

## Kurzfassung

In der heutigen Zeit ändern sich Anforderungen an Anwendungen laufend. Neben den Anforderungen haben sich natürlich auch die Hilfsmittel und Technologien weiterentwickelt. Moderne objektorientierte Sprachen wie z.B. UML und Java werden den heutigen Ansprüchen gerecht.

Die objektorientierte Programmiersprache Java, oder besser gesagt die Java Enterprise Edition, eignet sich hervorragend um verteilte Systeme, zu denen auch Webanwendungen zählen, zu entwickeln. UML 2.0 ist mit ihren aktuellen Diagrammartentypen ebenfalls sehr gut für die Modellierung von verteilten Anwendungen geeignet.

Diese Arbeit evaluiert das Zusammenspiel zwischen

UML 2.0 und Java Enterprise Anwendungen.

In UML 2.0 gibt es insgesamt 13 verschiedene Diagrammartentypen. Jedes einzelne Diagramm hat Stärken und Schwächen in Bezug auf Enterprise Anwendungen. Diese Arbeit zeigt, welche Teile einer Enterprise Anwendung sich mit welchem Diagramm in UML 2.0 modellieren lassen bzw. welche nicht.

Zusätzlich geht diese Arbeit auch auf die Möglichkeiten ein, allgemeine Aspekte einer Enterprise Anwendung zu modellieren, wie z.B. Design Patterns und Systemarchitekturen. Dabei werden für ein bestehendes Beispiel die einzelnen Diagramme, die UML 2.0 zur Verfügung stellt, modelliert.

Ein weiteres Ziel der Arbeit ist, einen generellen Überblick von UML 2.0 Diagrammartentypen im Rational Unified Process zu geben. Dabei wird evaluiert, welches Diagramm in welcher Phase des Unified Process in Bezug auf Enterprise Anwendungen hilfreich sein kann.

## (Java) Enterprise Applikationen

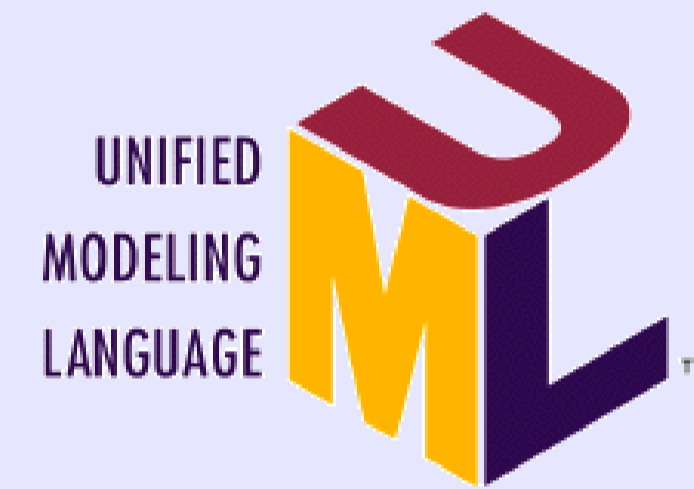


„Java Plattform, Enterprise Edition, abgekürzt Java EE oder früher J2EE, ist die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von in Java programmierten Anwendungen.“

Spezifikation einer SW-Architektur:

- Web-tier Technologien
- Web Service Technologien
- Enterprise Java Beans
- Persistenz Technologien

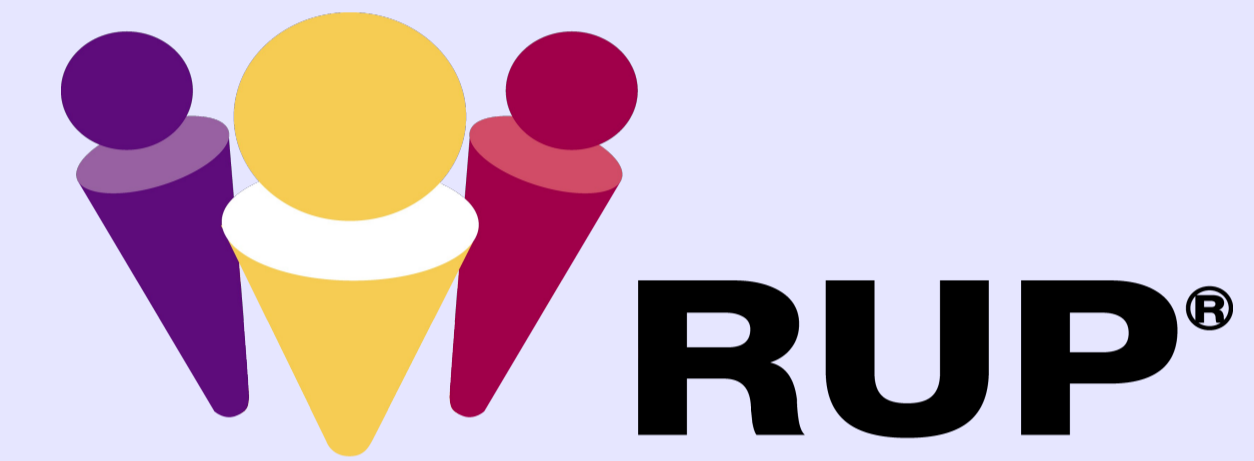
## UML 2.0 - Unified Modeling Language



von OMG im April 2005 verabschiedet

„verbreitete Notation, um Softwaresysteme zu analysieren und zu entwerfen“ [Rupp05]  
unabhängig von Vorgehensmodellen, Entwicklungswerkzeugen und Programmiersprachen [Hitz05]

## Rational Unified Process



objektorientiertes Vorgehensmodell

Eigenschaften [Zuse01]: anwendungsfallgesteuert; interaktiv / inkrementeller Prozess; architekturzentriert  
Arbeitsschritte: (Anforderungen), Analyse, Entwurf, Implementierung, Test, Inbetriebnahme und Wartung

**Anwendungsfalldiagramm:** System abstrakt, keine Schichten

**Aktivitätsdiagramm:** in UML 2.0 komplett überarbeitet; weit gestreut: Geschäftsprozessmodellierung, Anwendungsfallbeschreibung (Zusammenspiel Benutzer - System), Implementierung einer Operation

**Zustandsdiagramm:** Anwendungsfallbeschreibung, Beschreibung von Zuständen in Klassen, Protokollzustandsdiagramm

**Sequenzdiagramm:** Architekturbeschreibung / Beschreibung von Pattern; Zusammenspiel der Schichten; Nachteil: aufwendige Modellierung

**Kommunikationsdiagramm:** ähnliche Einsatzmöglichkeit wie Sequenzdiagramm, nicht so detailliert

**Zeitdiagramm:** genaue zeitliche Abfolgen; in Enterprise Applikationen kaum verwendet (Embedded Systems)

**Interaktionsübersichtdiagramm:** Zusammenspiel der Interaktionen; neues Diagramm, noch nicht etabliert

**Klassendiagramm:** mächtigstes Werkzeug in UML, natürlich auch für Java Enterprise Anwendungen

**Paketdiagramm:** guter Überblick über Schichten der Anwendung; verschiedene Gliederung möglich [Rupp05]: funktionale Gliederung, Gliederung nach Schichten

**Objektdiagramm:** Instanzen von Klassen; z.B. Domänenmodell oder Modellierung von Testdaten

**Kompositionsstrukturdiagramm:** Modellierung Einsatz von Pattern, Art Klassendiagramm aus Sicht einer Klasse (Kontextklasse)

**Komponentendiagramm:** Darstellung von Komponenten und Schnittstellen nach außen; gute Darstellung der Schichten, auch Artefakte (wie z.B. Konfigurationsdateien)

**Verteilungsdiagramm:** physische Verteilung der Komponenten; guter Überblick über Enterprise Anwendung

	Analyse	Entwurf	Implementierung	Test
Anwendungsfalldiagramm	1			
Aktivitätsdiagramm	1	2	2	
Zustandsdiagramm	1	2	2	
Sequenzdiagramm	2	1		1
Kommunikationsdiagramm	2	2		
Zeitdiagramm		3		
Interaktionsübersichtdiagramm	3	3		
Klassendiagramm	1	1	1	
Paketdiagramm	2	1		
Objektdiagramm	1			1
Kompositionsstrukturdiagramm	2	1		
Komponentendiagramm	2	1		
Verteilungsdiagramm		1		

1= häufig eingesetzt  
2 = benutzbar  
3 = wenige Einsatzmöglichkeiten

