

Beschreibung und Simulation ereignisdiskreter Systeme für Aufgabenstellungen aus dem Bereich des ATM und Airport Managements

P. Förster

Institut für Flugführung, Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)
Lilienthalplatz 7, 38108 Braunschweig, Deutschland

Zusammenfassung

Im Rahmen des Airport Collaborative Decision Managements (ACDM) ist ein umfassendes Situationsbewusstsein des einzelnen Akteurs für einen erfolgreichen Verhandlungsprozess von fundamentaler Bedeutung. Die bereits im Einsatz befindlichen Decision Support Systeme, wie die Anflug- und Abflugstaffelung, unterstützen den jeweiligen Handlungsträger dabei mittels Visualisierung der Daten und Optimierung der Prozessabläufe.

Durch die beim ACDM-Konzept erforderliche Einbindung weiterer Systemkomponenten des Flughafens wie Rollen, De-Icing und Turnaround, und der Bewertung von Auswirkungen verschiedener Handlungsstrategien, gewinnt die Simulation als Methodik an Bedeutung. Neben der beschriebenen „what-if“-Fähigkeit soll zudem ein erweitertes Systemverständnis des Nutzers durch die Möglichkeit zur strukturellen Veränderung des Modells gefördert werden. Die in diesem Dokument vorgestellte Modellierungs- und Simulationsumgebung basiert auf einer ereignisdiskreten Betrachtungsweise. Ziel ist es, unter Berücksichtigung von Expertenwissen eine schnelle Modellentwicklung für verschiedene Fragestellungen aus dem ACDM sowie ATM-Bereich zu ermöglichen und dabei dem Anwender bei der Formulierung eine beliebige Detailtiefe zu erlauben.

1. EINLEITUNG

Aufgrund des bisherigen Wachstums im Luftverkehr kommt einer verbesserten Ausnutzung bestehender Ressourcen bei der Vermeidung von Kapazitätsengpässen eine zentrale Bedeutung zu. In den letzten Jahren wurde dabei die Idee des Collaborative Decision Making (CDM) aus dem Bereich des Flussmanagements (in Anwendung auf das Problem des Ground Holding [1]) auf das Gesamtsystem Flughafen erweitert (Airport Collaborative Decision Management, ACDM [2]).

Ziel ist es mit Hilfe einer verlässlichen und für alle Handlungsträger zugänglichen Datenbasis ein verbessertes Situationsbewusstsein zu erreichen und über eine kooperative Entscheidungsfindung zu einer für alle Beteiligten vorteilhaften Lösung zu gelangen [3].

Hierbei sollen die einzelnen Flughäfen in einem gesamtheitlichen Verkehrsverbund betrachtet werden [4]. Die Vielzahl der beteiligten Akteure, wie bspw. der Flugsicherung, der Flusskontrolle, dem Flughafenbetreiber und den jeweiligen Airlines, führt bei dem System Flughafen durch die häufig konkurrierenden Zielstellungen auf ein hohes Maß an Komplexität.

Die Grundlage einer gemeinschaftlichen Entscheidungsfindung ist ein verbessertes Situationsbewusstsein des Einzelnen [5]. Neben der Visualisierung der Daten spielt dabei die Simulation der jeweiligen Prozesse eine wesentliche Rolle [6].

In diesem Dokument wird eine Modellierungs- und Simulationsumgebung vorgestellt, die bei Anwendung auf die jeweilige Fragestellung das Systemverständnis des Nutzers erhöhen und ihn so während des Entscheidungsprozesses unterstützen soll. Ziel ist ein tieferer Einblick des

Anwenders in die funktionalen Zusammenhänge und die Befähigung zur Integration der verfügbaren Daten. Der im Zusammenhang mit dem ATM bzw. ACDM typische organisatorisch-betriebliche Charakter führt dabei auf eine ereignisdiskrete Betrachtungsweise

2. ANSATZ EINER EREIGNISDISKRETEN MODELLIERUNGSUMGEBUNG ZUR UNTERSTÜTZUNG IM ACDM

Neben analytischen Modellen zur Untersuchung von Fragestellungen aus dem ATM Bereich wurde in den vergangenen Jahren eine Vielzahl von Simulationsumgebungen entwickelt. Diese unterscheiden sich im Grad der Detailliertheit von makro- bis mikroskopisch, dem Abbildungsumfang und einer dynamischen oder stochastischen Betrachtungsweise [7]. Berücksichtigt man eine ereignisdiskrete Arbeitsweise, so sind als wesentliche Programmsysteme zur Modellierung und Simulation von Luftverkehrsströmen *SIMMOD* und *TAAM* anzuführen. Hier bietet sich neben der Modellierung definierter Bereiche der Flughafen- und Luftraumstruktur wie Vorfeld, taxi way oder enroute auch die Möglichkeit zur Abbildung stochastischer Einflüsse [7] [8].

Im Ansatz der hier vorgestellten Modellierungsumgebung soll der vorgegebene Abbildungsumfang aufgebrochen und der Nutzer in die Lage versetzt werden, verschiedene Fragestellungen aus dem Bereich des ATM bzw. ACDM wie Rollverkehr, Piloten-Lotsen-Kommunikation oder dem De-Icing-Prozess unabhängig voneinander sowie in beliebiger Auflösung beschreiben zu können. Die Gestaltungsfreiheit auf struktureller Ebene in Verbindung mit einem freien Detaillierungsgrad soll dem Anwender einen verbesserten Einblick in die funktionalen Zusammenhänge der jeweiligen Problematik ermöglichen.

Aus den oben beschriebenen Eigenschaften und den Anforderungen des Nutzers an eine Modellierungsumgebung im Kontext des ACDM lassen sich Vorgaben an ein geeignetes Werkzeug ableiten. In diesem Zusammenhang soll auf die Bedeutung der Begriffe Modellierung und Simulation im weiteren Verlauf dieses Dokumentes hingewiesen werden. Als Simulation wird hierbei das nach der Systemabstraktion und Modellbildung durchgeführte Experiment und unter (ereignisdiskreter) Modellierung auch die mathematisch-analytische Beschreibung eines ereignisdiskreten Zusammenhanges verstanden. Da jedoch mit Modellierung im Allgemeinen die Modellentwicklung in Verbindung gebracht wird, sei an dieser Stelle zur Vermeidung von Missverständnissen auf die verschiedenartige Begriffsbedeutung hingewiesen.

Die Handhabbarkeit einer Simulationsumgebung trägt im wesentlichen Maße zur Akzeptanz durch den Anwender bei. Vor diesem Hintergrund wurde zur Erleichterung der Bedienung eine intuitive Arbeitsoberfläche erstellt. Neben den obligatorischen Anforderungen an die Umgebung wie Konsistenz, Transparenz, Fehlertoleranz und Flexibilität [9] wurde dabei besonderes Augenmerk auf die Bedeutung domänenspezifischer Funktionsblöcke gelegt. Untersuchungen zeigen, dass Domainexperten bei Vorgabe bestimmter Funktionen und vorhandenem Vertrauen in die Systemstruktur, schneller zu Ergebnissen gelangen als bei der Arbeit mit einer freien generischen Simulationsumgebung [10]. Im Folgenden sollen am Beispiel des Turnaroundprozesses die Eigenschaften der erstellten Simulationsumgebung erläutert werden.

2.1. Beispielanwendung Turnaround

Der Turnaround stellt als Schnittstelle zwischen ein- und ausgehenden Flügen aufgrund vieler nebenläufiger ressourcenverkoppelter und somit konkurrierender Prozesse ein komplexes Teilsystem des Gesamtsystems Flughafen dar. Der Bezug zum ACDM-Gedanken spiegelt sich schwerpunktmäßig in der Tatsache wieder, dass durch eine genauere Schätzung von Off-Block-Zeiten und der Flusskontrolle eine verbesserte Planung zu ermöglichen ist [5]. Eine detaillierte Abbildung der funktionalen Zusammenhänge des Turnaround wurde zudem in Simulationsverbünden nicht betrachtet [11].

Ausgangspunkt einer ereignisdiskreten Untersuchung von Fragestellungen aus dem Bereich des ATM bzw. ACDM bildet die Formulierung mit Hilfe der Elemente ereignisdiskreter Systeme. Im Folgenden sollen kurz die bei der Erarbeitung der Simulationsumgebung zugrunde gelegten Methoden erläutert werden, wobei grundlegendes Wissen über die Funktionsweise von Petrinetzen vorausgesetzt wurde.

Der Turnaround besteht aus einer Reihe serieller und paralleler Prozesse mit einer durch den Flugzeugtyp und die Airline festgelegten Prozessdauer. Die Darstellung der Abhängigkeiten erfolgt in einem Netzmodell. Bei einer Aufspaltung in verschiedene parallel ablaufende Teilprozesse kann der nächste serielle Prozess erst bei Beendigung sämtlicher Vorgängerprozesse aus den Parallelzweigen beginnen.

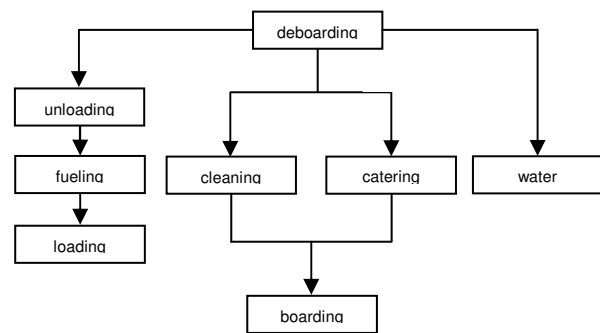


BILD 1 Ausschnitt Turnaround-Prozess

Als eine ingenieurnahe Methodik der vom Nutzer bevorzugten prozessorientierten Sichtweise eignen sich im Gegensatz zu zustandsorientierten Automaten, aufgrund der Möglichkeit zur Abbildung paralleler Prozesses, vor allem Petrinetze [12].

Aus Gründen der Übersicht bei der Darstellung und Beschränkungen in der Rechenkapazität soll der Turnaround-Prozess für alle Flüge an einem Flughafen in einem Modell abgebildet werden. Das für alle Flugzeuge gültige Netzmodell wird durch das „Übereinanderlegen“ der jeweiligen Netzmodelle (Faltung) zusammengefasst [13]. Da die einzelnen Flüge für den Nutzer unterscheidbar sein müssen und der Turnaround-Prozess einer Faltung unterzogen wurde, werden die (einfachen) Stellen- und Transitions-Petrinetze (S/T-Petrinetze) auf farbige Petrinetze erweitert.

Die vom Flugzeugtyp abhängigen jeweiligen Prozessdauern, die dem Verbleiben in einem Zustand entsprechen, werden hierbei noch nicht berücksichtigt und finden durch die Erweiterung auf zeitbewertete Petrinetze ihren Eingang. Dabei können entweder Plätze (verzögerte Aktivierung einer Transition) oder Transitionen mit einer Wartezeit beaufschlagt werden, wobei die Schaltbedingung für letztere neben der logischen Aktivierung durch entsprechende Token in den Vorplätzen durch eine Verzögerung der Transitionen selbst erreicht wird. Die oben beschriebene Struktur des Turnaround-Prozesses zeigt einen deterministischen Charakter, da die Prozessfolge eindeutig festgelegt ist und sich nur durch die Prozessdauer unterscheidet. In der Beschreibung mit Hilfe von Petrinetzen ist so jeder Stelle genau eine Eingangs- und Ausgangstransition zugeordnet. Die entsprechenden Standardsituationen *Aufspaltung* und *Synchronisation* bilden in ihrer Kombination den *Synchronisationsgraphen*, der bei Anbindung an äußere Bedingungen mit einem Eingang versehen werden kann [12].

Bild 2 zeigt den aus Bild 1 gewonnen zeitbewerteten Synchronisationsgraphen mit Eingang und einem ablaufendem *deboarding* Prozess (Markierung des Vorplatzes von t_1). t_1 und t_2 schalten im Gegensatz zu den anderen Transitionen nicht zeitverzögert., wobei t_2 die Synchronisation der beiden parallelen Prozesse *cleaning* und *catering* übernimmt.

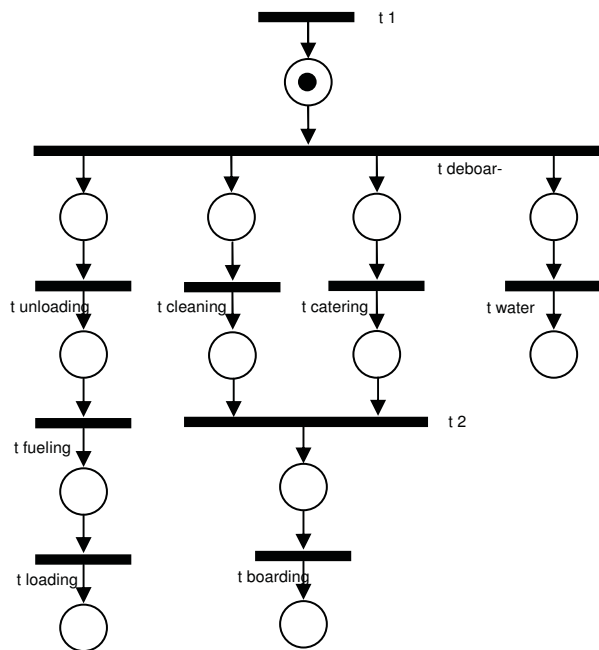


BILD 2 zu Bild 1 entsprechender Synchronisationsgraph

Um eine realistischere Abbildung des Turnaroundprozesses zu ermöglichen, sollen die verschiedenen Einflüsse auf die Dauer der jeweiligen Teilprozesse durch stochastische Verteilungen der Verweilzeiten beschrieben werden. Die entstandenen stochastischen Petrinetze ermöglichen jedoch nur bei exponentieller oder geometrischer Verteilung der Verweilzeiten (Gruppe der Markovschen Petrinetze) eine analytische Auswertung [13] - andernfalls kann mit Hilfe einer Monte-Carlo-Simulation eine Aussage über das Modell bzw. Systemverhalten getroffen werden [14]. Die Hinzunahme der Standardelemente *Auswahl* und *Begegnung* erlaubt die Modellierung von Ausfällen in der Prozesskette bzw. nichtdeterministisches Verhalten in der Ereignisfolge.

3. EIGENSCHAFTEN DER MODELLIERUNGSUMGEBUNG

Um der Zielsetzung der Anwenderfreundlichkeit gerecht zu werden, ist der relativ hohe Abstraktionsgrad bei der Bearbeitung einer Fragestellung mit Hilfe von Petrinetzen zu berücksichtigen. Zudem sind die Vorteile einer analytischen Auswertung Markovscher Petrinetze, wie Invarianten und der Darstellung aller erreichbaren Zustände über den Erreichbarkeitsgraphen und die damit einhergehende Warnung vor Verklemmungen den Vorteilen anderer ereignisdiskreter Modellierungssprachen gegenüberzustellen. Da sicherheitsrelevante Fragestellungen beim Turnaround eine untergeordnete Bedeutung einnehmen, sollen vor allem die Möglichkeit einer einfachen Modellerweiterung bzw. Strukturveränderung und die Erhöhung des Detailgrades bei der Auswahl des Beschreibungsmitels maßgeblich sein.

Kompositionale und hierarchische Modellbildung werden

bei der Beschreibung mit Hilfe von Petrinetzen ebenso unterstützt, wie die Möglichkeit der Simulation und der statistischen Auswertung. Da jedoch in einem weiteren Entwicklungsschritt die einzelnen Prozesszeiten des Turnaround über die jeweilige Arbeitsmenge und Ressourcenverfügbarkeit ermittelt werden sollen, steigt der Abstraktionsgrad bei strukturellen oder quantitativen Manipulationen durch den Anwender in beträchtlichem Maße an.

Die Entwicklung der hier vorgestellten Simulationsumgebung wurde mit Hilfe der ereignisdiskreten Modellierungssprache SIMEVENTS [15] auf der Basis des Programmsystems SIMULINK entwickelt. Diese transaktionsorientierte Sprache stellt zahlreiche Elemente ereignisdiskreter Modellierung wie Warteschlangensysteme (Server, Queue) und hilfreiche statistische Analyseverfahren zur Verfügung. Zudem nutzt sie den Funktionsumfang von MATLAB bzw. SIMULINK. Die transaktionsorientierte Sichtweise kann im Kern dem Funktionsprinzip der S/T Petrinetze in vergleichender Weise gegenübergestellt werden. Dynamische und statische Elemente, so genannte *Transaktionen* bzw. *Blöcke*, bilden die Modellstruktur, in der die Systemfunktionen und -bedingungen eingebunden werden. Durch das Zusammentreffen von Transaktion und Block wird ein Ereignis erzeugt [16]. In SIMEVENTS entsprechen die Transaktionen den so genannten Entities (mit Attributen belegbare, im Modell erzeugte temporäre Objekte), wobei die Blöcke durch bekannte Elemente ereignisdiskreter Sprachen wie Gates, Servern oder Switches vertreten werden [15].

Betrachtet man den Nutzer im Kontext des ACDM, so ergibt sich über den Ansatz eines gemeinsamen Operationszentrums, in dem die beteiligten Akteure die Lösungsverhandlung durchführen, die Forderung zur Ausführung des „what-if“-Probings, also der Möglichkeit zur Bewertung verschiedener möglicher Handlungsszenarien. Im Zusammenhang mit der Integration des Turnaround-Prozesses in einen Simulationsverbund, welche neben An- und Abflugstaffelung noch den Flug- wie Rollverkehr simuliert [17], muss die Simulationsumgebung somit in der Lage sein, parallel zur eigentlichen Simulation einen alternativen zukünftigen Verlauf abzubilden. Nur so können Ground Handler oder Airline-Verantwortliche die Auswirkungen getroffener Maßnahmen beurteilen.

Auch wenn die Sprache der Simulationsumgebung auf der transaktionsorientierten Sichtweise beruht, führt eine anwenderorientierte Gestaltung der Arbeitsblöcke auf das prozessorientierte Weltbild. Hier werden die objektbezogenen Aktivitäten und die zugehörigen Attribute zu einem Prozess zusammengefasst, wobei in den programmtechnisch gesehen aktiven Prozessphasen die Ereignisse nachgebildet und mit der inaktiven Phase die Prozessdauer beschrieben werden [16].

Soll bspw. das *deboarding* abgebildet werden, so wird ein Prozess definiert, der gleichzeitig zu seiner Deaktivierung die Reaktivierung nach einer bestimmten Zeitdauer vormerkt. Die Zustandsänderungen wird also über den Prozess abgebildet, der zwischen den Ereignissen *Beginn des deboarding* und *Ende des deboarding* programmtechnisch gesehen inaktiv bleibt. Die Phase zwischen diesen beiden Ereignissen entspricht dem Ablauf des Prozesses.

Die prozessorientierte Sichtweise eignet sich zur Modellierung nebenläufiger und komplex interagierender Prozesse. Ein weiterer Vorteil des Funktionsumfangs von MATLAB und SIMULINK liegt in der Möglichkeit zur Parameter-iterativen Optimierung des Gesamtprozesses.

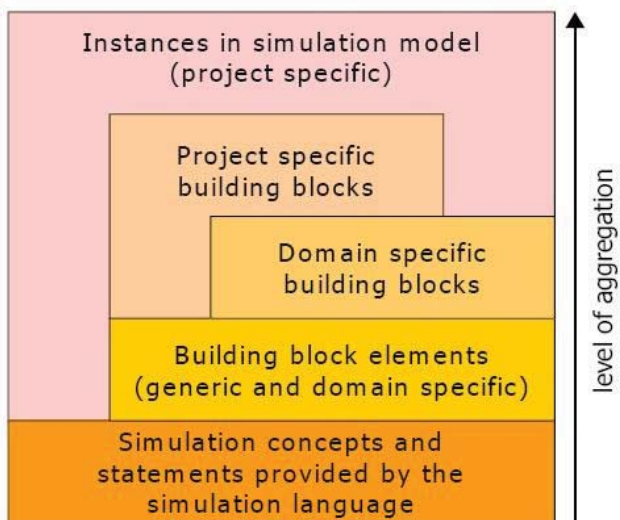


BILD 3 Building Blocks und Infrastruktur einer Sprache
(Quelle: Valentin, E. C. and Verbraeck, A.)

Um die für die verschiedenen Fragestellungen notwendigen Modellierungsanforderungen bereitstellen zu können, wurden in Anlehnung an die Theorie der Building Blocks [18] aus der Informatik so genannte *Funktionsblöcke* mit programmierbaren Blockeigenschaften erstellt. Diese sind auf beliebigen Hierarchieebenen einsetzbar und besitzen ähnlich wie Building Blocks domänen- und aufgabenspezifische Eigenschaften. So kann beispielsweise die Zeitsynchronisation mit einer umliegenden Simulationsumgebung als generische, und die Ressourcenkonfiguration für den Turnaround als domänenspezifische Funktionalität verstanden werden. Funktionsblöcke selber werden mit Elementen der Sprachen SIMULINK bzw. SIMEVENTS zu *Arbeitsblöcken* zusammengefasst, welche in Bezug auf

die prozessorientierte Sichtweise als Abstraktion der eigentlichen Prozesse betrachtet werden können und im Gegensatz zu den entwicklerorientierten Funktionsblöcken eher der Nutzerebene zuzurechnen sind. Arbeits- wie Funktionsblöcke können besondere oder allgemein benötigte Funktionen auf unterschiedlichen Ebenen in dem Modell übernehmen. Die Konfiguration eines Blocks findet dabei über ein jeweiliges Graphical User Interface (GUI) statt.

Bild 4 zeigt einen Ausschnitt aus einem mit der Simulationsumgebung erstellten Turnaround-Modell. Die graphische Verknüpfung unter SIMULINK erlaubt hierbei eine einfache Modellierung der seriellen und parallelen Abhängigkeiten der jeweiligen Prozesse (blau eingefärbte Arbeitsblöcke). Die Arbeitsblöcke stellen die verschiedenen Prozesse des Turnarounds auf der höchsten Nutzerebene dar, welche in Bezug zum Detaillierungsgrad auch als Makroebene betrachtet werden kann. Hellgrau eingefärbte Arbeitsblöcke symbolisieren eine Ereignisabhängigkeit. Das Ereignis *cabin crew on board* ist bspw. vom vorherigen deboarding und einer festgesetzten äußeren Zeit abhängig. Das neue Personal wird erst nach erfolgtem deboarding und spätestens zu einem, von der *Scheduled Off Block Time* abhängigen Zeitpunkt im Flugzeug eintreffen.

Die Flugzeuge durchlaufen, aufgrund des Signal verarbeitenden Charakters SIMULINKs als Integer abgebildet, die Netzstruktur, wobei die Flüge in parallel geführten Zweigen erst nach Freigabe der jeweiligen Arbeitsblöcke in der Struktur weiter voranschreiten können. Das Ereignis zum Fortfahren eines einzelnen Flugzeuges wird in den nachgeschalteten Arbeitsblöcken (*merger*) ermittelt. Beispielsweise wird der Arbeitsblock *security check* erst nach Beendigung der Arbeitsblöcke *cabin-board supply*, *offload catering*, *catering* und *cleaning* gestartet.

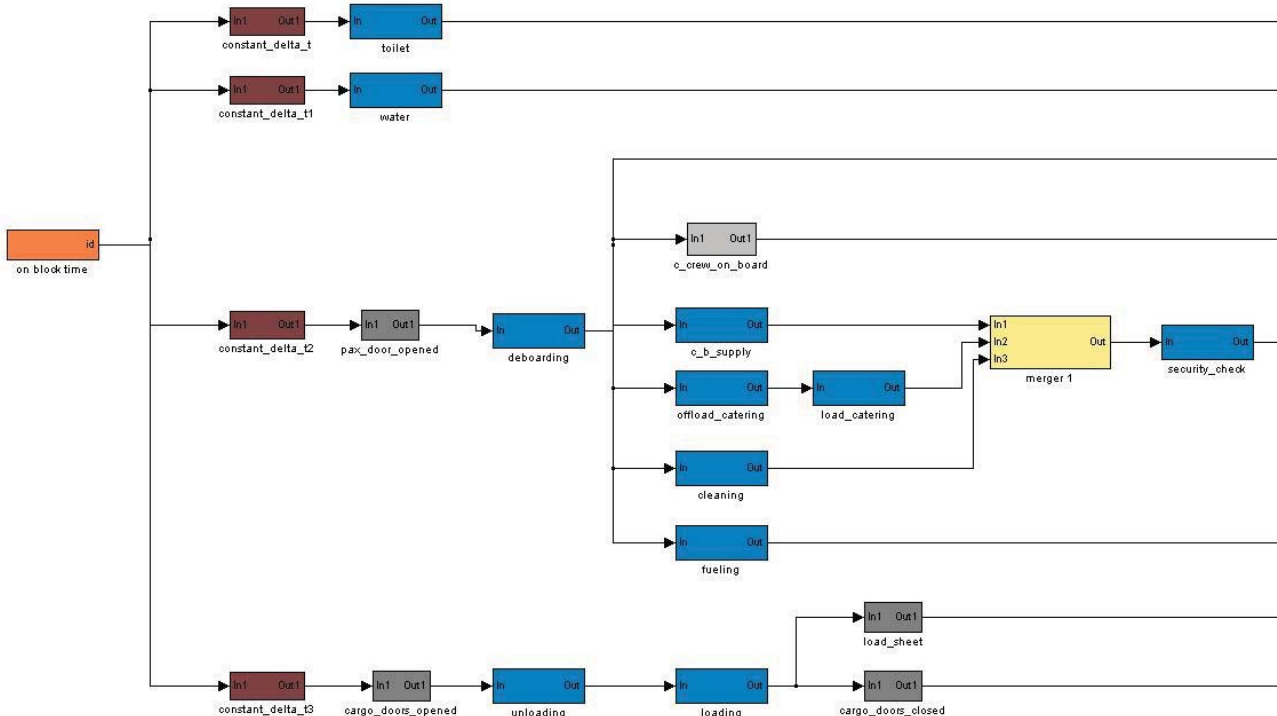


BILD 4 Ausschnitt Turnaround-Prozess

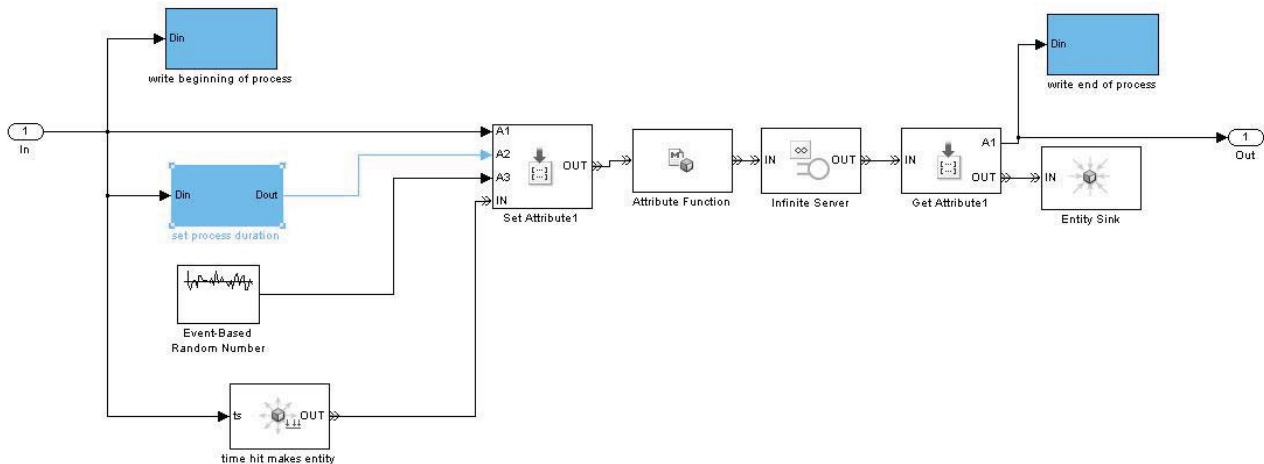


BILD 5 Ansicht eines Arbeitsblockes

Bild 5 zeigt auf einer tiefer liegenden Hierarchieebene, anhand des Inhaltes eines auf den Fall stochastisch verteilter Prozessdauern vereinfachten Arbeitsblockes, die Verknüpfung statischer Elemente der transaktionsorientierten Sichtweise mit den erstellten Funktionsblöcken. Zu Prozessbeginn erzeugte Entities werden durch die Struktur zu einem Server geleitet, welcher sie erst nach einem bestimmten Zeitraum wieder passieren lässt. Diese Zeit wird anschaulicher Weise als Prozess- oder Servicedauer aufgefasst

Aufgrund mehrerer gleichzeitig das Modell durchlaufender Entities (Flugzeuge) handelt es sich dabei um einen bis zu einer vorgegebenen Maximalanzahl von Entities parallel arbeitenden Server. Ein typisches Element der Sprache SIMEVENTS findet sich hierbei bei der Abbildung stochastischer Einflüsse, welche über den Block *Random numbers* eine Vielzahl möglicher kontinuierlicher wie diskreter Verteilungen zu Verfügungen stellen. Des Weiteren finden sich Ausgabeblocks, die vom Nutzer an beliebiger Stelle angebracht, genauere Informationen über den Status einzelner Flugzeuge innerhalb des Prozesses vermitteln können. Die bspw. zu Prozesseintritt (Deaktivierung des Prozesses bei prozessorientierter Weltsicht) sowie nach Ablauf (Reaktivierung des Prozesses bei prozessorientierter Weltsicht) eintretenden Ereignisse werden über konfigurierbare Funktionsblöcke in Abhängigkeit von Prozesstyp, Callsign und Zeitpunkt für den Anwender über eine Schnittstelle protokolliert (blaue eingefärbte Blöcke mit Umrandung). Im Falle des deboardings kann somit dem Nutzer leicht eine anschauliche Darstellung des Prozessfortschrittes zu Verfügung gestellt werden. Auf gleiche Weise findet die Ermittlung der betrieblich vorgegebenen Prozessdauern aus einer MySQL-Datenbank statt (blau eingefärbter Block ohne Umrandung).

Mit Hilfe spezifischer Arbeits- und Funktionsblöcken könnte eine detailliertere Modellierung des Reinigungsprozesses auf einer Mikroebene durchgeführt werden. Hierbei sei nochmals auf die Bedeutung des erhöhten Systemverständnisses durch Simulation und Einblick in funktionale Strukturen hingewiesen. Der Akteur ist angehalten, neben der Konfiguration und Änderung der Parameter

auch strukturelle Eingriffe in beliebiger Auflösung vorzunehmen.

Zurzeit wird das Turnaroundmodell in einen Simulationsverbund zur Erprobung des ACDM Ansatzes integriert.

4. ZUSAMMENFASSUNG

Eine ereignisdiskrete Simulationsumgebung zur Entscheidungsunterstützung von Akteuren bei Fragestellungen betrieblich-organisatorischen Charakters aus dem Bereich des ACDM und ATM wurde vorgestellt. Ziel ist ein verbessertes Situationsbewusstseins des Nutzers mittels Simulation der jeweiligen Fragestellungen.

Die Simulationsumgebung dient im Zusammenhang des ACDM neben der Modellierung auch der Vorhersage und Analyse. Sie findet in der Integration in einen übergeordneten Simulationsverbund sowie zur Unterstützung bestehender Systeme aus dem ATM-Bereich ihre Anwendung. Ausgehend von (stochastischen) zeitbewerteten farbigen Petrinetzen wurde unter Berücksichtigung der Anforderungen und der Komplexitätssteigerung bei Erhöhung des Detaillierungsgrades auf eine höhere transaktionsorientierte Modellierungssprache zurückgegriffen. Ausschlaggebend hierfür war das mit der Einbeziehung von Ressourcen verbundene höhere Modellierungskonstrukt der Ressourcensynchronisation.

Die Integration in andere Simulationsverbände, Parameter-iterative Optimierung sowie die anschauliche prozessorientierte Darstellung und die intuitive Benutzeroberfläche werden durch den Funktionsumfang der zugrunde liegende MATLAB / SIMULINK Umgebung wesentlich erleichtert. Die verwendete Sprache ermöglicht kompositionale und hierarchische Modellbildung wie auch Modularisierung. Bspw. kann ein in einem höheren Detaillierungsgrad beschriebene De-Icing Prozess mit der hier vorgestellten Turnaroundsimulation wie auch einer Rollsimulation verknüpft werden. Zudem erlaubt die erstellte Simulationsumgebung eine einfache strukturelle Manipulation des Modells bei einem hohen Parametrisierungsgrad und flexiblen Darstellungsmöglichkeiten für statistische Informationen. Die transparente Blocksstruktur der Simulationsumgebung trägt zur Akzeptanz durch den Nutzer bei und ermöglicht vor allem bei individueller Modellanpassung ein verbessertes Systemverständnis.

5. REFERENZEN

- [1] Burey, J. and Smith, R., CDM is the big success story of Free Flight Phase 1, says FAA. Inside FAA 3 (7), 1999
- [2] Martin, P., Hudgell, A. and Bouge, N., Improved Information Sharing: A Step Towards the Realisation of Collaborative Decision Making, 1998
- [3] EUROCONTROL, Airport CDM Applications Guide, Brussel, 2003
- [4] Völckers, U. and Meier, Ch. Technical Vision for Airport Airside Traffic Management, DLR-Institute of Flight Guidance, 2005
- [5] Günther, Y., Inard, A., Werther, B., Bonnier, M., Spies, G., Marsden, A., Temme, M., Böhme, D., Lane, R., and Niederstrasser, H., Total Airport Management (Operational Concept & Logical Architecture) Version 1.0., DLR-Institute of Flight Guidance, 2006
- [6] Chin, R.T.H., van Houten, S.-P., and Verbraeck, A., Towards a Simulation and Visualization Portal to support Multi-Actor Decision Making in Mainports, Proceedings of the 2005 Winter Simulation Conference M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.
- [7] Odoni, A. R., Bowman, J., Delahaye, D., Deyst, J. J., Feron, E., Hansman, R. J., Khan, K., Kuchar, J. K., Pujet, N. and Simpson, R. W., Existing and Required Modeling Capabilities for Evaluating ATM Systems and Concepts, International Center for Air Transportation Massachusetts Institute of Technology, 1997
- [8] Martinez, J. C., Trani, A. A. and Ioannou, P. G., Modeling Airside Airport Operations Using General-Purpose, Activity-Based, Discrete-Event Simulation. Transportation Research Record 1744, Paper No. 01-3476
- [9] Page, B., Diskrete Simulation - Eine Einführung mit Modula-2 Unter Mitarbeit von H. Liebert, A. Heymann, L.M. Hilty und A. Häuslein, Springer Verlag, Berlin, Heidelberg, 1991
- [10] Valentin, E. C. and Verbraeck, A., Requirements for domain specific discrete event simulation environments, Proceedings of the 2005 Winter Simulation Conference M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, eds.
- [11] Idris, H. R., Delcaire, B., Anagnostakis, I., Hall, W. D., Clarke, J.-P., Hansman, R. J., Feron, E. and Odoni, A. R., Observations of Departure Processes at Logan Airport to support the Development of Departure Planning Tools, 2nd USA/Europe Air Traffic Management R & D Seminar, Orlando, 1998
- [12] Lunze, J., Ereignisdiskrete Systeme - Modellierung und Analyse dynamischer Systeme mit Automaten, Markovketten und Petrinetzen., Oldenbourg Verlag, München, 2006
- [13] Kiencke, U., Ereignisdiskrete Systeme - Modellierung und Steuerung verteilter Systeme, Oldenbourg Verlag, München, 1997
- [14] Cassandras, C. G. and Lafortune, S., Introduction to Discrete Event Systems second edition, Springer Science+Business Media, New York, 2008
- [15] Apte, A., Modeling System Architecture and Resource Constraints Using Discrete-Event Simulation, The MathWorks Inc., Boston, 2008
- [16] Banks, J., Carson II, J. S., Nelson, B. L. and Nicol, D. M. Discrete - Event System Simulation, Prentice-Hall. Inc., New Jersey, 2001
- [17] Spies, G., Piekert, F., Marsden, A., Suikat, R., Meier, Ch. and Eriksen, P., Operational concept for an airport operations center to enable total airport management, 26th International congress of the aeronautical sciences, Anchorage, 2008
- [18] Valentin, E. C. and Verbraeck, A., Requirements for domain specific discrete event simulation environments, Proceedings of the 2005 Winter Simulation Conference M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, eds.