

Lernzettel

Verhaltensbeschreibung und Verhaltensspezifikation

Universität: Technische Universität Berlin
Kurs/Modul: Softwaretechnik und Programmierparadigmen
Erstellungsdatum: September 19, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Softwaretechnik und Programmierparadigmen

Lernzettel: Verhaltensbeschreibung und Verhaltensspezifikation

(1) Grundidee und Ziel.

Verhaltensbeschreibungen dokumentieren das gewünschte Verhalten eines Systems in typischen Situationen. Sie beschreiben beobachtbares Verhalten, Schnittstellen und zeitliche bzw. reaktive Eigenschaften, ohne Implementierungsdetails vorzudestillieren. Verhaltensspezifikationen dienen dazu, das Verhalten formal oder semi-formal festzuhalten, so dass Verifikation, Tests und Kommunikation über Anforderungen erleichtert werden.

(2) Notationen und Modelle.

Wichtige Modelle zur Verhaltensbeschreibung:

- Zustandsautomaten (State Machines): Zustände, Übergänge durch Ereignisse, ggf. Timings.
- Sequenzdiagramme und Aktivitätsdiagramme: Abläufe in typischen Interaktionen.
- Use Cases: typische Nutzungsszenarien aus Sicht der Akteure.
- Formale Spezifikationen (teilweise): Vorbedingungen, Nachbedingungen, Invarianten.

(3) Vorbedingungen, Nachbedingungen und Invarianten.

Formale Grundlage der Verhaltensspezifikation:

$$\{P\} S \{Q\}$$

- Precondition P : Zustand vor dem Ausführen von S .
- Postcondition Q : Zustand nach Beendigung von S .
- Invariant I : Bedingung, die während bestimmter Abläufe unverändert gilt (z.B. innerhalb einer Schleife).

Wichtige Regeln (Hoare-Kalkül):

$$\begin{aligned} &\{P\} x := E \{P[x := E]\} \\ &\{P \wedge B\} S_1 \{Q\}, \quad \{P \wedge \neg B\} S_2 \{Q\} \\ &\{P \wedge B\} \text{ if } B \text{ then } S_1 \text{ else } S_2 \text{ fi } \{Q\} \end{aligned}$$

(4) Verifikation und Tests.

Ziel ist es, sicherzustellen, dass das beschriebene Verhalten tatsächlich eintritt:

- Tests: Unit-, Integrations- und Akzeptanztests prüfen das beobachtbare Verhalten.
- Formale Verifikation: Hoare-Triple, Invarianten und Nachweisverfahren.
- Modellprüfung/Model Checking: automatische Überprüfung von Zustandsräumen gegen Eigenschaften.

Hinweis: Verhaltensbeschreibung dient als Grundlage für Testfälle und Verifikationspläne.

(5) Beispiel: einfacher Verhaltensfluss eines Login-Workflows.

Zustandenelemente:

- Zustände: Idle, Eingabe, Validierung, ZugriffGesteuert, Erlaubt, Fehler
- Ereignisse: `enterCredentials`, `submit`, `timeout`, `logout`

Beispiel-Übergänge (vereinfachte Darstellung):

- Idle $\xrightarrow{\text{enterCredentials}}$ Eingabe
- Eingabe $\xrightarrow{\text{submit}}$ Validierung
- Validierung $\xrightarrow{\text{Erfolg}}$ ZugriffGesteuert
- Validierung $\xrightarrow{\text{Fehler oder timeout}}$ Fehler
- Fehler $\xrightarrow{\text{logout}}$ Idle

Formale Spezifikation (Voraussetzung/Nachbedingung):

$\{ \text{authenticated} \Rightarrow \text{Zugriffserlaubt} \}, \quad \{ \neg \text{authenticated} \}$ führt zu Fehler/Timeout

Diese Beschreibung erlaubt es, Tests zu Ableitungen der erwarteten Zustandsübergänge und zugehörigen Ausgaben.

(6) Zustandsbasierte Verhaltensbeschreibung (Beispiel-Notation).

Zustände und Übergänge können auch kompakt so notiert werden:

Idle $\xrightarrow{\text{start}}$ Warten $\xrightarrow{\text{fertig}}$ Abgeschlossen

Invariantes Beispiel:

$\text{Balance} \geq 0$ während des Verarbeitungsprozesses

(7) Zusammenhang mit Architektur und Programmierparadigmen.

- Verhaltensbeschreibung klärt Schnittstellen und Interaktionen zwischen Modulen, unterstützt damit modulare Architektur.
- Objektorientierte, modellorientierte, funktionale oder logische Paradigmen profitieren von expliziten Verhaltensmodellen zur Spezifikation von Schnittstellen, Zuständen und Nebenwirkungen.
- Verhaltensspezifikationen dienen als Grundlage für Verifikation, Tests und Qualitätssicherung in unterschiedlichen Entwicklungsprozessen (V-Modell, agiles Vorgehen, modellbasierte Entwicklung).

(8) Qualitätsmetriken zur Verhaltensbeschreibung.

- Abdeckung: Anteil der relevanten Szenarien (Use Cases, Zustandsübergänge) abgedeckt durch Spezifikationen.
- Konsistenz: Widerspruchsfreiheit zwischen Spezifikationen, Vor-/Nachbedingungen und Invarianten.

- Nachvollziehbarkeit: Zuordnung von Spezifikationen zu Anforderungen und Tests.

(9) Weiterführende Hinweise.

Verhaltensbeschreibung ist kontinuierlich zu verfeinern, insbesondere bei Änderungen der Anforderungen oder der Architektur. Sie dient als lebendiges Dokument für Stakeholder, Testerinnen und Tester sowie Entwicklerinnen und Entwickler.