

UML – Unified Modeling Language

Objektorientierte Modellierung für die Praxis.

Helmut Schluderbacher

Wir schreiben das Jahr 2371. Dies ist ein Abenteuer des Raumschiffs Enterprise, das mit seiner Mannschaft unterwegs ist, neue Welten und unbekannte Zivilisationen zu entdecken und dabei modernste Arbeitsmethoden anzuwenden.

Wir befinden uns an Bord der Enterprise NCC-1701D und haben die Aufgabe ein Shuttle zu starten und Erkundungen innerhalb eines Nebels durchzuführen. Der Zeitdruck ist sehr groß und mit Schrecken stellen wir fest, daß auf Grund der starken Hintergrundstrahlung des Nebels der Bordcomputer teilweise gelöscht wurde. Doch glücklicherweise können wir auf die Ausbildung der Sternflottenakademie zurückgreifen und schnell ein Vorgehensmodell mit Hilfe von UML entwerfen.

Schritt 1 Anforderungen aufnehmen

Wichtig ist, daß wir die Anforderungen weitgehend unabhängig von der Implementierung formulieren. Das WAS? und nicht das WIE? Ist hier wichtig. Der für uns interessante Anwendungsfall heißt Shuttle starten. Auf Grund unseres Erfahrungsschatzes können wir jedoch sogleich zu Schritt 2 übergehen.

Schritt 2 Use-Case-Typen benennen (Use-Case-Modell)

Die wichtigsten Aktoren heißen Crewmitglied, denn irgendwer muß ja das Shuttle fliegen, Shuttle, Deflektorschild, Kraftfeld, Warpantrieb, Maschinenraum und Rampe. Die wichtigsten Anwendungsfälle sind neben dem Starten des Shuttles, das Aktivieren des Kraftfeldes der Shuttlerampe (Atmosphären-Eindämmungsfeld), die Steuern des Schleusensystemes und das Deaktivieren der hinteren Deflektorschilde.

Schritt 3 Use-Case-Instanzen festlegen (Objekt-Prozeß-Modell)

Um die Anwendungsfälle zu konkretisieren, werden nun im Objekt-Prozeß-Modell die relevanten Prozesse benannt und ihren Klassen zugeordnet.

Schritt 4 Szenarien definieren (Sequenzdiagramm)

Wir gehen in Gedanken noch einmal alles durch, was der Bordcomputer sonst bei jedem Start abfragt und erstellen dabei ein Sequenzdiagramm. Checkliste durchgehen, Resonanzfrequenz der Deflektorschilde anpassen, Tor öffnen, Startknopf drücken, Rampentor schließen, Schutzschilde passieren. Der Rest der Reise geht glücklicherweise wieder vollautomatisch.

Schritt 5 Logische Architektur festlegen (Klassenstrukturmodell)

Jetzt können wir die statischen Aspekte in den Vordergrund rücken. Welcher Typ das Shuttle ist, über welche Antriebsarten das Shuttle verfügt, die Größe der Besatzung, die komplexen Systeme der Kraftfelder und das Deflektorschildsystem.

Schritt 6 Persistenzmodell erstellen (Klassenstrukturmodell)

Jetzt muß entschieden werden welche Daten gespeichert werden sollen, also welche Klassen persistent werden sollen. Welche Klassen im Externspeicher hinterlegte Objekte haben dürfen, müssen wir aber glücklicherweise nicht festlegen und sparen damit wieder Zeit.

Schritt 7 Komponentenarchitektur festlegen (Komponentendiagramm)

In einem Diagramm wird die Architektur der Pakete und Module festgehalten. Dabei stellen wir fest, daß es eine ziemliche Abhängigkeit der verschiedensten Komponenten vom Modul Kraftfeld besteht. Eine Verbesserung merken wir uns für die nächste Release vor.

Schritt 8 Verteilungsstruktur bestimmen (Verteilungsdiagramm)

Im Verteilungsdiagramm sind die Verteilungsstrukturen zwischen physisch getrennten Komponenten des Systems anzugeben. Wir erkennen, daß das Shuttle für die Ermittlung der Schildfrequenz Informationen von der Enterprise benötigt. Dafür muß die Frequenzanpassung berechtigt sein auf die Daten im Mutterschiff zugreifen zu dürfen.

Dieses Beispiel ist in leicht abgewandelter Form direkt dem Buch "Unified Modeling Language" entnommen. Wie auch das nachfolgende Kommunikationsproblem welches gut in Schritt 1 passen würde.

Analyst: Do you have four-Volt, two-Watt bulbs?

Anwender: For what?

Analyst: No, two!

Anwender: Two what?

Analyst: Yes!

Anwender: No!

Analyst: ???

Rainer Burkhardt; Addison-Wesley; ISBN 3-8273-1226-4; 415 Seiten; ÖS 583,—; 1997, dem Buch liegt eine CD bei mit einer 90-Tage Lizenz für darauf enthaltene Software

