

Übung 01: Objektorientiertes Modellieren – erste Einführung in Enterprise Architect

Modellierungswerkzeuge erlauben die Modellierung der Analyse und des Entwurfs eines Software-Systems in der graphischen Modellierungssprache UML (Unified Modeling Language).

Ziel

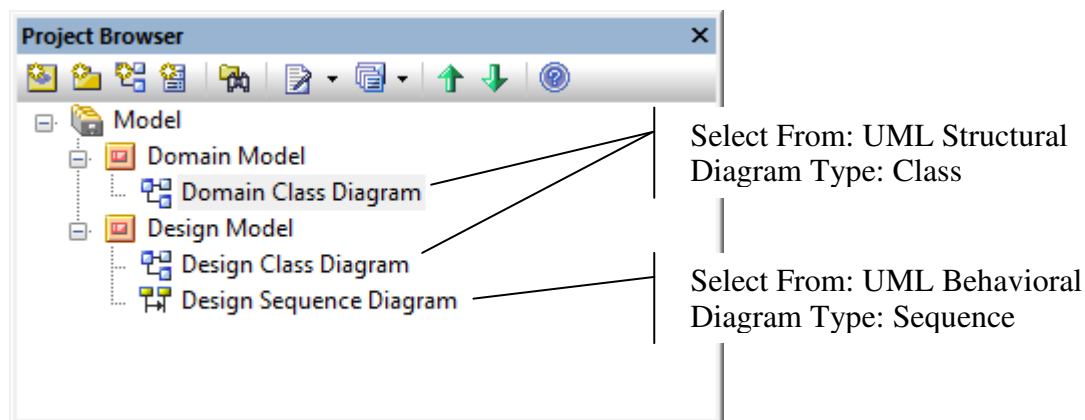
- Kennenlernen des UML Modellierungswerkzeug Enterprise Architect (EA)
- Anwenden von Enterprise Architect in einer kleinen, praktischen Übung

Aufgabe

Nehmen Sie das Beispiel „Würfelspiel“, d.h. Abschnitt 1.5 *A Short Example* aus dem Buch von Larman Seiten 8 bis 11 (es empfiehlt sich, auch die Seiten 3 – 7 zu lesen) und setzen Sie es in UML-Diagramme um. Benützen Sie dazu den Enterprise Architect.

Vorgehen

1. Die Seiten 8 bis 11 aus dem Buch von Larman lesen. Falls Sie das Buch noch nicht haben, steht auf dem Skripteserver ein PDF mit einer Leseprobe von *Ch01-Intro.pdf* (S.8-11) bzw. *K01-Einf.pdf* (S.43-46) des ersten Kapitels.
2. Hinweise zur Erstellung eines neuen Projektes: *Einführung EA.doc* lesen.
3. Erstellen Sie ein EA-Projekt *DiceGame* ohne Modelle auszuwählen. Es wird ein leeres Modell mit dem Namen *Model* erzeugt. Legen Sie in diesem Projekt die untenstehende Package- bzw. Diagramm-Struktur an:



4. Erstellen Sie im Diagramm *Domain Class Diagram* das Domainmodell1 gemäss Larman Figur 1.3.
 - Um ein Attribut hinzuzufügen: Klasse selektieren, rechte Maustaste *Attributes*
 - In einem Domainmodell stellen wir keine Sichtbarkeit der Elemente und meist auch keine Datentypen der Attribute dar. Um diese Dinge zu verbergen: Rechte Maustaste

1 Achtung, das Domainmodell ist sehr einfach (zu einfach) und in der deutschen Übersetzung die Multiplizität 2 bei DiceGame nicht sinnvoll.

auf den Hintergrund *Properties > Features: Show Qualifiers and Visibility Indicators* ausschalten und *Show Attribute Detail: Name Only* auswählen.

5. Erstellen Sie im Diagramm *Design Classes* das Design Klassendiagramm gemäss Figur 1.5.
 - Um die Navigationsrichtung auszuwählen: Doppelklick auf Assoziation öffnet Spezifikation: *Direction: Source -> Destination*
6. Erstellen Sie im Diagramm *Design Sequence Diagram* das Sequenzdiagramm gemäss Figur 1.4.
 - Eine Instanz bzw. Lifeline können Sie erzeugen, indem sie die entsprechende Klasse aus dem *Project Browser* ins Diagramm ziehen. Wählen Sie dabei *as Instance of Element*.
 - Fügen Sie zuerst einen Endpoint ein, von welchem die play()-Message ausgehen soll. Vom kleinen Pfeil des markierten *Endpoint* ziehen Sie auf das gewünschte Objekt, um die erste Message zu erzeugen. Den Namen der Message können Sie im Drop-down *Message* auswählen, da die Klasse *DiceGame* eine entsprechende Operation besitzt.

Codegenerierung, Reverse Engineering und Round-trip Engineering

Codegenerierung

Wählen Sie im *Project Browser* das Designmodell und über die rechte Maustaste *Code Engineering > Generate Source Code* und erzeugen Sie so aus Ihren Designklassen Sourcecode, genauer gesagt Code „Gerippe“, denn der Code innerhalb der Methoden muss von Hand ergänzt werden.

Reverse Engineering

Verändern Sie den generierten Code, in dem Sie zum Beispiel ein zusätzliches Attribut hinzufügen und in der Klasse *Die* den Inhalt der Methode `getFaceValue()` auscodieren. Wählen Sie im *Project Browser* das Designmodell und über die rechte Maustaste *Code Engineering > Synchronize Package with Code* und wählen dann im Pop-Up Fenster *Reverse Engineer*. Betrachten Sie das Modell, wurden die Codeänderungen aufgenommen?

Round-Trip Engineering

Ergänzen Sie nun wieder im Modell und wählen im *Project Browser* das Designmodell und über die rechte Maustaste *Code Engineering > Synchronize Package with Code*. Wählen Sie nun im Po-up Fenster *Forward Engineer*. Betrachten Sie den generierten Code. Ist Ihre Implementation von `getFaceValue()` noch vorhanden? Sie haben nun eine ganze Rundreise, einen Round-Trip durchgeführt.

Reverse Engineering eines komplexeren Java Projektes

Auch Reverse Engineering geht mit rechter Maustaste *Code Engineering > Import Source Directory*. Wählen Sie das Source Directory eines Ihrer letzten Java-Projekte. Ziehen Sie die Klassen auf ein Klassendiagramm, so werden auch Assoziationen sichtbar, also richtige Klassendiagramme aufgebaut. Aber Achtung: Solche mit Reverse Engineering erstellten Diagramme zeigen alle Details. In Architektur-/DesignDiagrammen werden in der Regel nur die wesentlichen Elemente dargestellt.