

ENTWICKLUNG EINER DOMÄNENSPEZIFISCHEN MODELLIERUNGSSPRACHE ZUR KONFIGURATION UND IMPLEMENTIERUNG EINES GENERISCHEN LANDEBAHNZUWEISUNGSSYSTEMS

D. Troegner, Institut für Flugführung, Deutsches Zentrum für Luft- und Raumfahrt e.V.,
Lilienthalplatz 7, 38108 Braunschweig und
Fachgebiet für Simulation, Institut für Systems Engineering, Fakultät für Elektrotechnik
und Informatik, Leibniz Universität Hannover, Welfenplatz 1, 30167 Hannover

Zusammenfassung

Die effiziente Nutzung von Landebahnkapazitäten gewinnt innerhalb des modernen Air Traffic Managements zunehmend an Bedeutung. In den letzten Jahren ist ein steigendes Verkehrsaufkommen zu beobachten, welches in Verbindung mit den begrenzten Kapazitäten an Flughäfen zu kostenintensiven Wartezeiten und einer zunehmenden Arbeitsbelastung der Fluglotsen führt. Unterstützungssysteme sollen auch bei einem hohem Verkehrsaufkommen eine strategische Entscheidungsfindung und damit eine effiziente Auslastung der Landebahnkapazitäten ermöglichen. Die Anpassung eines Unterstützungssystems an die flughafenspezifischen Randbedingungen, die geltenden gesetzlichen Bestimmungen und die Präferenzen des Anwenders sind bisher mit einem erheblichen Arbeitsaufwand verbunden. Die Entwicklung einer domänenspezifischen Modellierungssprache (DSL) für diesen Anwendungsbereich ermöglicht nun eine Entwicklung auf Domänen-ebene und in Verbindung mit einer darauf aufbauenden Codetransformation eine generische Erzeugung flughafenspezifischer Systeme. Im Folgenden wird die entwickelte Modellierungssprache AMAN-ML (Arrival Management Modeling Language) für die Komponente der Landebahnzuweisung eines Anflugplanungssystems in Aufbau und Funktionsweise vorgestellt.

1. EINLEITUNG

Das Wachstum des Luftverkehrs in Europa wird durch die an Flughäfen zunehmend auftretenden Kapazitätsengpässe und die damit verbundenen Wartezeiten stark beeinträchtigt. Neue Konzepte für die Planung und Führung des Luftverkehrs bilden daher einen Schwerpunkt aktueller Forschung im Bereich des Air Traffic Managements (ATM). Bodenbasierte Planungssysteme zur Unterstützung von Fluglotsen sind Teil dieser Konzepte, zu denen auch Anflugplanungssysteme (Arrival Manager - AMAN) gehören. Diese unterstützen Fluglotsen bei der Planung und Koordinierung des anfliegenden Verkehrs im Flughafennahbereich.

Die Landebahnzuweisung als Komponente der Anflugplanung nimmt eine zunehmend bedeutende Rolle hinsichtlich der Kapazitätsoptimierung von Flughäfen ein. Aufgabe der Landebahnzuweisung ist es, eine effiziente Verteilung von ankommenden Flugzeugen auf die verfügbaren Landebahnen zu gewährleisten. Ziel ist die Verringerung von Wartezeiten und die Erhöhung bzw. Auslastung der Kapazität, während flughafenspezifische Randbedingungen und gesetzliche Vorgaben beachtet werden müssen. Die domänenspezifische Modellierungssprache AMAN-ML (Arrival Management Modeling Language) ermöglicht eine intuitive Erstellung von Modellen für Flughäfen. Dies erlaubt die generische

Erstellung von lauffähigem Code für ein Landebahnzuweisungssystem unter der Beachtung der jeweils geltenden Randbedingungen. AMAN-ML wird derzeit zur Konfiguration und Implementierung der Komponente Landebahnzuweisung innerhalb des am Institut für Flugführung des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR) entwickelten Anflugplanungssystems 4D-CARMA (4-Dimensional Cooperative Arrival Manager) eingesetzt.

Die Motivation für die Erstellung einer graphischen Modellierungssprache zur Darstellung komplexer Zusammenhänge bei flughafenspezifischen Systemen, wie dem Anflugplanungssystem 4D-CARMA, wird zu Beginn dieses Beitrags in Kapitel 2 erläutert. Eine tiefer führende Beschreibung des Anflugplanungssystems und insbesondere der Komponente der Landebahnzuweisung erfolgt in Kapitel 3. Die Grundlagen domänenspezifischer Modellierung und ihre Vorteile werden dann in Kapitel 4 vorgestellt, bevor Aufbau und Funktionsweise der Modellierungssprache AMAN-ML ausführlich dargestellt werden. Abschließend erfolgen im letzten Kapitel eine Zusammenfassung bisheriger Einsatzergebnisse und ein Ausblick auf zukünftige Entwicklungen.

2. MOTIVATION

Anflugplanungssysteme dienen der Unterstützung von Fluglotsen bei der Planung und Koordinierung des anfliegenden Luftverkehrs innerhalb eines Flug-

hafennahbereichs (TMA). Das Planungssystem berechnet hierzu Anflugtrajektorien für alle Flugzeuge unter Berücksichtigung unterschiedlicher Randbedingungen, zu denen Sicherheitsabstände, geplante Ankunftszeiten und Landebahnuweisungen gehören. Grundlegend für diese Berechnungen sind insbesondere flughafenspezifische Daten, wie Referenzkoordinaten, Flugrouten, Navigationspunkte und Landebahnkonfigurationen. Ebenfalls ist die Planung von den jeweils geltenden rechtlichen Rahmenbedingungen abhängig. Zu diesen zählen Gesetze, Verordnungen und Bestimmungen, wie das Luftverkehrsgesetz (LuftVG), die Standards und Empfehlungen der International Civil Aviation Organisation (ICAO) oder auch das Luftfahrthandbuch (AIP). Insbesondere sind aber auch Erfahrungswissen, Heuristiken und Anwenderpräferenzen wichtige Bestandteile eines Anflugplanungssystems. Dementsprechend können zusätzlich verschiedene Landebahnuweisungsstrategien, Richtlinien von Airlines oder lotsenspezifische Verhaltensweisen, wie z.B. Abweichungen vom Standard-Routing, eine effiziente Planung nachhaltig beeinflussen.

Die Anpassung eines Unterstützungssystems an die flughafenspezifischen Randbedingungen, die gelten den gesetzlichen Bestimmungen und die Präferenzen des Anwenders sind bisher mit einem erheblichen Arbeitsaufwand verbunden. Umfangreiche Reengineering-Maßnahmen müssen vorgenommen werden, um problemspezifische Codeelemente zu extrahieren und an die neuen Anforderungen anzupassen. Eine Möglichkeit dies zu umgehen, liegt in der Analyse der Variabilität sämtlicher spezifischer Daten und Zusammenhänge und ihrer Verlagerung aus der Codierung heraus. Es entsteht ein generisches System, welches eine Trennung von statischen und spezifischen Codeelementen ermöglicht. Das Konzept der domänenspezifischen Modellierung setzt an diesem Punkt an und erlaubt eine Modellerstellung unter Verwendung von Domänenkonzepten, mit denen derartig spezifische Daten und Zusammenhänge dargestellt und direkt in die Codegenerierung integriert werden können. Neben einer Ersparnis an Arbeitsaufwand führt dieser Ansatz ebenfalls zu einer Visualisierung der domänenspezifischen Zusammenhänge und kann darüber hinaus auch eine Analyse des Flughafensystems und das Testen neuer Parameter vereinfachen.

3. LANDEBAHNUWEISUNG ALS KOMPONENTE DES ANFLUGPLANUNGSSYSTEMS 4D-CARMA

Der anliegende Luftverkehr eines Flughafennahbereichs wird von Fluglotsen überwacht, zu deren Aufgaben die Bildung einer Anflugsequenz, die Zuteilung eines Landezeitpunktes und einer Landebahn, die Vorausberechnung eines horizontalen und vertikalen Anflugweges (4D-Trajektorie) sowie die Umsetzung dieser Vorgaben in entsprechende Füh-

rungsanweisungen gehören. Das Anflugplanungssystem 4D-CARMA unterstützt Fluglotsen bei diesen Aufgaben, in dem es zunächst unter Berücksichtigung der aktuellen Luftraumstruktur die möglichen verschiedenen Anflugwege von der aktuellen Position der Luftfahrzeuge (LFZ) bis zur Landebahnschwelle berechnet. Die zu überfliegenden Wegpunkte unterliegen Beschränkungen bezüglich Höhe und Geschwindigkeit, die für jedes LFZ einzuhalten sind.

Anhand der festgelegten Flugwege, der Wetterdaten und der Leistungsdaten der einzelnen Flugzeuge (BADA) kann ein Zeitintervall für das Erreichen der Landebahnschwelle berechnet werden. Das Intervall aus frühest- und spätmöglicher Landezeit (Earliest/Latest Estimated Time of Arrival) wird unter Berücksichtigung des kürzesten und längsten verfügbaren Flugweges aufgespannt [1]. Innerhalb eines Planungsablaufes, wie er in BILD 1 dargestellt ist, wird diese Berechnung für jedes im System bekannten LFZ aufgerufen.

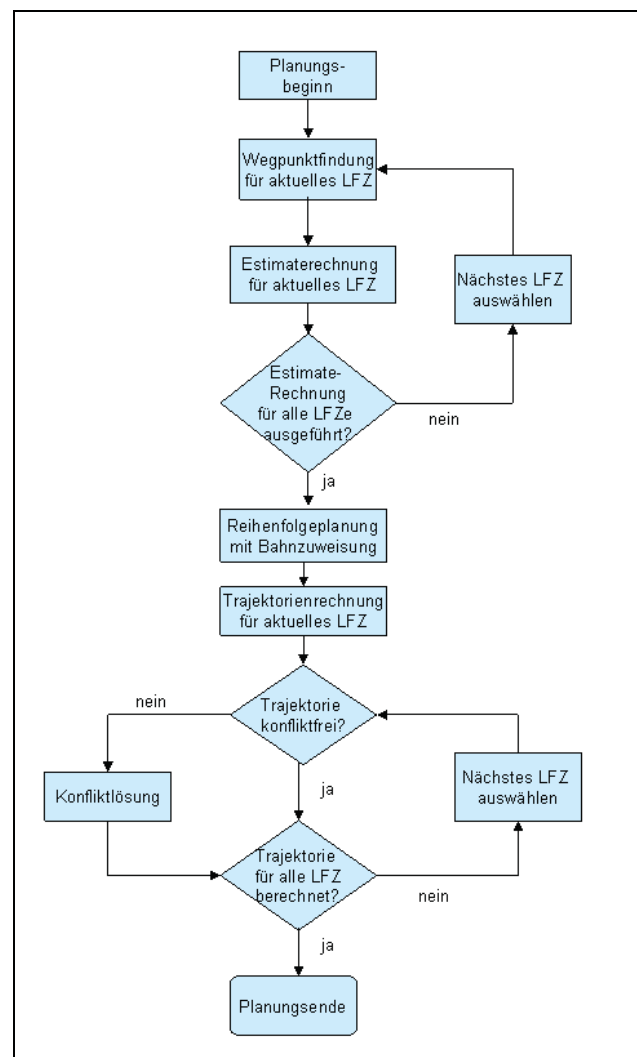


BILD 1. D4-CARMA Planungsablauf

Bei der Sequenzplanung werden verschiedene mögliche Landereihenfolgen bezüglich statischer und dynamischer Bewertungskriterien miteinander verglichen und eine optimale Landesequenz gebildet. Hierfür werden zunächst für jedes Luftfahrzeug einer betrachteten Landereihenfolge eine Landebahnzuordnung und eine Zielzeit (Estimate) ermittelt. Der Vergleich der Bewertungsergebnisse führt dann zur Etablierung einer Sequenz, die hinsichtlich der Kapazitätsauslastung des Flughafens optimiert ist.

Die Komponente Landebahnzuweisung wird von der Sequenzplanung verwendet. Als Eingabe dient die vorher bestimmte Landesequenz und die Aufgabe des Moduls ist, den Flugzeugen der Sequenz eine Landebahn und einen exakten Landezeitpunkt zuzuweisen. Diese Zuweisungen sind insbesondere abhängig von den zuvor berechneten Ankunftszeitintervallen und der gegebenen Landebahnkonfiguration. In die Berechnungen werden weitere Faktoren, wie z.B. die einzuhaltenden Separationen zwischen zwei aufeinander folgenden Flugzeugen und die geplanten Landebahnen der voraus fliegenden, oder bereits gelandeten Flugzeuge, mit einbezogen. Wird eine Landebahn auch für Abflüge eingesetzt, können sich Landebahnblockierungen ergeben, die in Form von anflugfreien Zeitintervallen (AFI) in die Berechnung eingehen. Eine effiziente Auslastung der Landebahnpazität wird über die Zuweisung einer Landebahn nach verschiedenen heuristischen Strategien realisiert.

Die Berechnung einer 4D-Anflugtrajektorie beinhaltet die Bestimmung eines zeitabhängigen horizontalen und vertikalen Flugprofils mit Angabe einzuhaltender Geschwindigkeiten für jedes Flugzeug. Jedes Flugzeug erhält damit genaue Anweisungen, wie von der aktuellen Position zur Landebahnschwelle geflogen werden sollte, um die von Reihenfolgeplanung und Bahnzuweisung bestimmte Landezeit einzuhalten.

4. DOMÄNENSPEZIFISCHE MODELLIERUNG

Als Domäne bezeichnet man ein Wissensgebiet, welches durch eine Menge von Konzepten und eine eigene Terminologie charakterisiert werden kann. Fachleute einer Domäne teilen diese Terminologie und das Wissen über ihre Konzepte [3]. Naheliegender Versuch, diese in der Softwareentwicklung nutzbar zu machen. Der Ansatz domänenspezifischer Modellierungssprachen ermöglicht dies über eine Kapselung des Domänenwissens als Sprachbestandteil innerhalb der DSL.

Eine Annäherung von Code und Problem-domäne wird durch eine domänenspezifische Modellierung (DSM) ermöglicht. Es erfolgt eine Abhebung des Abstraktionsniveaus über das Niveau der Programmierung hinaus, indem eine Lösung durch die unmittelbare Nutzung von Domänenkonzepten spezifiziert wird [2]. Die Modellierungssprache repräsentiert

neben den Konzepten auch die Semantik der betrachteten Domäne und verwendet die in der Domäne etablierte Notation. Es wird eine hohe Problemspezifität und ein hohes Abstraktionslevel erreicht, welches dem Domänenexperten eine intuitive Erstellung von Modellen ermöglicht. Die erstellten Modelle unterliegen den festgelegten Regeln der Domäne und stellen so bereits in einem frühen Entwicklungsstadium die semantische Gültigkeit des Ergebnisses einer Codegenerierung sicher.

Die Entwicklung einer domänenspezifischen Sprache beinhaltet vier wesentliche Aspekte:

- die Domänenkonzepte
- die zur Darstellung der graphischen Elemente verwendete Notation
- die Gültigkeitsbedingungen zur Steuerung des Modellierungsprozesses
- und die Abbildung der Modellkonstrukte in der Codegenerierung.

Die Domänenkonzepte und die Notation einer Modellierungssprache lassen sich direkt aus dem Fachwissen der Domäne ableiten. Sie richten sich in erster Linie nach der in der Domäne verwendeten Terminologie. Beispielsweise werden in der Domäne der Anflugplanung bekannte Konstrukte wie Luftraum, Flughafen und Landebahn verwendet.

Neben den Domänenkonzepten und der verwendeten Notation wird ebenfalls das Domänenwissen über Semantik und Relationen der einzelnen Konzepte benötigt. Die Gültigkeitsbedingungen für ein Modell stehen in einem engen Zusammenhang zu den Beziehungen zwischen den Konzepten und den angewendeten Regeln der Domäne.

Der Codegenerator spezifiziert, wie Informationen aus dem Modell entnommen und in Quellcode transformiert werden. Konzepte, Semantik und Gültigkeitsbedingungen der Modellierungssprache bestimmen hierbei den Prozess der Codeerstellung. Im einfachsten Fall kann jedem Modellierungselement ein bestimmtes, festgelegtes Codefragment zugeordnet werden. Die Attributwerte des Elements können dem Fragment als Parameter übergeben werden. In komplexeren Zusammenhängen kann auch verschiedener Code in Abhängigkeit zu Attributwerten, Beziehungen zwischen Elementen oder anderer modellinterner Informationen generiert werden [2].

Diese Art der Zusammenführung von Problem-domäne und Codierung ermöglicht eine Automatisierung in der Softwareentwicklung, die darin begründet ist, dass sowohl die Sprache als auch die Generatoren, die Anforderungen von lediglich einem Unternehmen und einer Anwendungsdomäne erfüllen müssen.

5. MODELLIERUNGSSPRACHE FÜR EIN LANDEBAHNZUWEISUNGSSYSTEM

Die Entwicklung von AMAN-ML folgt der Zielsetzung, eine Modellierungssprache zur Konfiguration und Implementierung eines generischen Anflugplanungssystems zu erstellen. Die Komponenten, die durch AMAN-ML abgebildet werden, sind u.a.

- flughafenspezifisch,
- unterliegen den jeweiligen rechtlichen Bedingungen,
- unterliegen den Präferenzen der Anwender oder
- sind durch Heuristiken bzw. Erfahrungswissen beschreibbar.

AMAN-ML ist demnach eine domänenspezifische Modellierungssprache zur Abbildung der oben genannten Komponenten, die in dieser Arbeit exemplarisch für ein Landebahnzuweisungssystem (Runway Operations Planning System – ROPS) vorgestellt wird.

Die entwickelte Modellierungssprache AMAN-ML besteht aus mehreren hierarchisch miteinander verbundenen Komponenten. Jede dieser Komponenten folgt ihrer eigenen Syntax und Semantik und lässt sich graphisch als *AirportFamily*, *RunwayConfiguration* und *AssignmentStrategy* modellieren.

Auf der höchsten Ebene befindet sich der Graph *AirportFamily*. Er enthält drei Gruppen von Objekten: Flughafenmodelle, Landebahnkonfigurationen und Zuweisungsstrategien. Im Modellierungsprozess wird zunächst ein Flughafenmodell erstellt. Dieses beinhaltet die für ein zu erstellendes Landebahnzuweisungssystem notwendigen Basisdaten eines Flughafens, zu denen u.a. IATA-Code, Referenzkoordinaten und Luftraumstruktur zählen. Des Weiteren enthält ein Flughafenmodell eine Zuordnung zu jeweils einem Element aus den Gruppen Zuweisungsstrategien und Landebahnkonfigurationen. Die Elemente der Gruppe Landebahnkonfigurationen sind per Dekomposition mit hierarchisch darunter liegenden Graphen vom Typ *RunwayConfiguration* verbunden, so dass ein Flughafenmodell direkt mit dem ihm zugehörigen Layout der Landebahnen verknüpft werden kann. Die Elemente der Gruppe Zuweisungsstrategien stehen für eigenständige Verteilungsheuristiken mit denen ankommende Flugzeuge auf verfügbare Landebahnen verteilt werden können. Auch hier können die Elemente per Dekomposition in einem eigenständigen Graphen vom Typ *AssignmentStrategy* weiter spezifiziert werden. Dies ermöglicht die Verwendung weiterer flughafenspezifischer Details bei der Codegenerierung, während im Fall einer fehlenden Dekomposition von Standarddaten ausgegangen wird. Ein mögliches Modell für die *AirportFamily* ist in BILD 2 dargestellt.

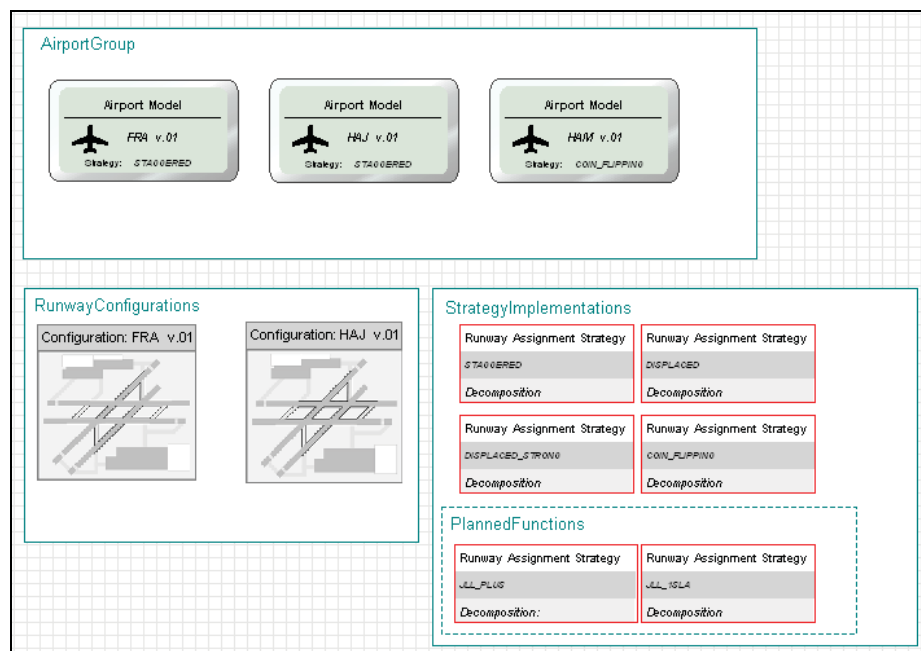


BILD 2. Modellierungsbeispiel für die Komponente *AirportFamily*

Der Objekttyp *RunwayConfiguration* bezieht sich auf eine hierarchisch per Dekomposition mit dem Flughafenmodell verbundene eigenständige Komponente. Die Landebahnkonfiguration beinhaltet die Layoutstruktur des Flughafens. Als Elemente sind insbesondere die Landebahnschwellen enthalten. Sie enthalten als Attribute Schwellenbezeichnung, Ausrichtung, Koordinaten, Verwendungsart und weitere, für spätere Berechnungen notwendige Daten. Die modellierten Schwellen können über verschiedene Beziehungsarten miteinander verbunden werden, um den Informationsgrad für den Modellierer und folglich für den Codegenerator zu erhöhen. Zwei gegenüberliegende Schwellen einer Landebahn können über die Beziehung *OnePhysicalRunway* mit der Kategorie *OppositeOperationalDirections* gekennzeichnet werden. Die Verwendung dieser Beziehung in der Kategorie *DualThreshold* ermöglicht die Erstellung einer versetzten Schwelle zu einer regulären Landebahnschwelle. Des Weiteren können Landebahnkreuzungen mit Angabe von Schnittpunktkoordinaten und Abhängigkeitsgrade zwischen parallel verlaufenden Landebahnen über Beziehungen dargestellt werden.

BILD 3 zeigt ein Modell der *RunwayConfiguration* für den Flughafen Frankfurt (Main). Abgebildet werden in diesem Modell neben der Struktur auch die Abhängigkeitsbeziehungen zwischen den Landebahnen. Bei parallelen bzw. im V-Layout verlaufenden Landebahnen wird in Abhängigkeit ihrer Distanz eine Beziehung vom Typ *Dependency* mit Angabe des jeweiligen Abstandes modelliert.

Eine weitere Komponente des Flughafenmodells, die *AssignmentStrategy*, dient der Erstellung von Strategien für die Landebahnzuweisung. Bekannte heuristische Verfahren wie bspw. Staggered, Displaced, CoinFlipping, TypeSplitting und Varianten des JoinTheLeastLoad können für ein Flughafenmodell ausgewählt werden.

- Staggered
Streng alternierende Zuweisung zwischen den verfügbaren Landebahnen. Bei einem Zweibahnensystem werden LFZ in einer Sequenz mit Nummer 1,3,5... Landebahn A und LFZ mit Nummer 2,4,6... Landebahn B zugeteilt.

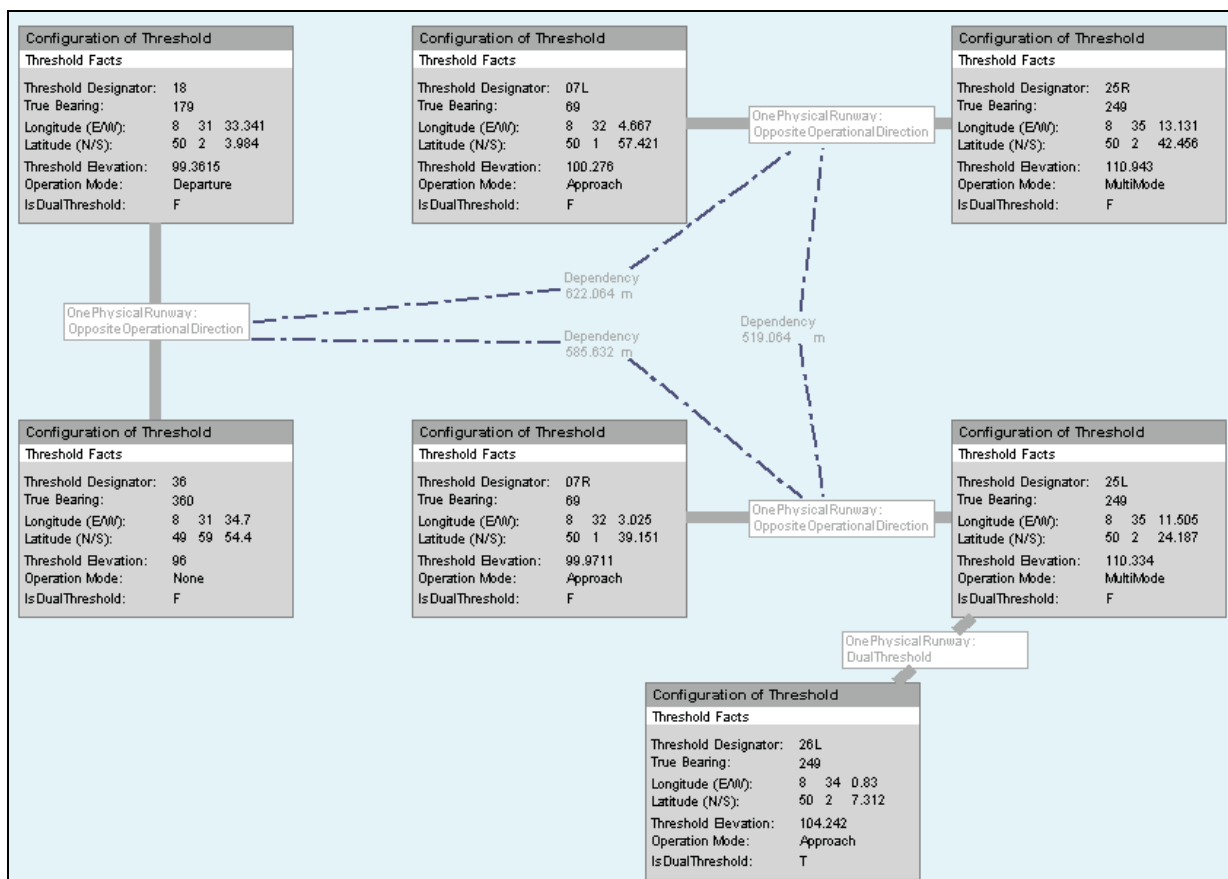


BILD 3. Modellierungsbeispiel für die Komponente RunwayConfiguration

- Displaced
Zuweisung unter Verwendung einer oder mehrerer versetzter Schwellen unter Beachtung von Wirbelschleppenkategorien und Landebahnzuweisung von aktuellem und voraus fliegendem Flugzeug.
- CoinFlipping
Landebahnzuweisung erfolgt gemäß einer Zufallszahlengeneration. Den Flugzeugen wird abhängig von der Anzahl der verfügbaren Landebahnen nach gleicher Wahrscheinlichkeit eine Bahn zugeteilt.
- TypeSplitting
Die Zuweisung erfolgt allein abhängig von der Gewichtsklasse des betrachteten Flugzeuges. Für jeden Flugzeugtyp erhält man eine Wahrscheinlichkeitsverteilung für die einzelnen Landebahnen.
- Varianten des JoinTheLeastLoad
Bei dieser Strategie wird der aktuelle Zustand jeder verfügbaren Landebahn betrachtet und daraus eine Arbeitslast berechnet. Der Zustand einer Landebahn besteht aus der verbleibenden Wartezeit, berechnet aus der Summe der Separationen der für diese Bahn geplanten Flugzeuge, und der Gewichtsklasse des LFZ am Ende der Warteschlange. Die Landebahn mit der geringsten Arbeitslast wird dem nächsten LFZ zugewiesen. Zusätzlich können bei Variationen dieser Strategie die Gewichtsklassen des aktuellen und des nachfolgenden LFZ in die Betrachtungen einbezogen werden [4].

Die beschriebenen Strategien können über eine per Dekomposition verbundene *AssignmentStrategy* an die individuellen Gegebenheiten und Richtlinien des Flughafens angepasst werden und ermöglichen demnach eine optimierte Landereihenfolge.

Der Modellierungsprozess wird durch ein implementiertes Model-Checking unterstützt. Für jeden Graph erfolgt eine grundlegende Überprüfung auf Modellebene. Das Modell wird auf seine Richtigkeit bezüglich der Spezifikationen im Metamodell überprüft. Hierzu gehören Datentypverletzungen und die Gültigkeit von Beziehungen zwischen Elementen hinsichtlich erlaubter Beziehungstypen und erforderlicher Kardinalität. Für die *RunwayConfiguration* erfolgt darüber hinaus eine Modellanalyse. Daten werden aus dem Modell ausgelesen und einer externen Model-Checking Applikation übergeben. Anhand der Modelldaten wird für die Landebahnkonfiguration zunächst die Distanz zwischen den einzelnen Landebahnen berechnet und das Abhängigkeitsverhältnis (Parallel, Near-Parallel, V-Layout oder Crossing) gemäß der rechtlichen Bestimmungen des ICAO Annex 14 (Aerodrome Design and Operations) be-

stimmt. Weitere Berechnungen erfolgen u.a. zum Winkelverhältnis und dem Schwellenversatz der Landebahnen [5]. Im Fall einer Landebahnkreuzung wird der Schnittpunkt (WGS 84 Koordinaten) der Landebahnen berechnet. Die Berechnungsergebnisse stehen dann modellintern zur Verfügung und werden darüber hinaus für die Codegenerierung verwendet.

Ein vollständiges Flughafenmodell mit erstelltem Landebahnlayout und einer ausgewählten Strategie kann dann über einen, der DSL angepassten Codegenerator, in lauffähigen Code übersetzt werden. Bei der Codegenerierung werden die generischen Elemente des Codes mit der aktuellen Situation im Modell verglichen und spezifischer Code für den jeweiligen Flughafen generiert. Der Prozess der Codegenerierung wird durch Attributwerte der Elemente (Objekte und Graphen), Beziehungen zwischen den Elementen und weiterer modellinterner Daten bestimmt. Die durch AMAN-ML erstellten Codedateien sind durch einen automatisch generierten Header gekennzeichnet. Dieser enthält die Angaben über die verwendete Version der AMAN-ML, Namen der verwendeten Konfiguration und die eingesetzte Strategie.

6. ZUSAMMENFASSUNG UND AUSBLICK

In dieser Arbeit wurde die domänenspezifische Modellierungssprache AMAN-ML vorgestellt. Die Motivation zur Erstellung von AMAN-ML liegt in der Möglichkeit, ein generisches Anflugplanungssystem zu entwickeln, welches mit geringem Aufwand an neue Flughäfen und deren spezifische Randbedingungen angepasst werden kann. Neben der Zeitersparnis, die die Anwendung von AMAN-ML ermöglicht, liegt der Nutzen der Applikation in der Möglichkeit, das vorhandene Flughafenlayout bezüglich verschiedener Parameter zu analysieren und die Ergebnisse direkt für die Codegenerierung zu verwenden. In Kapitel 3 wurde die Landebahnzuweisung als Komponente des Anflugplanungssystems 4D-CARMA dargestellt. Die Landebahnzuweisung (ROPS) ist hier als Aufgabe in die Sequenzplanung integriert und für die Zuweisung einer verfügbaren Landebahn und einer exakten Zielzeit zuständig. Diese Berechnungen sind von zahlreichen flughafenspezifischen, rechtlichen oder durch Erfahrungswissen bestimmten Randbedingungen abhängig. Die Entwicklung einer domänenspezifischen Modellierungssprache ermöglicht deren Darstellung. Die Grundlagen des DSM wurden in Kapitel 4 erläutert, in dem auf die notwendigen Aspekte einer Modellierungssprachenentwicklung kurz eingegangen wurde. AMAN-ML wurde entwickelt, um ein generisches Anflugplanungssystem zu konfigurieren und implementieren. Der Aufbau der Modellierungssprache wurde in Kapitel 6 beschrieben. Die einzelnen Graphen der Sprache sind hierarchisch miteinander verbunden, wobei auf höchster Ebene ein Flughafenmodell erstellt

wird, welches neben den Basisdaten des Flughafens per Dekomposition eine Landebahnkonfiguration und eine Zuweisungsstrategie enthält. Während in der Landebahnkonfiguration das Layout des Flughafens abgebildet wird, können über die Zuweisungsstrategien heuristische Verfahren zur Verteilung der Flugzeuge auf die verfügbaren Landebahnen bestimmt werden.

Exemplarische Codegenerierungen für verschiedene Flughäfen haben eine große Zeitersparnis durch die Verwendung von AMAN-ML bestätigt. Die Modellerstellung ist intuitiv gestaltet, da sie die Terminologie der Domäne verwendet und die Zusammenhänge eng an das Domänenwissen angelehnt sind. Anwendern mit geringen System- oder Codierungskenntnissen wird eine Codeerstellung durch die Abstraktion von der Programmebene erleichtert. Ein weiterer Nutzenaspekt von AMAN-ML liegt in der Trennung flughafenspezifischer Daten von der Programmebene, was zu einer Visualisierung der domänenspezifischen Zusammenhänge führt und darüber hinaus eine Analyse des Flughafensystems anhand neuer Parameter und Konzepte vereinfacht.

Nachdem für einen Flughafen das Landebahnlayout modelliert wurde, können anhand veränderlicher Modellparameter verschiedene Zuweisungsstrategien für einen Flughafen hinsichtlich einer optimalen Kapazitätsauslastung miteinander verglichen werden. Die Modellanalyse des Landebahnsystems erlaubt zusätzliche Aussagen über die Wirkungsweise von verschiedensten Layoutparametern auf die Kapazität des betrachteten Flughafens.

Erweiterungen der Modellierungssprache AMAN-ML sind im Bereich umfangreicherer Model-Checking Mechanismen geplant. Der Modellierer soll dann direkt im Modellierungsprozess unterstützt werden.

7. ABKÜRZUNGEN

4D-CARMA	4-Dimensional Cooperative Arrival Manager
AIP	Aeronautical Information Publication
AMAN	Arrival Manager
AMAN-ML	Arrival Management Modeling Language
ATM	Air Traffic Management
BADA	Base of Aircraft Data
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.

DSM	Domain Specific Modeling
DSL	Domain Specific Modeling Language
IATA	International Air Transport Association
ICAO	International Civil Aviation Organization
LFZ	Luftfahrzeug
ROPS	Runway Operations Planning System
TMA	Terminal Maneuvering Area

TAB 1. Abkürzungen

8. REFERENZEN

- [1] R. Hann, L. Christoffels, K. Muth, M.-M. Temme, M. Uebbing-Rumke, Zeitbasiertes Anflugmanagement mit 4D-CARMA zur Unterstützung von Dual Threshold Operations, Deutscher Luft- und Raumfahrtkongress 2008, Darmstadt, Deutschland, 23. – 25. September
- [2] J.-P. Tolvanen, Domänenspezifische Modellierung für vollständige Codegenerierung, JavaSpektrum 01/2006, S. 9 -12, SIGS-DATACOM
- [3] E. Evans: Domain-Driven Design. Tackling Complexity in the Heart of Software, Addison- Wesley, Pearson Education 2004
- [4] N. Bäuerle , O. Engelhardt-Funke, M. Kolonko, On the Waiting Time of Arriving Aircrafts and the Capacity of Airports with One or Two Runways, European Journal of Operational Research, 2007, Vol. 177, S. 1180-1196
- [5] ICAO, Annex 14, Volume I to the Convention on International Civil Aviation: Aerodrome Design and Operations, International Standards and Recommended Practices, International Civil Aviation Organization (ICAO), 2004