

# Technischer Bericht

3D-CLASS-OBJECT



VISUALIZATION

**Diplomarbeit SS 2007**

Autor: Dario Vonäsch

Betreuer: Thomas Letsch

## Dokumentinformationen

### Änderungsgeschichte

Datum	Version	Änderung	Autor
14.05.2006	0.1	Dokument erstellt	dav
17.05.2007	0.2	Dokument überarbeitet	dav
20.05.2007	0.3	Dokument erweitert	dav
21.05.2007	0.4	Dokument überarbeitet	dav
22.05.2007	0.5	Korrekturen gemäss Sitzung	dav
23.05.2007	0.6	Dokument abgeschlossen	dav

# 3D-CLASS-OBJECT VISUALIZATION

---

## 0. Inhalt

Dokumentinformationen .....	2
Änderungsgeschichte .....	2
0. Inhalt.....	3
1. Einleitung und Übersicht .....	4
2. Ergebnisse.....	5
3. Schlussfolgerungen.....	8

## 1. Einleitung und Übersicht

Wer sich zum ersten Mal mit Objektorientierter Modellierung auseinandersetzt, hat oft Mühe den Unterschied zwischen Klassen und Objekten zu verstehen. Im Normalfall werden diese in separaten Diagrammen dargestellt und der Zusammenhang ist nicht immer einfach zu erkennen. Die Idee dieser Arbeit ist es daher, ein Tool zu erstellen, das dies durch eine dreidimensionale Darstellung ermöglicht. Dazu soll in einer Ebene ein Klassendiagramm dargestellt werden und darüber, entsprechend angeordnet, das zugehörige Objektdiagramm. Dieser Ansatz macht die Zusammenhänge zwischen Klassen- und Objektdiagramm leichter erkennbar und somit einfacher zu verstehen.

Dazu sollte aufbauend auf den Resultaten der vorangegangenen Studienarbeit eine einsatzfähige Software erstellt werden, welche im Unterricht verwendet werden kann. Dazu gehört die Möglichkeit, komplexe und umfangreiche Diagramme interaktiv erstellen zu können, um diese danach durch Ändern des Beobachtungspunktes detailliert zu betrachten und die Zusammenhänge besser erläutern zu können.

Um den Einsatz zu erleichtern sollte auch das Speichern und Laden von einmal erstellten Projekten ermöglicht werden. Dazu sollen sowohl die Diagramme, mit den genauen Positionen der dazugehörigen Elemente, als auch die Lage des 3D-Modells in einer XML Datei gespeichert werden. Dies erlaubt es, interessante Positionen und Konstellationen abzuspeichern und bei einer Präsentation wieder abzurufen.

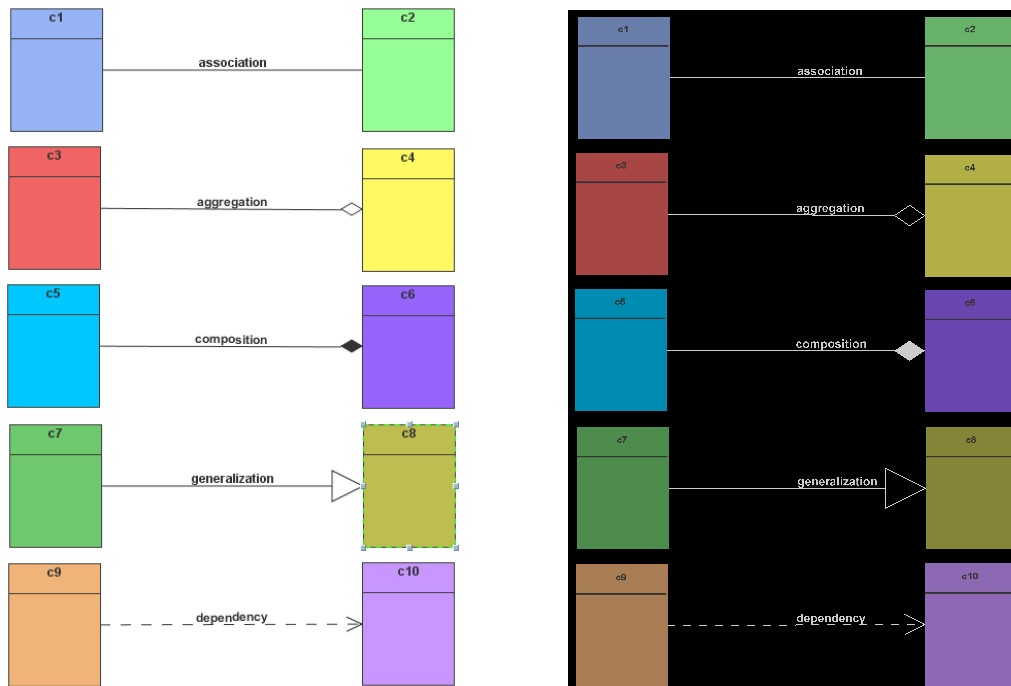
Nebst der Möglichkeit Klassendiagramme direkt in der Applikation zu erstellen, soll es auch ermöglicht werden, Diagramme, die in anderen Tools erstellt wurden, in die Applikation zu importieren. Dazu sollen diese durch das Einlesen von XMI Dateien in die Applikation geladen werden. Dort können sie dann bei Bedarf auch noch erweitert oder verändert werden.

Da die Applikation auch vor Publikum eingesetzt werden soll, ist es wichtig, dass die Bedienung möglichst einfach und effizient gestaltet ist. Es wird daher darauf geachtet, dass geübte Benutzer auch mit Hilfe von Abkürzungstasten arbeiten können.

Als Technologie ist JAVA, sowie als Entwicklungstools Eclipse und der Enterprise Architekt in der Aufgabestellung vorgegeben.

## 2. Ergebnisse

Die entstandene Applikation ist in zwei Teile aufgeteilt. Im einem Teil befindet sich der Klassendiagramm-Bereich, in dem zweidimensionale Klassendiagramme im herkömmlichen Sinne erstellt und bearbeitet werden können. Dazu können die üblichen Relationen wie Assoziationen, Aggregationen, Kompositionen, Generalisierungen und Dependencies eingesetzt werden. Es sind auch reflexive und parallele Relationen möglich. Die Relationen können zusätzlich mit Namen, Rollen und Multiplizitäten beschriftet werden.



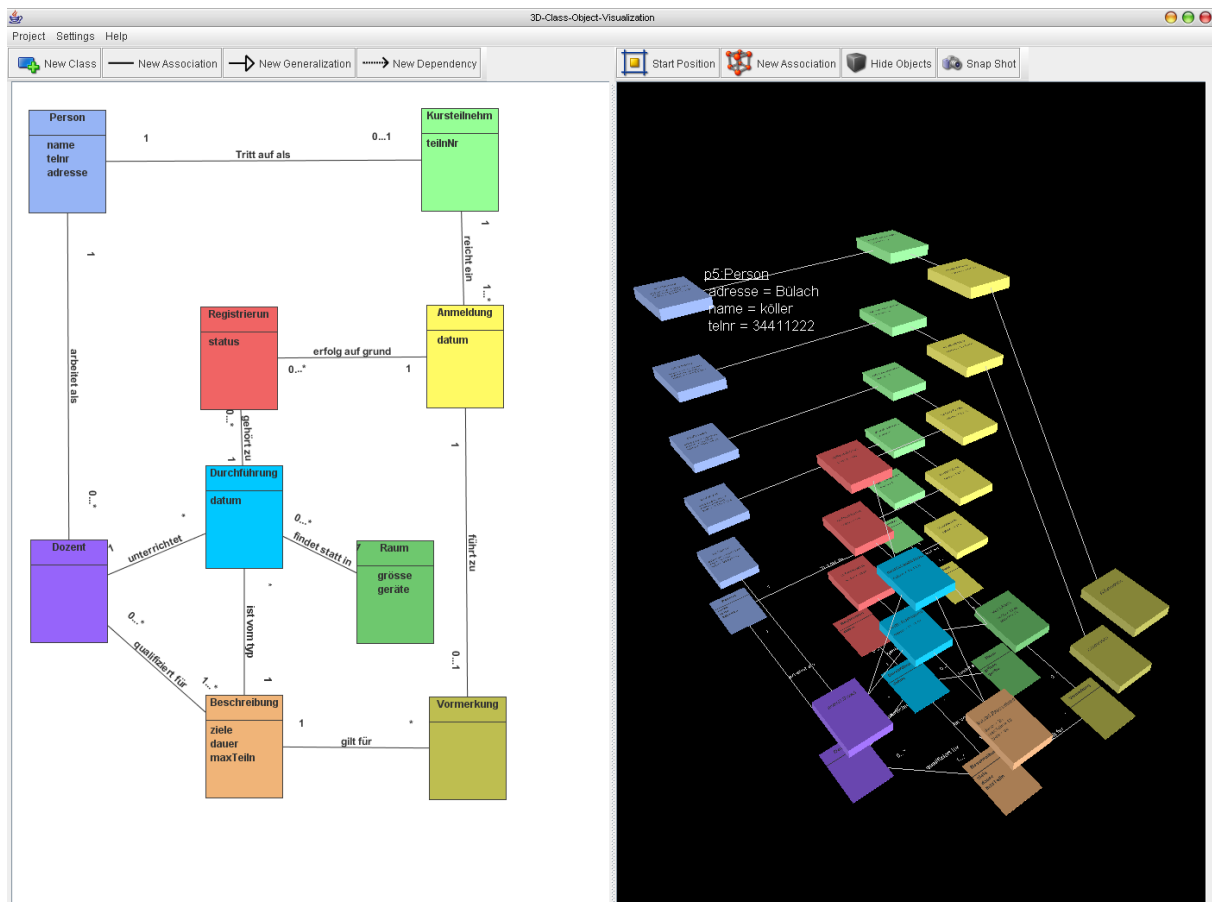
Die erstellten Klassendiagramme werden dabei automatisch auch im rechten Teil, dem 3D-Modell dargestellt und aktualisiert. Dabei stimmen die Elemente in Farbe und Dimension in den beiden Diagrammen überein, was die Identifizierung und Erkennung von Elementen sicherstellt. Die beiden Bereiche können je nach Bedarf einzeln oder nebeneinander dargestellt werden. Als besonders anspruchsvoll erwiesen, hat sich dabei die dreidimensionale Darstellung der Relationen, was einige Kenntnisse über Geometrie und Vektoren voraussetzt.

Von den vorhandenen Klassen können danach interaktiv Objekte erstellt werden und dabei mit entsprechenden Werten versehen werden. Die Objekte werden jeweils nur im 3D-Teil und direkt über der dazugehörigen Klasse dargestellt. Sie können in der Höhe beliebig verschoben werden, müssen jedoch stets über der entsprechenden Klasse bleiben. Bei der Erzeugung von Objekten werden spezielle Verhältnisse, wie Vererbung oder Assoziationsklassen, berücksichtigt und speziell dargestellt. So wird zum Beispiel ein Objekt einer Subklasse automatisch zusammen mit einem Objekt der Oberklasse dargestellt, welches die geerbten Attribute enthält. Dies funktioniert auch für Mehrfachvererbung.

Zwischen den einzelnen Objekten können nun beliebige Verbindungen erstellt werden. Ist die vorhandene Konsistenzprüfung aktiviert, so werden unzulässige Elemente verhindert und der Benutzer darüber informiert. Dabei werden auch die Multiplizitäten berücksichtigt. Diese Überprüfung kann jedoch optional ausgeschaltet werden, damit sind auch unzulässige Verbindungen möglich.

Wird mit der Maus über ein Objekt gefahren, erscheinen die Informationen dieses Objektes als Tool Tipp, da je nach Lage des Objektes die Oberfläche mit der Beschriftung nicht gut lesbar ist.

# 3D-CLASS-OBJECT VISUALIZATION



Screenshot der Applikation, links: das Klassendiagramm, rechts: das 3D-Modell mit Objekten

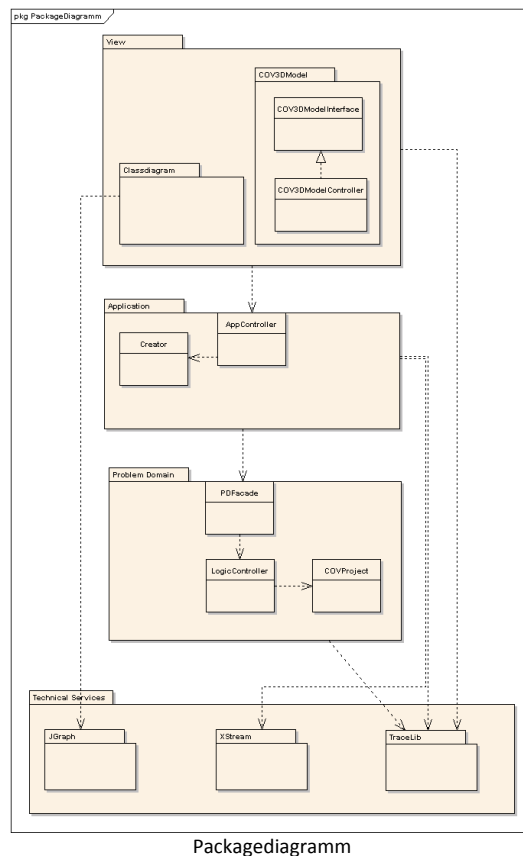
Einmal erstellte Projekte können als XML Datei an einem beliebigen Ort gespeichert und wieder geladen werden. Dies ermöglicht es, didaktisch sinnvolle Beispiele bereits im Voraus gezielt zu erstellen und später nur noch zu laden. Dabei werden die Daten beider Modelle als auch die genauen Positionen und die Lage des 3D-Modells festgehalten. Dadurch können auch mehrere Dateien mit unterschiedlichen Darstellungen des gleichen Projektes erstellt werden.

Obwohl das in der Applikation integrierte Case Tool bereits über viel Funktionalität verfügt, können Diagramme auch mit einem anderen Tool erstellt und danach als XMI Datei importiert werden. Da XMI ein vergleichsweise offener Standard ist, existieren pro Hersteller und XMI Version sehr unterschiedliche Umsetzungen. In dieser Arbeit wurde daher der XMI Import für Daten aus dem Enterprise Architect und in der XMI Version 1.1 ausgelegt. Die Architektur ist jedoch so gestaltet, dass durch Verwendung des Strategy-Patterns auf einfache Weise auch andere Tools und Versionen unterstützt werden können. Dazu wird zuerst die XMI Version des Files gelesen und, wenn vorhanden, der entsprechende Handler verwendet. Um weitere Versionen oder Tools zu unterstützen, muss also lediglich ein entsprechender Handler implementiert werden.

Um von einem interessanten 3D-Modell ein Bild zu erstellen, kann jederzeit eine Momentaufnahme erstellt und als Bild im PNG Format gespeichert werden. Dieses kann dann später in einem Skript oder einer Dokumentation als Illustration dienen.

Beim Entwurf der Architektur wurde darauf geachtet, dass diese für zukünftige Erweiterungen oder Änderungen vorbereitet ist. So wird in der Applikation ein zentrales Modell verwendet welches alle Elemente enthält, verwaltet und über eine zentrale Schnittstelle zugänglich macht. Von diesem Modell existieren dann zwei Sichten: das Klassendiagramm und das 3D-Modell, die die vorhandenen Daten auf ihre Weise darstellen. Es können jederzeit auch Sichten hinzugefügt oder entfernt werden. Das Modell informiert alle Sichten mittels detaillierten Nachrichten über Änderungen. Durch eine zusätzliche Schnittstelle zum 3D Modell, welche die benötigten Funktionen definiert, könnte auch ein anderes 3D-Framework verwendet werden.

# 3D-CLASS-OBJECT VISUALIZATION



Ebenfalls hilfreich bei der Weiterentwicklung ist die vorhandene Synchronität zwischen dem Source Code und den im Enterprise Architekt erstellten Klassendiagrammen, welche den Code dokumentieren. Dies erlaubt die automatische Generierung des Source Codes aus den Diagrammen und umgekehrt.

Um den Einstieg bei der Benutzung der Applikation zu erleichtern, wurde ein ausführliches Benutzerhandbuch erstellt, das die Bedienung erklärt.

Für die Darstellung des 3D-Modells wurde das JAVA 3D API verwendet, welches die Funktionalität zur Darstellung und Manipulation von dreidimensionalen Grafiken zur Verfügung stellt. Für die Interaktion wurden entsprechende Behaviors definiert, die die Steuerung des Modells mit der Maus oder der Tastatur erlauben. Dazu können die Geschwindigkeiten einzeln an die Bedürfnisse des Benutzers angepasst werden. Da sich das Modell immer um den jeweiligen Mittelpunkt dreht, muss dieser nach jeder Änderung des Modells neu berechnet werden. Als Hilfsmittel können dazu die aktuellen Modellachsen ein- und ausgeblendet werden.

### 3. Schlussfolgerungen

Das entstandene Tool stellt ein nützliches Hilfsmittel beim Vermitteln von Objektorientierter Theorie dar. Dabei kann es sowohl bei der Erläuterung von Grundlagen als auch für die Analyse bei besonders komplexen Beispielen oder Patterns verwendet werden. Es kann also auch für Fortgeschrittene eine hilfreiche Unterstützung sein.

Die Verwendung von JAVA 3D hat sich als gute Entscheidung erwiesen und gezeigt, dass auch ohne direkte Benützung einer hardwarenahen Programmiersprache anspruchsvolle 3D-Anwendungen realisiert werden können. Dabei muss jedoch vom bekannten Prinzip „Write once, run everywhere“, welches für normale JAVA Programme gilt, Abschied genommen werden. Denn der erstellte JAVA Code ruft grösstenteils Funktionen in der 3D-Engine der Grafikkarte auf und ist somit stark abhängig von der verwendeten Grafikkarte und deren Treiber. Es kann daher vorkommen, dass ein Programm auf verschiedenen Computern ein unterschiedliches Verhalten zeigt und damit auch eine gewisse Unvorhersehbarkeit mit sich bringt. Dies wurde durch das Erstellen einer zusätzlichen Schnittstelle entschärft, da damit auch ein anderes 3D Framework eingesetzt werden könnte. Es ist jedoch davon auszugehen, dass andere Frameworks ähnliche Probleme besitzen, was in Entwicklerforen auch mehrfach nachzulesen ist. Ansonsten sind die Erfahrungen mit JAVA 3D aber sehr gut.

In Zukunft sind verschiedene Erweiterungen denkbar. So wäre es beispielsweise möglich, das Tool so zu erweitern, dass es als Lernprogramm mit integrierten Aufgaben und Lösungen genutzt werden kann, oder die Applikation, durch Umwandlung in ein Applet, über das Internet einem breiten Publikum zugänglich zu machen.

Es wäre auch vorstellbar, den 3D-Teil soweit auszubauen, dass der bisherige linke Bereich, welcher das Klassendiagramm enthält, weggelassen werden könnte. Bei einer Präsentation kann dieser Teil jedoch von Vorteil sein, da hier die Details des Klassendiagramms besser zu erkennen sind.

Im Unterricht kann die Applikation aber schon in der jetzigen Form gut eingesetzt werden.