

Faktor-IPS UML-Integration Tutorial

(Dokumentversion: 171 | Version der Faktor-IPS UML-Integration: v1.0 | Datum: 06.11.08)

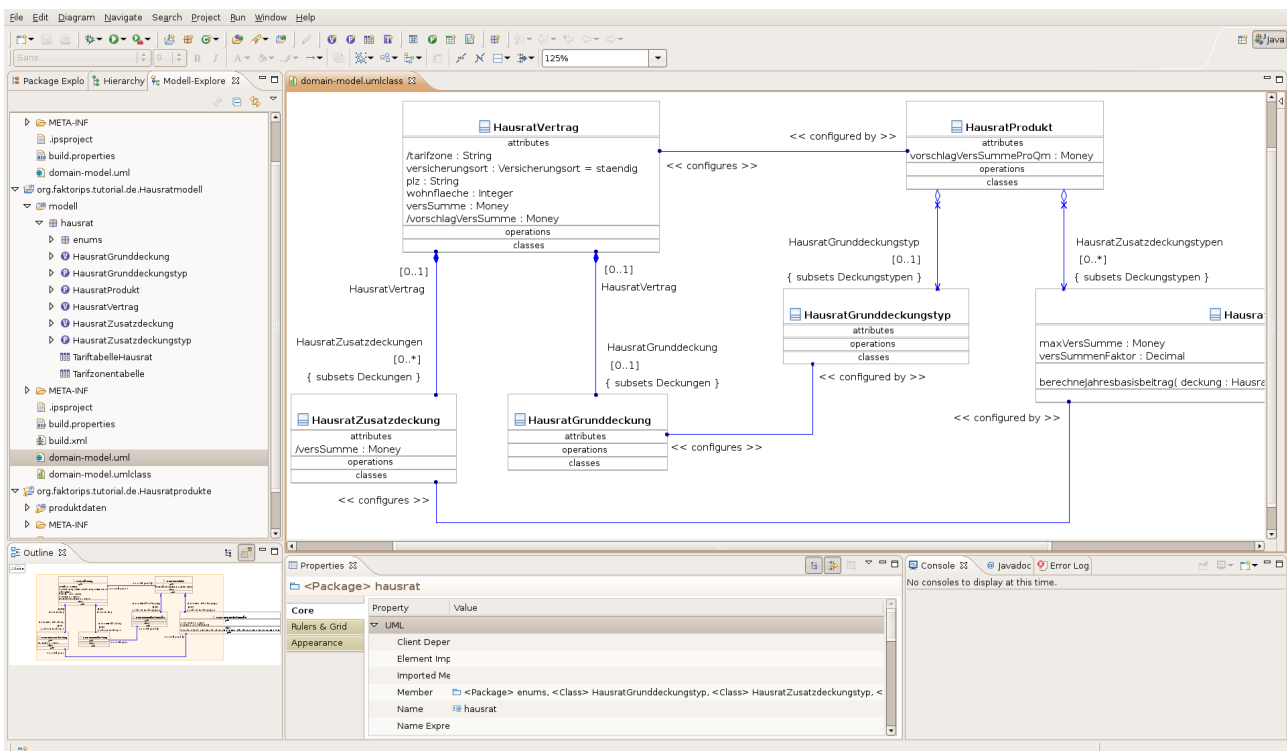


Abbildung 1: Faktor-IPS UML-Integration, verwendet mit Eclipse UML2-Tools

Inhaltsverzeichnis

1 Einleitung.....	3
1.1 Was ist die Faktor-IPS UML-Integration.....	3
1.2 Was ist Eclipse UML2.....	3
1.3 Inhalt und Aufbau dieses Tutorials.....	4
1.4 Wie Sie an die Tutorial-Projekte gelangen.....	5
 2 Installation.....	 6
 3 UML-Modelle für Faktor-IPS Modelle generieren.....	 7
3.1 UML-Modelle generieren.....	7
3.2 UML-Modelle aktualisieren.....	8
 4 UML-Diagramme erzeugen.....	 9
4.1 Welche Rolle haben Diagramme.....	9
4.2 UML-Diagramme erzeugen.....	10
 5 UML-Diagramme layouten und bearbeiten.....	 12
5.1 Elemente anordnen.....	12
5.2 Diagramm-Notizen hinzufügen.....	13
5.3 Elemente ausblenden.....	13
5.4 Elemente bearbeiten.....	13
5.5 Elemente aus anderen UML-Packages einblenden.....	14
5.6 Diagramme als Bild-Dateien exportieren.....	14
 6 Ausblick.....	 15
6.1 Faktor-IPS UML-Profil.....	15
6.2 UML-Modell → Faktor-IPS-Modell.....	15
6.3 Automatisches Synchronisieren.....	15

1 Einleitung

1.1 Was ist die Faktor-IPS UML-Integration

Die UML (Unified Modeling Language)¹ ist eine standardisierte Modellierungssprache, die in der Software-Entwicklung breite Anerkennung findet und einen hohen Bekanntheitsgrad erreicht hat. Mit der Faktor-IPS UML-Integration soll Faktor-IPS Entwicklern jetzt die Darstellung (und später auch die Bearbeitung) von Faktor-IPS Modellen mittels UML ermöglicht werden.

Die Faktor-IPS UML-Integration basiert auf dem *Eclipse UML2 Projekt*².

1.2 Was ist Eclipse UML2

Eclipse UML2 ermöglicht es, UML-Modelle über ein API zu erstellen und zu modifizieren. Diese Modelle können im XMI-Format persistiert werden.

Eclipse UML2 bringt außerdem einen einfachen Editor mit, mit dem vorhandene UML-Modelle eingesehen und auch bearbeitet werden können. Dabei werden die Modelle als einfache Baumstruktur angezeigt. Eine grafische Ansicht bzw. grafische Bearbeitung der Modelle ist dagegen ausdrücklich nicht Bestandteil des Eclipse UML2 Projektes.

Grafische Editoren werden von verschiedenen anderen Plugins zur Verfügung gestellt, die wiederum auf Eclipse UML2 aufsetzen³. Beispiele für solche Plugins sind unter anderen:

- Eclipse UML2 Tools⁴
- Papyrus⁵
- MOSKitt⁶
- TOPCASED UML⁷

1 <http://www.omg.org/UML/>

2 <http://www.eclipse.org/uml2/>

3 Siehe <http://wiki.eclipse.org/MDT-UML2-Tool-Compatibility> für eine ausführliche Liste UML2-kompatibler Tools

4 <http://wiki.eclipse.org/MDT-UML2Tools>

5 <http://www.papyrusuml.org/>

6 <http://www.moskitt.org/eng/moskitt0/>

7 <http://www.topcased.org>

Die aktuelle Version der Faktor-IPS UML Integration ermöglicht es bereits, Faktor-IPS Modelle in UML-Modelle zu konvertieren und im XMI Format zu speichern. Hierzu wird das API von Eclipse UML2 verwendet. Die folgende Abbildung zeigt die Zusammenhänge:

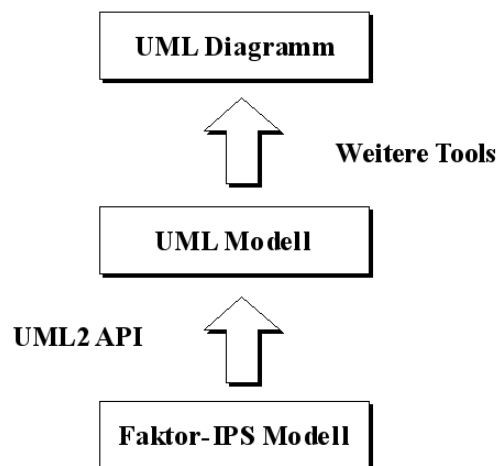


Abbildung 2: Zusammenhänge Faktor-IPS Modell, UML Modell, UML Diagramm

1.3 Inhalt und Aufbau dieses Tutorials

Mit diesem Tutorial möchten wir Ihnen zeigen,

- wie Sie die Faktor-IPS UML-Integration **installieren** (Kapitel 2).
- wie Sie **UML-Modelle** aus Faktor-IPS Modellen **generieren** (Kapitel 3).
- wie Sie für die generierten UML-Modelle **UML-Diagramme erzeugen** (Kapitel 4).
- wie Sie die erzeugten **UML-Diagramme layouten und bearbeiten** (Kapitel 5).
- was **für künftige Versionen** der Faktor-IPS UML-Integration **geplant** ist (Kapitel 6).

Zur grafischen Bearbeitung der Diagramme verwenden wir in diesem Tutorial das Plugin *Eclipse UML2 Tools*, da dieses Plugin bereits in der Modeling Distribution von Eclipse enthalten ist.

1.4 Wie Sie an die Tutorial-Projekte gelangen

Als konkretes Fallbeispiel verwenden wir in diesem Tutorial das Faktor-IPS Modell aus dem Faktor-IPS Tutorial. Wenngleich sich diese Anleitung für jedes beliebige Faktor-IPS Projekt anwenden lässt, möchten wir Ihnen an dieser Stelle trotzdem noch einmal kurz erklären, wie Sie an die genannten Faktor-IPS Projekte gelangen, um Schritt für Schritt folgen zu können.

Besuchen Sie dazu bitte die Webseite <http://www.faktorips.org/> und navigieren Sie im linken Menü zu *Dokumentation*. Unter *Einführungstutorial* finden Sie die Tutorial-Projekte für die aktuellste Faktor-IPS Version als ZIP-Datei.

Starten Sie nun Eclipse mit installiertem Faktor-IPS und installierter Faktor-IPS UML-Integration und importieren Sie die heruntergeladenen Tutorial-Projekte über *File* → *Import...* → *Existing Projects Into Workspace* in Ihren Eclipse Workspace.

2 Installation

Dieses Kapitel ist noch nicht verfasst, da UML-Integration noch nicht released.

3 UML-Modelle für Faktor-IPS Modelle generieren

3.1 UML-Modelle generieren

Stellen Sie sicher, dass entweder der Package-Explorer oder der Modell-Explorer geöffnet ist. Sie sehen die drei Tutorial-Projekte:

- *org.faktorips.tutorial.de.Grundmodell*,
- *org.faktorips.tutorial.de.Hausratmodell* und
- *org.faktorips.tutorial.de.Produktdaten*.

UML-Modelle lassen sich nur für Faktor-IPS Modell-Projekte erzeugen. Das Projekt *org.faktorips.tutorial.de.Produktdaten* ist kein Modell-Projekt. Sie können es also für dieses Tutorial gerne löschen oder ausblenden.

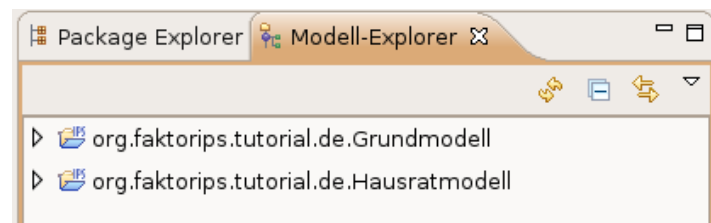


Abbildung 3: Modell-Explorer mit Tutorial-Projekten

Rechtsklicken Sie nun auf das Projekt *org.faktorips.tutorial.de.Grundmodell*. Es öffnet sich ein Kontext-Menü. Falls Sie im Modell-Explorer sind, können Sie jetzt direkt *Update UML Model* auswählen. Wenn Sie hingegen im Package-Explorer sind, finden Sie diesen Punkt im Untermenü *Faktor-IPS*.

Klicken Sie jetzt auf *Update UML Model*. Faktor-IPS erzeugt nun das UML-Modell und speichert es in der Datei *domain-model.uml*. Wiederholen Sie diesen Vorgang jetzt für das Projekt *org.faktorips.tutorial.de.Hausratmodell*. Da das Hausratmodell das Grundmodell referenziert ist es wichtig, dass Sie die Operation immer zuerst für das Grundmodell durchführen.

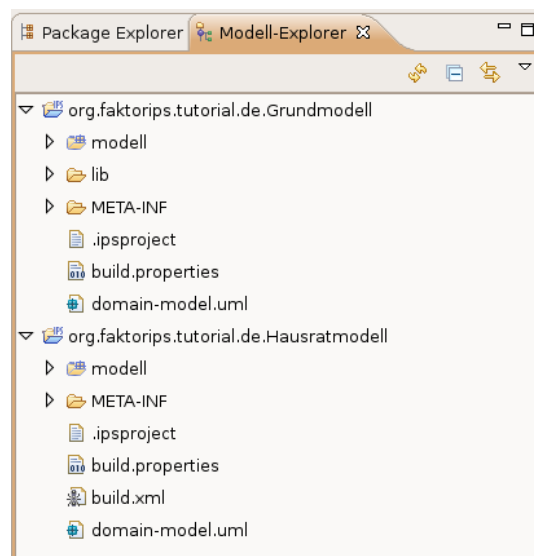


Abbildung 4: Modell-Explorer mit generierten UML-Modellen

3.2 UML-Modelle aktualisieren

Sobald Sie in Faktor-IPS Änderungen durchführen (zum Beispiel eine bestehende Vertragsklasse bearbeiten oder eine neue Vertragsklasse hinzufügen etc.), ist das UML-Modell nicht mehr auf dem aktuellen Stand. Änderungen am Faktor-IPS Modell werden also nicht automatisch auch in den bereits generierten UML-Modellen durchgeführt.

Wenn Sie aber bei einem Faktor-IPS Projekt die *Update UML Model* Operation anstoßen, für das bereits ein UML-Modell besteht, so wird der aktuelle Stand des Faktor-IPS Modells auf das UML-Modell übertragen.

Dabei werden, falls notwendig, alle seit der letzten Aktualisierung des UML-Modells

- neu hinzugekommenen Faktor-IPS Modellelemente im UML-Modell angelegt.
- veränderten Faktor-IPS Modellelemente dementsprechend im UML-Modell angepasst.
- gelöschten Faktor-IPS Modellelemente auch im UML-Modell gelöscht.
- neu hinzugekommenen UML Modellelemente gelöscht (falls für diese nicht auch eine Entsprechung im Faktor-IPS Modell hinzugekommen ist).

Es empfiehlt sich daher die *Update UML Model* Operation **immer** durchzuführen, **bevor** Sie mit einem bereits bestehenden UML-Modell arbeiten. Damit stellen Sie sicher, dass das UML-Modell das aktuelle Faktor-IPS Modell repräsentiert.

4 UML-Diagramme erzeugen

Hinweis

Dieses Kapitel ist spezifisch für das Eclipse UML2-Tools Plugin und geht deshalb davon aus, dass Sie dieses Plugin zur Einsicht und Bearbeitung von UML-Modellen verwenden. Sollten Sie ein anderes Plugin verwenden, könnten die folgenden Informationen für Sie keine Gültigkeit haben.

4.1 Welche Rolle haben Diagramme

Ein UML-Diagramm ist immer eine Sicht auf ein UML-Modell. Konkret bedeutet das zunächst einmal, dass Sie mehrere UML-Diagramme für ein und das selbe UML-Modell haben können. Jedes Diagramm ist dann eine andere Sicht auf das Modell.

Somit könnten Sie beispielsweise ein Diagramm erstellen, welches nur die Grundstruktur der Verträge und Produkte eines Modells zeigt, und ein weiteres Diagramm, welches zusätzlich auch Zuschläge und Nachlässe anzeigt. Ein drittes Diagramm könnte dann auch noch versicherte Risiken und Leistungsversprechen berücksichtigen und nochmal ein anderes Diagramm könnte schließlich für eine Präsentation speziell angeordnet und kommentiert sein.

Die UML-Diagramme werden separat zum UML-Modell abgespeichert. Bei den Eclipse UML2-Tools sind die UML-Diagramme aber sehr stark mit den darunter liegenden Modellen verknüpft. Dass ein UML-Diagramm immer eine Sicht auf ein UML-Modell ist, wurde bereits erwähnt. Es ist aber auch immer eine aktuelle Sicht. Eine Änderung im Modell wird automatisch in alle darauf basierenden Diagramme übertragen und umgekehrt.

Von dieser Synchronisierung sind nur spezielle Diagramm-Notizen ausgenommen, von denen man einem Diagramm beliebig viele hinzufügen kann. Diese eignen sich besonders dazu, komplizierte Zusammenhänge zu kommentieren. Diese Notizen sind wirklich Bestandteil des Diagramms und nicht des Modells. Sie gehen daher erst beim Löschen des Diagramms verloren.

Möchte man ein UML-Diagramm erstellen, welches sich nicht bei einem erneuten Anstoßen der *Update UML Model* Operation immer automatisch mit ändert, so muss man dieses Diagramm auf einer separaten UML2-Modelldatei aufsetzen.

4.2 UML-Diagramme erzeugen

Rechtsklicken Sie die *domain-model.uml* Datei, für die Sie ein Klassendiagramm erzeugen möchten, und wählen Sie im Kontextmenü *Initialize Class Diagram* aus.

Tun Sie das jetzt für das Hausrat-UML-Modell. Ein Wizard öffnet sich. Auf der ersten Seite des Wizards geben Sie an, wo Sie die Diagramm-Datei speichern möchten und wie diese heißen soll. Als Speicherort belassen Sie es bei *org.faktorips.tutorial.de.Hausratmodell*. Als Dateinamen tippen Sie bitte ein: *gesamtischt.umlclass*.

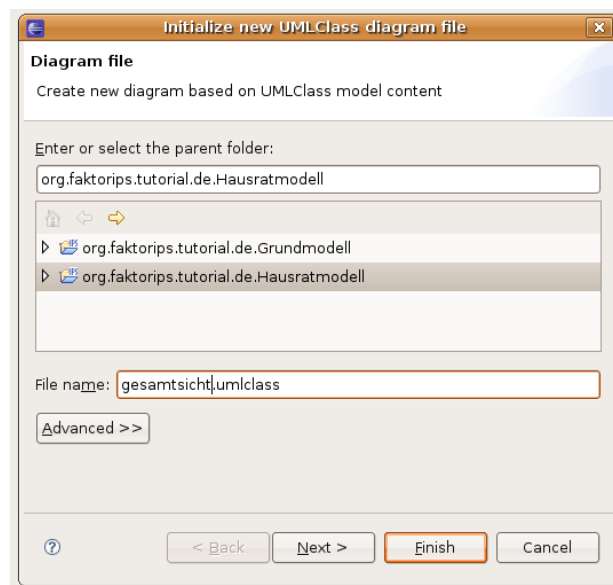


Abbildung 5: Wizard zur Erzeugung eines UML-Klassendiagramms, Seite 1

Klicken Sie dann bitte auf *Next >*.

UML-Diagramme können in den UML2-Tools immer pro UML-Package erzeugt werden. Auf der zweiten Seite des Wizards müssen Sie deswegen auswählen, für welches Package Sie ein Diagramm erzeugen möchten.

Bitte klappen Sie den Baum auf und wählen Sie das Package *hausrat* aus. Klicken Sie dann auf *Finish*. Im Hausratmodell-Projekt existiert nun die Datei *gesamtischt.umlclass*.

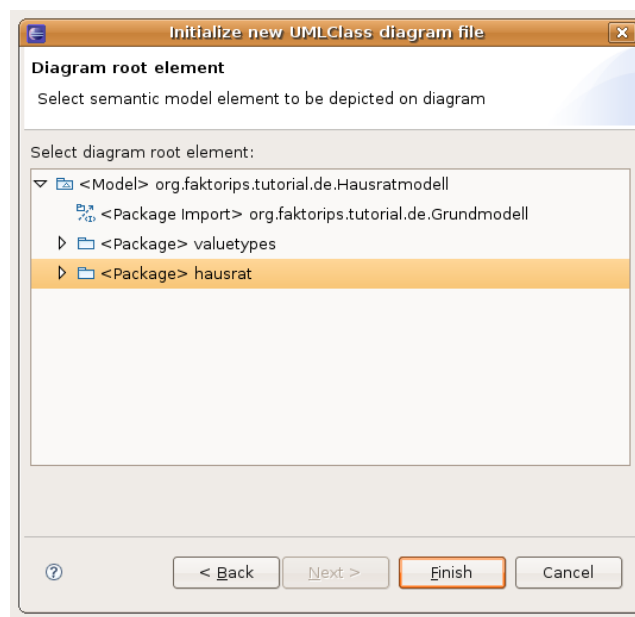


Abbildung 6: Wizard zur Erzeugung eines UML-Klassendiagramms, Seite 2

Hinweis

Auch das komplette Modell selbst ist ein Package. Wenn Sie das Modell als Package auswählen, sehen Sie ein Package-Diagramm mit allen Packages in Ihrem Modell. Falls Sie irgendwelche Modellelemente im Default-Package eingeordnet haben, so erscheinen diese ebenfalls dort.

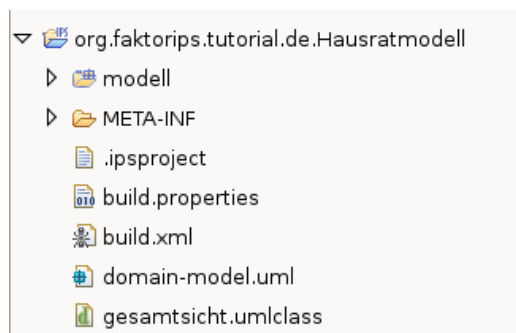


Abbildung 7: Modell-Explorer mit erzeugtem UML-Klassendiagramm

5 UML-Diagramme layouten und bearbeiten

Hinweis

Dieses Kapitel ist spezifisch für das Eclipse UML2-Tools Plugin und geht deshalb davon aus, dass Sie dieses Plugin zur Einsicht und Bearbeitung von UML-Modellen verwenden. Sollten Sie ein anderes Plugin verwenden, könnten die folgenden Informationen für Sie keine Gültigkeit haben.

5.1 Elemente anordnen

Doppelklicken Sie bitte auf die im vorherigen Kapitel erzeugte Datei *gesamtsicht.umlclass*. Nun öffnet sich der Editor zum Bearbeiten des Diagramms. Sie können die einzelnen Elemente jetzt ganz einfach per Drag & Drop anordnen.

Am oberen Bildschirmrand ist außerdem eine zusätzliche Toolbar erschienen. Sie finden dort unter anderem ein Drop-Down Feld, mit dem Sie den Zoom einstellen können, und Werkzeuge, die Ihnen beim Layouten behilflich sein können.



Abbildung 8: Eclipse UML2-Tools Toolbar

Die *Outline-View* hat sich ebenfalls geändert. Dort sehen Sie jetzt eine Komplettübersicht für das momentan geöffnete Diagramm. Das ist, vor allem bei großen Diagrammen, eine wichtige Navigationshilfe.

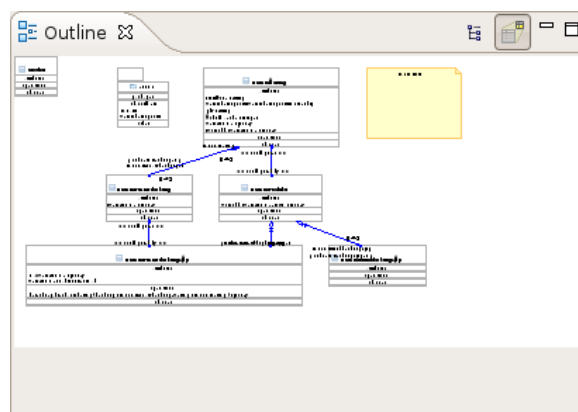


Abbildung 9: Eclipse UML2-Tools Outline-View

5.2 Diagramm-Notizen hinzufügen

Sie haben weiterhin die Möglichkeit, dem Diagramm Notizen hinzuzufügen. Die entsprechende Option finden Sie rechts in der Palette.



Abbildung 10: Eclipse UML2-Tools Werkzeug-Palette

Diese Notizen werden in der *.umlclass* Datei gespeichert und sind somit unabhängig vom Modell. Sie gehen also nicht durch Modelländerungen verloren, sondern erst, wenn Sie das Diagramm selbst löschen. Den Text der Notiz können Sie durch Doppelklicken auf das gelbe Feld ändern.

5.3 Elemente ausblenden

Manchmal möchten Sie nicht alle Elemente des gewählten UML-Modell-Packages, auf welches sich das Diagramm bezieht, sehen. Dennoch sollen die Elemente im Modell enthalten bleiben. Man darf diese also nicht einfach aus dem Diagramm löschen (das würde ja direkt das darunter liegende Modell ändern). Hier gibt es stattdessen die Möglichkeit, einzelne Elemente zu verbergen. Sie werden dann nicht aus dem Modell gelöscht, sondern nur im Diagramm unsichtbar gemacht.

Rechtsklicken Sie dazu auf das Element, welches Sie ausblenden möchten, und wählen Sie im Kontextmenü *Filters* → *Hide Selection*. Um die versteckten Elemente wieder „hervor zu holen“, rechtsklicken Sie irgendwo auf eine freie Fläche im Diagramm und wählen im Kontextmenü *Filters* → *Show All Hidden Parts*. In der Toolbar des Diagramm-Editors gibt es hierfür außerdem auch entsprechende Schaltflächen.

5.4 Elemente bearbeiten

Elemente hinzufügen können Sie ganz einfach über die Palette rechts. Wie Sie ein Element löschen können sollte selbsterklärend sein. Um Elemente bearbeiten zu können, müssen Sie die *Properties-View* öffnen. Wählen Sie dann das zu bearbeitende Element aus, um dessen Eigenschaften dort sehen und bearbeiten zu können.

5.5 Elemente aus anderen UML-Packages einblenden

Ein UML-Diagramm ist immer eine Sicht auf genau ein bestimmtes UML-Package. Manchmal möchte man aber auch Modellelemente aus anderen Packages bzw. anderen UML-Modellen in einem Diagramm einblenden.

Zum Beispiel könnte es sein, dass wir im Diagramm für das Hausratmodell die Klasse *Vertrag* aus dem Grundmodell einblenden möchten um anzuzeigen, dass *HausratVertrag* von dieser Klasse abgeleitet ist.

Um dies zu bewerkstelligen, rechtsklicken Sie einfach auf eine leere Fläche in Ihrem Diagramm und wählen im Kontextmenü *Create Shortcut...* aus. Ein Dialog erscheint, der alle Projekte in Ihrem Workspace anzeigt. Klappen Sie nun das Projekt *org.faktorips.tutorial.de.Grundmodell* und dann *domain-model.uml* → <Model> *org.faktorips.tutorial.de.Grundmodell* → <Package> *base* auf und wählen Sie <Class> *Vertrag* aus. Klicken Sie auf *Ok*.

Die Klasse *Vertrag* ist in Ihrem Diagramm erschienen und es wird sofort angezeigt, dass die Klasse *HausratVertrag* von *Vertrag* abgeleitet ist.

5.6 Diagramme als Bild-Dateien exportieren

Sie können die erstellten UML-Diagramme ganz einfach als Bild-Dateien exportieren.

Rechtsklicken Sie hierzu auf eine leere Fläche in Ihrem Diagramm. Wählen Sie im Kontextmenü die Option *Save As Image File...* im Untermenü *File* aus. Es erscheint folgender Dialog:

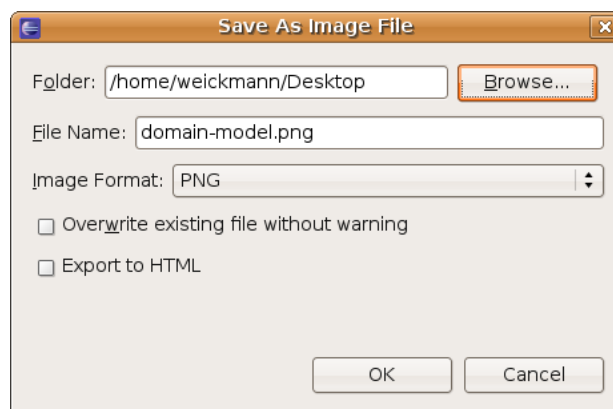


Abbildung 11: Dialog zum Exportieren eines UML-Diagramms als Bild-Datei

6 Ausblick

6.1 Faktor-IPS UML-Profil

Unser Hauptaugenmerk liegt momentan auf der Entwicklung und Einbindung eines UML-Profiles für Faktor-IPS. UML-Profile erlauben es, die UML um eigene Sprachkonstrukte zu erweitern, sogenannte Stereotypen. Momentan können wir beispielsweise Vertragsklassen und Produktklassen in UML nur allgemein als Klassen abbilden. Dabei geht Information verloren. Mit einem UML-Profil können wir hierfür eigene Stereotypen vorsehen. Die generierten UML-Modelle werden dadurch an die Faktor-IPS Domäne angepasst.

6.2 UML-Modell → Faktor-IPS-Modell

Nachdem wir mithilfe des Faktor-IPS UML-Profiles alle Eigenheiten der Faktor-IPS Modelle in den generierten UML-Modellen erfassen können, möchten wir aus den UML-Modellen auch Faktor-IPS Modelle generieren. Dies ermöglicht „Round-Trip-Modellierung“. Faktor-IPS Entwickler können dann jederzeit entscheiden, ob Sie ihre Modelle lieber mit Faktor-IPS oder mit einem beliebigen UML-Tool bearbeiten möchten.

6.3 Automatisches Synchronisieren

Ein weiteres geplantes Feature ist das automatische Synchronisieren der UML-Modelle mit den Faktor-IPS Modellen. Bei Aktivierung dieser Option soll bei Änderung eines Faktor-IPS Modells automatisch dessen UML-Modell aktualisiert werden und umgekehrt. Entwickler müssen dann nicht mehr jedes mal erneut die entsprechende Aktualisierung manuell anstoßen.