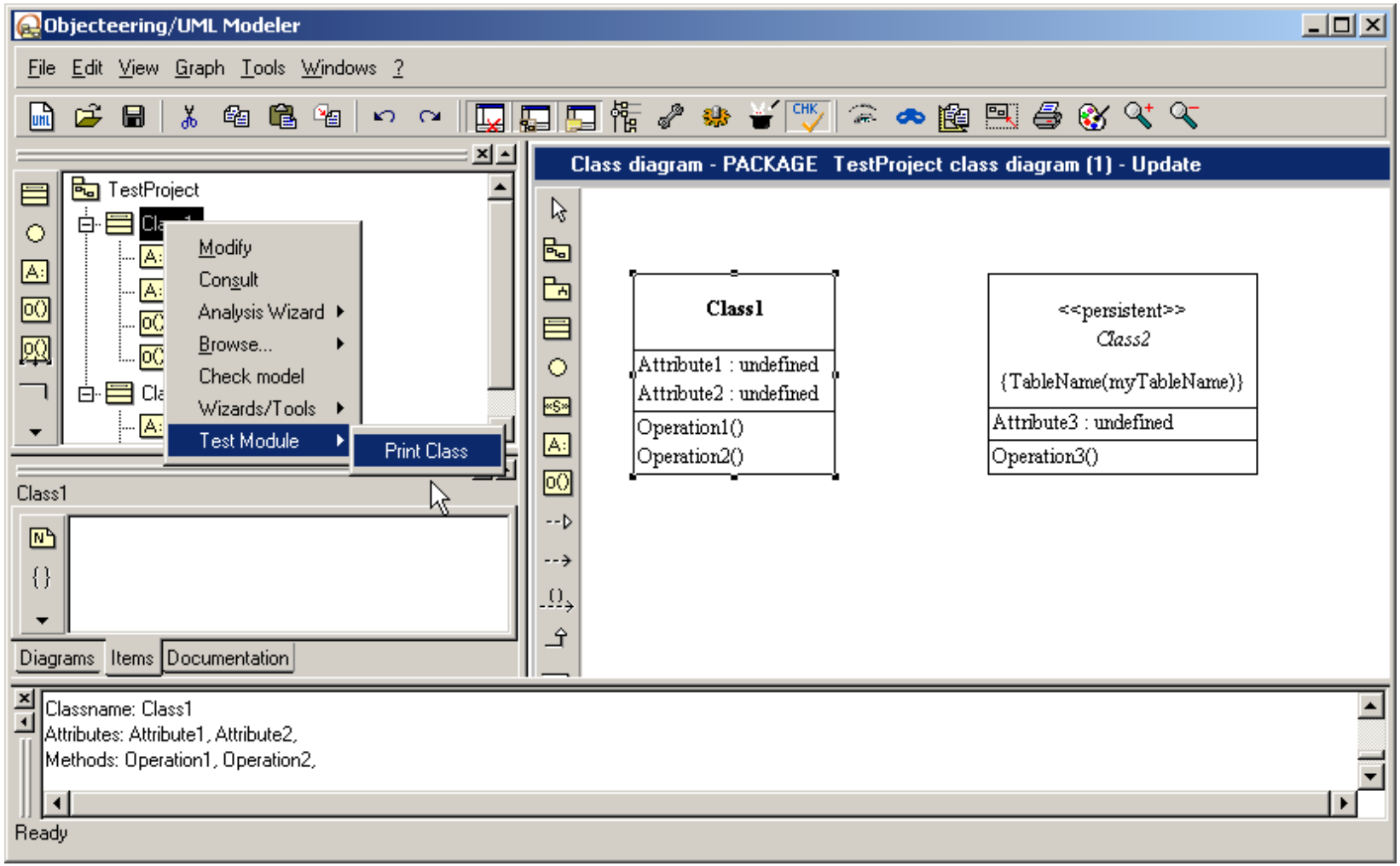
Definition einer J Method



Definition eines Commands



Objecteering UML



UML Profile Builder: Weitere Funktionalität

- Parameter zur Konfiguration des Profiles
- Work Products:
  - Repräsentieren externe Ergebnisse von Generierungsprozessen
  - Bsp.: Datei mit Code oder Dokumentation
- Generation Templates und Dokumentation Templates:
  - Beschreiben Struktur von textbasierten Objekten
  - Bestehen aus hierarchischem Baum von Textbausteinen
- Test von Modules direkt im UML Profile Builder möglich

Objecteering: Zusammenfassung

- UML Standard-Erweiterungselemente:
  - An UML 1.3 orientiert
  - Keine volle Unterstützung (z.B. keine Mehrfachvererbung von Stereotypes)
  - Keine graphische Repräsentation der Profile-Definitionen
- Breite proprietäre Unterstützung von Profiles zur Generierung von Dokumentation und Code

Gliederung

- Aufgabenstellung
- Einführung in UML Profiles
- Anforderungen an UML Profiles
- UML Profiles in UML 1.4
- UML Profiles im CASE-Tool „Objecteering“ von Softeam
- Zusammenfassung und Ausblick

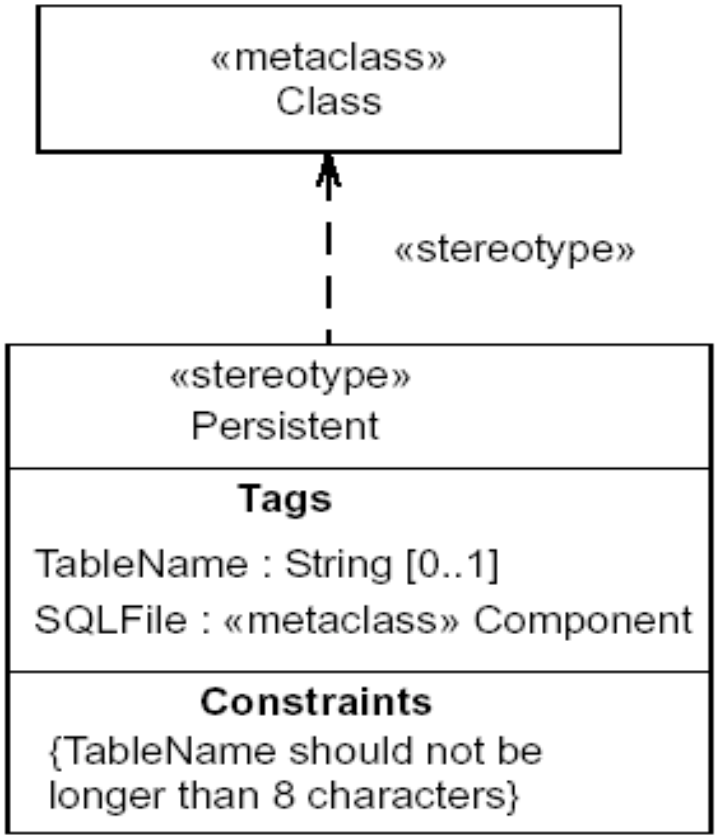
Zusammenfassung und Ausblick

- Unzureichende Festlegungen für:
  - Austauschformat
  - Generelle Spezialisierung einer Metaklasse im Profile
  - Graphische Stereotype-Definition
  - Komposition von Profiles
- Verbleibende Aufgaben:
  - Auswahl eines CASE-Tools
  - In engerer Wahl: Together (API), ArgoUML (Quellcode)
  - Empfehlungen zur Erweiterung des Tools und prototypische Umsetzung

UML 1.4: Änderungen

- Stereotypes:
  - Einem Modellelement Zuweisung mehrerer Stereotypes möglich
  - Mehrere Modellelemente als BaseClass
  - Konvention für graphische Definition von Stereotypes
- Tagged Values:
  - Metaklasse „Tag Definition“ zur Spezifizierung von Name, Typ und Anzahl der Werte
  - Typ: Datentyp oder Referenz auf Metaklassen
  - Soll nur noch Stereotypes zugeordnet sein

Graphische Definition eines Stereotypes



UML Profile Builder: Module verwendet Profiles

- Arten der Verwendung :
  - „reference“: Zugriff auf alle Elemente des Profiles
  - „use“: Zugriff nur auf Stereotypes, Tagged Value Types und Note Types des Profiles
  - „Installation Profile“: Profile, das Methoden für Umgang (Empfangen, Installation, ...) mit Module spezifiziert

UML Profile Builder: Module verwendet Profiles