

ANALYSE VON WARTUNGS- UND LOGISTIKPROZESSEN AUF DER BASIS AUSFÜHRBARER MODELLE

M. Schulz, V. Zerbe
Technische Universität Ilmenau
Ilmenau, D-98684
Deutschland

S. Marwedel
Airbus Deutschland
Hamburg, D-21129
Deutschland

ZUSAMMENFASSUNG

Der Entwurf auf der Basis ausführbarer Modelle (*model based design*) hat das Potential die Produktivität des Systementwurfes wesentlich zu erhöhen. Das Modell muss, beginnend in den frühen Entwurfsphasen, auf abstrakter Ebene entworfen und schrittweise verfeinert werden. Es versteckt Systemdetails, muss jedoch auch alle notwendigen Subsysteme und Komponenten auf allen Spezifikationsebenen repräsentieren. Auf dieser Basis können Systemleistungsbewertungen durchgeführt werden. Für die Modellierung komplexer, vernetzter, heterogener Systeme ist die Kombination verschiedener Berechnungsmodelle als auch die Verfügbarkeit und Erweiterbarkeit umfangreicher Modellbibliotheken notwendig. MLDesigner ist ein Modellierungs- und Simulationswerkzeug welches genau diese Anforderungen unterstützt. Im vorliegenden Papier werden Verfahren und Technologien vorgestellt, die eine Analyse und Optimierung von Wartungs- und Logistikprozessen auf der Basis ausführbarer Modelle erlauben. Präsentiert wird ein abstraktes Modell einer virtuellen Fluggesellschaft. Gezeigt werden Simulationsergebnisse für dieses Modell für verschiedene Verfügbarkeiten von Ressourcen.

1. EINFÜHRUNG

Das Ziel einer Fluggesellschaft ist es, die Transportkapazität zu maximieren und die Verfügbarkeit der Flotte zu erhöhen und das bei gleichzeitiger Minimierung der Kosten. Die Fragestellung ist: Wie können zum Beispiel Wartungs- und Logistikprozesse im Kontext eines komplexen Luftfahrtsystems analysiert und optimiert werden, um diese Zielstellung zu erreichen. Zunächst ist ein ausführbares Systemmodell zu erstellen, bevor Leistungsbewertungen durchgeführt und die Verfügbarkeit einer Flotte analysiert werden kann. Die Modellierung des ganzheitlichen Systems beginnt auf abstrakter Ebene. Eine strukturierte Vorgehensweise ist bei der schrittweisen Verfeinerung des Modells unerlässlich. Das komplexe Entwurfsproblem muss dabei in geordneter Weise in handhabbare Teilprobleme aufgeteilt werden, deren Komplexität eine Bearbeitung zulässt. Dabei ist es wichtig zu beachten, dass die Gültigkeit der Lösung eines Teilproblems stets nur im Rahmen der Lösung des gesamten Entwurfsproblems gegeben ist, d. h. die Lösung eines Teilproblems trägt nicht automatisch zur Lösung des Gesamtproblems bei, sondern wird stets an ihrem Beitrag zur Lösung des übergeordneten Entwurfsproblems gemessen. Bei dieser Art der Modellierung des Systems mit seinen Subsystemen und Komponenten über alle Abstraktionsebenen, bleiben interne Systemdetails auf den höheren Modellebe-

nen versteckt. Sie tauchen erst in den Blättern des hierarchischen Modellbaumes auf. Die Ausführung des Modells, die Simulation, erlaubt nun die Analyse des Systems für begrenzte, verfügbare Ressourcen und Systemanforderungen.

Für den Entwurf komplexer, heterogener, integrierter, vernetzter Systeme müssen verschiedene Berechnungsmodelle in die Systemarchitektur integriert werden. Ein Berechnungsmodell ist ein mathematischer Formalismus der eine Menge zulässiger Operationen für eine Berechnung definiert und von Implementierungsdetails abstrahiert. Somit können u. a. Nebenläufigkeiten, Datenflüsse, reaktive und kontinuierliche Systeme, Synchronisations- und Kommunikationsaspekte adäquat beschrieben werden. Jedes Subsystem oder jede Komponente des zu modellierenden Gesamtsystems sollte jedwedes Berechnungsmodell verwenden können.

Von besonderer Bedeutung ist die Integration des *top-down* und *bottom-up* Entwurfes für die Modellierung von Systemen. Die *top-down* Methodik erlaubt die Modellierung auf abstrakte Art und Weise. Der Entwurf kann überschaubar und beherrschbar gemacht werden. Der vorgegebene Entwurfsraum wird durch Komponentenerlegungen schrittweise verfeinert bis der angestrebte Detaillierungsgrad des Systems erreicht ist. Bei der *bottom-up* Methodik werden Subsysteme und Komponenten modelliert und

zu einem Gesamtsystem kombiniert. So können Erfahrungen aus vergangenen Systementwicklungen in den Entwurf eingebracht werden. Daraus folgt, dass beide Techniken für die Praxis kombiniert werden müssen. Diese Kombination von *top-down* und *bottom-up* ist bekannt als die *meet-in-the-middle* Strategie, siehe [1]. Um möglichst viele Alternativen untersuchen und optimale Entscheidungen treffen zu können, sollte die *top-down* Vorgehensweise die Hauptstrategie sein. Leistungsparameter können schrittweise verfeinert oder als Annotation durch den *bottom-up* Entwurf im Systemmodell repräsentiert werden. Diese Parameter sind hilfreich, um Leistungsbewertungen des Systems durchzuführen.

2. MODELLBASIERTER ENTWURF

Zurzeit werden derart komplexe Systeme, wie es zum Beispiel ein *avionics* System darstellt, rein anforderungsorientiert (*requirement based*) entwickelt. Aus verschiedenen Quellen werden Systemanforderungen zusammengetragen um Papierspezifikationen zu erstellen. Viele verteilte Entwicklerteams erarbeiten auf dieser Basis einen detaillierten Entwurf für die Subsysteme und Komponenten. Nach dem Erreichen eines akzeptierten Entwurfes erfolgen die Validierung und der Test des Gesamtsystems. Fehler im Designprozess können daher oft erst sehr spät entdeckt werden. Der Entwurf auf der Basis ausführbarer Modelle (*model based design*) hat das Potential die Produktivität des Systementwurfes wesentlich zu erhöhen. Diese Entwurfsstrategie ist modellgetrieben und beginnt bereits in den frühen Phasen des Entwurfes mit der Entwicklung einer ausführbaren Spezifikation. Diese ist verknüpft mit den Systemanforderungen. Im Zentrum des gesamten modellbasierten Entwurfsprozess steht das zu modellierende komplexe System, das ausführbare Modell. Es wird fortlaufend, schrittweise verfeinert. Durch die Verknüpfung des ausführbaren Modells mit den Systemanforderungen können durch Simulation Inkonsistenzen in den Systemanforderungen sehr zeitig gefunden werden. Es kann während des gesamten Entwurfsprozesses geprüft werden ob der Entwurf den Anforderungen entspricht und welche Auswirkungen eine vorgeschlagene Änderung in den Systemanforderungen bewirkt. Der modellbasierte Entwurf hat gegenüber dem anforderungsorientierten Entwurf [2], (auszugsweise) folgende hervorstechende Eigenschaften:

- Die Untersuchung des dynamischen Verhaltens des Systems wird bereits in den frühen Phasen des Entwurfsprozesses möglich.
- Entwurfsentscheidungen können auf der Basis simulierbarer Alternativen und zugehöriger *trade-off* Analysen getroffen werden.
- Es wird ein Modell auf verschiedenen Entwurfs-ebenen verwendet.

- Nichtfunktionale Anforderungen können gleichfalls modelliert und validiert werden.

3. MLDESIGNER

MLDesigner [3] ist ein Werkzeug für den Entwurf komplexer Systeme auf Missions- und Systemebene. Dieser Ansatz integriert Architektur, Funktion und Anwendungsszenarien in einer einzigen Entwicklungsumgebung auf sehr abstrakter Ebene. MLDesigner ist ein *multi-domain* Simulator und unterstützt die Modellierung in *discrete event* (DE), *synchronous data flow* (SDF), *continuous time* (CT), *finite state machine* (FSM) und anderen Berechnungsmodellen. Verschiedene Berechnungsmodelle können miteinander kombiniert werden, um ein System zu modellieren. Weiterhin kann ein Systemmodell mit seinen Komponenten in beliebiger Detailtiefe dargestellt werden. Auf dieser Basis können Systemleistungsbewertungen durchgeführt werden. Ein Vergleich und eine Bewertung unterschiedlicher Werkzeuge für den Entwurf auf Systemebene ist in [4] vorgenommen worden.

4. VIRTUELLES AIRLINE MODELL

Entwickelt wurde ein ausführbares Modell für eine virtuelle Fluggesellschaft mit Hilfe des Entwurfswerkzeuges MLDesigner. Es handelt sich bei dem zu modellierenden System um ein ereignisorientiertes System. Zu bestimmten Zeitpunkten starten und landen Flugzeuge oder werden Wartungsprozesse ausgelöst. Das Modell ist getrieben durch Ereignisse, ausgelöst zu bestimmten Zeitpunkten, d. h. Grundlage für die Modellierung ist das DE-Berechnungsmodell. Ziel ist neben der Untersuchung der Gesamtverfügbarkeit der Flotte unter dem Einfluss der Zuverlässigkeit der einzelnen Flugzeugsysteme auch die Optimierung der Wartungs- und Instandhaltungs- sowie Logistikprozesse. Wichtige Grundlage dafür ist natürlich das Verständnis des Flottenbetriebes. Es müssen räumlich verteilte Ressourcen (Hauptstation, entfernte Stationen) berücksichtigt und zwischen Boden- Flugzeug- und Kommunikationssegment unterschieden werden. Im Zentrum steht daher das zu modellierende, komplexe Gesamtsystem einer Fluggesellschaft, s. Bild 1.

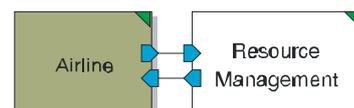


BILD 1. Top-Level Airline Modell

Für den Betrieb einer Fluggesellschaft notwendige Ressourcen (*Plane, FlightCrew, CabinCrew* und *Mechanic*) werden im Modul *Resource Management*

verwaltet. Zu berücksichtigen sind zum Beispiel Dienst- und Ruhezeiten von Besatzungsmitgliedern in- und außerhalb von Luftfahrtunternehmen oder die Qualifikationen des gesamten Personals oder der Kerosinverbrauch pro 100 Meilen in Litern oder die Entfernung zwischen Flughäfen. Innerhalb der Fluggesellschaft ist die Flotte, das Flugzeug, eingebettet in die Prozesse Flugplanung und Wartung sowie Engineering, s. Bild 2. Über ein Request bzw. Response können Ressourcen angefordert bzw. zugewiesen werden. *Planning* nimmt zum Beispiel Initialisierungen den Flugplan betreffend vor und weist die Ressourcen den in den Missionen spezifizierten Flügen zu.

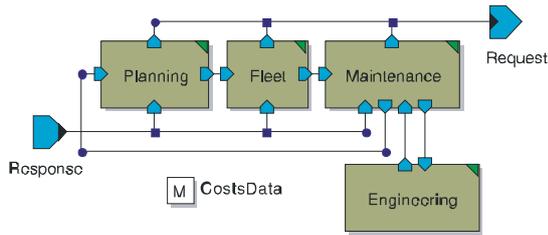


BILD 2. Second-Level Airline Modell

Bei der Modellierung wird ein virtueller Prototyp erstellt. D. h. modellierte Funktionskomponenten werden auf modellierte Architekturkomponenten abgebildet (*functional allocation*). Dieser virtuelle Prototyp ist zumeist hierarchisch über viele Ebenen (... , *aircraft, cabin, ...*) strukturiert, s. Bild 3, 4.

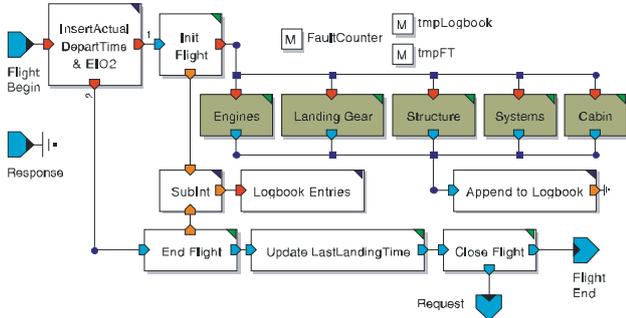


BILD 3. Fourth-Level Airline Modell, Aircraft Modell

Wesentliche Komponenten einer Kabine sind dabei beispielsweise die Toilette, das Kommunikationssystem, die Sitzreihen und die Bordküche.

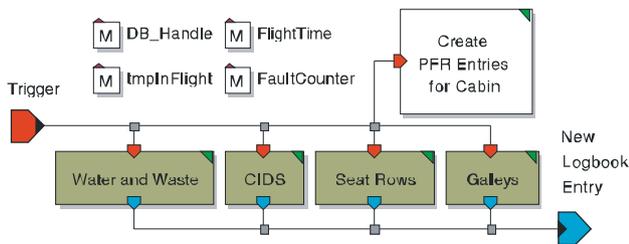


BILD 4. Fifth-Level Airline Modell, Cabin Modell

Zwischen den einzelnen Komponenten existieren komplexe funktionale (*smoke-detection-function*)

und nichtfunktionale (*availability, costs*) Zusammenhänge. Letztere definieren eine Gütefunktion, die mit Hilfe der ausführbaren Modelle für verschiedene Architekturen evaluiert wird. Es kann am Modell gezeigt werden, welche Entwurfsentscheidungen erforderlich sind, um das übergeordnete Entwurfsziel zu erreichen. Es kann gezeigt werden, welche Auswirkungen eine Änderung einer Komponente auf das Gesamtsystem hat, denn die Lösung eines Teilproblems trägt nicht automatisch zur Lösung des Gesamtproblems bei.

Während eines Fluges werden neben Logbucheinträgen auch *post-flight-report* Einträge (s. Bild 4) geschrieben. Die Logbucheinträge aktivieren ihrerseits Wartungs- und Logistikprozesse. Prozesse können effizient im FSM-Berechnungsmodell modelliert werden. Bild 5 zeigt in abstrakter Form den Wartungsprozess.

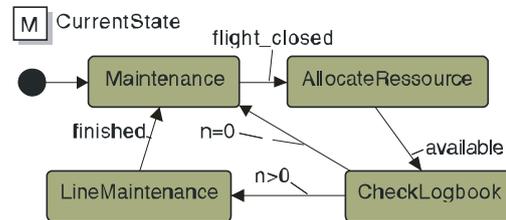


BILD 5. Wartungsprozess

Wird ein Flug geschlossen, werden für die Wartung notwendige Ressourcen alloziert. Sind keine Logbucheinträge geschrieben worden kann der Wartungsprozess abgeschlossen und das Flugzeug als Ressource freigegeben werden. Anderenfalls wird die Wartung durchgeführt. Die Eigenschleifen an den Zuständen der FSM ergeben sich formal aus den dargestellten Kanten. Der Graph ist vollständig und widerspruchsfrei. Die FSM selbst ist auch wieder hierarchisch modelliert. Für $n > 2$ Logbucheinträge sind entsprechend auch mehrere Wartungsprozesse abzuarbeiten. Auf analoge Art und Weise werden die Logistikprozesse modelliert. Die *multi-domain* Fähigkeit von MLDesigner erlaubt die Integration des FSM-Modells in das DE-Airline-Modell, lies zur Integration von FSM in DE dazu in [5]. Zudem gestattet MLDesigner, wie bereits erwähnt, die Integration von Ressourcen in das Modell, wodurch der Verbrauch, die Verfügbarkeit „messbar“ wird. Auf der Basis dieser ausführbaren Modelle können Wartungs- und Logistikprozesse im Gesamtkontext einer Fluggesellschaft analysiert werden.

Folgende Annahmen können als Eingabedaten für das Modell, die Simulation, getroffen werden. Eine Flotte besteht aus unterschiedlichen Flugzeugtypen, sowohl für die Kurz- als auch die Langstrecke. Anzugebende Parameter sind der Flugzeugtyp, die maximale Reichweite in Meilen, die maximale Anzahl verfügbarer Sitzplätze und der Kerosinverbrauch pro 100 Meilen in Litern, s. Tabelle 1.

| | | | |
|------|------|-----|-----|
| 747 | 7920 | 390 | 430 |
| A320 | 1980 | 165 | 500 |

TABELLE 1. lies: Flugzeugtyp, Reichweite, Sitzanzahl, Kerosinverbrauch

Der Airbus A320 hat eine Reichweite von 1980 Meilen, eine Sitzanzahl von 165 sowie einen Kerosinverbrauch von 500 Litern pro 100 Meilen.

Weitere notwendige Ressourcen sind neben der Flotte, die *FlightCrew*, *CabinCrew*, Mechaniker und Hangars. Für die *Crew's* werden Zulassungen für einen bestimmten Flugzeugtyp (747, A330, usw.) unterschieden und Mechaniker besitzen Lizenzen für die Ausführung bestimmter Tätigkeiten (Reparaturen am Flugzeug, Unterschriftsberechtigungen, etc.). Unterschieden werden hier die folgenden Kategorien: CatA, CatB1, CatB2 und CatC. Flugzeuge, *Crew's*, Mechaniker und Hangars sind Ressourcen die an unterschiedlichen Standorten zur Verfügung stehen. So können den Flughäfen, spezifiziert über den IATA-Code (International Air Transport Association), eine bestimmte Anzahl an Ressourcen zugewiesen werden, s. Tabelle 2.

| | | | |
|-----|------------|------|---|
| FRA | FlightCrew | A320 | 3 |
| HAM | Plane | A300 | 2 |

TABELLE 2. lies: Standort, Ressource, Typ, Anzahl

In Hamburg sind zur Zeit zwei Airbus-Maschinen vom Typ A300 verfügbar.

Im spezifizierten Flugplan können zurzeit über 50 Flughäfen in Europa angefliegen werden. Ein Flugplan setzt sich zusammen aus verschiedenen Missionen. Ein Mission wiederum entspricht einem Rundflug beginnend vom Flughafen A über B, ..., C und zurück zum Flughafen A. Eine Mission beginnend von Frankfurt über Barcelona, Düsseldorf zurück nach Frankfurt wäre wie folgt zu spezifizieren, s. Tabelle 3:

| | | | | |
|-----|-----|-------|-------|-----|
| FRA | BCN | 05:20 | 01:55 | 680 |
| BCN | DUS | 08:00 | 02:20 | 725 |
| DUS | FRA | 11:05 | 00:50 | 117 |

TABELLE 3. lies: Abflugort, Ankunftsart, Abflugzeit, Flugdauer, Flugstrecke

Berücksichtigung finden weiterhin die Abflugzeit, Flugdauer und Flugstrecke, d. h. der Flug von Barcelona nach Düsseldorf startet um 8.00 Uhr, dauert 2h 20 min und die Flugstrecke beträgt 725 Meilen.

Eine 5 Tage Simulation des Gesamtsystems bei einem gegebenen Flugplan liefert für jeden Standort die Ressourcenauslastung, s. Bild 6.

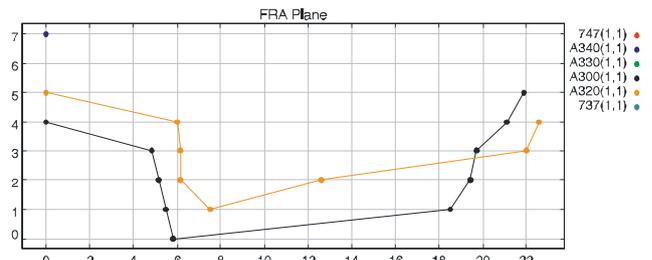


BILD 6. Ressourcenauslastung (Flugzeuge) in Frankfurt

Von besonderem Interesse für die Fluggesellschaft ist die Verfügbarkeit der Flotte. Wartungs- und Logistikprozesse haben erheblich Einfluss auf die Verfügbarkeit. Eine 5 Tage Simulation des Gesamtsystems ergibt für eine vorgegebene Ressourcenverteilung, gegebenem Flugplan und bei einer mittleren Reparaturzeit MTTR (*mean-time-to-repair*) von 14 Minuten über 80 Flugverspätungen. Wenn Wartungs- und Logistikprozesse und die dazu notwendige Architektur optimiert wird und somit eine Halbierung der Reparaturzeit erzielt werden kann, dann könnten bei gleicher vorgegebener Ressourcenverteilung und gleichem Flugplan die Verspätungen auf nur noch 12 reduziert werden. Siehe dazu die Simulationsergebnisse in Bild 7.

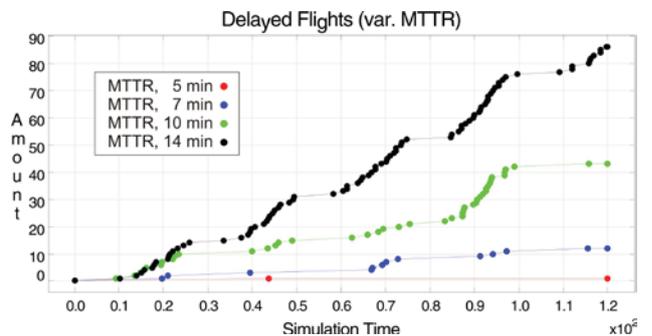


BILD 7. Flugverspätungen über der Simulationszeit für verschiedene MTTR

Welche Konsequenzen können daraus abgeleitet werden? Möglicherweise führt eine effizientere Fehlersuche zu einer Reduzierung der Reparaturzeit und schließlich zu einer höheren Verfügbarkeit der Flotte einer Airline. Eine effizientere Fehlersuche kann erreicht werden, wenn bei einem konkreten Wartungsvorgang die *post-flight-report* Einträge (PFR), die während eines Fluges kontinuierlich aufgezeichnet werden, in strukturierter und komprimierter Form Berücksichtigung finden. PFR's enthalten Zustandsinformationen aus den verschiedenen Systemen. Selbstverständlich führt auch eine andere Ressourcenverteilung zu einer unterschiedlichen Anzahl von Flugverspätungen. Das ausführbare Modell kann somit für die Dimensionierung einer

Architektur auf der jeweiligen Entwurfsebene verwendet werden.

5. RESÜMEE

Abschließend sei bemerkt, dass durch den Einsatz formaler, modellbasierter Methoden ein zu optimierendes Gesamtsystem modelliert und ausgeführt werden kann. Durch die Integration nichtfunktionaler Anforderungen kann das System über Architekturvarianten iteriert und somit Designentscheidungen bereits in den frühen Entwurfsphasen validiert werden. Im Modell vorbereitet ist die Integration von Ersatzteilen und Warenhäusern, so dass auch Logistikprozesse zukünftig optimiert werden können um die Verfügbarkeit der Flotte einer Fluggesellschaft zu erhöhen.

6. LITERATURVERZEICHNIS

- [1] K. D. Mueller Glaser, Systems Engineering in Microsystems Design. IFIP Workshop on Modelling of Microsystems. Stirling Scotland, 1997
- [2] J. Friedman, *Model-Based Design Helps Aerospace Engineers*. June 12 2009, http://www.designnews.com/article/print279138-Model-Based_Design_Helps_Aerospace_Engineers.php
- [3] <http://www.mldesigner.de>
- [4] A. de A. Agarwal, C.-D. Iskander, R. Shankar, G. Hamza-Lup. *System-Level Modeling Environment: MLDesigner*. SysCon 2008, IEEE International Systems Conference, Montreal, Canada, April 7-10 2008,
- [5] I. Paunovic, V. Zerbe. *Modeling and Simulation of Digital Systems in different Domains*. 3rd SSSS- Small Systems Simulation Symposium, Nis, Serbia, February 12-14 2010, p. 17-23
- [6] Flugverspätungen über der Simulationszeit für verschiedene MTTR