Constraint Generierung für domänenspezifische Modellierungssprachen

Dagi Geister

Institut für Systems Engineering, Leibniz Universität Hannover
Institut für Flugführung, Deutsches Zentrum für Luft- und Raumfahrt e.V.
dagi.geister@dlr.de

Abstract: Anflugplanungssysteme dienen der Unterstützung von Fluglotsen bei der Planung und Koordinierung des anfliegenden Luftverkehrs innerhalb eines Flughafennahbereichs. In dieser Arbeit wurde die domänenspezifische Modellierungssprache AMAN-ML (Arrival Management Modeling Language) entwickelt, mit der ein Anflugplanungssystem mit geringem Aufwand an veränderte Anforderungen angepasst werden kann. Auf der Grundlage von AMAN-ML wurde ein Ansatz verfasst, um das erfahrungsbasierte Wissen von Fluglotsen in den Modellierungsprozess einer Software zu integrieren. Eine im Rahmen dieser Arbeit formulierte Grammatik für natürliche Sprache dient als Basis für eine Transformation in die Constraint-Beschreibungssprache OCL (Object Constraint Language). Aspekte der Vagheit und Unbestimmtheit der natürlichen Sprache werden bei der Transformation erkannt und entsprechend bei der Generierung von Constraints an die Modellierungssprache berücksichtigt. Zur Abbildung von linguistischen Variablen und unscharfen Ausdrücken wurde das Konzept der Fuzzy Mengen herangezogen und in die Syntax der OCL integriert. Das Constraint System von AMAN-ML wird für die Verifikation (*Model Verification*) und die Vervollständigung (*Model Completion*) von Modellen eingesetzt.

1 Einführung

Ein Anflugplanungssystem wird für die Anflugkontrolle (engl. Approach Control) eines Flughafens eingesetzt und dient der Unterstützung der Flugverkehrskontrolle bei der sicheren, effizienten und kapazitätsorientierten Führung des Luftverkehrs. An ein Anflugplanungssystem werden hierbei hohe Anforderungen hinsichtlich einer korrekten und stets zuverlässigen Funktionalität gestellt. Diese Anforderungen basieren u.a. auf rechtlichen Vorschriften aus dem Geltungsbereich der Luftfahrt (u.a. LuftVO, ICAO-Dokumente), spezifischen Vorgaben und Informationen für einen Flughafen (Luftfahrthandbuch, Nachrichten für Luftfahrer) und den geltenden Verfahrensweisen der Flugverkehrskontrolle (BA-FVD). Die Umsetzung dieser Anforderungen in einem Anflugplanungssystem stellt sicher, dass einem Fluglotsen die für eine Planung und Koordinierung des Luftverkehrs erforderlichen Informationen und Unterstützungsfunktionen zur Verfügung gestellt werden können. Zudem sollte ein Anflugplanungssystem ohne erheblichen Aufwand an neue oder auch veränderte Rahmenbedingungen anpassbar sein. Dies bezieht sich zum einen auf die Anpassbarkeit von existierenden Funktionalitäten an neue Flughafentopologien und damit

der Übertragbarkeit eines Anflugplanungssystems auf weitere Flughafen, zum anderen auf die Erweiterbarkeit vorhandener Unterstützungsfunktionen hinsichtlich neu entstehender oder veränderlicher Aufgabenstellungen in der Flugverkehrskontrolle. Weiterhin ist eine Adaptierbarkeit des Anflugplanungssystems in Bezug auf individuelle Präferenzen und heuristische Verfahrensweisen der Fluglotsen von ebenso hoher Bedeutung. Diese Adaptierbarkeit stellt nicht nur eine Möglichkeit der Annäherung der Funktionalitäten des Anflugplanungssystems an die realen Handlungsweisen und Entscheidungen des Fluglotsen dar, sondern bildet bei diesen ebenfalls die Grundlage für die Bildung von Vertrauen und Akzeptanz.

Die Anpassung der Software eines Anflugplanungssystems an flughafenspezifische Vorgaben, geltende gesetzliche Vorschriften und mögliche Heuristiken von Fluglotsen ist bisher mit einem erheblichen Arbeitsaufwand verbunden. Umfangreiche Reengineering-Maßnahmen müssen vorgenommen werden, um problemspezifische Codekomponenten zu extrahieren und diese an neue Anforderungen anzupassen. Die hierbei geforderte Korrektheit muss über eine Vielzahl von generellen und flughafenspezifischen Vorgaben sichergestellt werden. Dies erschwert die Entwicklung flexibler und anpassungsfähiger Anflugplanungssysteme. Zum Zeitpunkt dieser Arbeit sind daher die meisten Anflugplanungssysteme auf die spezifischen Anforderungen eines einzelnen Flughafens ausgerichtet. In dieser Arbeit wird ein Ansatz vorgestellt, der eine flexible Anpassung von Anflugplanungssystemen erlaubt und zugleich eine Integration von Erfahrungswissen in den Entwicklungsprozess ermöglichen soll.

2 Motivation und Hintergrund

Als eine elementare Grundlage für die Entwicklung eines anforderungsgerechten Anflugplanungssystems gilt das formale und erfahrungsbasierte Wissen der Fluglotsen. Dieses Wissen ermöglicht ein umfassendes Verständnis über die Zusammenhänge und Abläufe in der Flugverkehrskontrolle. Das Erfahrungswissen wird meist in erzählender Form unter Verwendung von natürlicher Sprache weitergegeben. Natürliche Sprache hat die Eigenschaften von Vagheit und Unbestimmtheit, die sich lediglich über den situativen Kontext verstehen und in ihrer Bedeutung vervollständigen lassen. Dennoch ist Erfahrungswissen von elementarer Bedeutung, um das situative Verhalten eines Fluglotsen in einem Anflugplanungssystem bestmöglich nachbilden zu können. Bei der Konzeption eines Anflugplanungssystems ist es demzufolge nicht nur erforderlich eine Anpassungsfähigkeit an flughafenspezifische Vorgaben und gesetzliche Vorschriften vorzusehen, sondern ebenso wichtig das Erfahrungswissen von Fluglotsen für den Entwicklungsprozess und die entstehenden Unterstützungsfunktionen zu erfassen und zu formalisieren. Mögliche individuell formulierte Verfahrensweisen oder Heuristiken ermöglichen eine bessere Vorhersagbarkeit und Abbildbarkeit von situativ bedingten Entscheidungen eines Fluglotsen. Zudem kann das Erfahrungswissen auch als Anforderung an ein Anflugplanungssystem interpretiert und als Grundlage zur Bestimmung der Korrektheit des Systems verwendet werden. Dies kann zu einer höheren Akzeptanz der Fluglotsen beitragen, die über die Formulierung von erfahrungsbasierten Verfahrensweisen und Informationen ein Anflugplanungssystem ihren

Bedürfnissen anpassen und direkt Auswirkungen in Form adaptierter Planungen nach ihren eigenen Vorgaben beobachten können.

In dieser Arbeit wird insgesamt ein Ansatz zur Integration von Erfahrungswissen in den Modellierungsprozess und in die Komponenten eines Softwaresystems vorgestellt. Dieser Ansatz basiert auf der Zielsetzung

- eine funktionale und strukturelle Anpassbarkeit eines Softwaresystems (hier eines Anflugplanungssystems),
- eine Generierung von Constraints aus Erfahrungswissen,
- eine Formulierung von Constraints in natürlicher Sprache sowie
- eine Verwendung der generierten Constraints zur Verifikation und Vervollständigung von Modellen

zu ermöglichen. Hierfür wurden in dieser Arbeit verschiedene Konzepte erarbeitet, zu denen

- die Entwicklung einer Modellierungssprache zur Anpassung eines Anflugplanungssystems nach flughafen- und anwenderspezifischen Anforderungen,
- die Erweiterung bestehender Warteschlangenmodelle für eine generische Landebahnzuweisung,
- die Ableitung einer Grammatik für natürliche Sprache zur Abbildung von natürlichsprachigen Anforderungen an eine Modellierungssprache,
- die Erweiterung der Syntax der OCL zur Abbildung von unscharfen Ausdrücken der natürlichen Sprache sowie
- die Ausarbeitung eines Constraint Reasoning auf Basis unscharfer Ausdrücke der OCL zur Integration einer Modellverifikation und -vervollständigung

gehören. Diese einzelnen Konzepte wurden einerseits in einer Modellierungssprache und andererseits in einer Komponente zur Constraint Generierung umgesetzt und in ein bestehendes Anflugplanungssystem integriert. Das entstandene Gesamtsystem wurde auf Basis verschiedener Experimente validiert und wird derzeit operational im Deutschen Zentrum für Luft-und Raumfahrt eingesetzt.

3 Die Modellierungssprache AMAN-ML

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 Vorgaben und Randbedingungen angepasst werden kann. Darüber hinaus können neue Aufgabenstellungen in der Flugverkehrskontrolle in einem Modell

dargestellt und in die Unterstützungsfunktionalitäten des Planungssystems integriert werden. Neben der Zeitersparnis, die die Anwendung von AMAN-ML ermöglicht, liegt ein weiterer Vorteil der Modellierungssprache in der Visualisierung von domänenspezifischen Zusammenhängen. Abhängigkeiten und Beziehungen zwischen einzelnen Elementen des realen Systems und die mit ihnen verbunden rechtlichen, flughafentopologischen, verfahrensbezogenen und individuellen Vorgaben können modellbasiert abgebildet und analysiert werden. Die Elemente und Beziehungen eines realen Systems können in einem Modell abgebildet und als Basis für die Generierung von Programmkomponenten herangezogen werden. Die Auswirkung einer Veränderung einzelner Parameter kann innerhalb der Modelle und Programmkomponenten nachvollzogen werden. Die Modellierungssprache AMAN-ML besteht aus einzelnen Komponenten, die eine Abbildung von Flughafenmodellen, Landebahnsystemen, Zuweisungsstrategien, Luftraumstrukturen und heuristischer Verfahrensweisen ermöglichen. Bestehende Warteschlangenmodelle zur effizienten Verteilung von anfliegenden Luftfahrzeugen auf verfügbare Landebahnen wurden hierzu erweitert, um eine Zuweisung auf verschiedene Arten von Landebahnsystemen und die mit ihnen korrespondierenden Separationsvorschriften zu ermöglichen. In Anlehnung an eine optimierte Verteilung, wie in [BEFK06] für ein abhängiges Zweibahnensystem beschrieben, kann beispielsweise eine Zuweisungsstrategie für verschiedene Landebahnensysteme unter Beachtung aktueller Arbeitslasten definiert werden. Gemäß Gleichung 1 wird in der Zuweisungsstrategie δ^{JLL+} das ankommende Luftfahrzeug k der Landebahn r zugewiesen, auf der es die geringste Wartezeit $(u^r + b(i^r, k))$ zu erwarten hat. Hierzu werden die bestehenden Arbeitslasten u^r mit der notwendigen Separation $b(i^r, k)$ zwischen letztem Luftfahrzeug auf der Landebahn r und dem aktuellen Luftfahrzeug k addiert.

$$\delta^{JLL+}(u^r, i^r; k) = \{r \mid (u^r + b(i^r, k)) \leq min\{(u^I + b(i^I, k)), (u^{II} + b(i^{II}, k)), \ldots, (u^N + b(i^N, k))\}\} \quad (1)$$

Die einzelnen Komponenten der AMAN-ML lassen sich individuell an die spezifischen Vorgaben eines Flughafens und die Verfahrensweisen der Fluglotsen anpassen.

4 Constraint Generierung

Das Erfahrungswissen kann bei hier vorgestellten Ansatz zum einen innerhalb einer eigenständigen Komponente von AMAN-ML abgebildet und damit in die Funktionalitäten des Anflugplanungssystems direkt integriert werden, zum anderen kann ein Fluglotse individuelle Anforderungen an ein Anflugplanungssystem in natürlicher Sprache formulieren. Die natürlichsprachigen Formulierungen werden dann analysiert, interpretiert und in eine Constraint-Beschreibungssprache transformiert. Im Rahmen dieser Arbeit wird eine Grammatik zur Beschreibung der englischen Sprache (Standard-Sprache in der Flugverkehrskontrolle) zur Formulierung von Erfahrungswissen definiert, die den Einschränkungen einer kontextfreien Grammatik genügt. Dieser Ansatz wurde gewählt, da die englische Sprache keine kreuzenden Abhängigkeiten besitzt und die strukturellen Abhängigkeiten der Sprache über eine kontextfreie Grammatik ebenfalls dargestellt werden können [Cho57]

[Shi85]. Bei der Entwicklung der Grammatik G_{NL} wurde insbesondere Wert auf die geeignete Abbildung der besonderen Merkmale von natürlichen Sprachen, wie Kontextbindung, Variabilität, Ausdrucksfähigkeit und Vagheit, gelegt. Hierzu wurde basierend auf der Universalgrammatik von N. Chomsky [Cho57] sowie Arbeiten von Y. X. Wang [Wan09], in denen Produktionsregeln zur Definition grundlegender englischsprachiger Satzstrukturen erarbeitet wurden, die Grammatik G_{NL} ausgearbeitet. In der G_{NL} sind dabei gleichwertige Satzaussagen über unterschiedliche Satzarten, alternative Beschreibungsformen, mögliche Synonymbildungen und Formulierungen unter Verwendung unterschiedlicher Zeitformen möglich. Mit der entwickelten Grammatik G_{NL} ist es zudem durch die Definition von linguistischen Variablen möglich, die hohe Variabilität und Ausdrucksfähigkeit der natürlichen Sprache hinreichend genau nachzubilden. Die Formulierungen von Fluglotsen unterliegen somit kaum Einschränkungen in Bezug auf alternative Ausdrucksformen und eine verwendete Terminologie.

$$\Big[\,[From\ my\ experience]_{parenthesis}\,\Big[\Big[\,[two]_{determinative}\ [runways]_{noun}\,\Big]_{noun\_phrase} \tag{1}$$

$$\Big[\,[\,[should]_{auxiliary}\,[\,be\ considered]_{compound}\,]_{verb}\ [dependant]_{adjective}\,\Big]_{verb\_phrase}\Big]_{sentence}$$

$$[if]_{conj}\Big[\Big[\,[their]_{determinative}\ [distance]_{noun}\,\Big]_{noun\_phrase}\Big[\,[is]_{simplePresent}$$

$$\Big[[less\ than]_{comparison}\,[1035\ metres]_{numExpression}\Big]_{adverbial\_phrase}\Big]_{verb\_phrase}\Big]_{sentence}\Big]_{sentence}$$

Das Erfahrungswissen eines Experten und die situative Erläuterung von Zusammenhängen kann nicht immer anhand präziser Angaben formuliert werden. Eine natürlichsprachige Formulierung enthält daher oft unsichere und vage Beschreibungen, die zu einer Unbestimmtheit in der Satzbedeutung führen. Diese können innerhalb eines Satzes insbesondere durch eine Verwendung von linguistischen Werten, Gradpartikeln (z.B. *very, slightly*), Temporal- und Modaladverbien (z.B. *almost, often*) sowie Auxiliarverben (z.B. *could, should*) erkannt werden. Bei der Formulierung von Erfahrungswissen können derartige Unbestimmtheiten aufgrund von ungenauen Angaben, einem indirekten Situationsbezug oder fehlenden Informationen entstehen und müssen bei einer Formalisierung und Transformation in Constraints geeignet abgebildet werden. Die Constraint-Beschreibungssprache OCL wurde aus diesem Grund im Rahmen dieser Arbeit um ein mehrwertiges Konzept erweitert. Zur Abbildung von linguistischen Variablen und unscharfen Ausdrücken wurde das Konzept der Fuzzy Mengen herangezogen und formal in die Syntax der OCL integriert. Die hierbei entwickelte Grammatik $G_{OCLFuzzy}$ ermöglicht die Transformation der einzelnen Constraints in Konstrukte der Programmiersprache C++. Die Formulierung von Anforderungen an ein Anflugplanungssystem setzt eine Identifikation von linguistischen Variablen voraus, mit denen Zusammenhänge in der Anflugplanung beschrieben werden können. Die Elemente und Beziehungen von AMAN-ML werden in dieser Arbeit daher mit diesen linguistischen Variablen und ihren möglichen Werten versehen, welche sie am geeignetsten beschreiben. Den identifizierten linguistischen Variablen werden charakteristische Funktionen zugeordnet, die eine Zugehörigkeit eines gegebenen linguistischen Wertes zu der hierüber definierten Fuzzy Menge festlegen. Die Vagheit in einer

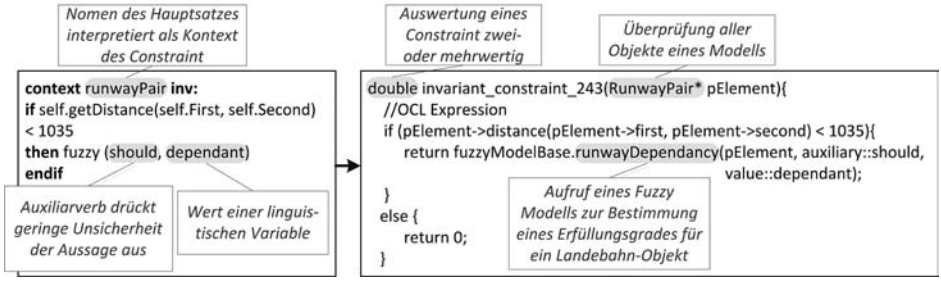

Abbildung 1: Transformation eines Constraints von OCL zu C++

natürlichsprachigen Formulierung kann über diesen Ansatz in eine Relation zu den Elementen und Zusammenhängen eines Modells in AMAN-ML gesetzt, entsprechend interpretiert und in OCL transformiert werden. Die Abbildung 1 enthält eine exemplarische Darstellung der Transformation eines Satzes in natürlicher Sprache (vgl. Satz 1) in eine Constraint-Beschreibungssprache sowie in einem nachfolgenden Schritt in Methoden einer Programmiersprache.

5 Constraint Reasoning

In AMAN-ML werden die definierten Constraints für zwei verschiedene Arten von Aufgaben in der Modellierung eingesetzt. Zum einen können die Constraints verwendet werden, um Anforderungen an die Gültigkeit von definierten Objekten, Beziehungen sowie deren Attribute zu formulieren. Die einzelnen Constraints werden hierbei mit Elementen des Metamodells assoziiert und stellen eine erweiterte Form der Spezifikation der Modelle dar. Zusätzlich lassen sich auf Basis der formulierten Constraints logische Schlussfolgerungen zur Vervollständigung von teilweise erstellten Modellinstanzen ableiten. Die grundlegende Aufgabe einer *Modell-Verifikation* ist die Überprüfung der Korrektheit eines Modells. Diese beinhaltet in AMAN-ML zunächst eine separate Verifikation der einzelnen Constraints, die den Elementen des Modells zugeordnet werden. Da die einzelnen Constraints in AMAN-ML nicht dem zweiwertigen Konzept entsprechen, wird der Auswertung eine Abbildung in eine scharfe Ergebnismenge auf Basis eines definierten Schwellenwertes zugrunde gelegt. Werden bei einer Verifikation fehlerhafte Modellelemente erkannt, ist es zudem von Bedeutung, den Ursprungsfehler sowie das verursachende Modellelement zu identifizieren. In den Modellen von AMAN-ML bestehen Hierarchien in der Fehlerfortsetzung, die als Basis zur Definition von Constraint Hierarchien gemäß [BDFBK87] herangezogen wurden. Die Verwendung von Constraint Hierarchien ermöglicht eine Analyse von möglichen Zusammenhängen in den Fehlern einer Modellinstanz und damit die Identifikation eines verursachenden Modellelements. Die *Modell-Verifikation* sollte weiterhin Methoden beinhalten, die eine Erkennung widersprüchlicher Constraints und damit

unbestimmter Constraint Systeme ermöglicht. Werden zu viele Constraints für ein Modell definiert bzw. enthalten die Constraints zu große Einschränkungen gültiger Werte, ist es möglich, dass keine gültige Lösung in Form einer Modellinstanz bestimmt werden kann. Bei der Formulierung von Constraints für eine Modellierungssprache ist es daher von Bedeutung, Constraints zu identifizieren, deren Anforderungen den Lösungsraum eines Constraint Systems zu stark einschränken. In dieser Arbeit wurde ein Verfeinerungs-Modell verwendet, welches ausgehend von einer gegeben Modellstruktur, erlaubte Werte auf Basis der Constraints sukzessive einschränkt. Enthalten die resultierenden Wertebereiche mehr als einen gültigen Eintrag, können diese verschiedene Wertebelegungen bilden und damit eine Menge an gültigen Modellinstanzen definieren.

Das Constraint System von AMAN-ML kann zudem eingesetzt werden, um teilweise erstellte Modelle zu einer vollständigen und korrekten Modellinstanz zu ergänzen. Auf Basis grundlegender Informationen über das zu erstellende Modell, werden die fehlenden Modellelemente und Beziehungen unter Beachtung der geltenden Anforderungen identifiziert und dem Modell hinzugefügt. Der Zeitaufwand für die Erstellung einer Modellinstanz wird deutlich reduziert und die resultierende Modellösung entspricht den geltenden Anforderungen. In dieser Arbeit wurden für die *Modell-Vervollständigung* zunächst einzelne Modellierungsschritte für die Komponenten von AMAN-ML festgelegt, die bei der Erstellung eines Modells regulär durchlaufen werden. Für jeden einzelnen Modellierungsschritt bestehen in AMAN-ML verschiedene Alternativen, die auf Basis der Constraints bewertet werden müssen. Hierzu wurden für jeden Modellierungsschritt geeignete FCSPs definiert, die eine Bewertung alternativer Modellierungsschritte vornehmen und die Bestimmung einer präferierten Lösung unter den alternativen Wertebelegungen ermöglichen.

Definition 1. $\vec{d_A} \geq_{\mathcal{P}}^{lex} \vec{d_A}$ *iff* $List(\vec{d_A}, \mathcal{P}) \geq_{lex} List(\vec{d_B}, \mathcal{P})$

Die Auswahl einer präferierten Lösung eines FCSP wird, wie in Definition 1 angegeben, über einen vollständigen lexikographischen Vergleich der geordneten Listen über die Erfüllungsgrade der einzelnen Constraints, basierend auf [FLS93], vorgenommen.

6 Das Gesamtsystem

Die Generierung von Constraints aus Erfahrungswissen und deren Integration in die Modellierungssprache sind in einen Entwicklungsprozess eingebettet, der sich aus zwei Phasen zusammensetzt. In der ersten Phase erfolgen die Spezifikation und die Modellierung des zu erstellenden Anflugplanungssystems, während der zweiten Phase die Integration der erzeugten Komponenten in das Gesamtsystem und deren Nutzung zugeordnet sind. In der ersten Phase erfolgen diejenigen Schritte des Entwicklungsprozesses, bei denen der Ausführungszeit keine Bedeutung zukommt. Die Generierung von Constraints an die Modellierungssprache aus formalem und erfahrungsbasiertem Wissen, die Erstellung flughafenspezifischer Modelle sowie deren Überprüfung und Vervollständigung gehören folglich in diese Phase. Hingegen gehören die Integration von erzeugten Codekomponenten und die Ausführung des erstellten Anflugplanungssystems der zweiten Phase an. Diese Trennung

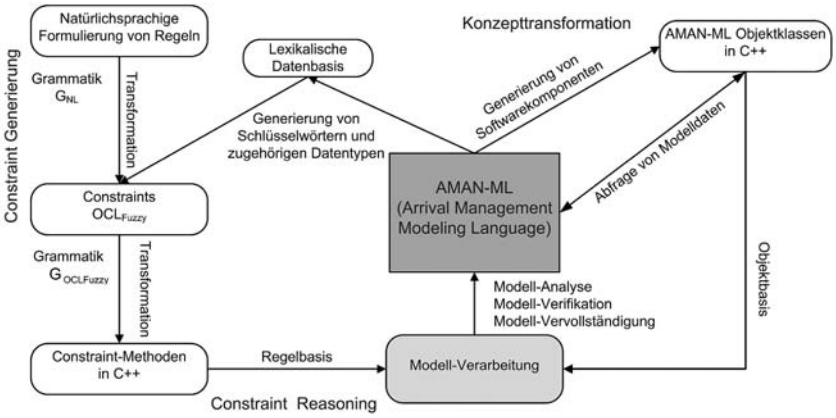

Abbildung 2: Systemkomponenten und Entwicklungsprozess

wurde in dieser Arbeit aufgrund der geforderten Echzeitfähigkeit für Anflugplanungssysteme vorgenommen. Der Entwicklungsprozess zur Generierung eines adaptierbaren Anflugplanungssystems sowie die in dieser Arbeit entwickelten Komponenten als Teil eines Gesamtsystems werden in Abbildung 2 dargestellt. Der Modellierungsprozess beginnt mit der Erstellung verschiedener Modellinstanzen zur Abbildung von Landebahnensystem, Luftraumstruktur, optimierender Zuweisungsstrategie und heuristischer Verfahrensweisen eines zu erstellenden Anflugplanungssystems. Die Grundlage für die Ermöglichung einer verlässlichen Modellverarbeitung und -auswertung bildet dabei die Transformation der Modellelemente und Constraints dieser Modellinstanzen in eine einheitliche Beschreibungsform. Die einzelnen Elemente (Objekte und Beziehungen) einer Modellkomponente werden daher auf Basis eines Transformationskonzepts für domänenspezifische Modellierungssprachen in korrespondierende Klassen einer objektorientierten Programmiersprache übersetzt. Eine Abfrage von konkreten Daten einer Modellinstanz aus AMAN-ML führt zu Objektinstanziierungen, die einer Objektgruppe für einen bestimmten Flughafen zugeordnet werden und damit eine Objektbasis für weitere Modellverarbeitungschritte repräsentieren (vgl. Abbildung 2).

Die Transformation von natürlicher Sprache wird in dieser Arbeit auf Basis einer Grammatik realisiert, deren Vokabular mit Schlüsselworten ergänzt wird, die aus dem Metamodell der Modellierungssprache abgeleitet werden. Mögliche Wortgruppen in einem natürlichsprachigen Satz können hierdurch besser eingegrenzt und erkannt werden. Ein formulierter Satz in natürlicher Sprache wird in dieser Arbeit zunächst in eine Constraint-Beschreibungssprache transformiert (Grammatik G_{NL}). Sprachliche Unbestimmtheiten werden dabei über die Verwendung verschiedener Sprachkonzepte und Worte angedeutet. Diese können wiederum erkannt und verwendet werden, um die Transformation in ein Zielformat entsprechend anzupassen. Die natürlichsprachigen Aussagen werden über die OCL formalisiert, deren standardisierte Syntax in dieser Arbeit um die Möglichkeit einer Verwendung von mehrwertigen Mengen (Fuzzy Sets) erweitert wurde. Vagheiten in einer

Aussage können daher direkt beschrieben und bei einer Verifikation berücksichtigt werden. In einem weiteren Schritt erfolgt eine Transformation der OCL-basierten Anforderungen in Methoden einer objektorientierten Programmiersprache (Grammatik $G_{OCLFuzzy}$). Diese Methoden beziehen sich gemäß der zugrunde liegenden OCL-basierten Anforderung auf eine bestimmte Gruppe von Objekten der Modellierungssprache und werden in einer gemeinsamen Regelbasis hinterlegt.

Die Objektbasis und die Regelbasis gehen in die *Modell-Verarbeitung* ein. Die einzelnen Elemente einer Modellinstanz werden gegen die, über die Regelbasis definierten, Anforderungen verifiziert. Mögliche Fehler in der Modellierung können auf diese Weise identifiziert werden (*Modell-Verifikation*). Darüber hinaus kann ein teilweise erstelltes Modell anhand der Regelbasis über mehrere Schritte zu einem gültigen Modell vervollständigt werden (*Modell-Vervollständigung*). Diese Aufgaben erfordern eine Schlussfolgerung (*Constraint Reasoning*) anhand einer gegebenen Modellinstanz und der darauf anwendbaren Teilmenge des Constraint Systems. Die einzelnen Komponenten zur Generierung von Constraints auf Basis von natürlicher Sprache sowie deren Integration in die Verarbeitung von Modellen werden in Abbildung 2 abgebildet.

7 Fazit und Ausblick

Die einzelnen Komponenten von AMAN-ML wurden von Fluglotsen validiert und in einem iterativen Prozess an die Erfordernisse der Anflugplanung angepasst. Verschiedene Experimente auf dem entstandenen System auf Basis der generierten Programmkomponenten wurden durchgeführt und mit vorab definierten Prognosen über das Systemverhalten verglichen. Die Grammatik wurde gegen verschiedene natürlichsprachige Formulierungen und die dabei erfolgte Erkennung spezifischer Merkmale validiert.

In den durchgeführten Experimenten und Studien konnte eine hohe Akzeptanz der Benutzer hinsichtlich der Verwendung von AMAN-ML festgestellt werden. Die Modellerstellung ist intuitiv gestaltet, da diese die Terminologie der Domäne verwendet und die Strukturen von AMAN-ML eng an die Elemente und Zusammenhänge des realen Systems angelehnt sind. Eine Generierung von Programmkomponenten wird Benutzern mit geringen Kenntnissen in Bezug auf das Anflugplanungssystem durch die Abstraktion von der Programmebene erleichtert. Dies konnte in den Validierungsstudien belegt werden. Die Fluglotsen ordneten die Modellierungssprache als gut verständlich ein und konnten bereits nach kurzer Zeit eigenständig erste Modelle erstellen. Die Methodik zur Integration von Erfahrungswissen in den Modellierungsprozess wurde von den Fluglotsen ebenfalls gut angenommen. Insbesondere die Möglichkeit zur Abbildung von erfahrungsbasierten Verfahrensweisen in einem Modell und deren Integration in die Planungen des Anflugplanungssystems wurde positiv bewertet. Die Modellierungssprache AMAN-ML wird seit 2010 am Deutschen Zentrum für Luft- und Raumfahrt e.V. operational eingesetzt. Dabei haben exemplarische Modellierungen für verschiedene Flughäfen eine hohe Zeitersparnis durch die Verwendung von AMAN-ML bestätigt. Während eine Anpassung des Anflugplanungssystems 4D-CARMA (4-Dimensional Coorperative Arrival Management System) für einen neuen Flughafen sonst mit einem Zeitaufwand von mehreren Monaten

verbunden ist, konnten im Rahmen dieser Arbeit verschiedene flughafenspezifische Modelle mit AMAN-ML bereits in wenigen Stunden erstellt und implementiert werden. Anhand des definierten Constraint Systems konnte zudem die korrekte und stets zuverlässige Funktionalität der generierten Programmkomponenten sichergestellt werden. Während der Experimente zeigten sich die Fluglotsen sehr interessiert an dem vorgelegten System und zeigten eine hohe Bereitschaft an der Mitarbeit zu dessen Verbesserung. Insbesondere die Möglichkeit zur Erfassung von Erfahrungswissen wurde positiv aufgenommen und bewertet. Dabei wurde jedoch von den Fluglotsen oft angemerkt, dass es wünschenswert wäre, neue Anforderungen auch während des operationalen Betriebs formulieren und direkt in die Planungen des Systems integrieren zu können.

Die hier entwickelte Grammatik für die natürliche Sprache wurde im Rahmen eines Projektes zur Spracherkennung an die Phraseologie von Fluglotsen im Sprechfunk angepasst und wird seither zur Erfassung von Anweisungen für eine Erkennung von Lotsenabsichten in der Anflugplanung eingesetzt.

Literatur

[BDFBK87] A. Borning, R. Duisberg, B. Freeman-Benson und A. Kramer. Constraint Hierarchies. In *Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA'87)*, 1987.

[BEFK06] N. Bäuerle, O. Engelhardt-Funke und M. Kolonko. On the Waiting Time of Arriving Aircrafts and the Capacity of Airports with One or Two Runways. *European Journal of Operational Research*, 177 (Issue 2):1180–1196, 2006.

[Cho57] N. Chomsky. *Syntactic Structures*. Mouton de Gruyter, 1. Auflage, 1957.

[FLS93] H. Fargier, J. Lang und T. Schiex. Selecting preferred solutions in Fuzzy Constraint Satisfaction Problems. In *Proceedings of EUFIT'93*, 1993.

[Gei12] D. Geister. *Constraint Generierung für domänenspezifische Modellierungssprachen*. Forschungsbericht 2012-15, Deutsches Zentrum für Luft- und Raumfahrt e.V., 2012.

[Shi85] S. M. Shieber. Evidence against the context-freeness of natural languages. *Linguistics and Philosophy*, 8:333–343, 1985.

[Wan09] Y. X. Wang. A Formal Syntax of Natural Languages and the Deductive Grammar. *Fundamenta Informaticae*, 90(4):353–368, 2009.

Dagi Geister hat ihr Studium der Wirtschaftswissenschaften mit Schwerpunkt Informatik an der Leibniz Universität Hannover im April 2005 abgeschlossen. Ihre Diplomarbeit verfasste sie zum Thema XML-basierte Datenaustauschformate im B2B-Bereich in Kooperation mit der IBM Deutschland. Seit 2007 arbeitet sie am Deutschen Zentrum für Luft- und Raumfahrt e.V. und hat dort ihre Promotionsarbeit über die natürlichsprachige Anpassung von Assistenzsystemen für Fluglotsen angefertigt. Seitdem ist sie u.a. für die Entwicklung neuer Konzepte für die Luftfahrt, insbesondere Unbemannter Luftfahrzeugsysteme (UAS) und deren Integration in den zivilen Luftraum, verantwortlich.