



Spezifikation von Softwarekomponenten auf Qualitätsebene

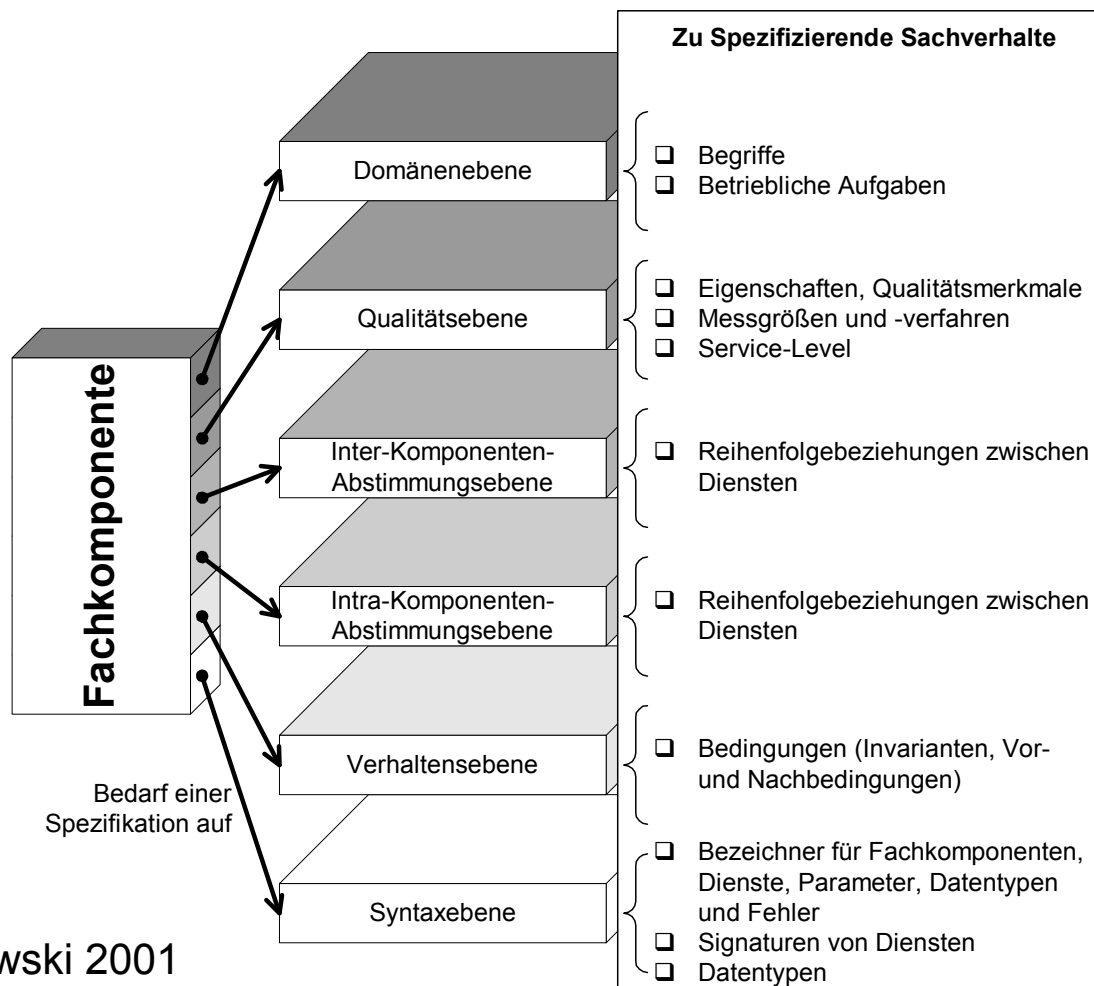
Andreas Schmietendorf, Reiner Dumke

2. Workshop "Modellierung und Spezifikation von Fachkomponenten"
im Rahmen der Fachtagung vertIS 2001
05. Oktober 2001, Bamberg



Agenda

- Einführung und Motivation
- Qualitätseigenschaften von Softwarekomponenten
- Meßansätze von Komponentenarchitekturen
- Spezifikation von Qualitätsmerkmalen
- Zusammenfassung und weitere Entwicklungen



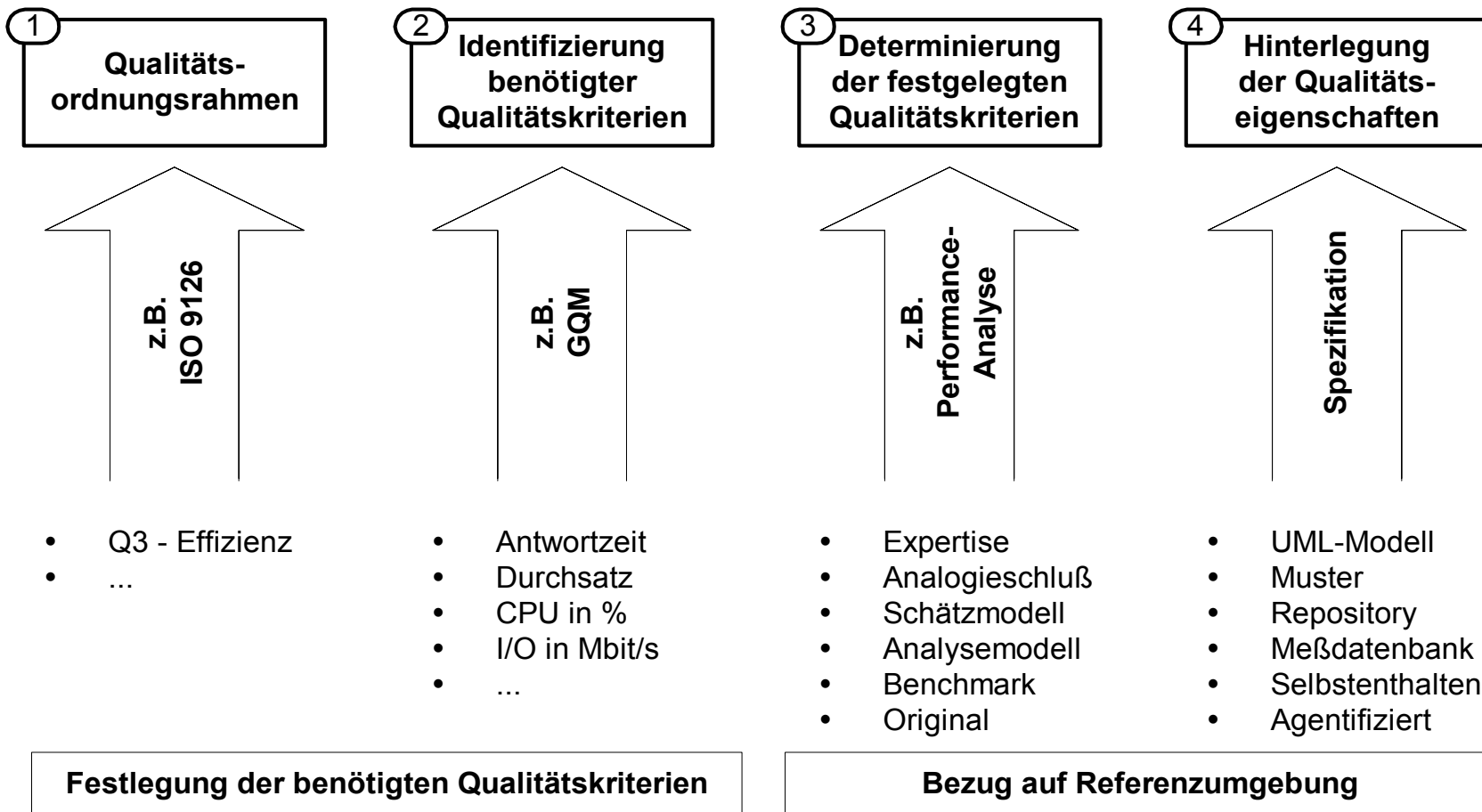
Quelle: Turowski 2001



Randbedingungen einer Spezifikation

- Widerspruchsfrei und eindeutig
- Verwendung von wenigen Notationen
- Gütesiegel im Falle einer COTS-Komponente
- Bezug auf die abgebildete Domaine bei Fachkomponenten
- Umsetzbar mit minimalen Aufwand
- Konvergent innerhalb betroffener Ingenieur-Disziplinen

Schritte zur Qualitäts-Spezifikation





ISO 9126 als Standard für ein Qualitätsmodell vgl. FCM

- *Übertragbarkeit Q1*: Anpassbarkeit, Installierbarkeit,...
- *Anwendbarkeit Q2*: Erlernbarkeit, Beherrschbarkeit, ...
- *Effizienz Q3*: Raum-, Zeit- und Ressourcenbezogen
- *Funktionalität Q4*: Angemessenheit, Interoperabilität, ...
- *Zuverlässigkeit Q5*: Fehlertoleranz, Fehlerhäufigkeit,...
- *Wartbarkeit Q6*: Analysierbarkeit, Änderbarkeit, Stabilität, ...

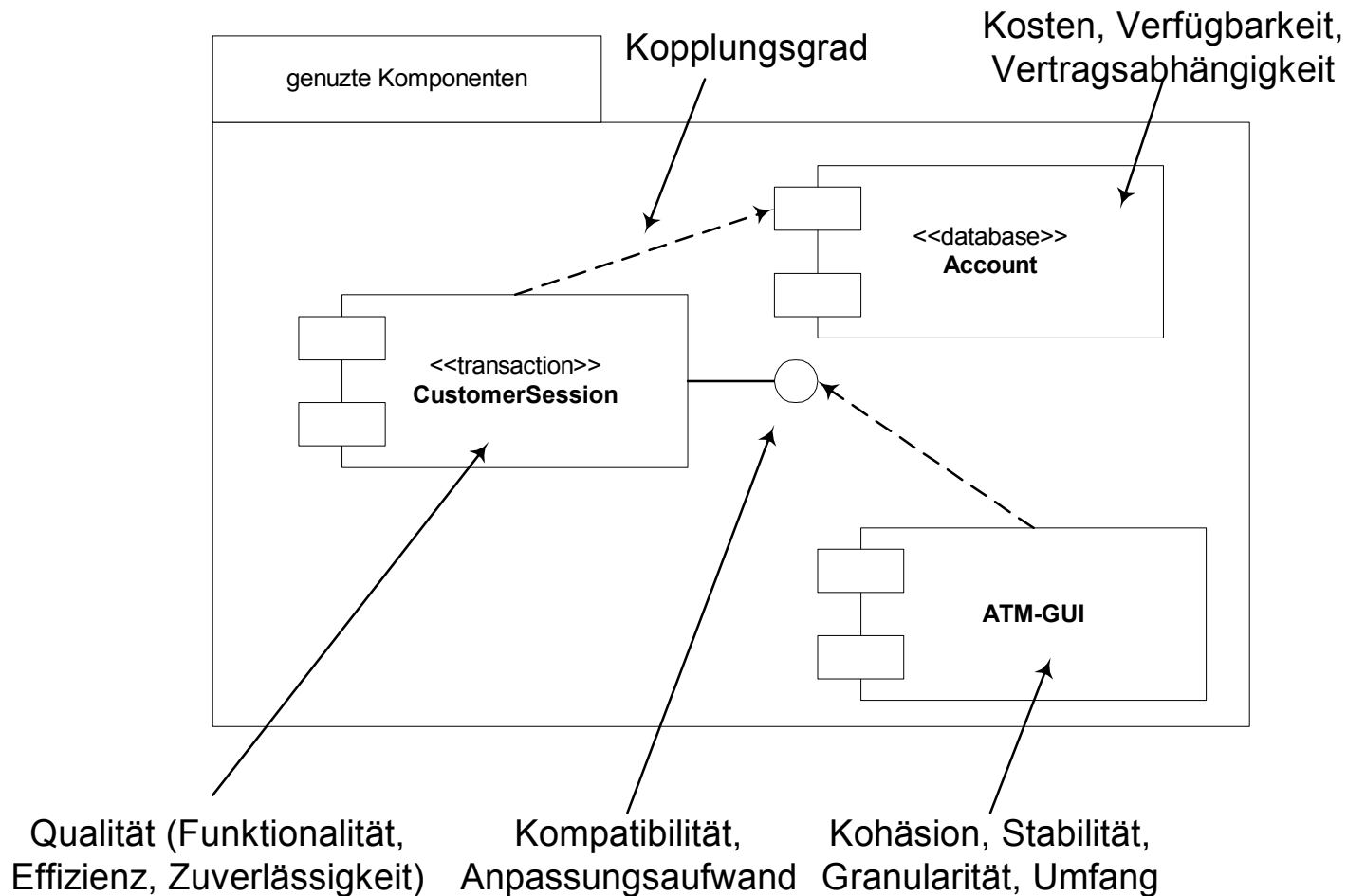


Identifikation von Qualitätskriterien

- GQM-Pardigma (Goal Question Metric)
 - Goal - formulieren von Qualitätszielen (übernommen aus ISO 9126)
 - Question - wie können Qualitätsziele erreicht werden
 - Metric - Quantifizieren der beantworteten Fragen

- Erfolgskriterien für die Beantwortung der Fragen
 - Sichtweise: Komponentenentwickler, Komponentenanwender,...
 - Anwendungsbereich: spezielles Projekt oder auch Produktklasse
 - Zweck: Analyse, Verständnis,...
 - Kontext: z.B. im Rahmen eines ausgewählten Entwicklungsteams

Mögliche Analyse- bzw. Meßansätze





Technischer Report des SEI (Klein 1996):

No standard of practice exist for how to evaluate the performance of a software component. As reliance on COTS increases, the risk of discovering performance problems after it is too late may increase for systems for which performance predictability is critical.



Verfügbare Arbeiten

- EJB-Vergleichs-Projekt (Charles University of Prague)
 - Funktionalität, Interoperabilität und Performance
 - Overhead der Transaktionsverarbeitung
- ECPeefTM Benchmark von SUN für J2EE-Server
 - Test des Containers hinsichtlich Ressourcenbedarf
 - Performanceanalyse des Bean-Life-Cycle (inkl. Pooling)
 - Einfluß von Umfang und Komplexität des Beans
- EJB-Server-Vergleich (TU Dresden, Prof. Schill)
- ...

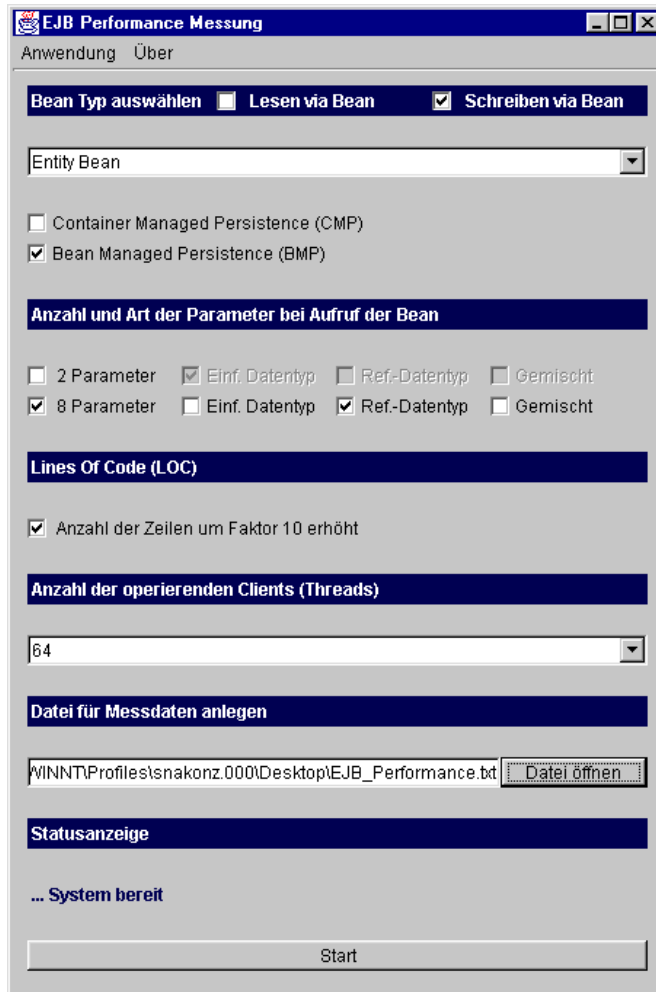


■ Zielstellungen

- Antwortzeit und Durchsatz angebotener Methoden des Remote-Interface
- Einfluß der übergebenen Parameter auf die Performanceeigenschaften
- Performanceanalyse unterschiedlicher EJB-Typen
- Einfluß von Umfang und Komplexität des Beans

■ Zu lösende Problemstellungen

- Definition einer Referenzumgebung (HW und SW)
- Abbildung des Ressourcenverbrauchs
- Verwendung eines generischen Lastmodells
- Genauigkeit der Zeitmessung unter Java



Konfiguration:

- EJB-Typen (EB, SFS, SLS)
- Schreibende und lesende Zugriffe
- Variation der Parameteranzahl (2 oder 8)
- Variation der Parametertypen
- Variation der Bean-Größe über LoC
- Persistenzmechanismen (CMP, BMP)
- Anzahl aufrufender Clients

Ergebnisse:

- Zeit für die Bean-Deserialisierung
- Zeit für die Bean-Instanziierung
- Zeit zur Methodenausführung

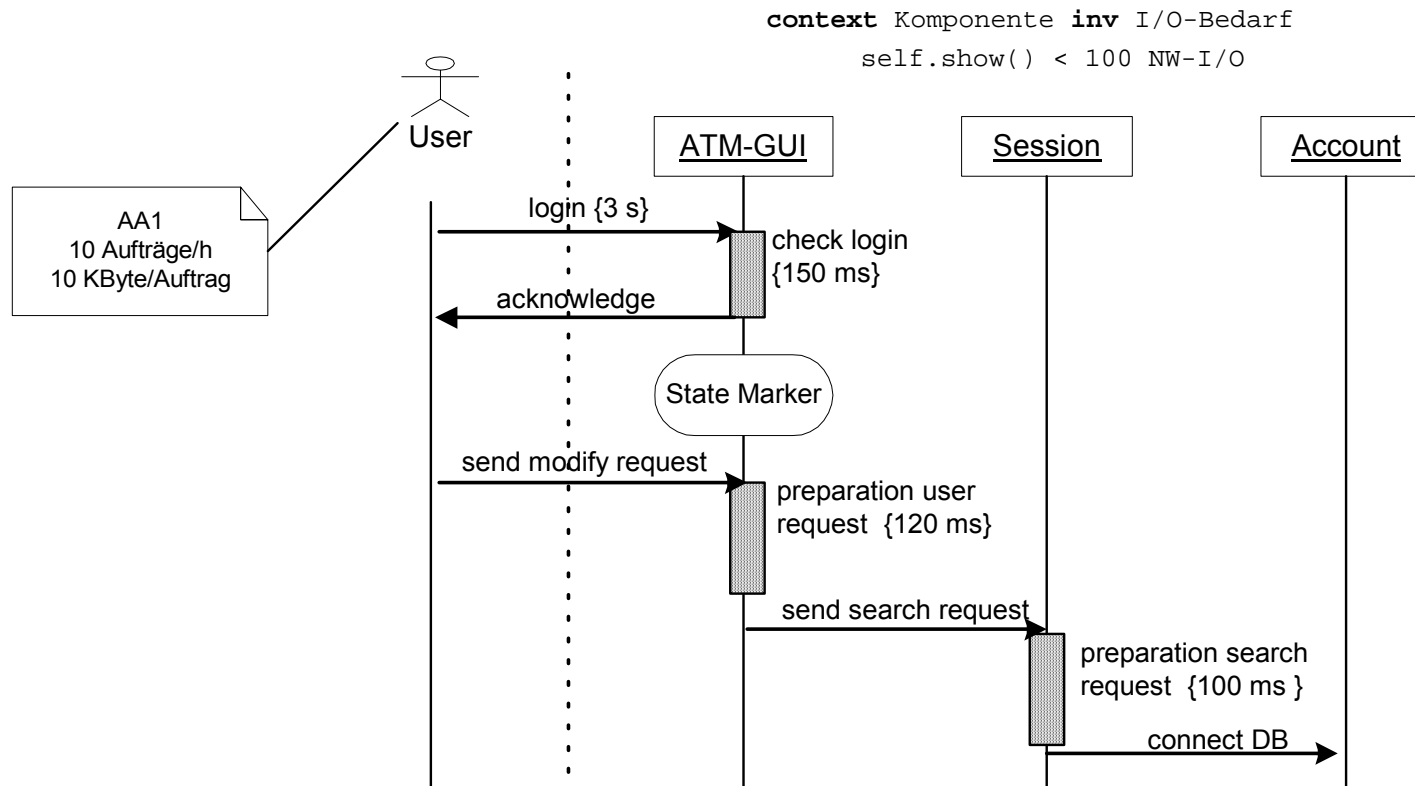


UML-Notation im Kontext der Effizienz

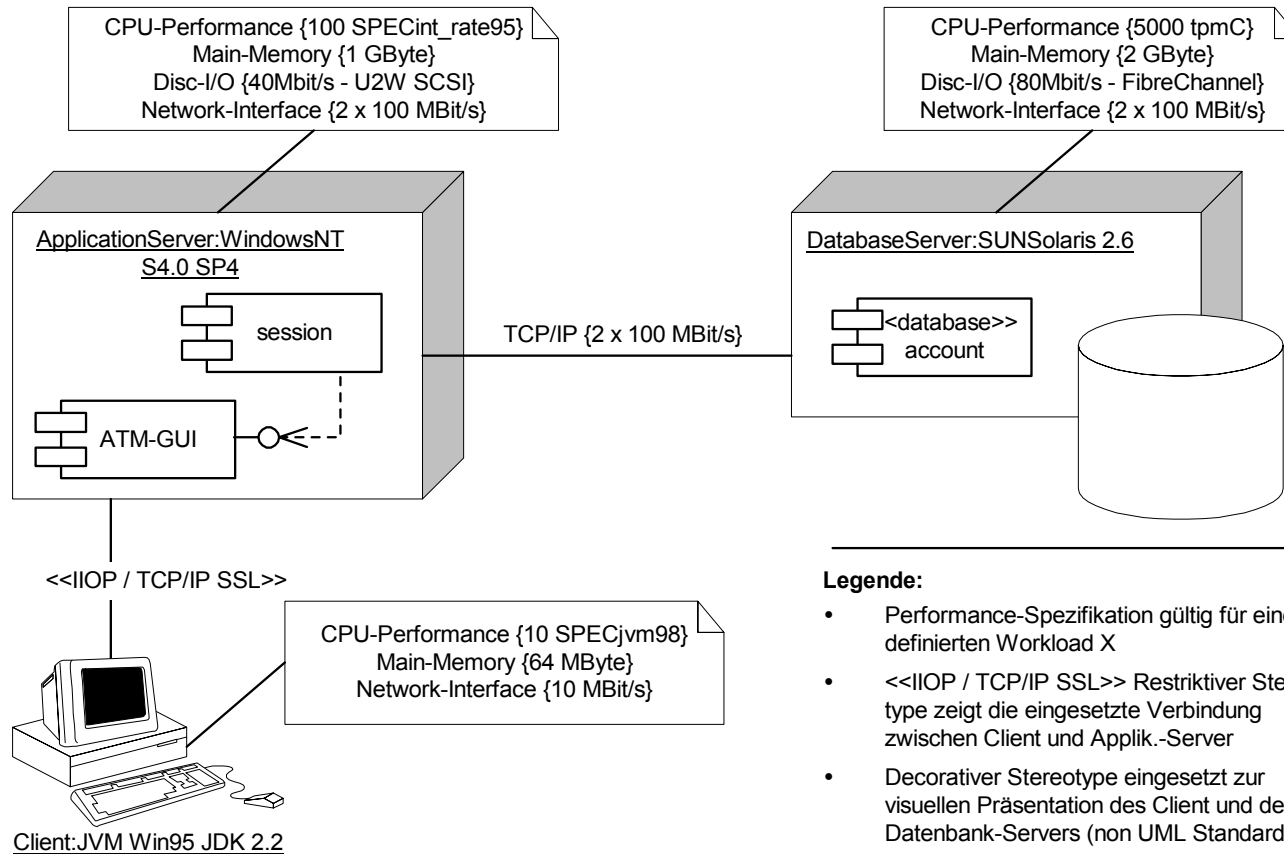
- **Verwendbare UML-Diagrammtypen**
 - Interaktionsdiagramm (Lastmodell, Zeit- und Ressourcenverhalten)
 - Paketdiagramm (Performance der Schnittstellen)
 - Zustandsdiagramm (mögliche Zustände von Komponenten)
 - Verteilungsdiagramm (abstrakte Zuordnung der Komponenten zu Ressourcen)
- **Verwendung von nativen UML-Erweiterungsmechanismen**
 - Stereotypen (z.B. annotative und restriktive Sterotypen)
 - Tagged Values (Zeitverhalten)
 - Annotationen (Beschreibung des Schnittstellenverhaltens)
 - Constraints (benötigte Ressourcen oder auch Zeitverhalten über OCL)
- **Berücksichtigung weiterer Entwicklungen der OMG**



Zeitliches Interaktionsverhalten



Verteilung der Hardwareressourcen



Legende:

- Performance-Spezifikation gültig für einen definierten Workload X
- <<IOP / TCP/IP SSL>> Restriktiver Stereotype zeigt die eingesetzte Verbindung zwischen Client und Applik.-Server
- Decorativer Stereotype eingesetzt zur visuellen Präsentation des Client und des Datenbank-Servers (non UML Standard)



Vorschlag für das Memorandum

- Schritte zur Spezifikation allgemein darstellen
- Darstellung eines FCM-Modells (z.B. ISO 9126)
- Schritt zur Identifizierung notwendiger Qualitätskriterien
- Schritt der Analyse als Beispiel in Bezug auf Performance
- Möglichkeiten der Spezifikation kurz umreißen
 - UML-Notation wie hier dargestellt inkl. Verwendung der OCL
 - Muster inkl. eines entsprechenden Anwendungsbeispiels



Ausblick auf weitere Arbeiten

- weitere Möglichkeiten zur Spezifikation
 - Ablage im Rahmen eines Repository
 - Messdatenbank vs. Experience Factory
 - Selbstenthalten z.B. innerhalb des EJB-DD
 - Agentifiziert (dynam. Reaktion >> SLA/SLO)
- Verbreitung des empirischen Erfahrungshintergrunds
- Abbildung der Referenzumgebung
- CURE - COTS usage risk evaluation vs. Qualitätsebene?
- Analyse der internen und externen Metriken des ISO 9126
- Aufwandsproblematik bei der Spezifikation



Weitere Informationen über:

- **Andreas Schmietendorf**

andreas.schmietendorf@t-systems.de

schmiete@ivs.cs.uni-magdeburg.de

- **Reiner Dumke**

dumke@ivs.cs.uni-magdeburg.de