Modellierung und prototypische Implementierung eines Zustandsautomaten für Aircraft States im System AMAN

Praxisphase 2, Wintersemester 2023/24, Dominik Dietz

Projektmotivation

Ausgangssituation

- Die vom System geplanten Flüge besitzen viele **Eigenschaften & Zustände**
- Daraus ergibt sich ein **Planungszustand**, auch „**Aircraft State**“ → siehe **Abb.1**
- Viele Faktoren können **Zustandsübergänge** auslösen

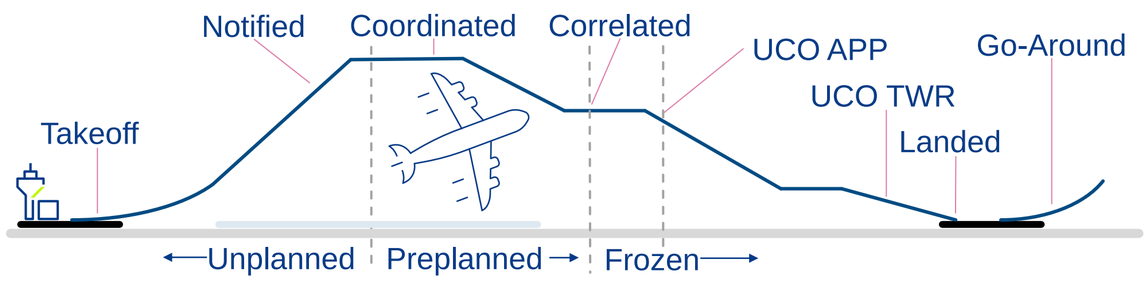

Abb. 1: Exemplarisches Flugprofil mit einer Auswahl an Zuständen und Eigenschaften

Problem

- Zustände sind nicht explizit definiert, sondern leiten sich aus **Kombination** verschiedener **Eigenschaften** ab
- Systemverhalten hängt maßgeblich von diesen Eigenschaften ab
- Beteiligung vieler Komponenten
- Abhängigkeit der Module

→ Hohe Komplexität, Wartungsaufwand

→ **Implizites State Handling**

Lösungsansatz

- Zustandsautomat als zentrale Stelle für das Management der Planungszustände
- Modellieren des Automaten aus Anforderungen
- Definition diskreter Zustände
- Event-getriebene Zustandsübergänge

→ **Explizites State Handling**

Was ist Arrival Management?

- Konzept zur Verkehrsflusssteuerung in der Anflugkontrolle
- Erstellung einer **optimalen Anflugsequenz** für hoch frequentierte Flughäfen oder komplexe Lufträume

→ effiziente Nutzung der Landebahnen, verringert Warteschleifen

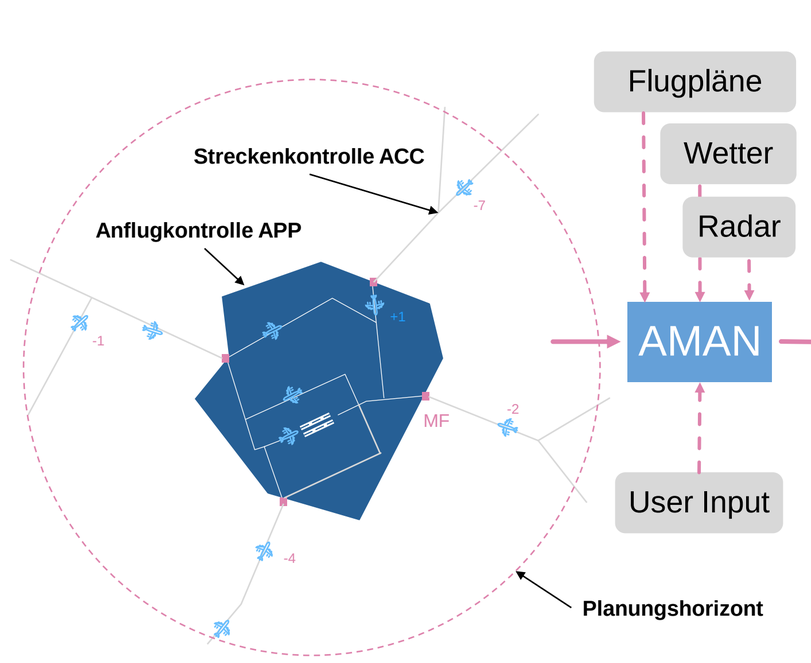

Abb. 2: Schematische Darstellung Luftraum und Systemumfeld

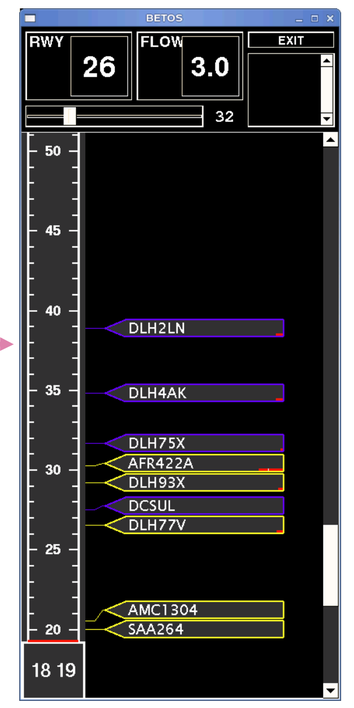

Abb. 3: AMAN GUI

Ziel: Verbesserung der Software hinsichtlich **Verständlichkeit**, **Wartbarkeit**, **Erweiterbarkeit**, **losere Kopplung**

Methodik

Anforderungsanalyse & Modellierung

- Analyse der **Systemanforderungen** des größten Kunden/Projekts
- **Tabellarische (hierarchische) Strukturierung** von Zuständen und Zustandsübergängen
- Graphische Darstellung in Form mehrerer **UML-Zustandsdiagramme**
  - → **Strukturierung** durch **hierarchische Komponenten-Ansicht**
  - → siehe **Abb. 6** und **Abb. 7** für eine **Auswahl**
- **Iteratives Vorgehen:** Regelmäßiger Austausch mit dem Entwickler-team & Präsentation der Zwischenergebnisse beim Kunden

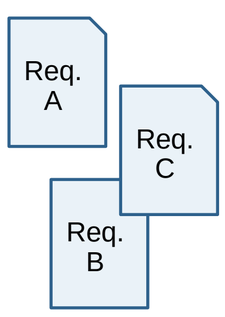

Abb. 4: Anforderungen

Source	Event	Guard	Action	Target

Abb. 5: Zustandsinformationen

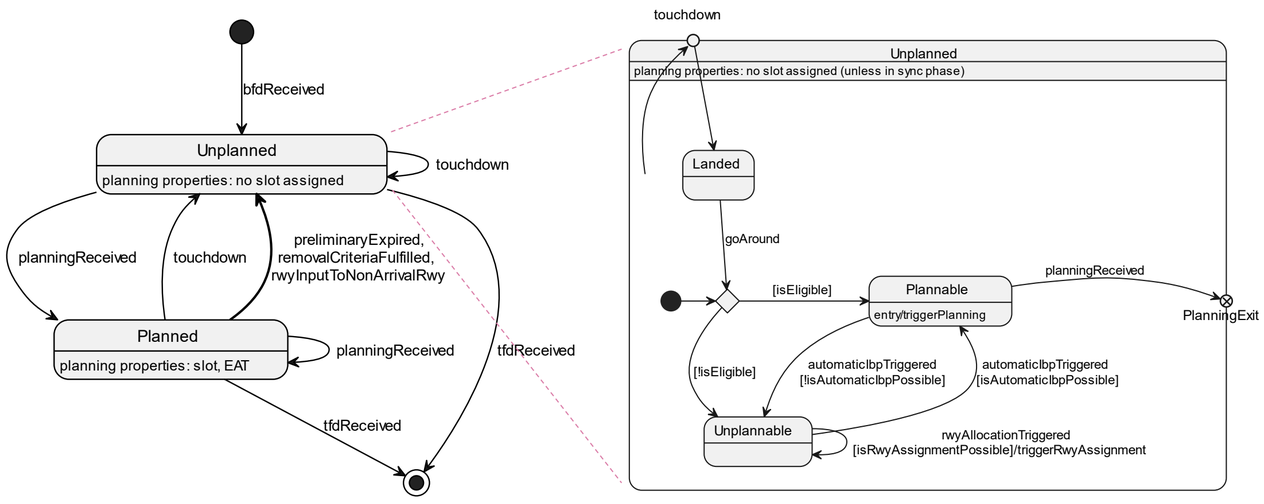

Abb. 6: UML Zustandsdiagramm der obersten Ebene des Automaten

Abb. 7: UML Zustandsdiagramm des Superzustands / Subautomaten „Unplanned“ mit weiteren Subzuständen

```
struct transition_table : vector<
    Row<Unplannable, automaticIbpTriggered,  Plannable,   isAutomaticIbpPossible>,
    Row<Plannable,   automaticIbpTriggered,  Unplannable, Not_<isAutomaticIbpPossible> >,
    Row<Unplannable, rwyAllocationTriggered, Unplannable, isRwyAssignmentPossible>,
    Row<Landed,      goAround,               EligibilityDecision> > {}; ...
```

Abb. 8: Ausschnitt des Source Codes. Die „Boost Meta State Machine” Library erlaubt eine tabellarische Notation für Zustandsübergangstabellen

Umsetzung des Modells in C++ State Machine Library

- Entscheidung für die „***Boost Meta State Machine***“ Library
- Prototypische Implementierung in separatem Projekt zur **schnelleren Iteration**
  - → Geringere Compile-Zeiten
  - → Vermeidung von Nebeneffekten
- **Validierung** der benötigten Features
- Vollständige Implementierung des modellierten Automaten

Integration in AMAN

- Integration des Automaten **als eigene C++ Library**
- **Event-getriebene** Zustandsänderungen können im Planungsalgorithmus getriggert werden
- **Logging-Infrastruktur** im plantUML-Format zur **Visualisierung** von Zustandsinformationen
- Verifizierung mit **Regressions- & Integration-Tests**
  - → Build Checks, CI-Pipeline mit über 450 Tests
- Eigene **Testszenarien** mittels **Robot Framework**

→ **Proof of Concept anhand einzelner Szenarien**

Ergebnisse

- ✓ **Dokumentation und Modellierung** des zustand- & event-spezifischen Systemverhaltens als **nützliches Dokument** für das Entwicklungsteam
- ✓ **Zustandsautomat in C++** im Kern von AMAN als **Werkzeug** zur Restrukturierung von eventgetriebenem Verhalten
- ✓ **Visualisierung** der durchlaufenen Zustände durch Logging
- ✓ **Proof of Concept** eines Zustandsautomaten im System AMAN
- ✓ **Funktionsfähiger Entwicklungsstand** zur Übergabe an Projektteam oder Nachfolger

h_da hochschule darmstadt informatik

member of EUT+ EUROPEAN UNIVERSITY OF TECHNOLOGY

DFS Aviation Services

Praxisprojekt 2
Informatik Dual
2024

Gruppe A