



Spezifikation eines Musikautomaten (Jukebox) mit UML

Luca M. Schmidt

Prüfungsteil 1 (Software Engineering) am Fachbereich AI
der HS Fulda

Matrikelnummer: 1540963

Projektcode: Automatentyp Nr. 0 - Musikautomat (Jukebox)

Erstbegutachtung: Prof. Dr. Cathrin Möller

Eingereicht am 04.06.2025

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	I
1 Funktionsbeschreibung des Musikautomaten	1
2 UML-Diagramme	1
2.1 Aktivitätsdiagramm: Songauswahl und -wiedergabe	2
2.2 Sequenzdiagramm: Songwiedergabe und Interaktion mit externem Gerät	2
3 Fazit	3
3.1 Diagrammqualität und funktionale Abdeckung	3
3.2 Methodische Limitationen	3

Abbildungsverzeichnis

1 Technische Skizze eines modernen Musikautomaten	1
2 Aktivitätsdiagramm: Songauswahl und -wiedergabe mit Nutzer-System- Interaktionen. (Diagramm erstellt von: Luca M. Schmidt)	2
3 UML-Sequenzdiagramm: Zeitliche Abfolge der Komponenten-Interaktionen bei Songwiedergabe	2

1 Funktionsbeschreibung des Musikautomaten

Der Musikautomat ist ein moderner, coin-betriebener Automat für öffentliche Bereiche. Nutzer werfen einen speziellen Coin ein, um Zugang zu einem digitalen Musikkatalog zu erhalten. Die Bedienung erfolgt über ein Touchscreen-Display zur Navigation durch Kategorien (Genre, Künstler, Charts) oder eine Suchfunktion.

Nach der Songauswahl werden Titel einer Warteschlange (Queue) hinzugefügt und in der Reihenfolge abgespielt. Das System zeigt während der Wiedergabe Metadaten wie Interpret und Titel an. Steuerungsoptionen wie Pause oder Skip sind bewusst deaktiviert, um Missbrauch zu vermeiden.

Der Automat ist permanent mit einem externem Gerät des Besitzers vernetzt. Diese ermöglicht die Fernaktualisierung des Musikkatalogs, Konfiguration von Lautstärke und Betriebszeiten sowie die Sammlung von Nutzungsstatistiken für Lizenzabrechnung. Wartung erfolgt sowohl digital (Software-Updates, Katalog-Synchronisation) als auch physisch (Reinigung, Coin-Entleerung, Hardware-Reparaturen).

TODO: Erstelle eine detaillierte technische Skizze des Musikautomaten mit folgenden Elementen: (1) Großes Touchscreen-Display (ca. 21-24 Zoll) mit Benutzeroberfläche zur Songauswahl, (2) Coin-Einwurf Schlitz mit Validierungseinheit, (3) Hochwertige Stereo-Lautsprecher, (4) Verschießbare Service-Zugangsklappe für Wartungspersonal, (5) Netzwerkanschluss/WLAN-Antenne, (6) Status-LED-Anzeigen. Die Skizze sollte sowohl Frontal- als auch Seitenansicht zeigen.

Abbildung 1: Technische Skizze eines modernen Musikautomaten

Die primäre Nutzerinteraktion konzentriert sich auf das zentrale Touchscreen-Display (1), das sowohl zur Navigation im Musikkatalog als auch zur Anzeige von Systeminformationen dient. Der Coin-Einwurfmechanismus (2) ist mit einem integrierten Validierungssystem ausgestattet, das gefälschte oder beschädigte Coins automatisch erkennt und zurückweist. Die Audio-Ausgabe erfolgt über professionelle Stereo-Lautsprecher (3), die für den Dauerbetrieb in öffentlichen Umgebungen optimiert sind. Der Service-Zugang (4) ist durch ein Sicherheitsschloss geschützt und ermöglicht autorisiertem Personal den Zugriff auf interne Komponenten für Wartungs- und Reparaturarbeiten.

2 UML-Diagramme

Die folgenden beiden UML-Diagramme zeigen verschiedene Aspekte des Musikautomaten und ergänzen sich in ihrer Darstellung.

2.1 Aktivitätsdiagramm: Songauswahl und -wiedergabe

Das Aktivitätsdiagramm in Abbildung 2 stellt den typischen Nutzungsprozess vom Coin-Einwurf bis zur Songwiedergabe dar. Swimlanes trennen die Verantwortlichkeiten zwischen 'Nutzer' und 'Jukebox-System'.

TODO: Erstelle ein UML-Aktivitätsdiagramm mit Swimlanes für 'Nutzer' und 'Jukebox-System'. Nutzer-Aktivitäten: 'Coin einwerfen', 'Songs durchsuchen', 'Song auswählen', Entscheidung 'Weiteren Song wählen?'. System-Aktivitäten: 'Coin validieren', Entscheidung 'Coin gültig?', 'Auswahlmodus aktivieren', 'Song-Verfügbarkeit prüfen', Entscheidung 'Song verfügbar und Queue nicht voll?', 'Song zu Queue hinzufügen', 'Fehlermeldung anzeigen', Fork für parallele Wiedergabe: 'Song abspielen', 'Metadaten anzeigen', 'Status an Zentrale melden'. Objektknoten: 'Coin', 'Song-Queue', 'Ausgewählter Song'.

Abbildung 2: Aktivitätsdiagramm: Songauswahl und -wiedergabe mit Nutzer-System-Interaktionen. (Diagramm erstellt von: Luca M. Schmidt)

Ablauf: Der Nutzer wirft einen Coin ein, das System validiert ihn und aktiviert bei Gültigkeit den Auswahlmodus. Nach der Songauswahl prüft das System die Verfügbarkeit und fügt verfügbare Songs zur Queue hinzu. Parallele Wiedergabe-Aktivitäten umfassen Abspielen, Metadaten-Anzeige und Zentrale-Kommunikation.

2.2 Sequenzdiagramm: Songwiedergabe und Interaktion mit externem Gerät

Das Sequenzdiagramm in Abbildung 3 fokussiert sich auf die zeitliche Choreographie der Nachrichten und Interaktionen zwischen den verschiedenen Systemkomponenten während des kritischen Prozesses der Songwiedergabe. Besondere Aufmerksamkeit gilt dabei der Kommunikation mit dem externem Gerät des Besitzers, welches für Lizenzmanagement, Nutzungsstatistiken und Systemüberwachung verwendet wird.

TODO: Erstelle ein UML-Sequenzdiagramm mit folgenden Lifelines: Nutzer, Jukebox-UI, Jukebox-Steuerung, LokaleMusikdatenbank, Überwachungszentrale. Zeige den zeitlichen Ablauf der Nachrichten: `selectSong()`, `spieleSongAnfrage()`, `getSongStream()`, `songDatenLiefern()`, `playSongRequest()`, `showCurrentSong()`, `meldeSongGespielt()`, `confirmReceipt()`. Verwende Aktivierungsbalken und korrekte UML-Notation.

Abbildung 3: UML-Sequenzdiagramm: Zeitliche Abfolge der Komponenten-Interaktionen bei Songwiedergabe

Interaktionsanalyse:

Der Ablauf beginnt mit der Songauswahl durch den Nutzer (`selectSong()`) über die

UI, die eine `playSongRequest()` an die Steuerung sendet. Diese fragt parallel Audiodaten von der lokalen Datenbank ab (`getSongStream()`, `deliverSongData()`) und validiert die Lizenz über das externe Gerät (`validateLicense()`, `licenseConfirmation()`). Nach erfolgreicher Validierung startet die Wiedergabe (`startAudioPlayback()`) und die UI wird aktualisiert (`showCurrentSong()`). Abschließend wird die Wiedergabe protokolliert (`reportSongPlayed()`, `confirmReceipt()`) während kontinuierliches Monitoring zwischen Steuerung und dem externem Gerät erfolgt (`sendStatusUpdate()`, `configurationUpdate()`).

3 Fazit

Die UML-basierte Spezifikation des Musikautomaten mittels Aktivitäts- und Sequenzdiagramm bietet unterschiedliche Perspektiven auf die Systemfunktionalität und ermöglicht eine strukturierte Analyse der komplexen Nutzer-System-Interaktion.

3.1 Diagrammqualität und funktionale Abdeckung

Das Aktivitätsdiagramm (2) visualisiert den nutzerzentrierten Hauptprozess transparent durch Swimlanes und erfasst sowohl optimale Abläufe als auch wesentliche Verzweigungen.

Das Sequenzdiagramm (3) ergänzt dies durch detaillierte Darstellung zeitlicher Abhängigkeiten und Nachrichtenaustausche, insbesondere mit externen Systemen für Lizenzvalidierung und Compliance.

Vollständig abgedeckte Aspekte: Kompletter Nutzungsworkflow, interaktive Songauswahl, Lizenzvalidierung, grundlegende Fehlerbehandlung, bidirektionale Kommunikation mit Überwachungsinfrastruktur und Echtzeit-Statusübermittlung.

Teilweise abgedeckte Funktionalitäten: Erweiterte Fehlerbehandlung (Netzwerkausfälle, Hardwaredefekte) und Wartungsprozesse sind nur konzeptuell erfasst, was jedoch für die primär manuellen Offline-Tätigkeiten vertretbar ist.

3.2 Methodische Limitationen

Die gewählten Diagrammtypen decken die Kernfunktionalitäten umfassend ab, stoßen jedoch bei folgenden Aspekten an Grenzen:

- **Systemzustände:** Betriebsmodi und Transitionen wären durch Zustandsdiagramme präziser modellierbar
- **Architektur:** Strukturelle Aspekte erfordern Komponenten- oder Verteilungsdiagramme
- **Datenmodell:** Persistente Strukturen benötigen Klassendiagramme

- **Nebenläufigkeit:** Parallele Prozesse sind in den gewählten Diagrammen nur eingeschränkt darstellbar

Die Diagramme erfüllen ihren Zweck als prozess- und interaktionsorientierte Spezifikation, während strukturelle und zustandsbasierte Aspekte durch ergänzende UML-Diagrammtypen vollständiger erfasst werden könnten.