

Code Generieren mit UML2



Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>



Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Über uns



- Philipp Auersperg
 - Mitbegründer der Bluedynamics
 - seit 20 Jahren in der objektorientierten Softwareentwicklung
 - Seminare für Beratungsfirmen über UML
 - Initiator des ArchGenXML Projektes (Vorgänger von AGX)
 - seit den Anfangszeiten von Plone in der Community aktiv
- Robert Niederreiter
 - Seit 2005 selbstständiger Software Entwickler
 - Mitglied der BlueDynamics Alliance
 - Initiator des AGX Projekts
 - hat maßgeblich bei der Entwicklung von ArchGenXML mitgewirkt
- BlueDynamics Alliance
 - Kooperation von 8 Firmen der DACH Region

Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>



Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Begriffsklärung



- Plone Content Types
- Archetypes
- Dexterity (Designierter Nachfolger von AT)
- UML
- ArchGenXML
- Python Eggs
- Buildout

Geschichte & Lessons Learned



- 2003 startet Vorgängerprojekt ArchGenXML mit dem aus UML1 Diagrammen Plone Produkte mit Archetypes generiert werden können („Produktive Faulheit“)
- ArchGenXML verbreitet sich schnell, die Funktionalität wird ständig erweitert, ist aber auf Plone-Archetypes fixiert
- Mit Python Eggs, GenericSetup, und Dexterity kommt ArchGenXML aufgrund seiner statischen Strukturen an seine Grenzen in Sachen Erweiterbarkeit
- 2008 startet AGX, einem erweiterbaren Ansatz basierend auf UML2

Geburtsstätte von ArchGenXML



Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>



Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Motivation



- Boilerplate Code erstellen braucht Zeit
- Copy & Paste ist fehleranfällig
- Statische Templategeneratoren wie Zopeskel sind unflexibel und erzeugen oft nicht lauffähigen Code
- Best-Practice Code

Vorteile



- Grafische Dokumentation der Entitäten
- Niedrige Einstiegshürde für Plone Ein- und Umsteiger
- Prozesse und Strukturen können in UML elegant gemeinsam mit dem Kunden erarbeitet werden
- Generatoren können gut an aktuellen Best-Practice Standards angepasst werden.
- Update der Software durch erneutes generieren des Modells
- Dadurch wenig oder nur geringfügige „Handarbeit“ notwendig, lästiger „Fitzelkram“ beim Projektsetup bleibt erspart

Nachteile



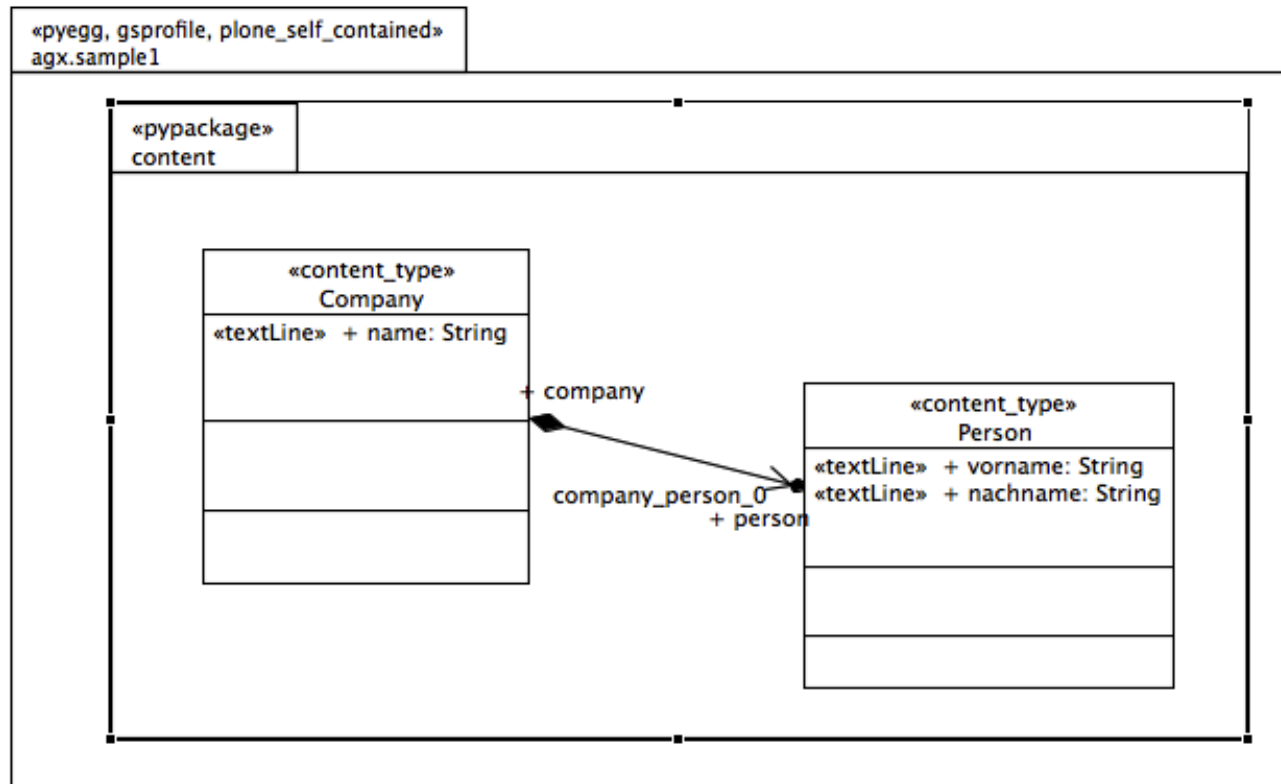
- Kollaboration erfordert von allen Kontributoren den Einsatz ähnlicher Tools (zumindest einen UML2-konformen Editor)
- UML Modelle können in der Versionsverwaltung nicht zusammengeführt werden, dadurch keine parallele Bearbeitung des Modells möglich - zeitliche Abstimmung der Beteiligten erforderlich

UML2 und UML-Profile



- Die Unified Modelling Language (UML) ist eine grafische Modellierungssprache zur Spezifikation, Konstruktion und Dokumentation von Software-Teilen und anderen Systemen
- Die UML2 ist selbst im UML2-Metamodell durch Metaklassen beschrieben
- Mit UML-Profilen kann man das UML2-Metamodel erweitern, um die Semantik einer bestimmten Anwendungsdomäne zu formalisieren

Einfaches Modell-Beispiel



Stereotypen



- Die Erweiterung des UML2-Metamodells erfolgt über Stereotypen
- Die Stereotypen werden im UML-Profil definiert
- Ein Stereotyp spezifiziert, wie eine bereits im UML2-Metomodell definierte Metaklasse für eine Anwendungsdomäne erweitert werden kann
- Ein Stereotyp kann Eigenschaften (Tagged Values) besitzen

AGX und UML-Profile



- AGX besteht aus einer Reihe von Generatoren
- Diese Generatoren sind über ein UML-Profil an die jeweilige Anwendungsdomäne gebunden
- Jeder Generator ist verantwortlich, den generierten Code um seine anwendungsspezifischen Fragmente zu erweitern
- Beim Erstellen von UML-Modellen kann man nun durch Anwenden ein oder mehrerer UML-Profile das Modell um einzelne Anwendungsdomänen erweitern
- Das Steuern der Generatoren erfolgt über die Anwendung der Stereotypen auf Modell-Elemente, die in den UML-Profilen spezifiziert sind

Anwendung von AGX



- Grundsätzlich zwei Möglichkeiten
 - AGX als Kommandozeilentool
 - benötigt lediglich UML2-Tool
 - AGX als Eclipse-Plugin
 - Eclipse Indigo
 - Eclipse Modeling Tools
 - Papyrus UML Editor
 - AGX stellt Wizard zur Verfügung
 - Generiert auf Knopfdruck lauffähigen Code

AGX Eclipse Wizard



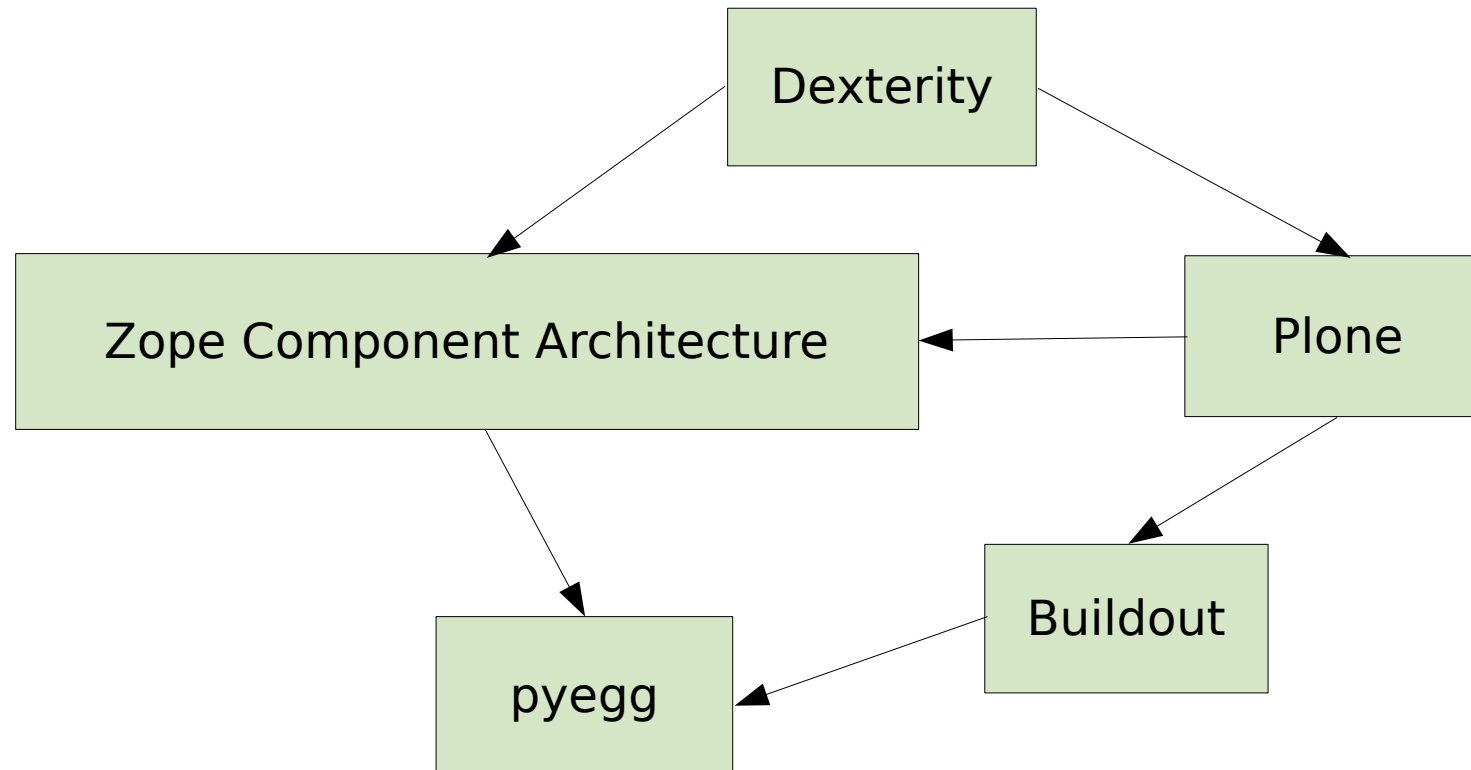
- Stellt eine Auswahl von Modell-Vorlagen zur Verfügung
 - Python Egg
 - Plone Product
 - Dexterity Product
 - Dexterity Example

Auswahlkriterien für UML Editor



- Papyrus ist Open Source
- Konsistente UML2 Unterstützung
- Eclipse Integration
- Aktives Projekt
 - Papyrus befindet sich in der Inkubations-Phase und wird somit Teil des offiziellen Eclipse-Projekts
 - Ist über Eclipse Modeling Edition installierbar

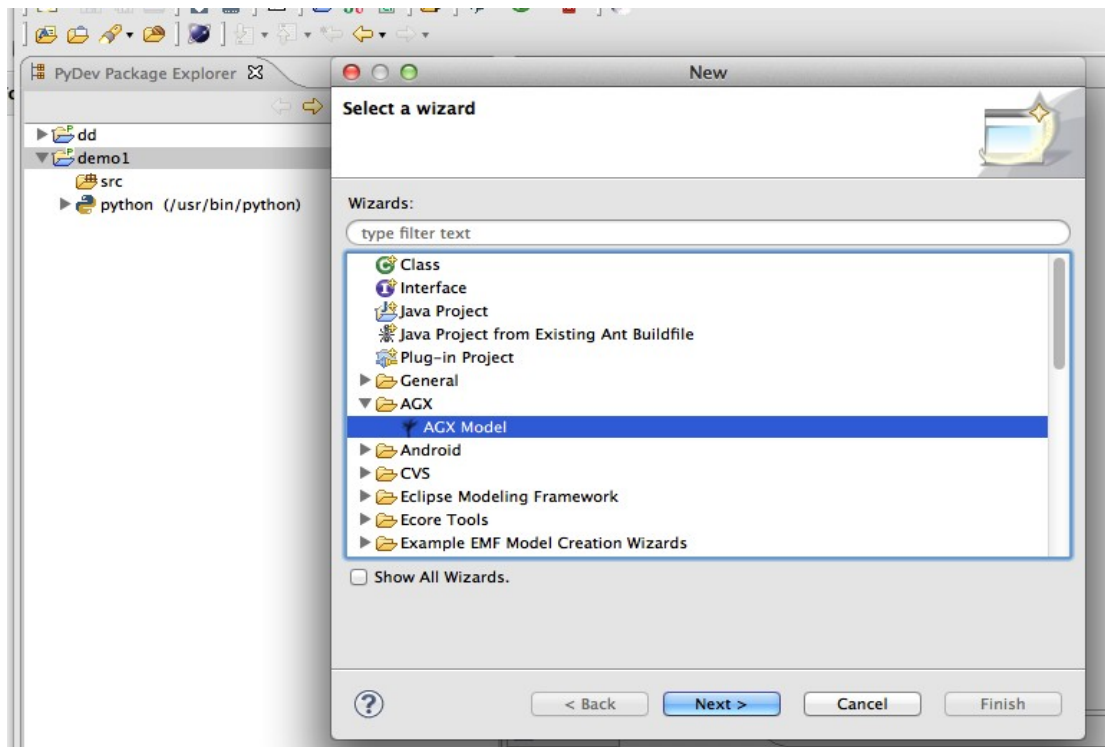
Abhängigkeiten der Generatoren



Projekt und Modell einrichten



- Aufruf des Modell-Wizards



Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>

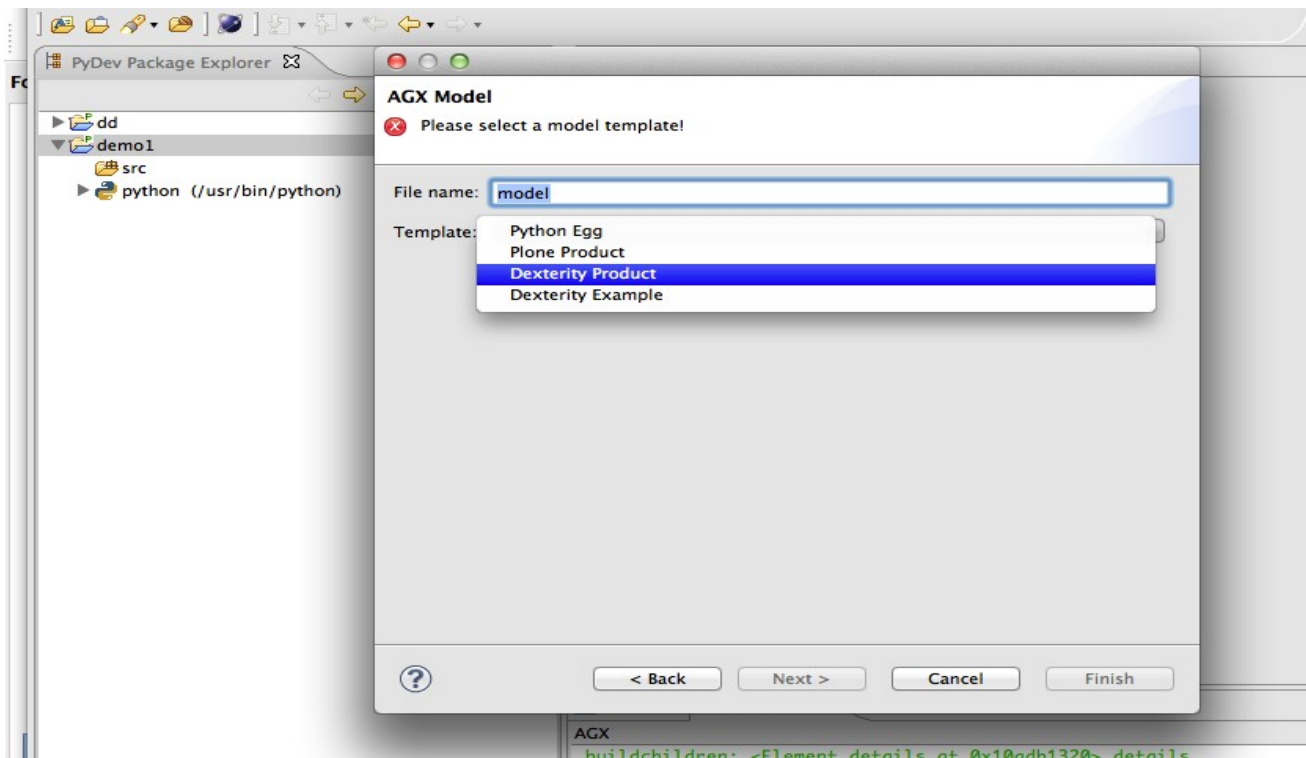


Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Der Modell-Wizard



- Modellvorlage auswählen



Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>

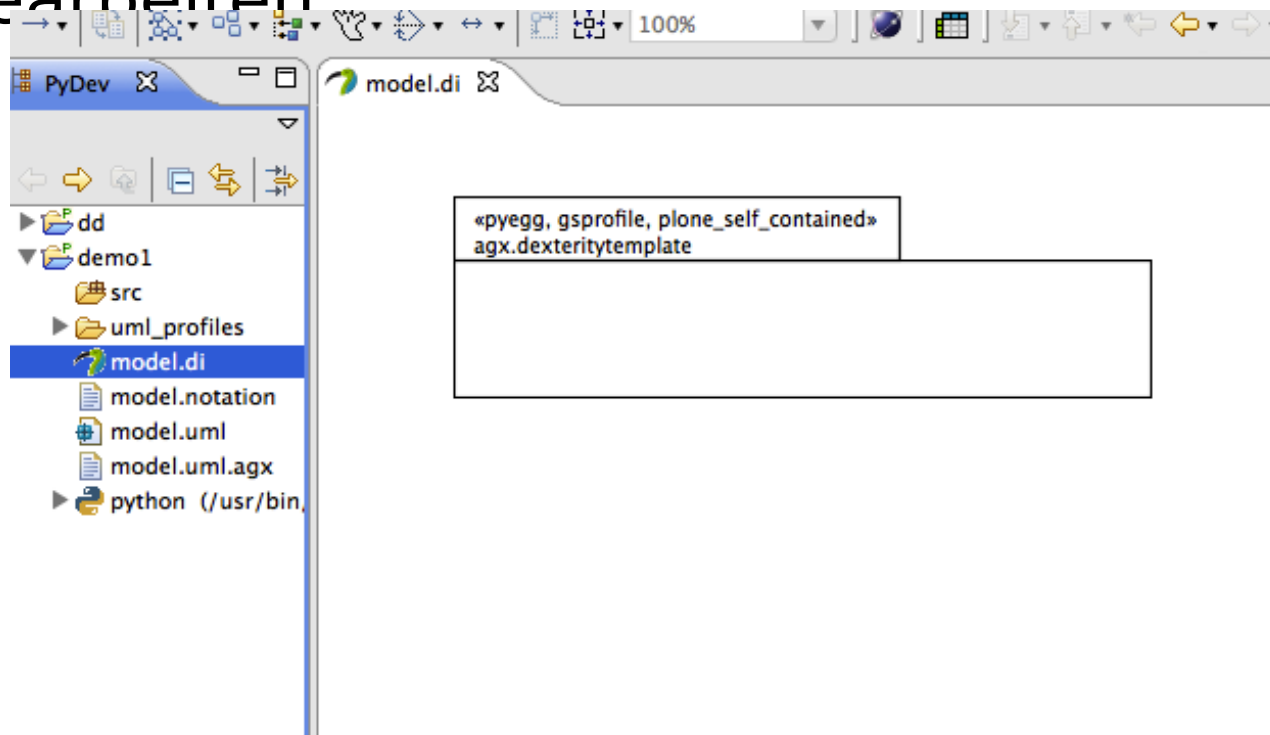


Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Dexterity Modell-Vorlage



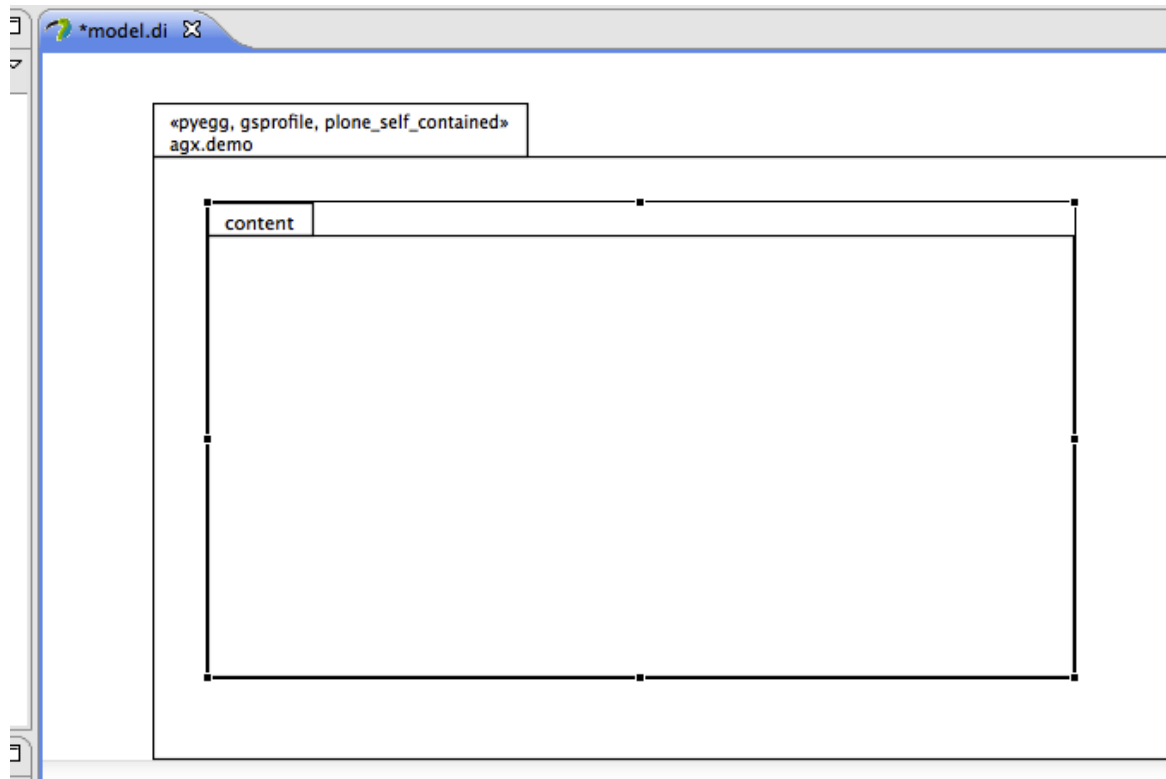
- Das frisch erstellte Modell ist bereit zum Bearbeiten



Modell erweitern



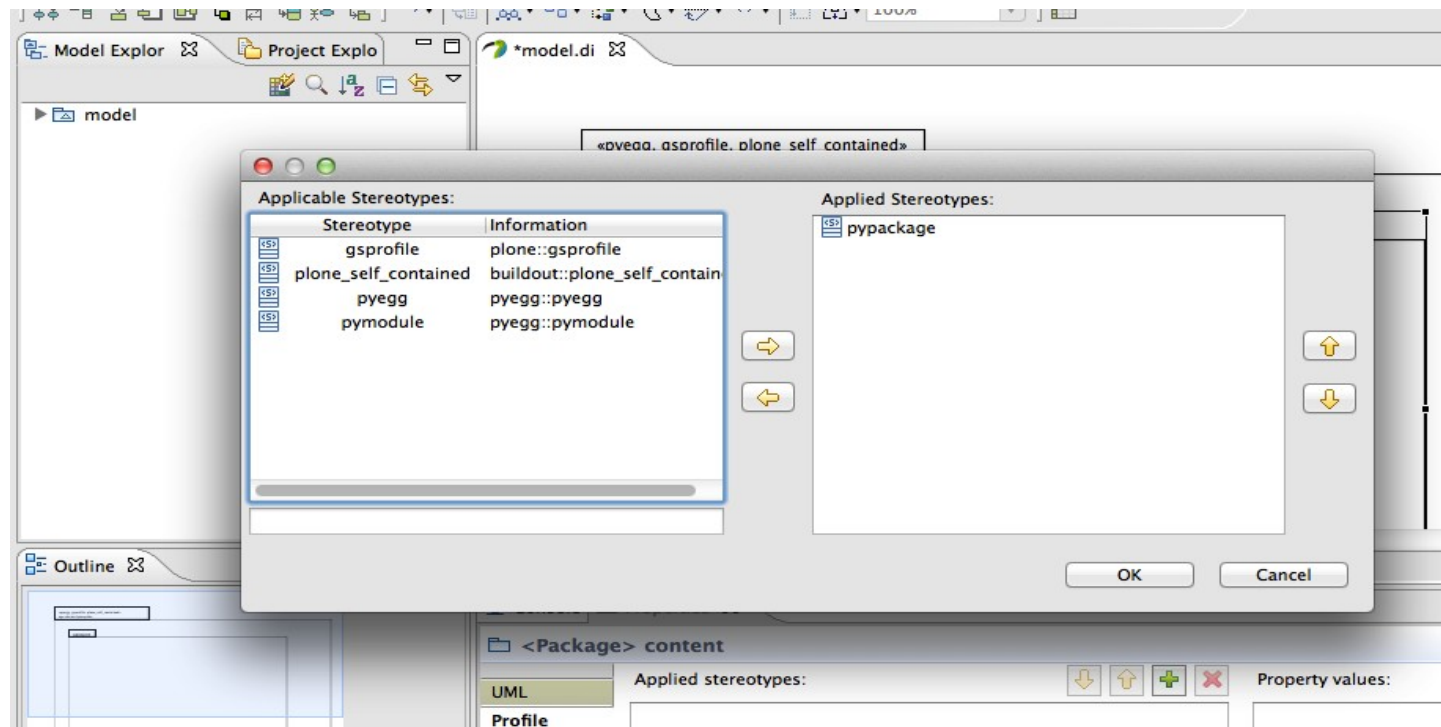
- Erstellen eines Packages im Python Egg



UML Package wird Python Package



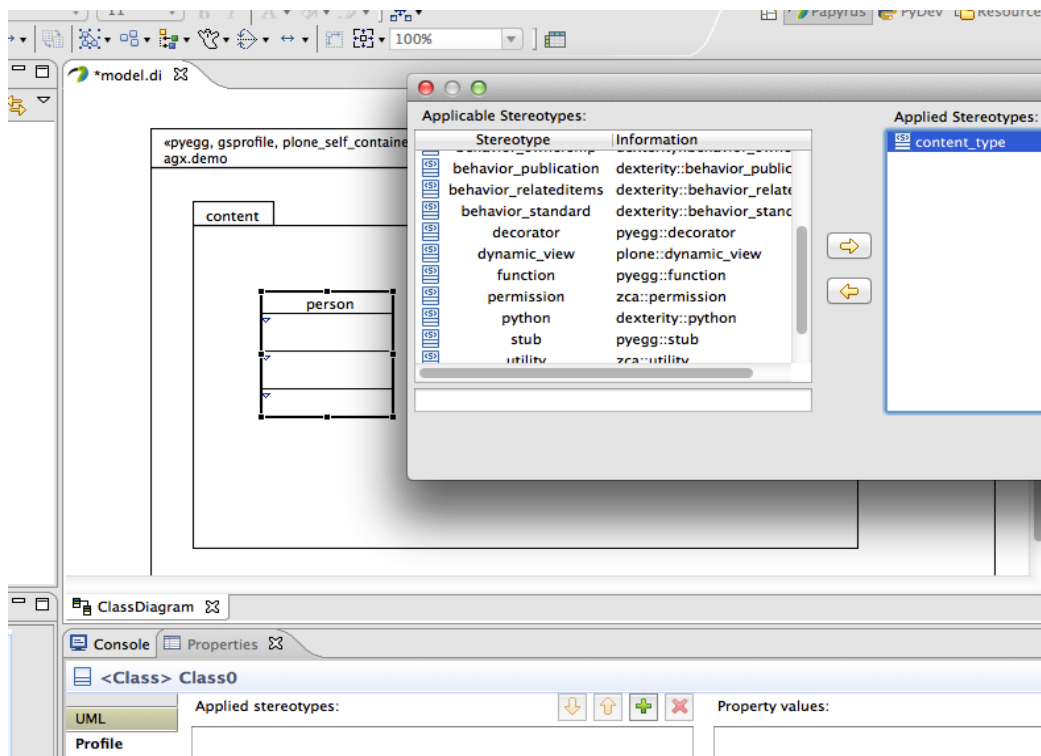
- Durch Anwenden des Stereotyps <<pypackage>> wird das UML Package ein Python Package



Contenttype-Klasse anlegen



- Durch Anwenden des Stereotyps <<content_type>> wird die UML Klasse zum Contenttype



Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>

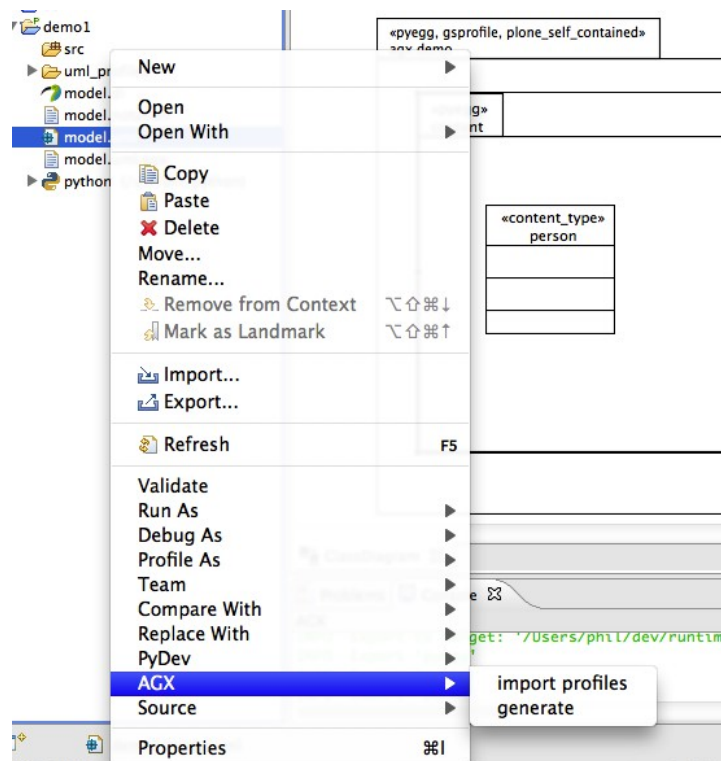


Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Generator ausführen



- Über das Kontextmenü der UML-Datei wird der Generator gestartet und der Sourcecode generiert



Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>



Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Ausblick



- Implementierung weiterer Generatoren
 - Generator-Generator
 - Workflows
 - SQLAlchemy
 - Buildout (vom Deployment-Diagramm)
 - Pyramid
 - etc...
- Bedienungskomfort verbessern
 - Automatische Installation von AGX über das Eclipse-Plugin

Ressourcen



- <http://agx.me>
- <http://github.com/bluedynamics/agx.dev>
- <http://bluedynamics.eu>

Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>



Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz

Nerds Are Artists

- Donations welcome



Philipp Auersperg <phil@bluedynamics.com>
Robert Niederreiter <office@squarewave.at>



Creative Commons Namensnennung-
Keine kommerzielle Nutzung-
3.0 Österreich Lizenz