



ARCHITEKTURKONZEPTE



Zeinab Barakat, Hilal Cubukcu und
Sun-Min Yook



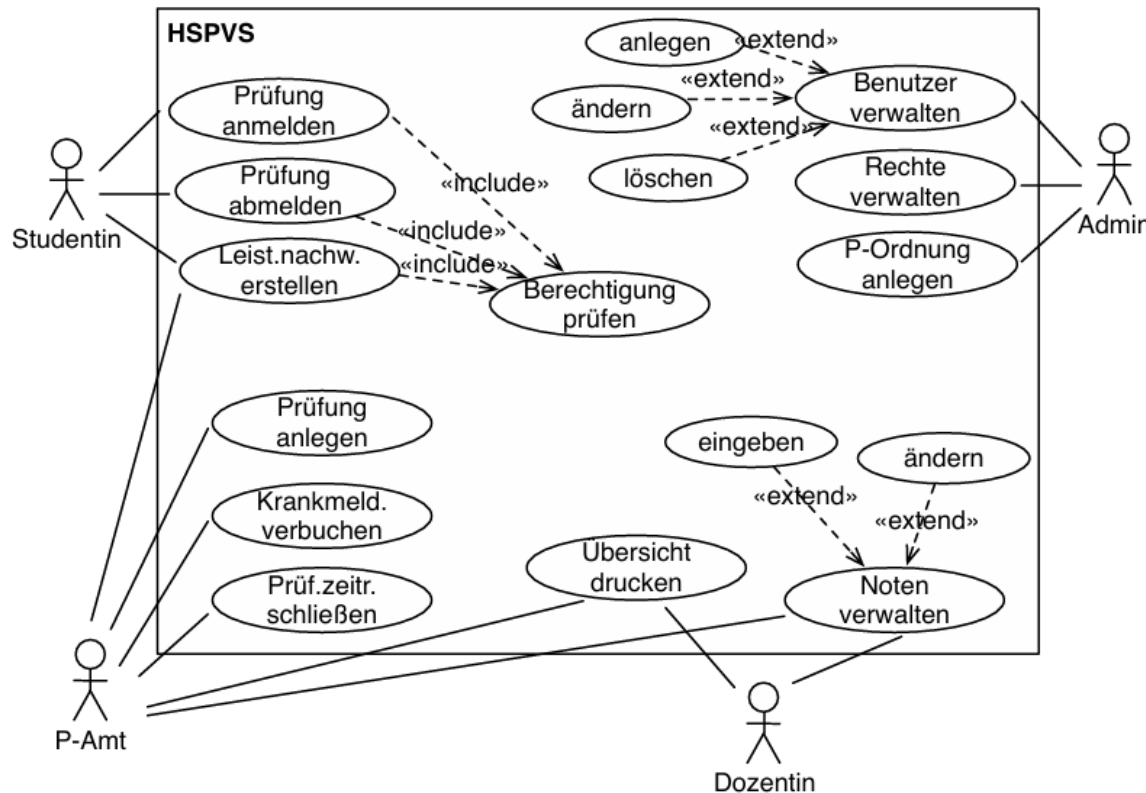
KONTEXTABGRENZUNG / KONTEXTSICHT

- zeigt den Zusammenhang des Systems mit seiner Umgebung → alle beteiligten Benutzer und Fremdsysteme werden dargestellt
- auf technischer Ebene und auf funktionaler Ebene (z.B. mit Nachbarsystemen oder Akteure kommunizieren) möglich
- beinhaltet folgende Verantwortlichkeiten: Was leistet das System, was leistet es nicht? Wo verläuft die Systemgrenze und welche Fremdsysteme müssen integriert werden?

UML Beispiel: Kontextsicht

Use-Case-Diagramm

Hochschulprüfungsverwaltungssystem



- stellt verschiedene Benutzerrollen (Akteure) und ihre Interaktionen mit den Funktionen eines Systems dar
- zeigt das Umfeld eines Systems und wie die externen Benutzer mit den Funktionalitäten interagieren

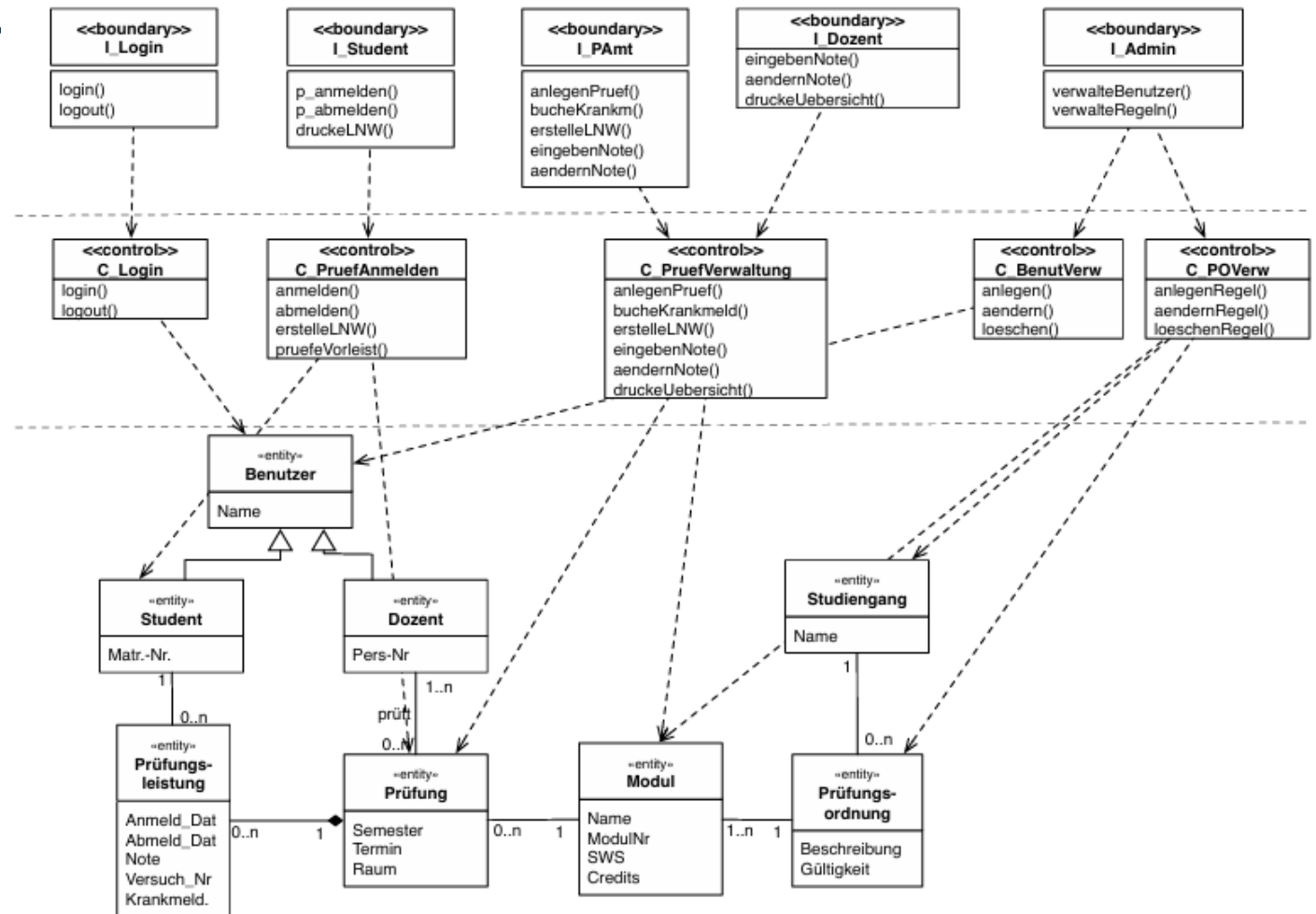
BAUSTEINSICHT / STRUKTURSICHT

- dient hauptsächlich den Entwicklern und beschreibt die statische Struktur der Architekturbausteine des Systems, der Subsysteme, Komponenten und deren Schnittstellen zueinander
 - zeigt die Beziehungen zwischen den Komponenten und welche Kommunikationswege bestehen
- Bausteine bilden einzelne Funktionalitäten ab
- Notation z.B. durch UML-Klassendiagramme & Paketdiagramme

UML-Beispiel: Bausteinsicht

UML-Klassendiagramm Hochschulprüfungsver- waltungssystem

- zeigt die verschiedenen Komponenten (Bausteine) des Systems, darunter Boundary-, Control und Entity- Klassen
- sind durch Assoziationen und Abhängigkeiten miteinander verbunden, die die Kommunikationsbeziehungen darstellen



LAUFZEITSICHT / VERHALTENSICHT

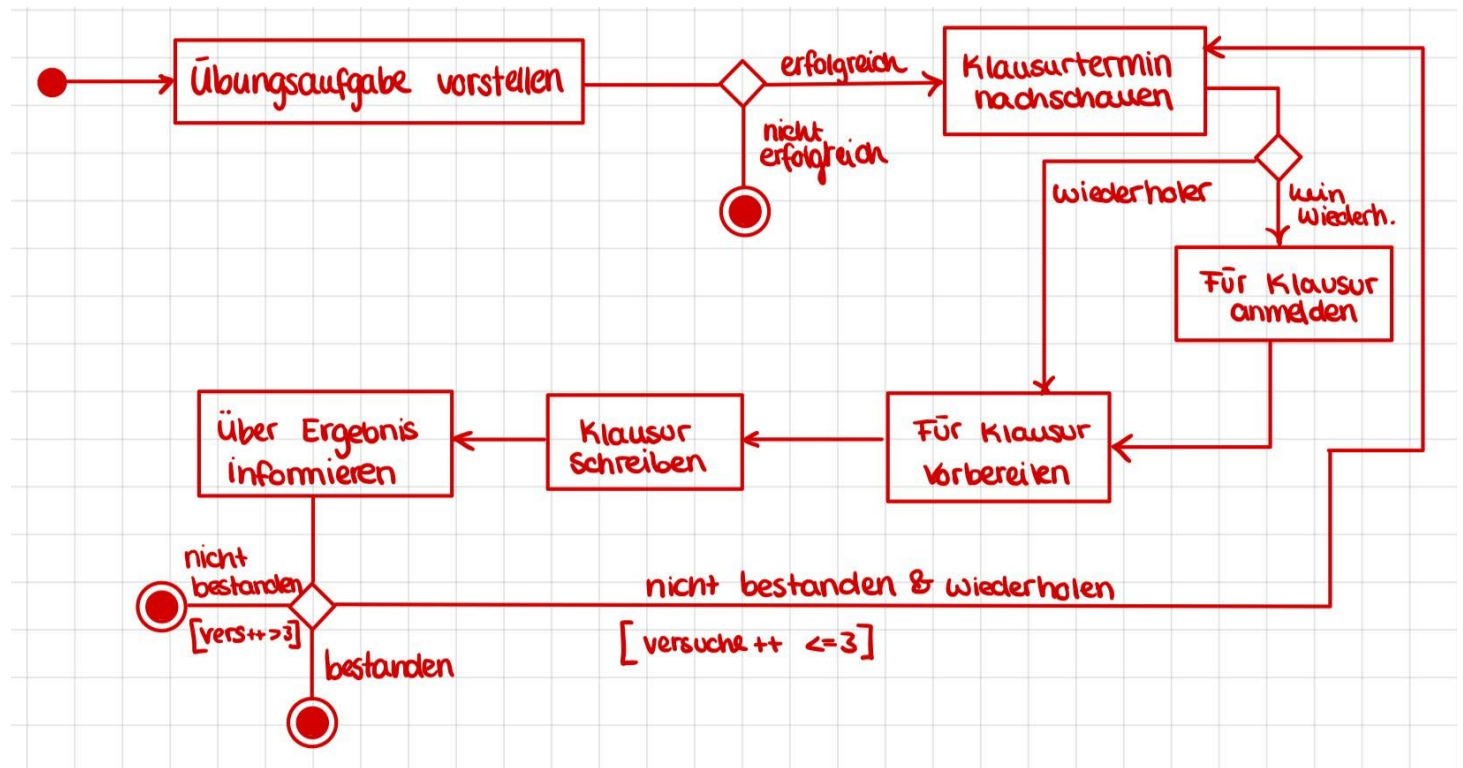
- beschreibt, wie die Bausteine des Systems zur Laufzeit interagieren und sich verhalten
- ist dynamisch und zeigt, wie Objekte miteinander kommunizieren
- Visualisierung der Interaktionen und Zustandsübergänge zwischen den Komponenten
- Notation z.B. durch UML-Sequenz- oder Aktivitätsdiagramme

UML Beispiel: Laufzeitsicht

Aktivitätsdiagramm

Teilnahme an SE1-Klausur

- zeigt die Abfolge von Aktivitäten und Entscheidungen, die ein Student durchläuft, wenn er eine Klausur vorbereitet, schreibt und sich bei Nichtbestehen erneut anmeldet
- ist eine Darstellung von Abläufen, die zur Laufzeit des Systems stattfinden



VERTEILUNGSSICHT / ABBILDUNGSSICHT

- Art Landkarte, in der alle Hardwarekomponenten und die dazugehörigen Protokolle eingezeichnet werden
- stellt alle physikalischen Geräte wie Server, Prozessor, Netzwerke, Firewalls, Router, Speicher usw. dar
- beschreibt technische Verteilung der Softwarekomponenten auf die Hardware: technische Infrastruktur

Physische Verteilung der Softwareschichten

	Vorteile	Nachteile
Zwei-Schichten-Architektur	<ul style="list-style-type: none">- einfach und kostengünstig zu entwickeln und zu betreiben- Direkte Verbindung zw. Client und Server	<ul style="list-style-type: none">- Änderungen am Client oder Server haben oft Auswirkungen auf gesamte Anwendung- Sicherheitsrisiken, da Client direkten Zugriff auf Datenbank haben kann

	Vorteile	Nachteile
Drei-Schichten-Architektur	<ul style="list-style-type: none"> - Gut skalierbar, da die einzelnen Schichten logisch voneinander getrennt sind - Erhöhte Sicherheit, da Clients keinen direkten Zugriff auf Datenbank haben 	<ul style="list-style-type: none"> - Höherer Entwicklungsaufwand und komplexere Infrastruktur
Mehr-Schichten-Architektur	<ul style="list-style-type: none"> - Hohe Skalierbarkeit und Flexibilität - Sehr gute Sicherheit, da die Datenzugriffsschicht isoliert ist - Für große, verteilte Systeme geeignet 	<ul style="list-style-type: none"> - Sehr komplex in Entwicklung und Wartung, erfordert umfangreichere Ressourcen und Infrastruktur - Kostspieliger in der Entwicklung

Quellen

- [4 - Softwarearchitekturen.key](#)
- [Kontextsicht | IT und Enterprise Architektur](#)
- [Artikel Softwarearchitektur \(Teil IV\) - Architektursichten – Object Systems GmbH](#)
- [Schichtenarchitektur – Wikipedia](#)