

Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages

Aircraft Total Maintenance Operations, Solution & Technologies Program – Germany @MOST-G

**Teilvorhaben Service Oriented Architecture Application
for Airlines & Aircraft – SO4A**

Schlussbericht



**TECHNISCHE UNIVERSITÄT
ILMENAU**

Technische Universität Ilmenau
Fachgebiet System- und Software-Engineering

Projektlaufzeit

01.1.2009 – 31.06.2012

Schlussbericht nach NKBF 98 Nr.8.2

ZE:	Technische Universität Ilmenau	Förderkennzeichen: 20K0801D
Vorhabenbezeichnung:	Service Oriented Architecture Application for Airlines & Aircraft	
Laufzeit des Vorhabens:	01.1.2009 – 31.06.2012	
Techn. Berichtersteller:	Prof. Dr.-Ing. Armin Zimmermann, Dipl.-Inf. Mario Schulz	
Projektleiter:	Prof. Dr.-Ing. Armin Zimmermann Technische Universität Ilmenau Fakultät für Informatik und Automatisierung Fachgebiet System- und Software-Engineering PF 100 565 D-98693 Ilmenau	

Das diesem Schlussbericht zugrundeliegende Forschungsvorhaben wurde mit Mitteln des Bundesministeriums für Wirtschaft und Technologie (BMWi) unter dem Förderkennzeichen 20K0801D gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Berichtersteller. Die Form des Berichtes entspricht den Nebenbestimmungen auf Kostenbasis des Bundesministers für Wirtschaft und Technologie an Unternehmen der gewerblichen Wirtschaft für Forschungs- und Entwicklungsvorhaben (NKBF98).

Wir danken unseren Forschungspartnern und insbesondere dem Projektträger für die vertrauensvolle Zusammenarbeit.

Prof. Dr.-Ing. Armin Zimmermann

Inhaltsverzeichnis

1. ÜBERBLICK ÜBER DAS VORHABEN	4
1.1 AUFGABENSTELLUNG.....	6
1.2 VORAUSSETZUNGEN DER DURCHFÜHRUNG	7
1.3 PLANUNG UND ABLAUF DES VORHABENS.....	8
1.4 STAND DER TECHNIK	13
1.5 ZUSAMMENARBEIT MIT ANDEREN STELLEN	15
2. DARSTELLUNG DER WISSENSCHAFTLICH-TECHNISCHEN ERGEBNISSE	19
2.1 DETAILLIERTE DARSTELLUNG DER ERGEBNISSE	19
2.1.1 AP150 – MODELLIERUNG UND SIMULATION.....	19
2.1.2 AP151 – MODELLIERUNG UND SIMULATION VON INFORMATION-, MATERIAL-, UND KONTROLLFLUSS.....	20
2.1.2.1 Ein Hierarchisches Airline-Modell	21
2.1.2.2 Flottenmodell	22
2.1.2.3 Aircraft-Modell.....	23
2.1.2.4 Cabin-Modell.....	24
2.1.2.5 Planungsmodell	25
2.1.2.6 Wartungs- und Logistikmodell.....	27
2.1.2.7 Kompensationsalternativen.....	28
2.1.3 AP152 – DATENANALYSE	29
2.1.3.1 Realistische Eingangsdaten.....	30
2.1.3.2 Simulationsergebnisse.....	32
2.1.3.3 Animation des Simulationsablaufs	37
2.1.4 AP153 – ENTERPRISE APPLICATION INTEGRATION	37
2.1.4.1 Werkzeugarchitektur	38
2.1.4.2 Datenbank und Simulationsszenarien	39
2.1.4.3 Ersatzteilbevorratung	41
2.1.5 AP154 – LOGISTISCHES MODELL.....	41
2.1.6 SONSTIGE ZUARBEITEN.....	49
2.2 VERWENDUNG DER ZUWENDUNG, NOTWENDIGKEIT UND ANGEMESSENHEIT DER ARBEIT ...	51
2.3 NUTZEN DES PROJEKTS UND VERWERTUNG DER ERGEBNISSE.....	53
2.4 WÄHREND DES VORHABENS BEKANNT GEWORDENE ERGEBNISSE DRITTER	54
2.5 VERÖFFENTLICHUNG DER ERGEBNISSE	55
3. REFERENZEN	56

1. Überblick über das Vorhaben

Aufgrund einer sich verändernden Marktsituation, sowohl im Bereich Flugzeugflottenbetrieb als auch im Bereich Flugzeugherstellung, stellt der Betrieb einer modernen Flugzeugflotte hohe Anforderungen nicht nur an das Flugzeug selbst, sondern ebenso an die für den Betrieb nötige Infrastruktur. Einerseits zwingen sinkende Preise für Flugtickets, Veränderung im Verhalten von Passagieren sowie steigende Treibstoffkosten die Luftfahrtgesellschaften dazu, jede erdenkliche Effizienzreserve zu nutzen, während auf der anderen Seite die Flugzeughersteller die Voraussetzungen für deren Nutzung schaffen müssen.

Ein heutiger Flottenbetrieb besteht im Wesentlichen aus drei Teilen:

1. Flugbetrieb (Flugplanung und -durchführung)
2. Passagierabfertigung (Einstieg/Ausstieg, Betreuung während des Fluges)
3. Wartung und technische Unterstützung zur Aufrechterhaltung des Betriebes.

Der Flottenbetrieb ist dabei umso effizienter, je kürzer ein Flugzeug am Boden steht und je länger es in der Luft unterwegs ist. Bodenstandzeiten sind i.d.R. entweder aus betrieblichen Gründen (Ein- und Ausstieg von Passagieren, Versorgen des Flugzeugs mit Treibstoff, Be- und Entladen) oder aufgrund von Wartungsarbeiten erforderlich. Gegenstand des Projektes ist es, für den Bereich Wartung und technische Unterstützung als Teil des Flottenbetriebs neue Lösungen zu entwickeln, um bisher nicht nutzbares Potential zur Steigerung der Effizienz zu aktivieren.

Um dieses Ziel zu erreichen, ist es erforderlich, den Bereich Wartung und technische Unterstützung als System aufzufassen und für dieses System eine an den Erfordernissen der Effizienz ausgerichtete Architektur zu entwerfen. Es handelt sich dabei um ein komplexes System, das aus einer Reihe verschiedener miteinander verbundener Teilsysteme besteht, die untereinander Informationen austauschen. So gehören z.B. das Flugzeug und verschiedene Bodensysteme, sowie Organisationen wie Disposition und Reparaturbetrieb ebenso dazu wie die Luftfahrtgesellschaft und der Flugzeughersteller.

Das im Bereich informationsverarbeitender Systeme etablierte Prinzip einer dienstorientierten Architektur (engl. SOA = Service-Oriented Architecture) ist gut geeignet, ein solches System zu beschreiben. Bei einer dienstorientierten Architektur werden die einzelnen Funktionen des Systems als Dienste aufgefasst, die eine genau beschriebene Dienstleistung erbringen. Zur Definition eines Dienstes gehören neben seinen funktionalen Eigenschaften auch nichtfunktionale Aspekte, wie z.B. Dienstgüte und Verfügbarkeit. Wesentliches Element einer solchen Architektur ist es, dass alle Dienste über festgelegte, allgemeingültige Schnittstellen erreichbar sind.

Dieses Verfahren erlaubt ein größeres Maß an Flexibilität. So können z.B. neue Dienste einfach integriert, bzw. durch die Kombination vorhandener Dienste neue Funktionen realisiert werden. Da die Schnittstellen eines solchen Dienstes im Wesentlichen über die auszutauschende Information und nicht über den zugrundeliegenden Mechanismus definiert sind, ist die dienstorientierte Architektur sehr gut geeignet, hybride Systeme wie im vorliegenden Fall zu beschreiben.

Auf diese Weise können bisherige starre Strukturen flexibilisiert und z.Zt. nicht genutzte Effizienzreserven mobilisiert werden. Zum Beispiel wird es möglich, schneller auf sich verändernde Bedingungen, wie z.B. Veränderungen im genannten Marktsegment, Einführungen neuer gesetzlicher Bestimmungen u.a. zu reagieren. Die zu entwerfende Systemarchitektur umfasst eine Reihe unterschiedlicher Bereiche mit unterschiedlichen Voraussetzungen und Randbedingungen. Dazu zählen z.B. Systeme an Bord des Flugzeuges, informationsverarbeitende Systeme am Boden, Disposition von Betriebsmitteln (technische Ausrüstung, Personal), organisatorische Abläufe usw. Im Rahmen des Vorhabens soll eine einheitliche Beschreibungsweise für eine dienstorientierte Architektur definiert werden, die diese unterschiedlichen Bereiche umfasst. Aufgrund der Komplexität des Problems kann diese Beschreibung nur in Form von Modellen erfolgen, anhand derer sich die verschiedenen Aspekte der zu entwerfenden Systemarchitektur studieren lassen.

Die Entwicklung von geeigneten Modellen für komplexe Systeme ist ein sehr wichtiger Gegenstand aktueller Forschung, denn die Erfahrung hat gezeigt, dass mit zunehmender Komplexität herkömmliche Verfahren für die Entwicklung von Systemarchitekturen versagen. Die Erstellung und das Verständnis solcher Modelle sind der Schlüssel für die Fähigkeit, komplexe Systeme entwerfen zu können. Erfahrung und Wissen auf diesem Gebiet können als wichtiger Wettbewerbsvorteil für Unternehmen im Bereich der Hochtechnologie angesehen werden.

Zu Beginn der Projektlaufzeit war die TU Ilmenau mit ihrem Unterauftragnehmer Mission Level Design zunächst als Unterauftragnehmer von Airbus Deutschland in das Verbundprojekt integriert. Im Sommer 2010 erfolgte eine größere Umstrukturierung, die mit einer Unterbrechung der Arbeiten bis Dezember 2010 einherging. In der Folge wurde die TU Ilmenau eigenständiger Verbundpartner. Außerdem wurde als zusätzlicher Unterauftragnehmer die Firma Philotech mit der Übernahme von Teilaufgaben betraut, die nach der Umstrukturierung nicht durch einen anderen Projektpartner übernommen werden sollten. Dieser Bericht beschreibt die Arbeiten der TU Ilmenau und ihrer Unterauftragnehmer in der Projektlaufzeit.

1.1 Aufgabenstellung

Hauptaufgabe des Teilprojektes ist die Entwicklung eines ganzheitlichen ausführbaren Systemmodells einer service-orientierten Architektur. Das Ziel ist es, Technologien zu entwickeln, mit denen ein gekoppeltes System von Flugzeugsegment, Kommunikationssegment und Bodensegment als Ganzes entwickelt und zu optimiert werden kann. Zu berücksichtigen sind dabei verschiedene Abstraktionsebenen (z.B. Airline, Aircraft, Cabin, Galley). Dieses Forschungsvorhaben ist von besonderer Bedeutung, da es zum ersten Mal ein Lufttransportsystem auf Flottenebene inklusive Flugzeug, Wartung und Reparatur sowie Betrieb und Logistik optimiert. Die zu entwickelnde Optimierungsmethode soll die Kosten einer geforderten Transportkapazität minimieren. Um dieses Ziel zu erreichen, müssen Architektur und Funktion von Ressourcen im Flugzeug und Services angepasst werden.

Ein System dieser Komplexität kann mit vertretbarem Risiko nur auf Grundlage modellbasierter Methoden und ausführbarer Spezifikationen entwickelt werden. Da fundamentale Verbesserungen nur durch Änderung und Optimierung der Architektur erzielt werden können, ist es erforderlich, dass die Architektur aller Untersysteme mit standardisierten Methoden modelliert wird. Dadurch wird das Einlesen, automatische Erzeugen von Modellen sowie automatische Zuordnen von Funktionen auf Architekturelemente möglich.

Folgende technisch-wissenschaftlichen Hauptziele werden verfolgt:

- Bestimmung der Optimierungsziele, Randbedingungen und Bewertungskriterien
- Aufnahme aussagekräftiger Daten und Formalisierung
- Modellierung aller Untersysteme mit standardisierten Methoden und Validierung dieser Modelle mit gemessenen Daten. Diese Modelle müssen alle Ressourcen enthalten, die zu den Bewertungskriterien beitragen.
- Integration der Untermodelle in ein einheitliches ausführbares Gesamtmodell. Das Referenzmodell stellt dabei eine formale Beschreibung der zu entwickelnden service-orientierten Architektur dar.
- Optimierung des Referenzmodells nach den Bewertungskriterien zur Bestimmung einer validierten ausführbaren Spezifikation der Gesamtarchitektur

Abschnitt 1.3 erläutert die Aufteilung der Arbeiten in Pakete sowie die Einbettung in das Gesamtprojekt. Auf die detaillierten Ziele der Arbeitspakete gehen die Unterabschnitte von 2.1 ein.

1.2 Voraussetzungen der Durchführung

Das Fachgebiet System- und Software-Engineering (SSE) (und sein Vorgänger System- und Steuerungstheorie) der Technischen Universität Ilmenau ist seit mehr als 15 Jahren erfolgreich auf dem Gebiet des Entwurfes vernetzter eingebetteter Systeme in den Bereichen Avionik, Automotive sowie fehlertoleranter sicherheitskritischer Systeme tätig. Die Forschungsarbeiten beschäftigen sich insbesondere mit der Methodik, dem modellbasierten Entwurf sowie der Konzeption geeigneter Werkzeuge und Toolketten für die Entwicklung komplexer vernetzter Systeme. Die Konzentration liegt dabei insbesondere auf den frühen Entwicklungsphasen, wo heute mehr als 60% aller kritischen Fehler auftreten und typischerweise mehr als 85% der kritischen Entwurfsentscheidungen getroffen werden können.

Ein weiterer Aspekt der Forschung konzentriert sich darauf, die Lücke zwischen Entwurf auf Systemebene und Implementierung zu überwinden. Die entwickelten Methoden, Entwurfs- und Entwicklungstools integrieren den Entwurfsprozess von der Konzeptentwicklung bis zur Implementierung. SSE unterstützt die Mission Level Design GmbH bei der Entwicklung des Softwaresystems MLDesigner [AIS08, S00], das weltweit für die Entwicklung komplexer Systeme eingesetzt wird. Mit der Fachgebietsausgründung der Mission Level Design GmbH existiert ein strategischer Kooperationspartner, mit dem die Ergebnisse der Forschungsarbeiten in zahlreichen erfolgreichen Industrieprojekten gemeinsam umgesetzt werden. Beispielprojekte wurden im Bereich der Vernetzung von elektronischen Komponenten im Flugzeug bearbeitet. Zu nennen sind hier die Projekte zur Realisierung einer Entwicklungsumgebung zur Optimierung vernetzter Avionik-Systemarchitekturen, Arbeiten im Rahmen des Fiber Optical Virtual Network Design sowie Arbeiten zum Engineering Process. Aktuell ist das Team integriert in die Projekte LuFo @MOST SO4A, CabWiSe und DAPHNE. Wichtige Problemstellungen, die auch hier gelöst werden, betreffen den modellbasierten Entwurf, die Entwicklung von Toolketten und die Berechnung optimaler Architekturen. Der nach der Umstrukturierung hinzugewonnenen Partner Philotech hat große Erfahrungen in den Gebieten System-Engineering, Software und technische Logistik mit Simulation im Avionik-Bereich.

1.3 Planung und Ablauf des Vorhabens

Abbildung 1 zeigt die Themen und Arbeitsteilung des Teilvorhabens SO4A in der Planung vom Beginn des Projektes. Die Technische Universität Ilmenau leitet das Arbeitspaket 150 Technologien, Modellierung und Simulation, und leistet hauptsächliche Beiträge in den Unterarbeitspaketen 151 Modellierung u. Simulation von Informations-, Material- und Kontrollfluss und 152 Datenanalyse. Im Abschnitt 2.1 werden die Arbeitspakete und ihre Ergebnisse im Einzelnen beschrieben.

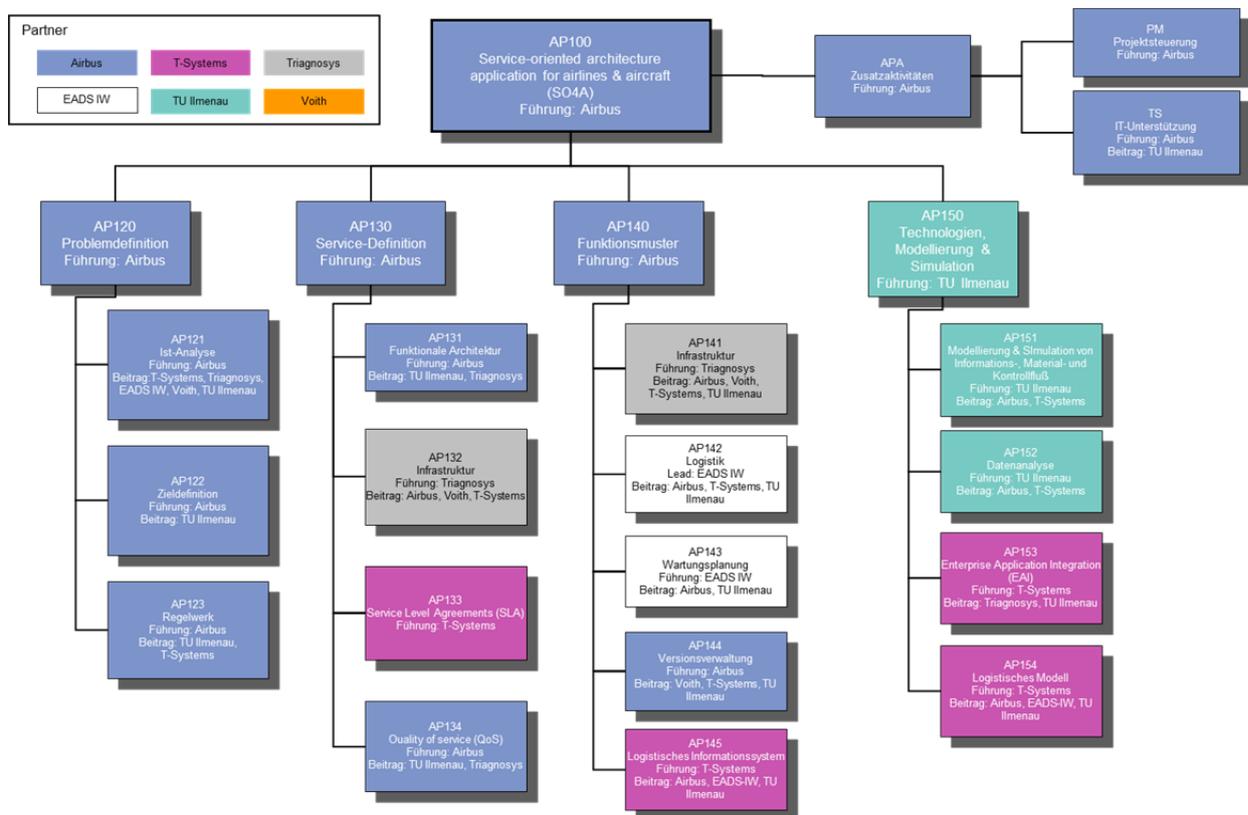


Abbildung 1: Arbeitspakete des Teilprojektes SO4A

Die von der TU Ilmenau bearbeiteten Arbeitspakete hatten die folgenden Ziele:

AP151 Modellierung & Simulation von Informations-, Material- und Kontrollfluss

- Weiterentwicklung des Modellkonzeptes der serviceorientierten Architektur, der Teilmodelle z.B. zur Abbildung der Logistik, Ressourcen und Prozesse, der wartungsbezogenen Teilmodelle (Wartungsobjekte, Ressourcen und Dienste) mit dem Ziel, segmentübergreifende (Aircraft segment, Communication segment, Ground segment), durchgängig integrierte und optimale Serviceprozesse zu entwickeln.
- Umsetzung der entwickelten Modellkonzepte (MLDesigner-Modell), Entwicklung von Modell-Bibliotheken
- Entwicklung der Modellumgebung (Interfaces, GUI, Präsentation, Toolkopplungen, Lastszenarien, Visualisierung, Datenbankkopplung)
- Simulationen zur Validierung und Optimierung des Modellkonzeptes (Architecture blueprint) auf der Basis zu modellierender Lastszenarien, sowie Performance- und Optimierungszielen
- Entwicklung der Modellumgebung zu einem integrierenden Systems, der Entwicklungsumgebung/Simulator und der Funktionsmuster (Hardware-in-the-Loop Simulation).
- Dokumentation des methodischen Vorgehens (Vorgehensmodell) und des validierten Architecture blueprint (Referenzmodell). Evaluation, Auswahl, Anpassung bzw. Erweiterung und Integration von Werkzeugen zur Unterstützung des methodischen Vorgehens.

AP152 Datenanalyse

- Analyse und Aufbereitung von Wartungs- und Logistikdaten in Kooperation mit den Verbundpartnern und Airbus SMO (Hamburg-Fuhlsbüttel).
- Formalisierung der Daten für die Integration in das Gesamtsystemmodell, Definition von Ressourcenmodellen und Datenstrukturen
- Entwicklung generischer Modelle, um Benutzungsfreundlichkeit und Verwertbarkeit wesentlich zu verbessern
- Analyse zur Integration der Modellumgebung in eine spätere Unternehmensplattform

AP153 Enterprise Application Integration

Das Arbeitspaket 153 sollte ursprünglich von T-Systems geleitet und hauptsächlich bearbeitet werden. T-Systems war nach der Umstrukturierung aber nur noch in geringem Maße an der Projektdurchführung beteiligt. Die für die Fortführung des Projekts auf der Ebene der Modellierung und Simulation wichtigen Teilaufgaben wurden auf den Unterauftragnehmer Philotech übertragen. Aufgabe des Arbeitspaketes ist es, eine Architektur unter Integrationsaspekten zu bestimmen, die die Anforderungen von SO4A/@MOST hinsichtlich des serviceorientierten Ansatzes des Gesamtprojektes erfüllt.

- Definition von Anforderungen und Bewertungskriterien für Integrationstechnologien, die eine Implementierung der vorgesehenen service-orientierten Architektur aus Sicht der Modellierung und Simulation befördern
- Identifizierung und Auswahl benötigter bzw. geeigneter Komponenten im Sinne einer SOA/BPM Referenzinfrastruktur
- Durchführung einer Studie zur Evaluierung und Definition von universellen Datenaustauschwegen und -standards für die @MOST- Architektur
- Durchführung einer Machbarkeitsstudie zur Integration von weiteren neuen oder bereits vorhandenen Systemen und Produkten des Logistik- bzw. Luftfahrtmarktes
- Definition einer durchgängigen service-orientierten Architektur zur Anbindung und Integration anderer Teilsysteme und Unternehmensanwendungen im @MOST und SO4A Gesamtumfeld sowie des Partnernetzwerkes.
- Erarbeitung einer Architekturbeschreibung, wie sich die Integrationsschicht in SO4A mit dem @MOST Gesamtprogramm und dem Partnernetzwerk verbindet.

AP154 Logistisches Modell

Das Arbeitspaket 154 sollte ursprünglich ebenfalls von T-Systems bearbeitet werden und wurde nach der Umstrukturierung durch Philotech übernommen. Dieses Arbeitspaket dient der Erstellung von Modellen, die den Bereich der Logistik abdecken und dann als Basis für die Implementierung der geplanten Logistikplattform verwendet werden sollen.

- Definition relevanter Logistikfunktionen aus Sicht der Modellierung und Simulation, insbesondere aus Sicht der ausführbaren Modelle, Definition von Modellierungsschnittstellen
- Modellierung von Diensten zur integrierten logistischen Unterstützung im Bodensegment als Grundlage für die Entwicklung eines Service-orientierten Logistischen Informationssystems (SoLIS) im Rahmen der @MOST/SO4A Gesamtarchitektur
- Definition aller relevanten Bereiche der integrierten logistischen Unterstützung inkl. Beschreibung der entsprechenden Entitäten, Funktionen und Komponenten sowie des internen Workflows für dieses Service-orientierte Logistische Informationssystem
- Definition von Diensten/Funktionen zur Kopplung mit anderen Disziplinen und Systemen im Rahmen der @MOST Gesamtarchitektur sowie als Input für die Erstellung eines dynamischen ausführbaren Modells für die Simulation, da sich nur so wesentliche Aussagen für den operationellen Flottenbetrieb und der diesbezüglichen Optimierungspotentiale durch die SO4A-Ansätze ableiten und zeigen lassen.

Sonstige Arbeitspakete

Zuarbeiten mit geringeren Umfängen leistet die Technische Universität Ilmenau innerhalb der Arbeitspakete AP140 Funktionsmuster für die Arbeitspakete mit den Unterthemen 141 Infrastruktur, AP142 Logistik, AP143 Wartungsplanung, AP144 Versionsverwaltung und 145 Logistisches Informationssystem, sowie IT-Unterstützung.

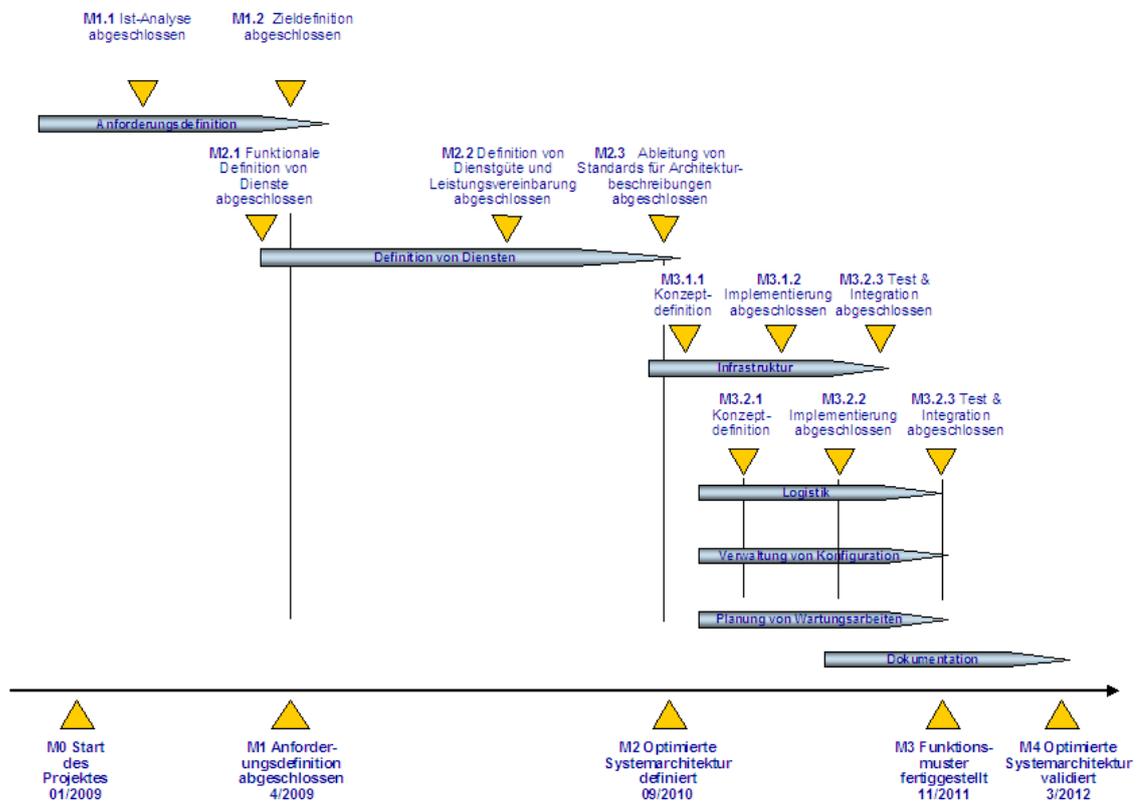


Abbildung 2: Ursprünglicher Zeitplan der Arbeiten

Der Zeitplan der Arbeiten und Meilensteine ist in Abbildung 2 dargestellt. Aufgrund der Projektumstellung verzögerte sich das Projekt um einige Monate. Die Einbettung der Unterauftragnehmer Philotech und Mission Level Design können den Angeboten entnommen werden. Abschnitt 2.2 enthält Informationen zum Mitarbeiterereinsatz.

1.4 Stand der Technik

Der Entwurf auf der Basis ausführbarer Modelle (model-based design, [F09, S04, Z07]) hat das Potential, die Produktivität des Systementwurfes wesentlich zu erhöhen. Klassische Methoden wie UML-Diagramme (Beschreibung der technischen Systeme) oder ARIS Darstellungen (Beschreibung der Prozesse) können die Abläufe eines solchen Systems beschreiben. Sie enthalten jedoch keine Ressourcenbeschreibungen und sind für die Kostenminimierung nicht geeignet. Das ausführbare Modell muss – beginnend in den frühen Entwurfsphasen – auf abstrakter Ebene entworfen und schrittweise verfeinert werden [FS11b, JWZ1]. Es versteckt Systemdetails, muss jedoch auch alle notwendigen Subsysteme und Komponenten auf allen Spezifikationsebenen repräsentieren. Auf dieser Basis eines ausführbaren Systemmodells können Systemleistungsbewertungen durchgeführt werden [MFS10, Z07].

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. 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 Modellierung des Systems mit seinen Subsystemen und Komponenten über alle Abstraktionsebenen bleiben interne Systemdetails auf den höheren Modellebenen 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 und vernetzter Systeme müssen verschiedene Berechnungsmodelle [PZ10] 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.

Von besonderer Bedeutung ist die Integration des top-down und bottom-up Entwurfes [S00] 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 Sub-

systeme und Komponenten modelliert und zu einem Gesamtsystem kombiniert. So können Erfahrungen aus vergangenen Systementwicklungen in den Entwurf eingebracht werden. 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.

Zurzeit werden derart komplexe Systeme, wie es zum Beispiel ein Avionik-System [SF08] darstellt, rein anforderungsorientiert (requirements-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. Das führt zu lokalen Lösungen, die schwer zu integrieren sind. Nach dem Erreichen eines akzeptierten Entwurfes erfolgen Validierung und Test des Gesamtsystems. Fehler im Designprozess können daher oft erst sehr spät entdeckt werden.

Der Entwurf auf der Basis ausführbarer Modelle hat dagegen das Potential, die Produktivität des Systementwurfes wesentlich zu erhöhen [Z03]. Im Zentrum des gesamten modellbasierten Entwurfsprozess steht ein ausführbares Modell des zu entwerfenden komplexen Systems. 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 [ZSZ10, SZM10a]. Während des gesamten Entwurfsprozesses kann geprüft werden, ob der Entwurf den Anforderungen entspricht und welche Auswirkungen eine vorgeschlagene Änderung in den Systemanforderungen bewirkt. Der modellbasierte Entwurf [F09] hat gegenüber dem anforderungsorientierten Entwurf unter anderem 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 Entwurfsebenen verwendet.
- Nichtfunktionale Anforderungen können gleichfalls modelliert und validiert werden.

MLDesigner [S00, AIS08] ist ein Modellierungs- und Simulationswerkzeug, welches diese Anforderungen unterstützt. Es wird von der am Projekt beteiligten Firma Mission Level Design weiterentwickelt.

1.5 Zusammenarbeit mit anderen Stellen

Eine ausführliche Beschreibung des Verbundvorhabens @MOST-G mit seinen beiden Teilvorhaben SWAN und SO4A wird durch den Verbundführer Airbus vorgenommen. Die Technische Universität Ilmenau ist Verbundpartner innerhalb des Verbundvorhabens @MOST-G, Teilvorhaben SO4A.

Das Fachgebiet System- und Software Engineering der TU Ilmenau ist Know-How-Träger auf den Gebieten modellbasierter Entwurf sowie Modellierung und Simulation auf Architekturebene für komplexe Systeme. In enger Kooperation mit der Mission Level Design GmbH hat sich auf diesem Arbeitsgebiet im letzten Jahrzehnt eine strategische Partnerschaft entwickelt. Das Modellierungstool MLDesigner, entwickelt durch die Firma Mission Level Design GmbH ist prädestiniert für die Lösung der Projektaufgaben im Verbundvorhaben @MOST-G SO4A. MLDesigner ist ein grundlegendes Werkzeug für Airbus, das in zahlreichen Projekten für den modellbasierten Entwurf eingesetzt wird. Auf Grund der Kompetenzen der Mission Level Design GmbH auf dem Gebiet der modellbasierten Entwicklung sowie der Werkzeugentwicklung sollten von ihr insbesondere die Analyse und Entwicklung von Schnittstellenfunktionen zu anderen Werkzeugen sowie die Entwicklung und Implementierung von Basismodulen und Modellbibliothekskomponenten übernommen werden.

Praktisch erfolgte die Zusammenarbeit mit Mission Level Design (MLD) durch die räumliche Nähe sehr einfach und in enger Abstimmung. Die bei MLD im Projekt arbeitenden Mitarbeiter waren mehrere Tage in der Woche am Standort der TU tätig, wo genug Platz für die gemeinsame Arbeit am Projekt vorhanden war. Abstimmungen mit Philotech erfolgten über Telefonate, Arbeitstreffen und den Austausch von Dokumenten.

Neben der TU Ilmenau und ihren Unterauftragnehmern waren für den Kontext unserer Arbeiten besonders die Partner Airbus, TriaGnoSys und EADS Innovation Works wichtig. Mit ihnen gab es Arbeitstreffen, Telefonkonferenzen, Dokumentenaustausch sowie ein zentrales Projekt-Wiki, in dem inhaltliche und organisatorische Informationen abgelegt wurden.

Als Verbundführer war die Zusammenarbeit mit Airbus Operations von besonderer Bedeutung. Die Ergebnisse und weitere Ausrichtung der Arbeiten wurden besprochen. Darüber hinaus wurden die jeweils aktuellen an der TU Ilmenau entwickelten Modelle bei Airbus getestet und Erweiterungen ergänzt. Dazu gehörte vor allem die Ersatzteilbedarfsplanung (initial provisioning). Ein bei Airbus entwickeltes Werkzeug kann ausgehend von sogenannten recommended spare parts list berechnen, wo die Ersatzteile gelagert werden sollten. Diese Werte werden im Flottenmodell benutzt. Auch die Modelldokumentation wurde gemeinsam durchgeführt.

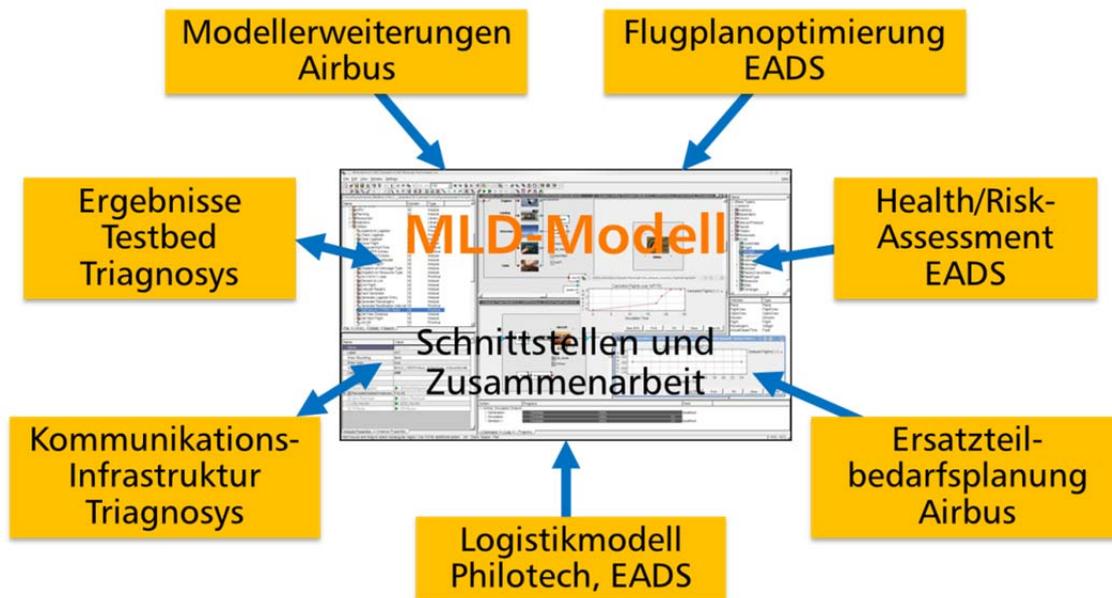


Abbildung 3: Fachliche Zusammenarbeit im Projekt

Aus Sicht der Arbeiten der TU Ilmenau war das zu entwickelnde Modell der zentrale Punkt, an dem sich die Zusammenarbeit orientierte. Das umgebende Softwarewerkzeug MLDesigner und das entwickelte Modell bilden eine Integrationsplattform für den Entwurf, ein „virtuelles Labor“ für Untersuchungen an zukünftigen Architekturvarianten. Abbildung 3 zeigt symbolisch die Zusammenarbeit mit den Projektpartnern rund um das Modell.

Als Beispiel für die enge Zusammenarbeit auf konzeptioneller und technischer Ebene wurde durch die TU Ilmenau das von Philotech entwickelte Flugplanungskonzept in MLDesigner als Planungsmodell umgesetzt und damit verschiedene Szenarien simuliert. Dabei wurden unterschiedliche Möglichkeiten zur Neuplanung eines abgesagten Fluges implementiert. Eine Datenbank wurde hinterlegt, um beispielsweise initial provisioning (mit Daten von Airbus) zu berücksichtigen. Aus simulierten Daten von TriaGnoSys wurde eine Kommunikationsschnittstelle definiert und in das Modell implementiert. In einer Präsentation des Projektfortschritts konnte so dem Projektträger eine Simulation gezeigt werden, bei der die Ersatzteilbeschaffung per Flugzeug oder Auto erfolgt. Dabei zeigen die Ergebnisse, dass die Versorgung auf der Straße aufgrund des beschränkten Flugplans der Airline besser ist. Abschließend wurde das Modell anhand einiger Beispiele live präsentiert.

Die Zusammenarbeit mit Philotech betraf vor allem den Ablauf der Wartung und Ersatzteillogistik (AP154, siehe Abschnitt 2.1.5). Die Aufgabenstellung sah eine beratende Tätigkeit vor, bei der die logistischen Prozesse im Rahmen eines Airline-Betriebes erkannt und für die spätere

Modellierung aufgearbeitet werden sollten. Der Schwerpunkt dieser Arbeit sollte dabei auf drei Themengebieten liegen:

- Nach welchen Kriterien finden die Abläufe des Spare Part Repair Cycles statt?
- Wie kann die Kapazität von Lagerhäusern angemessen realistisch simuliert werden?
- Welche Kriterien werden benötigt um vorgegebene Angaben zum Initial Provisioning im Modell korrekt abzubilden?

Nach dem Kick-Off Meeting nach der Projektumstrukturierung 2011 bei Airbus in Buxtehude wurden diese Fragestellungen in einem ersten Workshop in Anfang Februar in Ilmenau erarbeitet. Die Themen wurden dabei jeweils als UML-Diagramme visualisiert werden um eine möglichst schnelle Umsetzung im bereits existenten MLDesigner-Modell zu gestatten.

Durch die Komplexität des Projektes und die vielen beteiligten Arbeitsgruppen wurden dafür zu Beginn der Arbeit verschiedene Termine mit den unterschiedlichen Teams durchgeführt, um so einen möglichst genauen Überblick über den aktuellen Kenntnisstand zu bekommen und die unterschiedlichen Anforderungen der Gruppen an das Modell zusammenzuführen. So konnte über die Zusammenarbeit mit EADS IW und mehrere Gespräche mit Airbus direkt die Aufgabenstellung noch einmal konkreter eingegrenzt werden. So verlangt der Projektrahmen, dass die Simulation eine richtige Balance zwischen zu hoher Komplexität auf der einen und angemessener Realitätsnähe auf der anderen Seite findet. Nur so lässt sich gewährleisten, dass die Ergebnisse im vorhandenen Zeitrahmen umgesetzt werden können und auch gegen reale Vorgaben validiert werden können.

Zu Bearbeitung der drei genannten Kernfragen war es notwendig, diejenigen Parameter eines Ersatzteils zu identifizieren, die maßgeblichen Einfluss auf die logistischen Abläufe haben. Hierfür wurde auf die bei Philotech vorhandene Expertise im Bereich der Logistik und des Ersatzteil Managements zurückgegriffen, aber auch wie bei der Analyse der zentralen Fragestellungen, auf das Wissen anderer Projektgruppen eingegangen. Darüber hinaus fand ein Informationsaustausch mit der Hochschule für Angewandte Wissenschaften in Hamburg statt.

Später fand ein weiteres Treffen mit den unterschiedlichen Arbeitsgruppen im Rahmen der Vorstellung der Arbeitsergebnisse vor dem DLR in München statt. Die benannten Themen wurden in Form von UML-Diagrammen visualisiert und die Arbeitsergebnisse im Rahmen eines Workshops Anfang Mai 2011 vorgestellt und durch die Teilnehmer diskutiert. Im Anschluss wurden bestimmte Änderungswünsche durch Airbus in weiteren Meetings geklärt, um den Entwurf für die spätere Modellierung entsprechend anzupassen. Philotech erstellte das Wartungs- und Lo-

gistikmodell (siehe Abschnitt 2.1.2.6), das dann durch die Ilmenauer Arbeitsgruppe in das Gesamtmodell integriert wurde.

Die Zusammenarbeit mit dem Projektpartner TriaGnoSys beschäftigte sich thematisch mit der Untersuchung von möglichen Kommunikationsnetzwerken. Dazu gehört die Definition der Anforderungen und der Herstellung einer Kommunikationsschnittstelle mit MLDesigner. Der Begriff SWIM wird dabei in diesem Bereich häufig verwendet. Die Nutzung von SWIM erfolgt durch die klassische Abfolge subscribe / publish / notification. Dazu wurde eine sogenannte Testbed-Architektur definiert, in der als Beispiel unterschiedliche Flugzeuge an diesem Netzwerk teilhaben. Innerhalb des SWIM-Umfelds wurden nur die Wartungsaspekte berücksichtigt. Das Projekt wurde auf einem Rechnersystem implementiert, Anwendung des Systems gestartet und anschließend die Kommunikationsschnittstellen überwacht, um einen Überblick über die Nutzung (Ressourcen, Datenrate etc.) zu bekommen. Das Software-Testbed wurde mit Anforderungsdaten belastet, die aus dem Ilmenauer Flottenmodell generiert wurden. Dafür wurde in Telefonkonferenzen und ein Austauschformat für die Daten definiert. Die sich ergebenden Verzögerungszeiten der Kommunikationsschnittstelle im Testbed wurden von TriaGnoSys bestimmt und diese Daten dann wiederum an Ilmenau gegeben, wo sie Eingang in das vereinfachte Modell der Kommunikations-Infrastruktur fanden. Das Modell wird so realistischer, und der Einfluss der Kommunikation auf den Gesamtprozess kann simuliert werden.

Mit dem Projektpartner EADS IW fand Austausch zum Thema Flugplanoptimierung statt. Außerdem wurden Erfahrungen mit dem bei EADS entwickelten Health/Risk-Assessment Modul sowie zu Logistikprozessen (siehe oben, mit Philotech) genutzt. Die von EADS IW entwickelte Erweiterung des Wartungsmodells um ein Abnutzungsklassen- oder Alterungs-Modell wurde im MLDesigner-Modell implementiert. Damit ist es möglich, die Wartung für eine Komponente bei bekannter Alterung so zu optimieren, dass eine pünktliche Vorhersage des Versagens möglich ist.

2. Darstellung der Wissenschaftlich-Technischen Ergebnisse

Im Folgenden werden die inhaltlichen Ergebnisse des von der TU Ilmenau in Zusammenarbeit mit den Unterauftragnehmern Mission Level Design und Philotech bearbeiteten Teilprojektes ausführlich beschrieben.

2.1 Detaillierte Darstellung der Ergebnisse

Dieser Abschnitt stellt für die im Teilprojekt bearbeiteten Pakete AP150 bis 154 jeweils Zielstellung und Ergebnisse dar.

2.1.1 AP150 – Modellierung und Simulation

Das Arbeitspaket AP150 ist die übergeordnete Einheit für alle Modellierung und Simulation betreffenden Pakete AP151 bis AP154. In diesen werden die eigentlichen spezifischen Arbeiten ausgeführt, wie in den folgenden Abschnitten 2.1.2 bis 2.1.5 jeweils beschrieben.

Die grundlegenden Ziele des Arbeitspakets beinhalteten:

- den modellbasierte Entwurf (Modellierung und Simulation) und die Verifikation und Validierung einer dienstorientierten Architektur
- die Dokumentation der entwickelten Basistechnologien im Kontext Modellierung und Simulation

Das zu entwickelnde Gesamtsystemmodell (Referenzmodell) sollte die Realisierung und Validierung der Funktionsmuster unterstützen. Dabei werden Schnittstellen definiert, wie die Funktionsmuster in die Simulation integriert werden können. Notwendig ist der direkte Austausch von Daten (reale Messwerte, Simulationsdaten) für die Validierung der avisierten SO4 Architektur und des zu entwickelnden Referenzmodells.

Des weiteren wurden Anforderungen für die Unterarbeitspakete 151-154 definiert und die Ergebnisse der Partner validiert. Im Rahmen der Koordinierung und Leitung dieses Arbeitspaketes fielen daneben administrative Arbeiten wie Planung, Projektkontrolle, Steuerung, Berichtswesen und technische Unterstützung an.

2.1.2 AP151 – Modellierung und Simulation von Informations-, Material-, und Kontrollfluss

Ziel dieses Arbeitspaketes sind die Modellkonzeption, Erstellung einer Modellbibliothek, Entwicklung der Modellumgebung, Simulation und Optimierung der Architektur und die Integration der Funktionsmuster in die Simulation für die angestrebte serviceorientierte Architektur. Innerhalb dieses Arbeitspaketes fiel der Hauptteil der Arbeitsaufwände für die TU Ilmenau zusammen mit ihrem Unterauftragnehmer und Partner Mission Level Design an.

Aus den Ergebnissen der Ist- und Datenanalyse werden ein Vorgehens- und ein Referenzmodell entwickelt. Letzteres ist eine formale Beschreibung typischer segmentübergreifender Abläufe (Segmente: Flugzeug, Kommunikation, Boden; Abläufe: Logistik, Wartungsplanung, Konfigurationsverwaltung) innerhalb der Architektur. Auf Basis dieser Erkenntnisse, Expertenwissen und der Grundlage erzielter Simulationsergebnisse wird das Referenzmodell ständig weiterentwickelt, verifiziert und validiert. Das Referenzmodell, das aus Struktur-, Prozess- und Ressourcenmodellen besteht, ist Grundlage für die spätere Implementierung der Architektur. Die wesentlichen Komponenten des Gesamtsystems werden mit dem Modellierungs- und Entwurfssystem MLDesigner [S00, AIS08, S04] beschrieben. Ausführbare Spezifikationen erlauben die Simulation, Analyse, Validierung und Verifikation der zu entwickelnden Architektur. MLDesigner als integrierendes Werkzeug stellt Schnittstellen und Bibliotheken zur Verfügung. Aufgaben im Einzelnen sind vor allem:

- Entwicklung des Modellkonzeptes, wartungsbezogener Teilmodelle (Objekte, Ressourcen u. Dienste), von Teilmodellen zur Abbildung der Logistik, Ressourcen und Prozesse, des Modells der dienstorientierten Architektur
- Modellimplementierung mit MLDesigner und Entwicklung von Bibliotheken, Entwicklung der Modellumgebung (Schnittstellen, GUI, Visualisierung, Toolkopplungen)
- Modellierung von Lastszenarien, Untersuchung von Performance- und Optimierungszielen
- Simulationen zur Validierung und Optimierung des Modellkonzeptes

Im Ergebnis des Arbeitspaketes wurde ein den Anforderungen entsprechende umfangreiches Modell einer Luftfahrtflotte entwickelt, in dem erstmals alle genannten Segmente und Vorgänge integriert betrachtet werden. In den folgenden Abschnitten 2.1.2.1 bis 2.1.2.4 werden die hierarchisch verfeinerten Ebenen des entwickelten Modells im Überblick dargestellt, so wie sie in einem Top-Down-orientierten Entwurfsprozess erstellt werden würden. Anschließend wird auf

Teilmodelle eingegangen, die die Aspekte Planung, Wartung und Logistik, sowie Kompensationsalternativen bei Ausfällen von Flugzeugen behandeln. Ein Vorversionen des Modells und beispielhafte Ergebnisse der Bewertung durch Simulation wurden in [ZSZ10, SZM10b] veröffentlicht.

2.1.2.1 Ein Hierarchisches Airline-Modell

Das zentrale Simulationsmodell einer Airline ist aufgrund seiner Komplexität hierarchisch und modular unterteilt. Abgebildet wurden die folgenden fünf Ebenen:

- L0 – Airline Airline
- L1 – Fleet Planning, Fleet, Maintenance, Engineering
- L2 – Aircraft Aircraft
- L3 – Component Engines, Landing Gear, Structure, Systems, Cabin
- L4 – Assembly Galleys, SeatRows, CIDS, Water and Waste

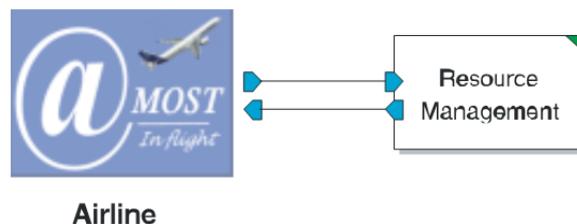


Abbildung 4: Top-level Airline Simulationsmodell

Abbildung 4 zeigt das entsprechende MLDesigner-Modell auf der obersten Ebene, die sogenannte System Airline Simulation. Das Hauptmodul ist „Airline“.

In der zweiten Hierarchieebene des Hauptmoduls Airline befinden sich die vier Module Planning, Fleet, Maintenance, und Engineering, siehe Abbildung 5. Über globale Variablen können Kosten für den Betrieb berechnet werden, DB_HANDLE organisiert die Anbindung an die Datenbank (siehe auch Abschnitt 2.1.4.2).

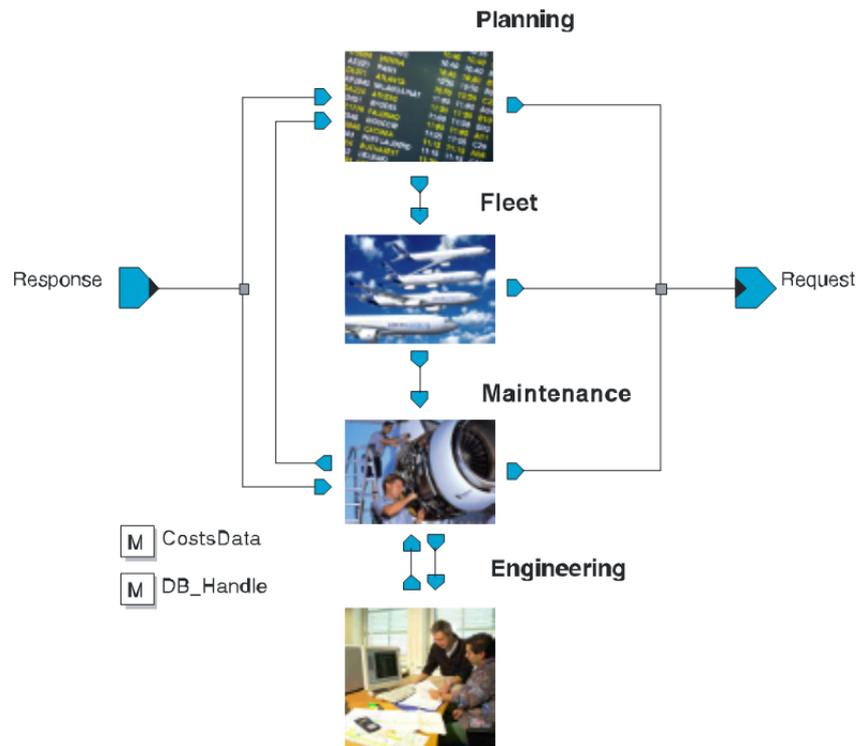


Abbildung 5: Airline-Modell

Ist der Flug geschlossen, wird der Flieger an das Modul Maintenance übergeben. Im Falle von Triebwerksfehlern (Engines) und Fehlern, deren Behebung einen Hangar benötigen, wird das Flugzeug in einem Hangar zur Reparatur abgestellt. Die Aufenthaltsdauer ist über Werte in einer Datenbank abgebildet. Sie verändert sich, falls noch Logbucheinträge abgearbeitet werden müssen. Verfügt der aktuelle Flughafen nicht über eine Ressource Hangar, dann ist der Flieger „Blocked on Ground“ und steht zunächst nicht mehr zu Verfügung. Für alle anderen Fälle wird über das RectificationInterval entschieden, ob der Fehler sofort zu beheben ist oder verschoben werden kann. Falls eine Reparatur durchgeführt werden muss, wird eine benötigte Ersatzteilliste aufgestellt und gegeben falls Bestellungen ausgelöst. Gleichzeitig wird das notwendige Mechaniker Team zusammengestellt und die Reparatur durchgeführt. Verzögerungen können entstehen, falls nicht ausreichend Ressourcen verfügbar sind.

2.1.2.2 Flottenmodell

Es sei nun die dritte Hierarchieebene von Airline Simulation betrachtet. Das Modul Fleet unterteilt sich in diesem Modell nur in das Modul Aircraft, s. Abbildung 6. Über die Clock und Synchronize Primitiven kann die Simulationsgeschwindigkeit eingestellt werden, um zum Beispiel externe Werkzeuge zu synchronisieren. Der Parameter SyncTimeScale auf der Ebene Toplevel definiert die Zeit in Sekunden pro Simulationsschritt. Die Anzahl der Simulationsschritte ist im Parameter RunLength in Stunden angegeben.

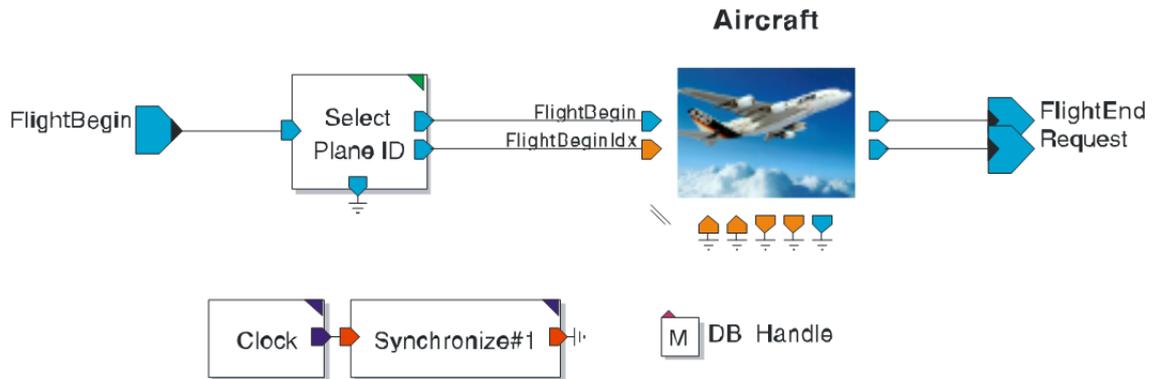


Abbildung 6: Fleet Model

Das Modul Maintenance unterteilt sich in die Hauptmodule MRO und Facility. Der Block „On the way to Hangar“ stellt hier nur eine Zeitverzögerung dar, welche die Überführung des Flugzeuges vom Gate zum Hangar modelliert. Das Modul Engineering ist noch nicht weiter verfeinert.

2.1.2.3 Aircraft-Modell

In der in Abbildung 7 dargestellten vierten Hierarchieebene des Modells unterteilt sich das Modul Aircraft in die Hauptmodule Engines, Landing Gear, Structure, Systems und Cabin.

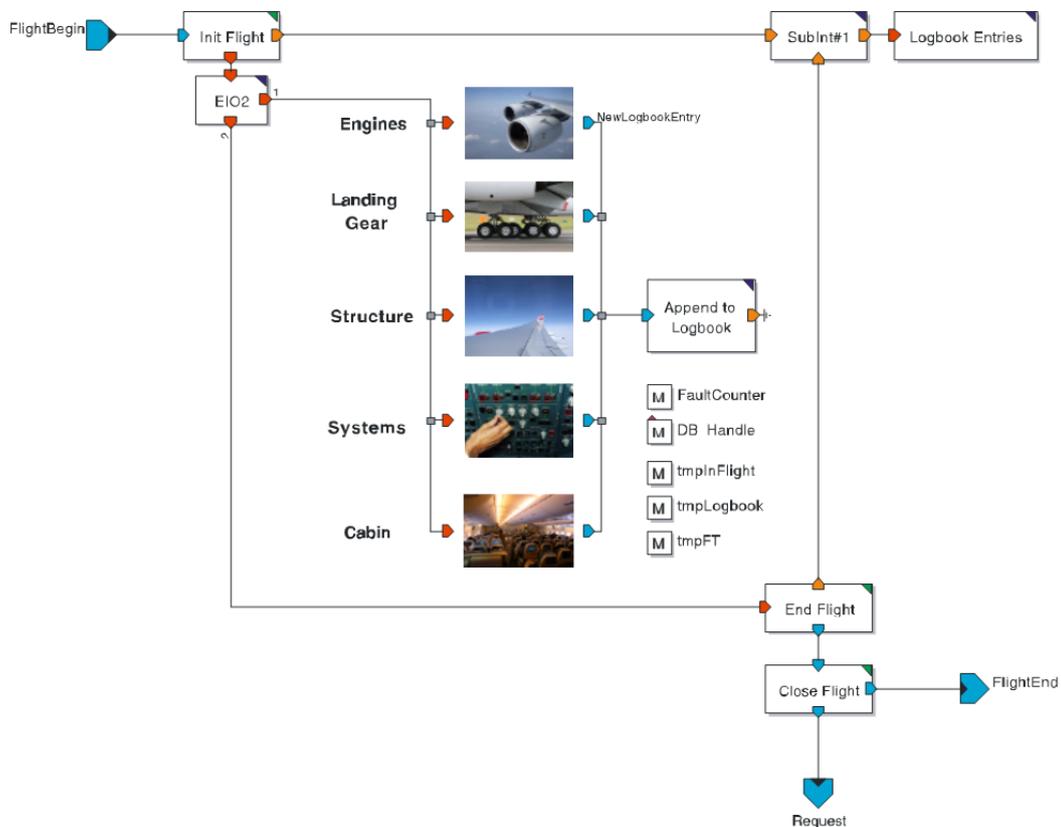


Abbildung 7: Aircraft Model

In den Hauptmodulen (außer Cabin) sind Fehlergeneratoren hinterlegt, d.h. während eines Fluges kann es dort zu Ausfällen und Defekten kommen. Einstellbar sind in den Fehlergeneratoren die MTBF – Mean Time Between Failure und das RectificationIntervalDistributions. Die MTBF gibt den durchschnittlichen, zeitlichen Abstand zwischen zwei Fehlern in Stunden an. Allgemein werden Fehler nach ihrer Schwere bzw. Dringlichkeit unterschieden:

- Kategorie A (1): Der Fehler ist in einem vorgeschriebenen Intervall zu beheben,
- Kategorie B (2): Der Fehler ist in den nächsten 3 Kalendertagen zu beheben,
- Kategorie C (3): Der Fehler ist in den nächsten 10 Kalendertagen zu beheben,
- Kategorie D (4): Der Fehler ist erst in den nächsten 120 Kalendertagen zu beheben.

Im RectificationIntervalDistributions wird in Arrayform die Wahrscheinlichkeitsverteilung angegeben.

Das Modul Facility unterteilt sich in der vierten Ebene nur in den Modul Hangar. Nur Flugzeuge mit Triebwerksschaden (Engines) werden im Hangar abgestellt.

2.1.2.4 Cabin-Modell

Das Hauptmodul Cabin ist in der fünften Hierarchieebene unterteilt in die Komponenten Galleys, SeatRows, CIDS und Water and Waste, siehe Abbildung 8. In den einzelnen Komponenten sind Fehlergeneratoren hinterlegt, die genau wie bereits genannt über MTBF und RectificationIntervalDistributions einstellbar sind. Außerdem ist die eigentliche CORBA-Schnittstelle über das Modul Create_PFR_Entries modelliert.

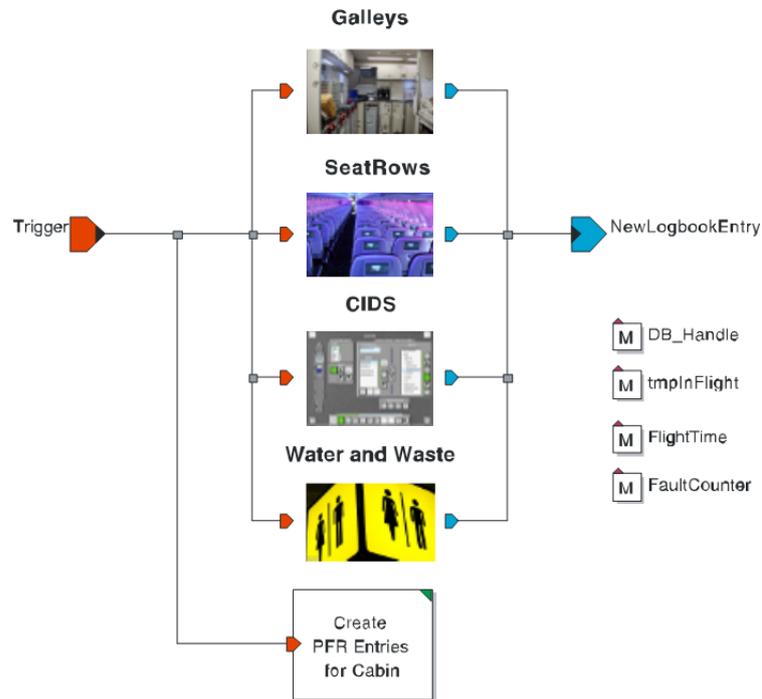


Abbildung 8: Cabin Model

2.1.2.5 Planungsmodell

Neben den im Überblick beschriebenen Modellen, die die Struktur und hierarchische Aufteilung einer Airline beschreiben, gibt es dynamische Abläufe, die mit eher prozessorientierten Modellen abgebildet werden. Da es sich um Verhalten von Ressourcen handelt, sind diese Vorgänge auch bei den entsprechenden Strukturelementen modelliert. Das erhöht die Übersichtlichkeit des komplexen Gesamtmodells.

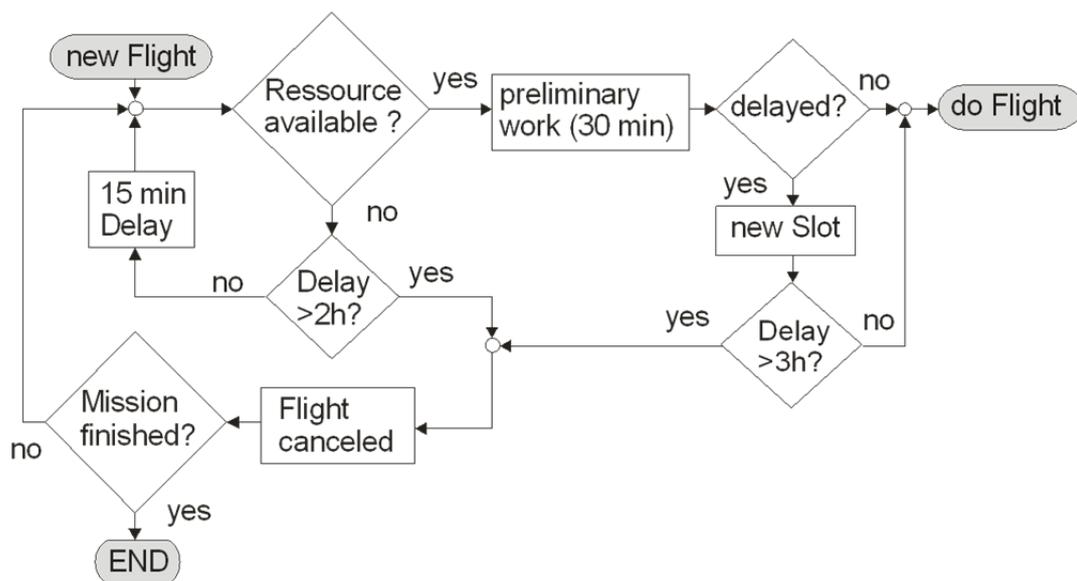


Abbildung 9: Beispielhafter Ablauf einer Flugplanung

Ein wichtiger Vorgang, der die Effizienz der Airline beeinflusst, ist die Flugplanung. Sie entscheidet darüber, welche Ressourcen (Aircraft, Besatzung) wann welchen Flug durchführt. An diesem Modellteil können verschiedene Planungsverfahren implementiert werden, um ihren Einfluss auf Kosten, Effizienz und Zuverlässigkeit des Flugbetriebs zu untersuchen. Abbildung 9 zeigt den groben Ablauf einer Flugplanung. Die oberste Seite der konkreten Umsetzung in einem MLDesigner-Modell ist in Abbildung 10 dargestellt. Dies ist auch ein Beispiel für die Verfeinerbarkeit komplexer Abläufe und ihr Verständnis durch die grafische Darstellung und aussagekräftige Bezeichner in MLD.

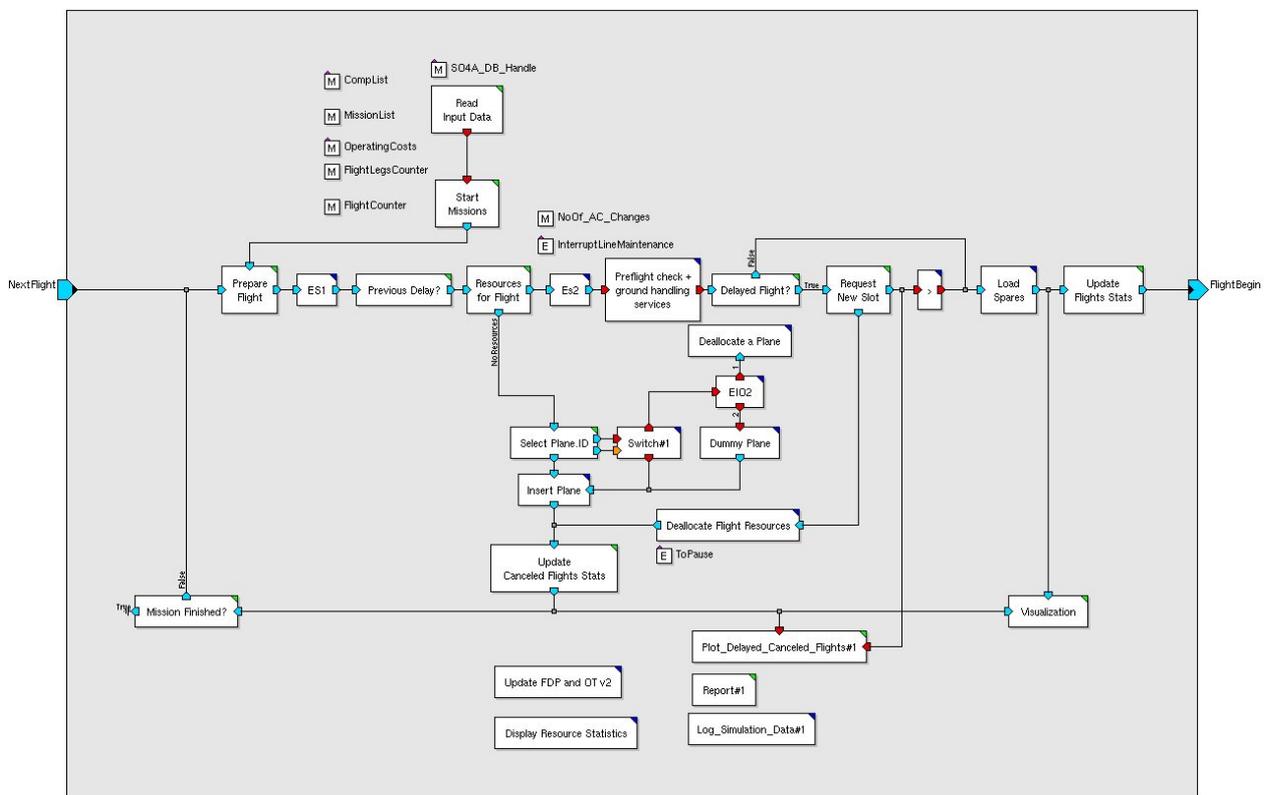


Abbildung 10: Planungsmodell in MLDesigner

Teil der Entscheidung über Ressourcen sind verschiedene Eingriffsmöglichkeiten, die beispielsweise Reaktionen auf Störungen beinhalten. Diese sogenannten Kompensationsalternativen (siehe Abschnitt 2.1.2.7) sind daher als verfeinerte Details in diesem Planungsmodell enthalten. Weitere Ideen und Regeln, wie mit Störungen oder fehlenden Ressourcen umgegangen werden soll, können hier implementiert und dann per Simulation bewertet werden.

2.1.2.6 Wartungs- und Logistikmodell

Im späteren Verlauf des Projekts wurden die Modelle der Fluggesellschaften tiefer detailliert. Damit beschäftigte sich das von Philotech bearbeitete Arbeitspaket AP154, siehe auch Abschnitt 2.1.5. Das Teilmodell, welches die Wartung modelliert, wurde umstrukturiert und auf Basis der durch den Projektpartner Philotech erarbeiteten Konzepte neu gestaltet. Wegen der Komplexität der Wartungs- und Logistikprozesse ist es nötig, Informations-, Material und Kontrollfluss voneinander zu trennen. Es existieren mehrere Modellierungsansätze zu diesem Thema. Einer dieser Ansätze ist das *Production Authorization Card (PAC)* Konzept. Bei diesem Konzept wird der Informationsfluss zur Steuerung des Materialflusses einheitlich durch sogenannte Tags abgebildet und das unterschiedliche Systemverhalten durch Variation weniger Steuerungsparameter definiert. Aufgrund der wenigen Steuerungsparameter wurde entschieden, das PAC-Konzept in das MLDesigner Modell zu integrieren. In Abbildung 11 ist das umstrukturierte Wartungsmodell zu sehen.

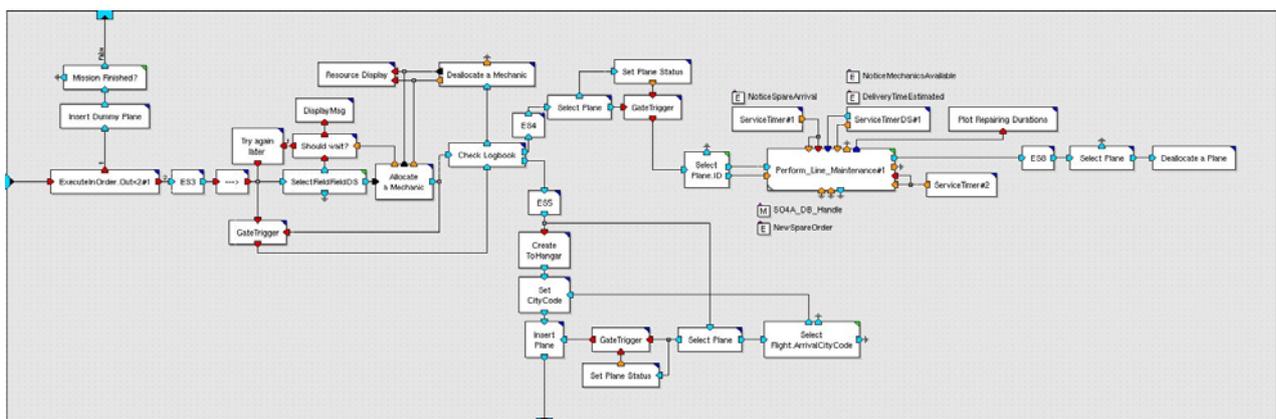


Abbildung 11: Neues Wartungsmodell

Das Logistikmodell, welches auf den Ansätzen des Production Authorization Card (PAC) Konzeptes basiert, wurde fertiggestellt und in das vorhandene Flottenmodell integriert. In Abbildung 12 ist das neue Logistikmodell im Überblick mit den dazugehörigen Events zusehen. Über diese Events erfolgt der Material- und Informationsfluss mit dem Flottenmodell. Eine detailliertere Darstellung der Abläufe in diesem Modellteil ist in Abschnitt 2.1.5 enthalten.

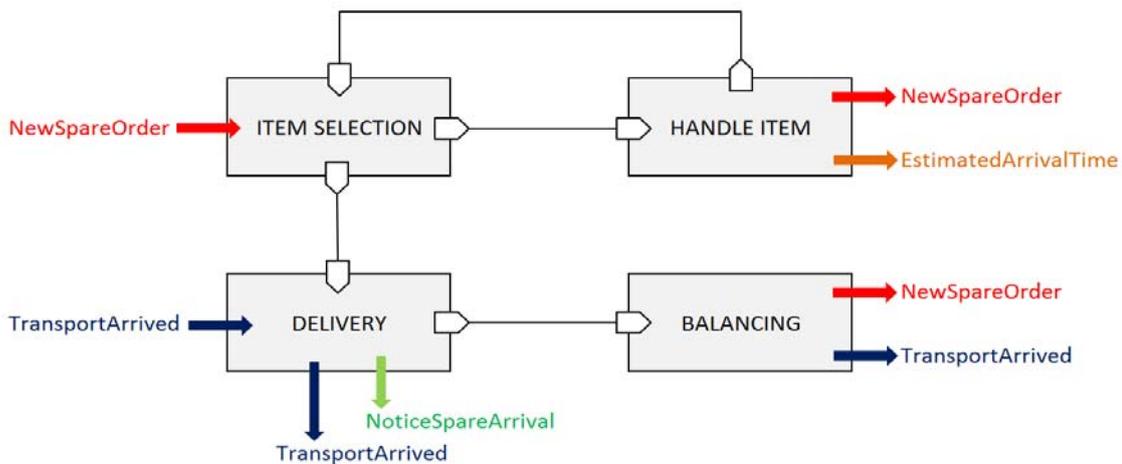


Abbildung 12: Struktur des Logistikmodells mit Events

2.1.2.7 Kompensationsalternativen

Hier wurden neben der Anpassung des Detaillierungsgrades verschiedene Kompensationsmethoden in das Modell eingearbeitet. Eine Kompensationsmethode ist beispielhaft in Abbildung 13 dargestellt. Das sogenannte „Swapping“ findet Anwendung, falls ein bestimmter Flug aufgrund eines verspäteten Flugzeuges (z.B. wegen Reparaturen) erst verspätet beginnen kann.

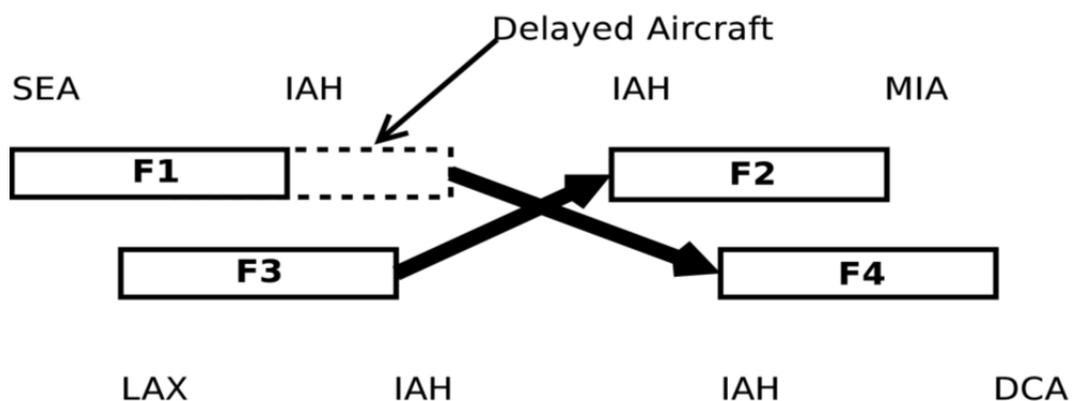


Abbildung 13: Vorgehen beim Swapping

Das Swapping bietet die Möglichkeit, ein sich im Flugbetrieb befindliches Flugzeug für diesen Flug einzusetzen. Das verspätete Flugzeug übernimmt dann den Flug des Ersatzflugzeuges. Danach gibt es zwei Möglichkeiten, wie es nach dem Tausch (Swapping) weitergeht. Entweder läuft der Flugbetrieb mit dieser Konfiguration weiter, oder es erfolgt bei der nächsten Möglichkeit ein Rücktausch der beiden Flugzeuge. Im Modell ist zurzeit nur die erste Variante implementiert.

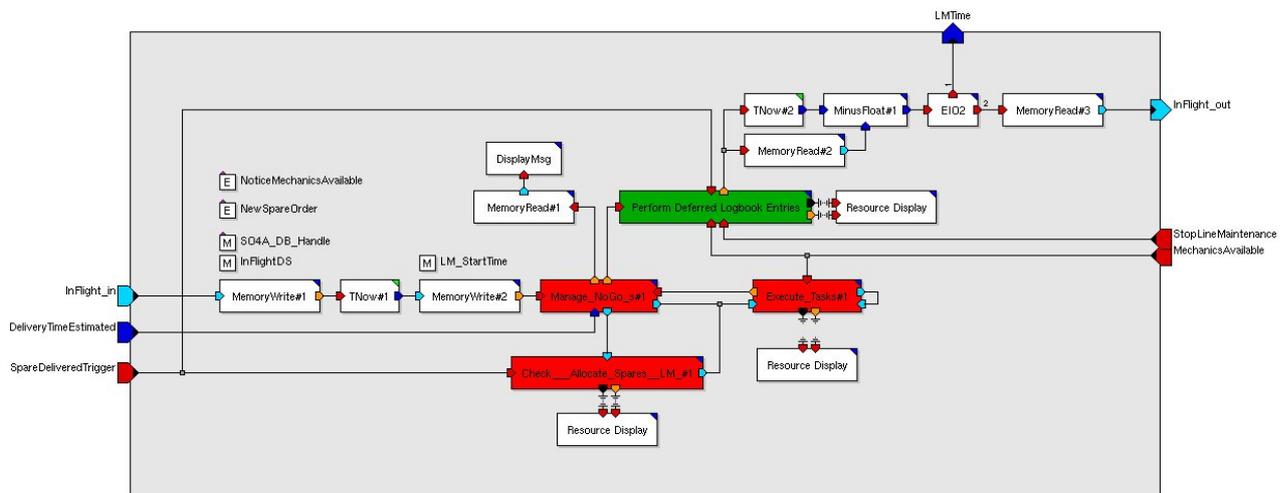


Abbildung 14: Modul Perform_Line_Maintenance

Eine weitere Kompensationsmethode ist die Reparatur eines Flugzeuges während eines Overnight-Stops. Dabei werden fehlerhafte Komponenten repariert, die kein unmittelbares No-Go zur Folge hätten. Dies hat den Vorteil, dass nicht so viele fehlerhafte Komponenten auflaufen und diese dann später enorme Reparaturzeiten verursachen. In Abbildung 14 ist das gesamte Modul für die Maintenance dargestellt. Der grün eingefärbte Block (Mitte oben) ist dabei für die oben genannten Overnight Reparatur verantwortlich. Der Prozess für diese Reparatur wird nur ausgeführt, wenn das Flugzeug bis zum nächsten Flug mindestens acht Stunden Zeit hat.

2.1.3 AP152 – Datenanalyse

Voraussetzung für aussagekräftige Simulationsergebnisse sind neben korrekten Modellen die numerischen Eingangsparameter. Ziel dieses Arbeitspaketes war die Analyse und Aufbereitung von Wartungs- und Logistikdaten. Diese Daten müssen ggf. konvertiert bzw. mit anderen Daten kombiniert werden, damit eine sinnvolle Anwendung in der Simulation möglich ist. Damit beschäftigt sich Abschnitt 2.1.3.1.

Weiterhin sollten die Analysen der Simulationsergebnisse im Rahmen dieses Arbeitspaketes durchgeführt werden. Dazu zählen z.B. Extraktion der Daten aus den jeweiligen Quellen, Aufbereitung und ggf. Verknüpfung der Daten für eine konkrete Auswertung, sowie die Erzeugung entsprechender graphischer Darstellungen (vergleiche Abschnitt 2.1.3.2). Der Nutzen einer Simulation für den Modellierer kann durch eine Animation des Verhaltens während des Simulationslaufes verbessert werden. Ergebnisse zu dieser Aufgabe stellt Abschnitt 2.1.3.3 dar.

2.1.3.1 Realistische Eingangsdaten

Nur mit Hilfe realistischer Daten ist es möglich, das Modell zu validieren. Dazu ist es notwendig, die Menge der zur Verfügung stehenden Daten sinnvoll zu filtern und zu analysieren. Folgende Daten werden vor allem benötigt:

- Flugplan / Missionsplan
- Wartungsplan
- Verbrauchsstatistik für Ersatzteile
- Ausfallraten für Ersatzteile
- Lieferzeiten / Lieferwege
- Organisationsstrukturen

Mit Hilfe eines Analyseprogramms wurde ein konkreter realistischer Airline-Flugplan in einen dazu gehörigen Missionsplan überführt. Dieser Missionsplan bildete dann die Grundlage für die Simulation. Mit Hilfe der Datenbank, welche an das Modell angebunden ist, ist es möglich für alle betrachteten Ersatzteile eine Ausfallrate zu hinterlegen. Diese Ausfallrate entspricht den realen Ausfallraten. Des Weiteren ist es möglich, die Lieferzeiten und Wege für jedes Ersatzteil einzeln zu pflegen. Die Verbrauchsstatistik wird am Ende benötigt, um das Modell zu validieren.

Im Bereich der Datenanalyse wurden Beispieldaten des Flugstatus (Fehler, Start, Landung) mit Hilfe der Simulation erzeugt. Diese dienten als Eingabewerte für das vom Projektpartner TriaG-noSys erstellte Kommunikationsmodell. In Abbildung 15 ist das Schema dieser Kommunikationsarchitektur dargestellt. Die Resultate der Kommunikationssimulation im Testbed beim Projektpartner wurden dann in das Gesamtsystemmodell integriert. Dadurch ist es möglich, Aussagen über die Veränderung der Reparaturzeit in Abhängigkeit von einer Kommunikationsinfrastruktur zu treffen.

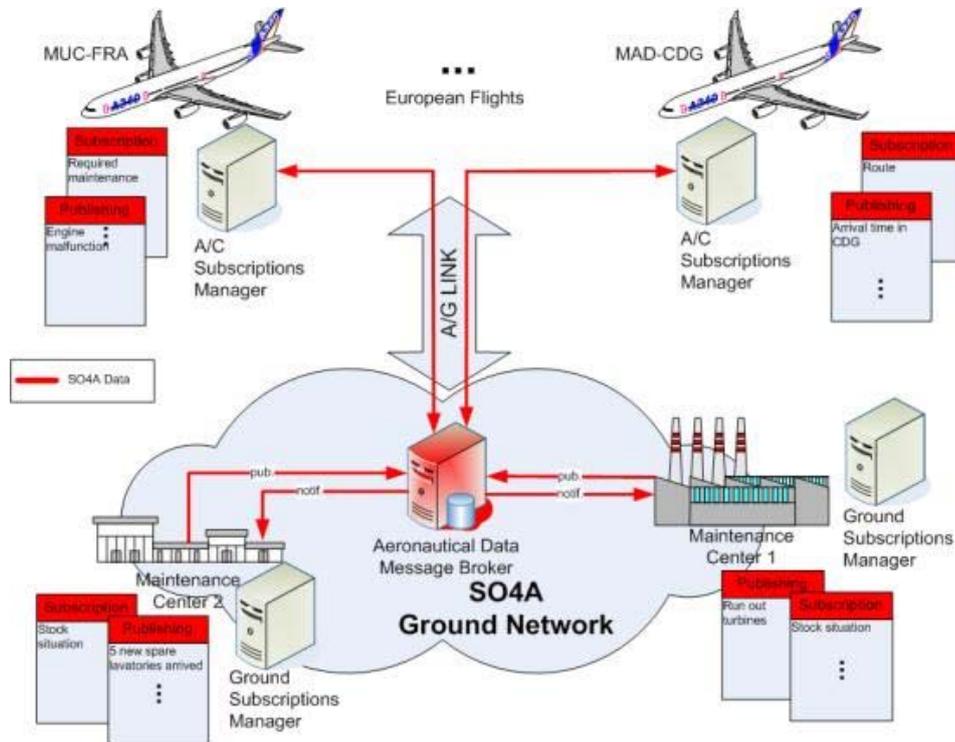


Abbildung 15: Elemente der Kommunikations-Architektur (TriaGnoSys)

Im Folgenden werden einige beispielhafte Ergebnis-Auswertungen gezeigt.

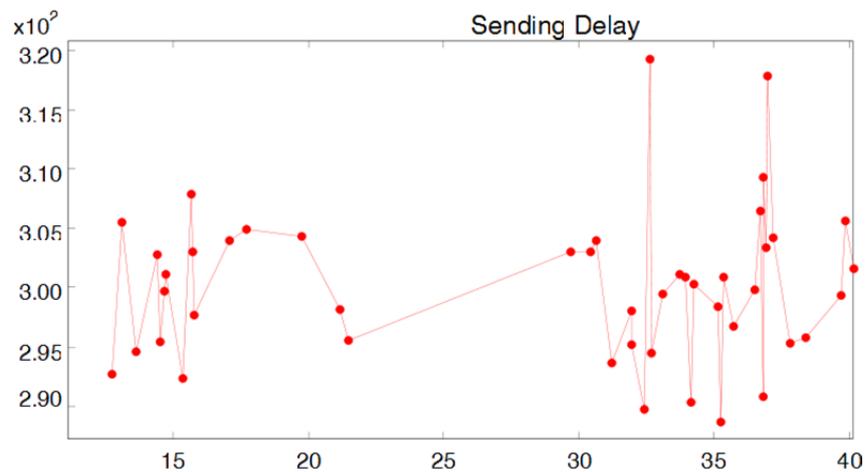


Abbildung 16: Verzögerungszeiten von Nachrichten

Abbildung 16 zeigt die Sendeverzögerung einzelner Nachrichten im Modell gegenüber der Simulationszeit. Hier können beispielsweise Trends und Ausreißer festgestellt werden, die aufgrund der gewählten Kommunikations-Infrastruktur entstehen.

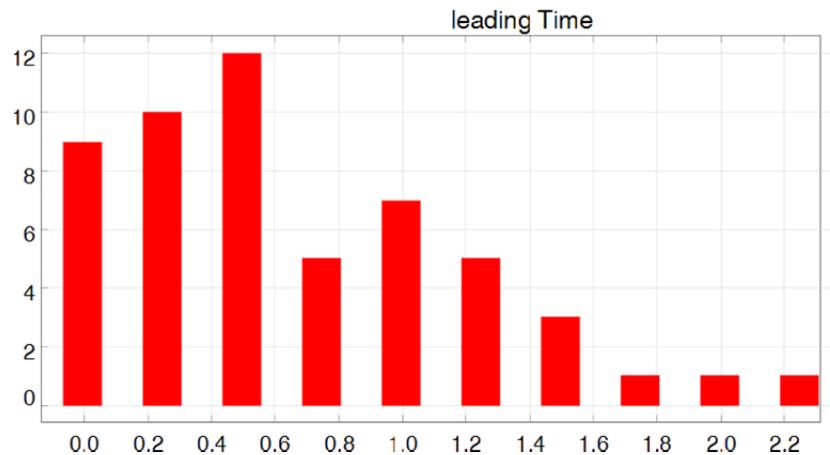


Abbildung 17: Zeitvorsprung durch Vorab-Fehlermeldungen

Letztendlich soll durch eine zukünftige Kommunikationsarchitektur eine bessere Wartungsplanung möglich werden, indem bereits während eines Fluges Fehlermeldungen zum Zielflughafen oder einer Wartungs-Zentrale geschickt werden. Mit diesen Daten kann dann die anstehende Wartung und ggf. Kompensationsalternativen geplant werden. Zahlreiche Varianten sind hier denkbar: es ist klar, dass einerseits eine schnelle und auch kleinere Fehler betreffende Sendung zu besserer Vorausplanung und damit geringeren Gesamtkosten führen kann, andererseits aber zusätzlichen Kommunikationsaufwand und -bandbreite mit entsprechenden Folgen für die nötige Architektur erfordert. Eine „optimale“ Abwägung zwischen diesen gegenläufigen Einflüssen ist ein typischer „Trade-off“ im Systems Engineering, der in so komplexen Fällen nur mit einem (Simulations-)Modell untersucht werden kann. In unserem Beispiel ist zunächst die Frage zu klären, wie groß der zeitliche Vorsprung durch das Vorab-Senden der Fehlermeldungen in der Realität zu erwarten ist. Abbildung 1 zeigt eine entsprechende Häufigkeitsverteilung für das Modell, aus dem der theoretisch nutzbare Zeitvorsprung gegenüber der hergebrachten Methode als Ausgangspunkt für Verbesserungen ablesbar ist.

2.1.3.2 Simulationsergebnisse

Mit dem Ende der Simulation werden unter anderem folgende Informationen ausgegeben: Canceled Flights, Delayed Flights, Blocked on Ground, End Report und PFR Entries. Eine detaillierte Beschreibung der in den Meldungen enthaltenen Informationen ist in den Zwischenberichten gegeben.

Beispielsweise wäre in Delayed Flights exemplarisch die Information

```

7:Airbus_WP120:Root.WP120.InFlight{{1,A300,8:Airbus_WP120:Root.
  WP120.LogbookEntry{0,New,1,0,0,Cabin}}, {42,A300}, {55,A300},
  {5,7:Airbus_WP120:Root.WP120.Flight{0,"","",0,0,0}},
  {2,FRA,GOT,6.5,1.6666666666666667,548}}
  
```

zu interpretieren als: der Flug mit der ID 2, von Frankfurt nach Göteborg, mit der Startzeit 6:30 Uhr, einer Flugdauer von 1h 40min (1.6666), einer Flugstrecke von 548 Meilen, in der Mission 5, bestehend aus 7 Flügen, startet erst um 7:00 Uhr. Erkennbar sind in der Datenstruktur weitere Informationen wie z. B. die Anzahl der Logbucheinträge (hier: 8) oder die ID und der Typ des Flugzeuges (hier: 1, A300).

Alle Ausgaben werden in eine Datenbank geschrieben (vgl. Abschnitt 2.1.4.2), aus der dann alle weiteren Analysen erfolgen können.

Als weitere Ausgabe stehen in der Konsole des Simulationswerkzeugs die Gesamtkosten der Airline zur Verfügung. Die Gesamtkosten sind in folgende Teilkosten aufgeteilt:

- Kosten für Mechaniker
- Kosten für Flugzeuge
- Kosten für FlightCrew und CabinCrew
- Kosten für verspätete und gestrichene Flüge
- Kosten für Hangars
- Einnahmen durch Flugtickets

Auf der obersten Ebene sind im Parameterfeld von MLDesigner weitere Ausgaben/Plots aktivierbar.

- *ShowHistograms* zeigt die Anzahl der Logbucheinträge (x-Achse) im Zusammenhang mit der Anzahl der Flüge (y-Achse), z. B. 18 Flüge mit 2 Einträgen (siehe Abbildung 18) und die Reparaturdauer (x-Achse) im Zusammenhang mit der Anzahl der Flüge (y-Achse), z. B. 9 Flüge haben eine mittlere Reparaturdauer von 0.25 h, was einer Dauer von 15 min entspricht (siehe Abbildung 19).

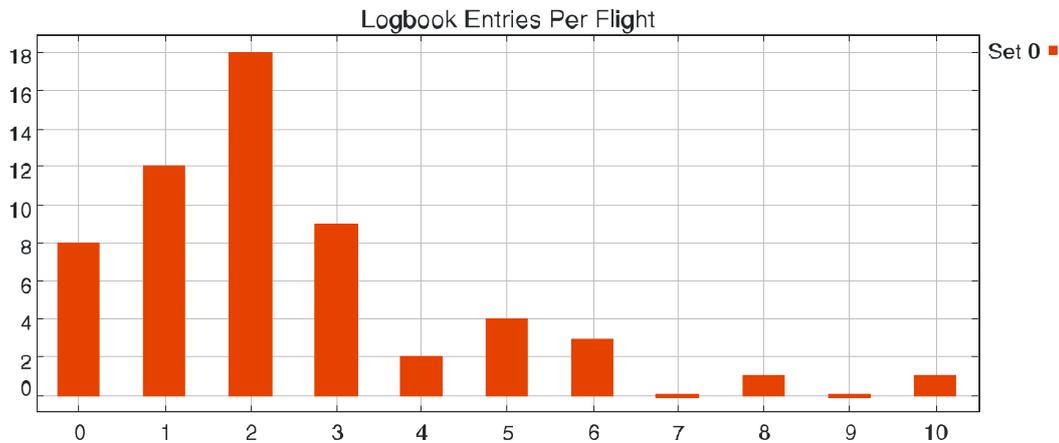


Abbildung 18: Logbook Entries per Flight

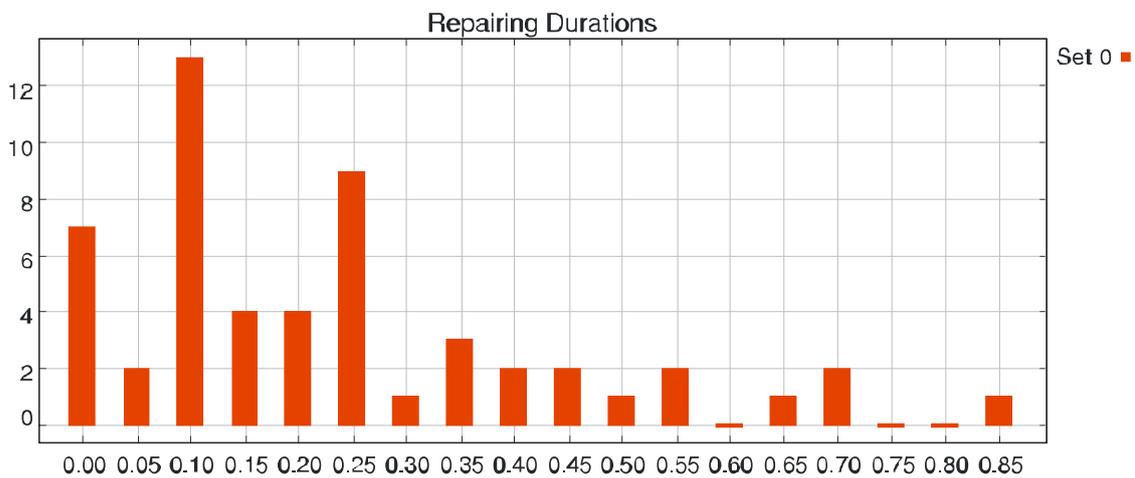


Abbildung 19: Repairing Durations

- *ShowHangarStats* zeigt die Belegung der Hangars an den einzelnen Flughafen-Standorten.

Über das Modul Resource Management kann eine Auswertung an den einzelnen Standorten aktiviert werden. Mit Hilfe der Parameter `Display_CabinCrews`, `Display_FlightCrews`, `Display_Mechanics`, `Display_Planes` kann die entsprechende Ressourcenauslastung an einem Standort angezeigt werden. Die Angabe des Standortes erfolgt über den CityCode (IATA-Code). Bei gewünschter Ressourcenauslastung an mehreren Standorten werden diese durch Leerzeichen getrennt angegeben.

Exemplarisch seien einige Plots, siehe Abbildung 20 bis Abbildung 24, kurz erläutert. Die Simulationszeit beträgt 24 h (angegeben in der obersten Ebene, Parameter `RunLength`). Betrachtet sei die Auslastung der Ressourcen:

- *Plane* in Frankfurt: in Frankfurt sind zunächst drei Flugzeugtypen (A300, A320, A340) verfügbar. Die Startzeit des ersten Fluges ist um 5:20 Uhr, d.h. der Flieger steht 30 min vorher am Gate bereit und eine Ressource *Plane* wird belegt. Kurz nach 18:00 Uhr steht ein A300 wieder zum Abflug zur Verfügung.
- *FlightCrew* in Frankfurt: prinzipiell ist die Auslastung ähnlich wie der Auslastung *Plane*. Landen allerdings Flieger kann die Crew auch wechseln, was z.B. gegen 12:00 Uhr der Fall ist.
- *Mechanic* in Frankfurt: initial steht in Frankfurt ein *Mechanic* der Kategorie *CatA* zur Verfügung. Kurz nach 12:00 Uhr landet eine Maschine in Frankfurt, die Ressource *Mechanic* wird belegt und nach Logbuchcheck und Reparatur wieder freigegeben.
- *CabinCrew* in Frankfurt: siehe *FlightCrew*.

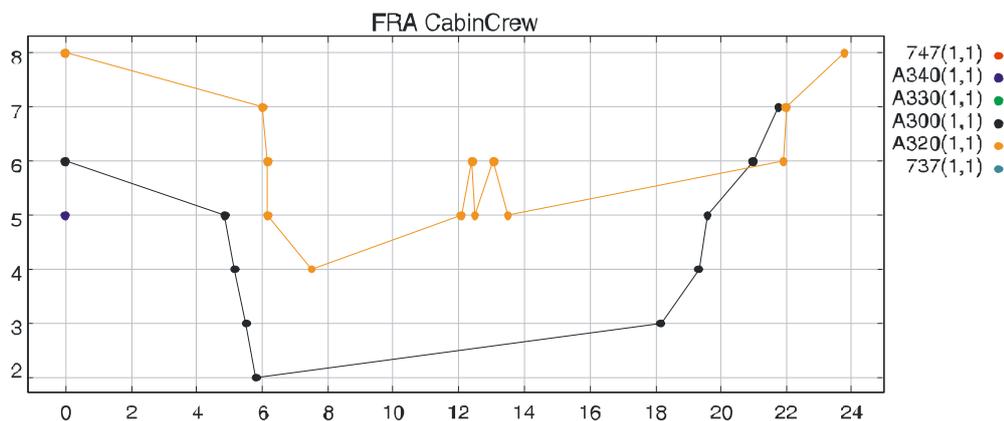


Abbildung 20: Ressource Cabin Crew in Frankfurt

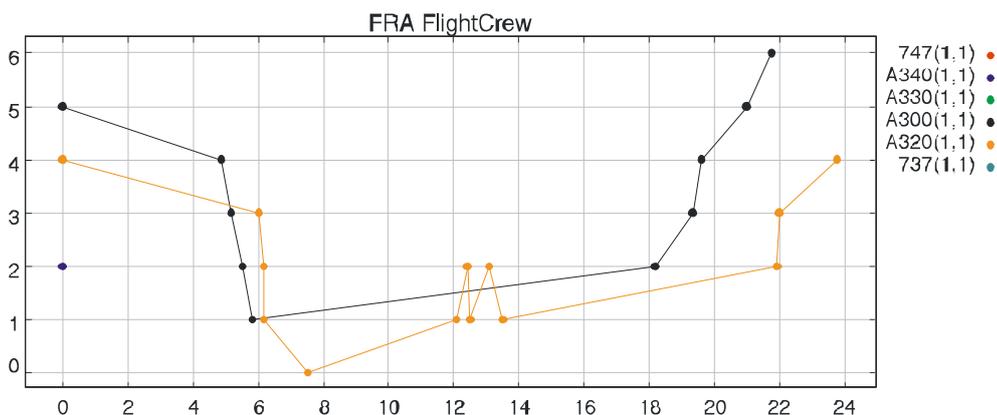


Abbildung 21: Ressource Flight Crew in Frankfurt

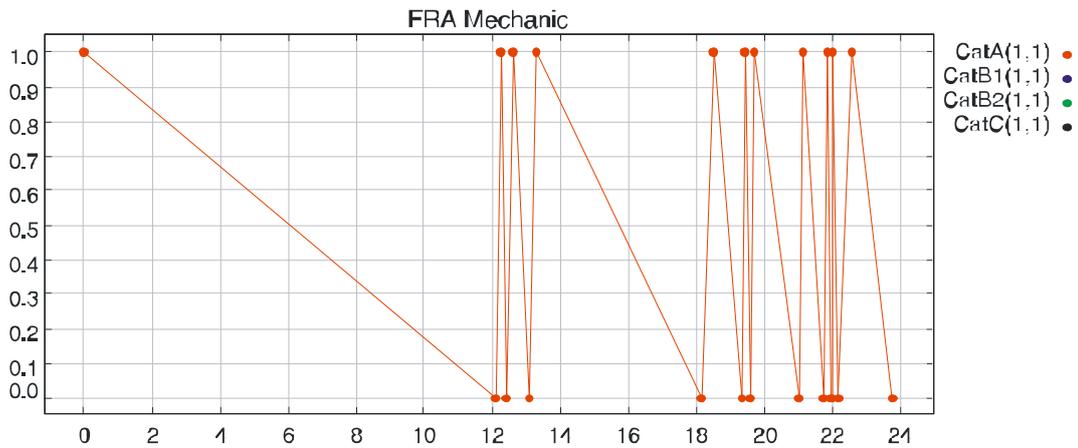


Abbildung 22: Ressource Mechanic in Frankfurt

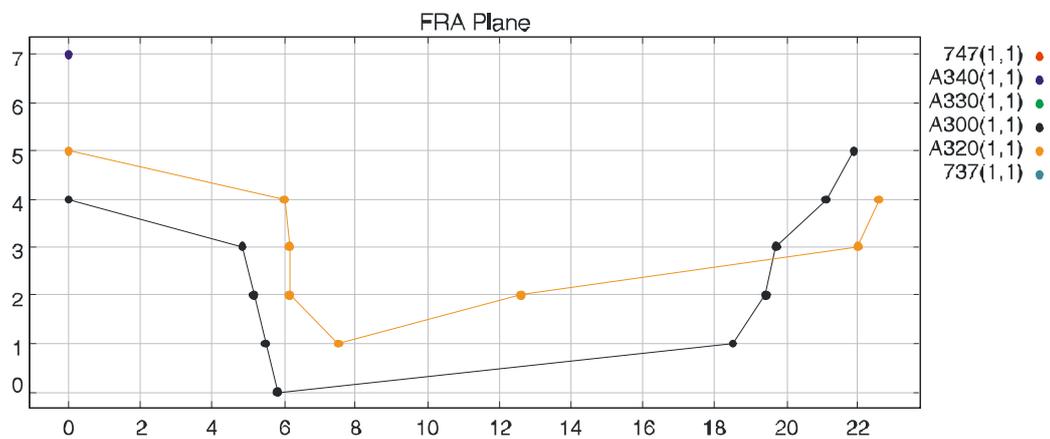


Abbildung 23: Ressource Plane in Frankfurt

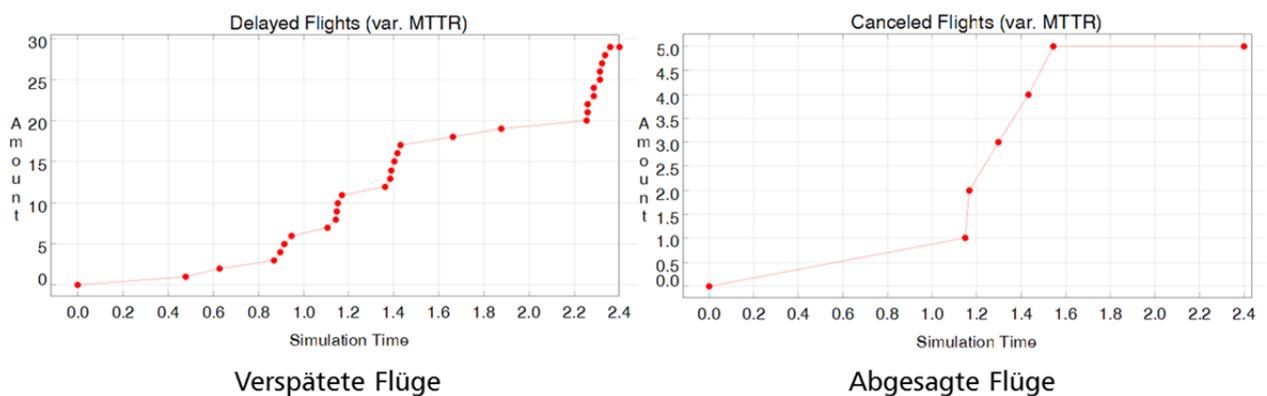


Abbildung 24: Verspätete und abgesagte Flüge in einer Simulation

Abbildung 24 zeigt die Anzahl verfügbarer Flugzeuge in einem Flughafen über der Simulationszeit dar. Eine Frage könnte nun sein, wie sich verschiedene Varianten der Ersatzteil-Logistik (z.B. Anlieferung per Straße oder Flug) auf das Verhalten auswirken. Für eine Variante ergäben sich dann beispielsweise als Ergebnisse für die Zuverlässigkeit des Flugbetriebs die Anzahl und

Zeiten des Auftretens von verspäteten und abgesagten Flügen, wie sie in Abbildung 25 zu sehen sind.

2.1.3.3 Animation des Simulationsablaufs

Im Projekt wurde eine speziell für die Fragestellungen von Airlines geeignete Präsentation der Ergebnisse während der Simulation entwickelt. Im Gesamtsystem existiert dafür ein Modul „Visualization“ zur Animation der vorhandenen Flüge während der Simulation. Auf der obersten Ebene ist im Parameterfeld von MLDesigner die Animation über den Parameter ShowPlaneMovement aktivierbar. Die Nichtverfügbarkeit von Ressourcen an den Standorten (Flughäfen) wird in der Animation durch Markierung des betroffenen Flughafens signalisiert. Die geografische Lage der Flieger und evtl. Signalisierung von Problemen wird direkt in einer (austauschbaren) Landkarte grafisch angezeigt. Hier wurde USA gewählt, da realistische Daten einer Airline für dieses Gebiet vorlagen. Abbildung 25 zeigt einen beispielhaften screen shot des Werkzeugs während einer Simulation. Die wichtigsten Ereignisse werden auch oben in vier Fenstern textuell dargestellt.

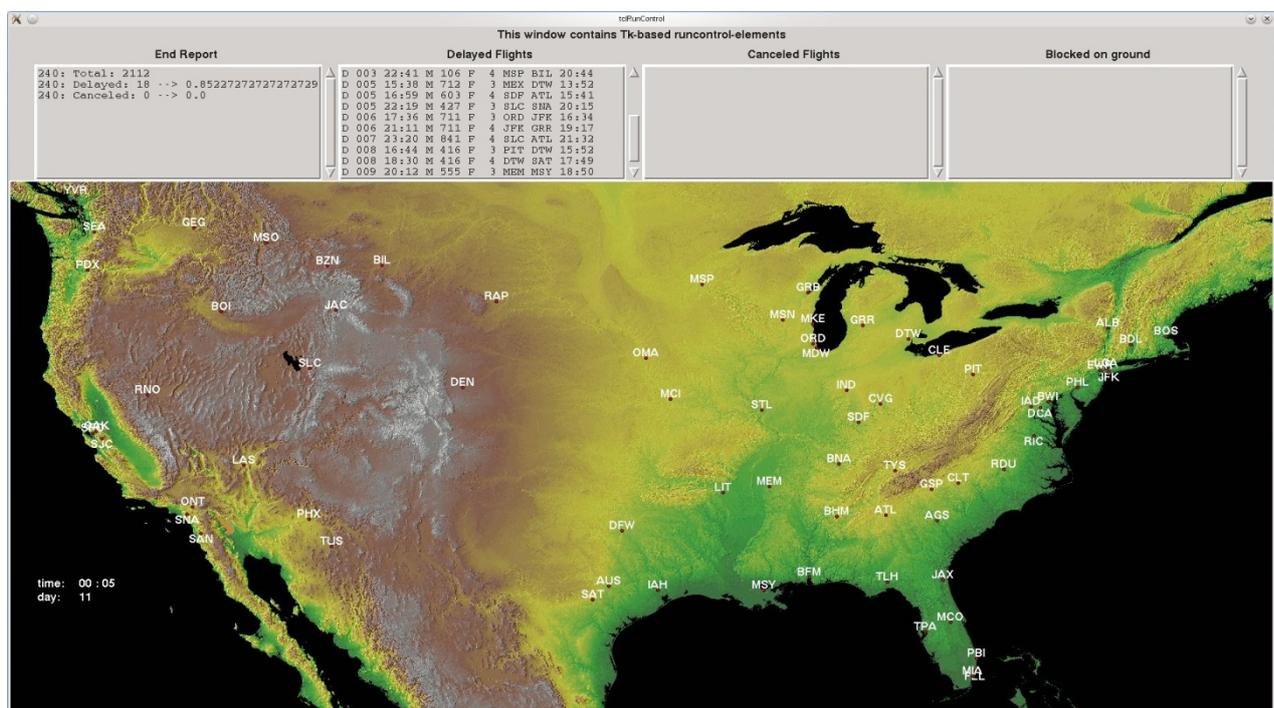


Abbildung 25: Ergebnispräsentation während der Simulation

2.1.4 AP153 – Enterprise Application Integration

Hauptziel des Arbeitspaketes 153 war der Entwurf einer Architektur unter Integrationsaspekten, die die Anforderungen von SO4A/@MOST hinsichtlich des serviceorientierten Ansatzes des Schlussbericht @MOST-G SO4A

Gesamtprojektes erfüllt. Nach der Neustrukturierung des Projekts und dem Ausscheiden des Projektpartners T-Systems aus der Verantwortung für die Arbeitspakete 153 und 154 wurden Teilbereiche dieser Aufgabe durch die TU Ilmenau und MLD durchgeführt. Im Ergebnis wurde im erstellten umfangreichen Modell (siehe Abschnitt 2.1.2) die Integration und das Zusammenspiel verschiedener Elemente der Logistik- und Wartungsarchitektur abgebildet, die durch den Unterauftragnehmer Philotech innerhalb des Arbeitspaketes 154 erarbeitet wurden (vergleiche Abschnitt 2.1.5).

Die Einbettung eines simulationsbasierten Planungs- und Bewertungswerkzeugs für zukünftige serviceorientierte Architekturen im Luftfahrtbereich muss nach der konzeptionellen Arbeit technisch umgesetzt werden. Zu lösende Teilaspekte betreffen die Architektur eines solchen Werkzeugs (siehe Abschnitt 2.1.4.1) sowie die Verwaltung von Eingangsdaten, Szenarien und Ergebnissen der Simulation, die hier mit einer Datenbank realisiert werden (siehe Abschnitt 2.1.4.2). Ein solches Werkzeug kann nicht allein alle Teilprobleme im Detail lösen, sondern muss erweiterbar sein. Beispielhaft wird in Abschnitt 2.1.4.3 die Anbindung an eine vorhandene Methode zur Lösung des Problems der anfänglichen Ersatzteilbevorratung (*initial provisioning*) dargestellt.

2.1.4.1 Werkzeugarchitektur

Abbildung 26 zeigt die Kopplung der aktuell im Projekt verwendeten Softwarewerkzeuge. Zentrales Tool ist MLDesigner, welches mit einer SQLite-Datenbank gekoppelt ist.

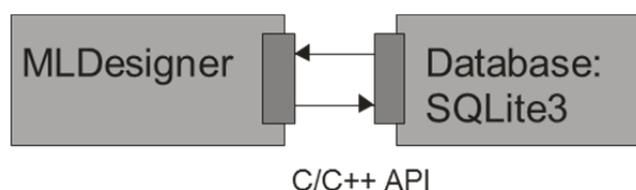


Abbildung 26: Architektur des Werkzeugs

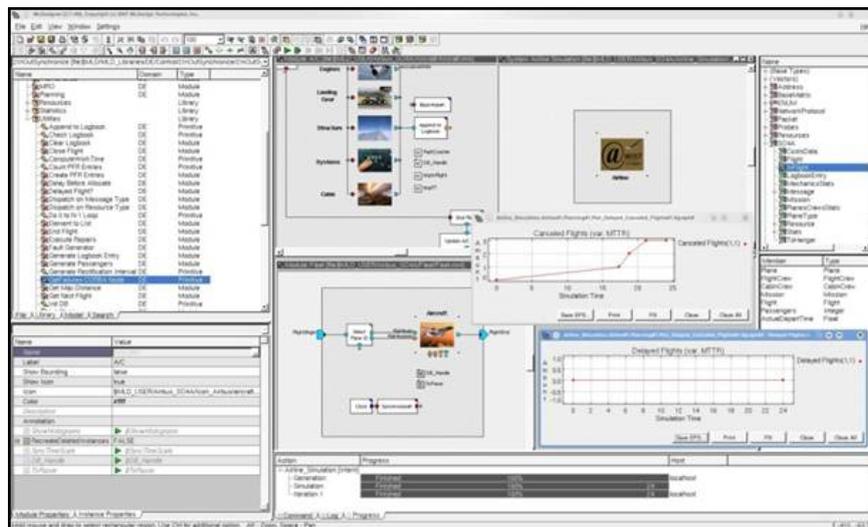


Abbildung 27: Screen shot von MLDesigner

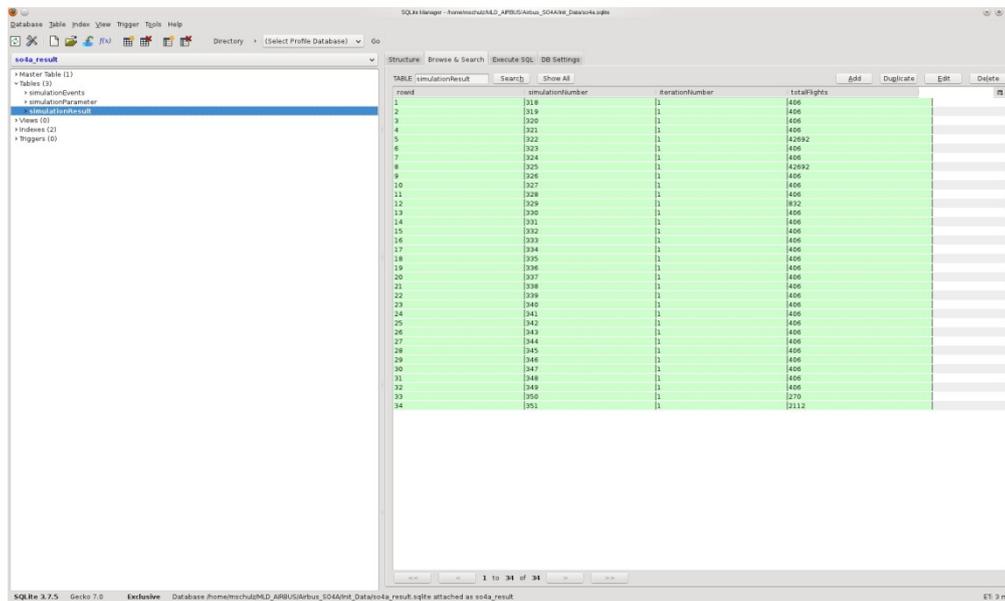
Sämtlich Modelldaten und Parameter werden in der Datenbank verwaltet. Die Kopplung erfolgt über eine C/C++ API (application programming interface). Die Integration mit dem anfänglich außerdem noch verwendeten Werkzeug CaHMoS war für die im späteren Projektverlauf betrachteten Analysen nicht mehr notwendig.

Die Modelldaten und Parameter wurden vollständig in die Datenbank ausgelagert, um sie einfacher veränderbar zu gestalten und sie besser als Parametersätze abzuspeichern. Lesbarkeit und das Verständlichkeit des Modells erhöhen sich ebenfalls. Für Details der genaueren Datenbankstruktur wird auf den Zwischenbericht zum 3. Meilenstein vom Juni 2010 verwiesen. Die während eines Simulationslaufs erzeugten Ergebnisse werden ebenfalls in einer Datenbank abgespeichert, um sie im Anschluss analysieren zu können.

Eine prototypisch als Designstudie entwickelte Erweiterung ermöglicht die Eingabe von Daten für geografisch verteilte Systeme in einer 3D-Umgebung, ihre Evaluierung und letztlich Optimierung von Architekturvarianten [SZYa, SZYb]. Die prinzipielle Möglichkeit der Beschreibung von komplexen Avionik-Systemen mit Struktur- und Verhaltensmodellen in der Object Process Methodology [D02, ODS11] und die Übersetzung dieser Modelle in MLDesigner zur Simulation und Leistungsbewertung wurden in [SZM10] untersucht.

2.1.4.2 Datenbank und Simulationsszenarien

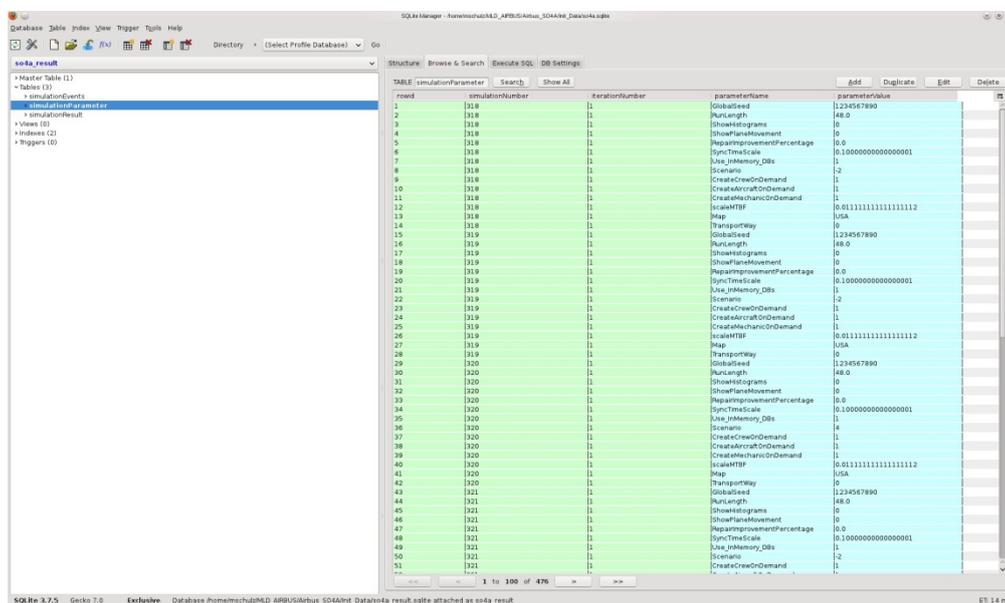
Im Projekt wurden Strategien entwickelt, um die Menge an Eingabeparameter, Eingabedaten und vor allem die Simulationsergebnisse zu handhaben. Dies ermöglicht spätere Auswertungen über mehrere Iterationen der Simulation. Hierzu wurde eine Datenbankstruktur geschaffen bzw. entsprechend erweitert. Für die Menge an Simulationsergebnissen wurde eine eigene Datenbank aufgesetzt, die drei verschiedene Tabellen enthält (siehe Abbildung 28).



rowid	simulationNumber	IterationNumber	totalFlights
1	318	1	406
2	319	1	406
3	320	1	406
4	321	1	406
5	322	1	42892
6	323	1	406
7	324	1	406
8	325	1	42892
9	326	1	406
10	327	1	406
11	328	1	406
12	329	1	892
13	330	1	406
14	331	1	406
15	332	1	406
16	333	1	406
17	334	1	406
18	335	1	406
19	336	1	406
20	337	1	406
21	338	1	406
22	339	1	406
23	340	1	406
24	341	1	406
25	342	1	406
26	343	1	406
27	344	1	406
28	345	1	406
29	346	1	406
30	347	1	406
31	348	1	406
32	349	1	406
33	350	1	270
34	351	1	2112

Abbildung 28: Beispielhafte Ergebnis-Tabelle

In diesen Tabellen werden alle Resultate, aber auch die entsprechenden Parametereinstellungen der Simulation gespeichert (siehe Abbildung 29). Diese Datenbankbindung ermöglicht eine offene Schnittstelle, mit der Simulationsläufe angepasst werden können, ohne die grafische Benutzungsoberfläche zu verwenden.



rowid	simulationNumber	IterationNumber	parameterName	parameterValue
1	318	1	GlobalSeed	1234567890
2	318	1	RunLength	480
3	318	1	ShowHistograms	0
4	318	1	ShowPlaneMovement	0
5	318	1	RepairPowerPercentage	0.0
6	318	1	SyncTimeScale	0.1000000000000001
7	318	1	UseMemory_Dbs	1
8	318	1	Scenario	-2
9	318	1	CreateAircraftOnDemand	1
10	318	1	CreateMechanicOnDemand	1
11	318	1	IsCableTBP	0.0111111111111112
12	318	1	Map	USA
13	318	1	Transportway	0
14	318	1	GlobalSeed	1234567890
15	318	1	RunLength	480
16	318	1	ShowHistograms	0
17	318	1	ShowPlaneMovement	0
18	318	1	RepairPowerPercentage	0.0
19	318	1	SyncTimeScale	0.1000000000000001
20	318	1	UseMemory_Dbs	1
21	318	1	Scenario	-2
22	318	1	CreateAircraftOnDemand	1
23	318	1	CreateMechanicOnDemand	1
24	318	1	IsCableTBP	0.0111111111111112
25	318	1	Map	USA
26	318	1	Transportway	0
27	318	1	GlobalSeed	1234567890
28	318	1	RunLength	480
29	318	1	ShowHistograms	0
30	318	1	ShowPlaneMovement	0
31	318	1	RepairPowerPercentage	0.0
32	318	1	SyncTimeScale	0.1000000000000001
33	318	1	UseMemory_Dbs	1
34	318	1	Scenario	-2
35	318	1	CreateAircraftOnDemand	1
36	318	1	CreateMechanicOnDemand	1
37	318	1	IsCableTBP	0.0111111111111112
38	318	1	Map	USA
39	318	1	Transportway	0
40	318	1	GlobalSeed	1234567890
41	318	1	RunLength	480
42	318	1	ShowHistograms	0
43	318	1	ShowPlaneMovement	0
44	318	1	RepairPowerPercentage	0.0
45	318	1	SyncTimeScale	0.1000000000000001
46	318	1	UseMemory_Dbs	1
47	318	1	Scenario	-2
48	318	1	CreateAircraftOnDemand	1
49	318	1	CreateMechanicOnDemand	1
50	318	1	IsCableTBP	0.0111111111111112
51	318	1	Map	USA
52	318	1	Transportway	0
53	318	1	GlobalSeed	1234567890
54	318	1	RunLength	480
55	318	1	ShowHistograms	0
56	318	1	ShowPlaneMovement	0
57	318	1	RepairPowerPercentage	0.0
58	318	1	SyncTimeScale	0.1000000000000001
59	318	1	UseMemory_Dbs	1
60	318	1	Scenario	-2
61	318	1	CreateAircraftOnDemand	1
62	318	1	CreateMechanicOnDemand	1
63	318	1	IsCableTBP	0.0111111111111112
64	318	1	Map	USA
65	318	1	Transportway	0
66	318	1	GlobalSeed	1234567890
67	318	1	RunLength	480
68	318	1	ShowHistograms	0
69	318	1	ShowPlaneMovement	0
70	318	1	RepairPowerPercentage	0.0
71	318	1	SyncTimeScale	0.1000000000000001
72	318	1	UseMemory_Dbs	1
73	318	1	Scenario	-2
74	318	1	CreateAircraftOnDemand	1
75	318	1	CreateMechanicOnDemand	1
76	318	1	IsCableTBP	0.0111111111111112
77	318	1	Map	USA
78	318	1	Transportway	0
79	318	1	GlobalSeed	1234567890
80	318	1	RunLength	480
81	318	1	ShowHistograms	0
82	318	1	ShowPlaneMovement	0
83	318	1	RepairPowerPercentage	0.0
84	318	1	SyncTimeScale	0.1000000000000001
85	318	1	UseMemory_Dbs	1
86	318	1	Scenario	-2
87	318	1	CreateAircraftOnDemand	1
88	318	1	CreateMechanicOnDemand	1
89	318	1	IsCableTBP	0.0111111111111112
90	318	1	Map	USA
91	318	1	Transportway	0
92	318	1	GlobalSeed	1234567890
93	318	1	RunLength	480
94	318	1	ShowHistograms	0
95	318	1	ShowPlaneMovement	0
96	318	1	RepairPowerPercentage	0.0
97	318	1	SyncTimeScale	0.1000000000000001
98	318	1	UseMemory_Dbs	1
99	318	1	Scenario	-2
100	318	1	CreateAircraftOnDemand	1
101	318	1	CreateMechanicOnDemand	1
102	318	1	IsCableTBP	0.0111111111111112
103	318	1	Map	USA
104	318	1	Transportway	0
105	318	1	GlobalSeed	1234567890
106	318	1	RunLength	480
107	318	1	ShowHistograms	0
108	318	1	ShowPlaneMovement	0
109	318	1	RepairPowerPercentage	0.0
110	318	1	SyncTimeScale	0.1000000000000001
111	318	1	UseMemory_Dbs	1
112	318	1	Scenario	-2
113	318	1	CreateAircraftOnDemand	1
114	318	1	CreateMechanicOnDemand	1
115	318	1	IsCableTBP	0.0111111111111112
116	318	1	Map	USA
117	318	1	Transportway	0
118	318	1	GlobalSeed	1234567890
119	318	1	RunLength	480
120	318	1	ShowHistograms	0
121	318	1	ShowPlaneMovement	0
122	318	1	RepairPowerPercentage	0.0
123	318	1	SyncTimeScale	0.1000000000000001
124	318	1	UseMemory_Dbs	1
125	318	1	Scenario	-2
126	318	1	CreateAircraftOnDemand	1
127	318	1	CreateMechanicOnDemand	1
128	318	1	IsCableTBP	0.0111111111111112
129	318	1	Map	USA
130	318	1	Transportway	0
131	318	1	GlobalSeed	1234567890
132	318	1	RunLength	480
133	318	1	ShowHistograms	0
134	318	1	ShowPlaneMovement	0
135	318	1	RepairPowerPercentage	0.0
136	318	1	SyncTimeScale	0.1000000000000001
137	318	1	UseMemory_Dbs	1
138	318	1	Scenario	-2
139	318	1	CreateAircraftOnDemand	1
140	318	1	CreateMechanicOnDemand	1
141	318	1	IsCableTBP	0.0111111111111112
142	318	1	Map	USA
143	318	1	Transportway	0
144	318	1	GlobalSeed	1234567890
145	318	1	RunLength	480
146	318	1	ShowHistograms	0
147	318	1	ShowPlaneMovement	0
148	318	1	RepairPowerPercentage	0.0
149	318	1	SyncTimeScale	0.1000000000000001
150	318	1	UseMemory_Dbs	1
151	318	1	Scenario	-2
152	318	1	CreateAircraftOnDemand	1
153	318	1	CreateMechanicOnDemand	1
154	318	1	IsCableTBP	0.0111111111111112
155	318	1	Map	USA
156	318	1	Transportway	0
157	318	1	GlobalSeed	1234567890
158	318	1	RunLength	480
159	318	1	ShowHistograms	0
160	318	1	ShowPlaneMovement	0
161	318	1	RepairPowerPercentage	0.0
162	318	1	SyncTimeScale	0.1000000000000001
163	318	1	UseMemory_Dbs	1
164	318	1	Scenario	-2
165	318	1	CreateAircraftOnDemand	1
166	318	1	CreateMechanicOnDemand	1
167	318	1	IsCableTBP	0.0111111111111112
168	318	1	Map	USA
169	318	1	Transportway	0
170	318	1	GlobalSeed	1234567890
171	318	1	RunLength	480
172	318	1	ShowHistograms	0
173	318	1	ShowPlaneMovement	0
174	318	1	RepairPowerPercentage	0.0
175	318	1	SyncTimeScale	0.1000000000000001
176	318	1	UseMemory_Dbs	1
177	318	1	Scenario	-2
178	318	1	CreateAircraftOnDemand	1
179	318	1	CreateMechanicOnDemand	1
180	318	1	IsCableTBP	0.0111111111111112
181	318	1	Map	USA
182	318	1	Transportway	0
183	318	1	GlobalSeed	1234567890
184	318	1	RunLength	480
185	318	1	ShowHistograms	0
186	318	1	ShowPlaneMovement	0
187	318	1	RepairPowerPercentage	0.0
188	318	1	SyncTimeScale	0.1000000000000001
189	318	1	UseMemory_Dbs	1
190	318	1	Scenario	-2
191	318	1	CreateAircraftOnDemand	1
192	318	1	CreateMechanicOnDemand	1
193	318	1	IsCableTBP	0.0111111111111112
194	318	1	Map	USA
195	318	1	Transportway	0
196	318	1	GlobalSeed	1234567890
197	318	1	RunLength	480
198	318	1	ShowHistograms	0
199	318	1	ShowPlaneMovement	0
200	318	1	RepairPowerPercentage	0.0
201	318	1	SyncTimeScale	0.1000000000000001
202	318	1	UseMemory_Dbs	1
203	318	1	Scenario	-2
204	318	1	CreateAircraftOnDemand	1
205	318	1	CreateMechanicOnDemand	1
206	318	1	IsCableTBP	0.0111111111111112
207	318	1	Map	USA
208	318	1	Transportway	0
209	318	1	GlobalSeed	1234567890
210	318	1	RunLength	480
211	318	1	ShowHistograms	0
212	318	1	ShowPlaneMovement	0
213	318	1	RepairPowerPercentage	0.0
214	318	1	SyncTimeScale	0.1000000000000001
215	318	1	UseMemory_Dbs	1
216	318	1	Scenario	-2
217	318	1	CreateAircraftOnDemand	1
218	318	1	CreateMechanicOnDemand	1
219	318	1	IsCableTBP	0.0111111111111112
220	318	1	Map	USA
221	318	1	Transportway	0
222	318	1	GlobalSeed	1234567890
223	318	1	RunLength	480
224	318	1	ShowHistograms	0
225	318	1	ShowPlaneMovement	0
226	318	1	RepairPowerPercentage	0.0
227	318	1	SyncTimeScale	0.1000000000000001
228	318	1	UseMemory_Dbs	1
229	318	1	Scenario	-2
230	318	1	CreateAircraftOnDemand	1
231	318	1	CreateMechanicOnDemand	1
232	318	1	IsCableTBP	0.0111111111111112
233	318	1	Map	USA
234	318	1	Transportway	0
235	318	1	GlobalSeed	1234567890
236	318	1	RunLength	480
237	318	1	ShowHistograms	0
238	318	1	ShowPlaneMovement	0
239	318	1	RepairPowerPercentage	0.0
240	318	1	SyncTimeScale	0.1000000000000001
241	318	1	UseMemory_Dbs	1
242	318	1	Scenario	-2
243	318	1	CreateAircraftOnDemand	1
244	318	1	CreateMechanicOnDemand	1

2.1.4.3 Ersatzteilbevorratung

Eine wichtige Entscheidung mit Auswirkungen auf Kosten und Zuverlässigkeit (also Quality of Service aus einer serviceorientierten Sicht der Airline) ist die Abwägung, welche und wie viele Ersatzteile in den einzelnen Maintenance-Stationen bevorratet werden sollen. Auch hier gibt es wieder einen klassischen trade-off: viele Ersatzteile erhöhen die Verfügbarkeit, kosten aber auch Lager- und Wartungsaufwände. Diese Entscheidung betrifft das sogenannte “initial provisioning”. Für diesen Bereich wurde vom Projektpartner Airbus eine Methode zur Berechnung des voraussichtlichen Bedarfs und der verbesserten Verteilung auf die Wartungsstationen entwickelt, die im Rahmen des Projekts als MLD-Modell implementiert und in das hier beschriebene Gesamtmodell integriert wurde. Die Verwaltung der Daten über die vorteilhafte Ersatzteilverteilung wird ebenfalls über die angekoppelte Datenbank realisiert und ist daher leicht zu verändern oder durch ein externes Werkzeug zu überschreiben.

2.1.5 AP154 – Logistisches Modell

Im Rahmen der Arbeitspakete 153 und 154 wurden durch Philotech die logistischen Prozesse im Rahmen eines Airline-Betriebes analysiert und für die spätere Modellierung aufgearbeitet. Der Schwerpunkt dieser Arbeit lag auf den Themengebieten:

- Nach welchen Kriterien finden die Abläufe des Spare Part Repair Cycles statt?
- Wie kann die Kapazität von Lagerhäusern angemessen realistisch simuliert werden?
- Welche Kriterien werden benötigt, um vorgegebene Angaben zum Initial Provisioning im Modell korrekt abzubilden?

Die Themen wurden zunächst wie weiter unten in diesem Abschnitt dargestellt als UML-Diagramme visualisiert, um eine möglichst schnelle Umsetzung im bereits existenten MLDesigner-Modell zu gestatten. Über die Zusammenarbeit mit dem Projektpartner EADS IW und in mehreren Gesprächen mit Airbus konnte die Aufgabenstellung noch einmal konkreter eingegrenzt werden. Die Simulation musste die richtige Balance zwischen zu hoher Komplexität auf der einen und angemessener Realitätsnähe auf der anderen Seite finden.

Zur Bearbeitung der drei genannten Kernfragen war es notwendig, diejenigen Parameter eines Ersatzteils zu identifizieren, die maßgeblichen Einfluss auf die logistischen Abläufe haben. Hierfür wurde auf die bei Philotech vorhandene Expertise im Bereich der Logistik und des Ersatzteil Managements zurückgegriffen, aber auch das Wissen anderer Projektgruppen genutzt. Darüber hinaus fand ein Informationsaustausch mit der Hochschule für Angewandte Wissenschaften in Hamburg statt.

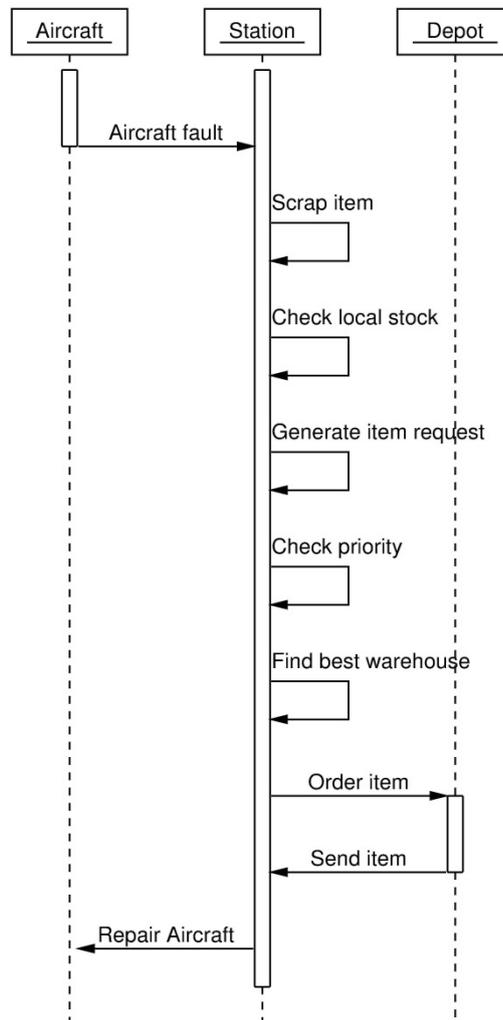


Abbildung 30: Sequenzdiagramm für den Ablauf nach einem Aircraft Fault

Die Abbildung 30 zeigt das Sequenzdiagramm für den Anwendungsfall Aircraft Fault: Ein Flugzeug erzeugt eine Wartungsanforderung an die Station. Für die Wartung wird in dem Beispiel nur ein Ersatzteil benötigt. Das für die Wartung benötigte Ersatzteil ist nicht reparierbar und nicht im lokalen Lager der Station vorhanden. Die Station bestellt das Ersatzteil bei einem Depot und repariert das Flugzeug, sobald das benötigte Ersatzteil verfügbar ist.

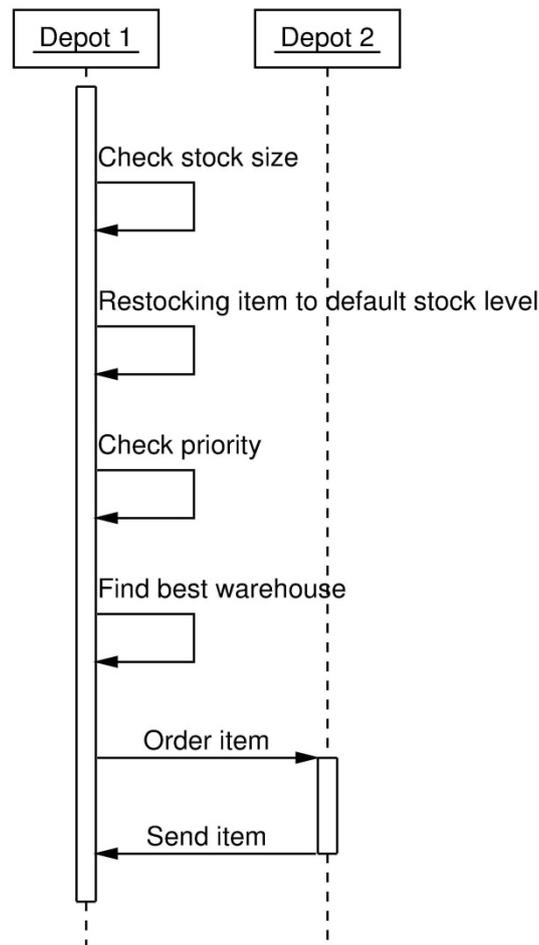


Abbildung 31: Sequenzdiagramm für den Vorgang Lagerauffüllung

Die Abbildung 31 zeigt das Sequenzdiagramm für den Anwendungsfall Lagerauffüllung (Restocking). Depot 1 stellt im Beispiel fest, dass die festgesetzte Mindestanzahl für ein bestimmtes Ersatzteil unterschritten ist. Das Depot 1 sucht das Depot mit den geringsten Transportkosten, in dem das Ersatzteil noch in ausreichender Menge vorhanden ist. Von diesem Depot wird dann das Ersatzteil Menge bestellt, so dass der Lagerbestand wieder die geforderte Menge erreicht.

Im Bereich des logistischen Modells wurden auf Grundlage der vorher im Projekt erstellten UML-Diagramme die logistischen Prozesse in das Flottenmodell integriert. Der Schwerpunkt lag dabei auf folgenden Themen

- Implementierung des neuen Logistikmodelles
- vollständige Integration des neuen Logistikmodelles
- Validierung des neues Logistikmodelles

Innerhalb des Logistikmoduls (vergleiche Abschnitt 2.1.2.6) werden die grundlegenden Prozesse der Ersatzteillogistik beschrieben. Das Logistikmodell (siehe Abbildung 12) besteht aus vier

Modulen, die zusammen die Abläufe an einer einzelnen Station beschreiben. Im Folgenden werden die Elemente dieses Modells genauer erläutert. Die einzelnen Module sind ITEM SELECTION, HANDLE ITEM, DELIVERY und BALANCING. Diese behandeln das Eintreffen von Bestellungen, die Überprüfung der Lieferbarkeit von Ersatzteilen, Annahme und Versand von Lieferungen und die Überprüfung der Lagerbestände. Bei der betrachteten Station kann es sich sowohl um die Anlagen an einem Flughafen, ein Lagerhaus, einen Repair Shop oder einen Supplier handeln.

Die Kommunikation zwischen Flottenmodell und Logistikmodul erfolgt über drei verschiedene Events:

- das Flottenmodell teilt den Logistikmodell mit, dass eine neue Ersatzteilbestellung vorliegt [*NewSpareOrder*]
- Sobald die Logistik eine Abschätzung zum Lieferzeitpunkt machen kann, wird dieser an das Flottenmodell gesendet. (Dies ermöglicht es zum Beispiel das bei längeren Wartezeiten auch andere unkritische Reparaturen durchgeführt werden können.) [*EstimatedArrivalTime*]
- das Logistikmodell benachrichtigt das Flottenmodell das die Lieferung vollständig eingetroffen ist [*NoticeSpareArrival*]

Innerhalb des Logistikmodells wird ein weiteres Event für das Eintreffen einer Ersatzteillieferung verwendet. [*TransportArrived*]

Die Kommunikation zwischen den verschiedenen Station erfolgt auf zwei voneinander getrennten Wegen: Dem Warenfluss und dem Informationsfluss. Der Warenfluss entspricht den Lieferungen von Ersatzteilen. Beim Abschicken einer Lieferung erzeugt eine Station ein verzögertes Event (*TransportArrived*); sobald dieses eintritt, weiß die Empfängerstation, dass sie eine neue Lieferung empfangen hat.

Der Informationsfluss erfolgt über Bestellungen (*NewSpareOrder*), durch die eine Station von einer anderen Station Teile anfordern kann.

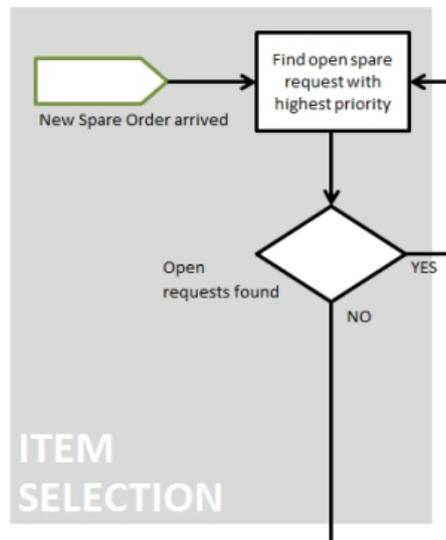


Abbildung 32: Modul ITEM SELECTION

Das Modul ITEM SELECTION wird durch ein Event (neue Bestellungen) vom Flottenmodell oder einer anderen Station gestartet (siehe Abbildung 32). Aus der Datenbank wird die offene Bestellung gesucht, die die höchste Priorität hat. Diese wird dann an das Modul HANDLE ITEM übergeben. Ist die Bearbeitung der Bestellung abgeschlossen, wird das Modul ITEM SELECTION erneut aufgerufen und nach den nächsten offenen Bestellungen gesucht. Dies wird solange durchgeführt, bis keine offenen Bestellungen an der aktuellen Station gefunden werden können. Danach wird für die aktuelle Station das Modul DELIVERY aufgerufen, die weiter unten beschrieben ist.

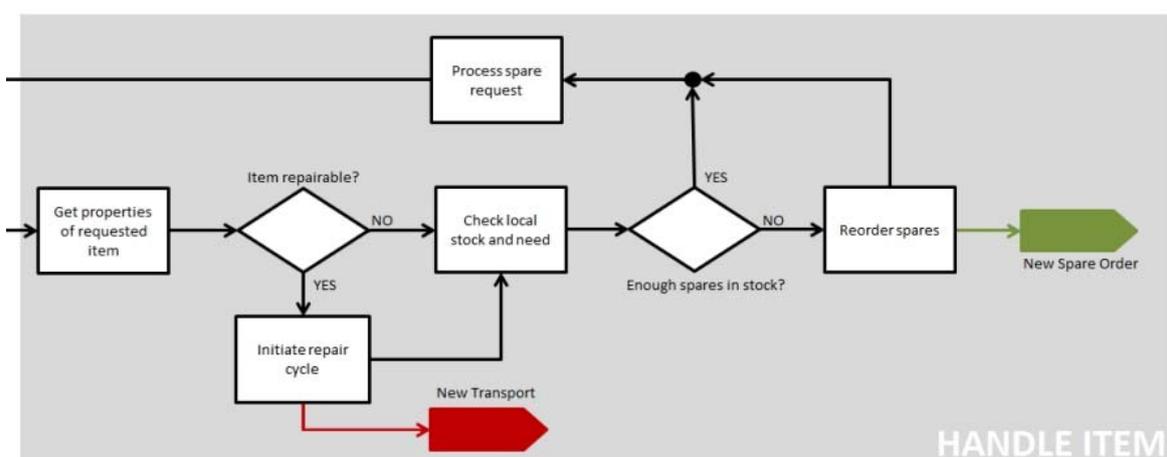


Abbildung 33: Modul HANDLE ITEM

Das in Abbildung 33 gezeigte Modul HANDLE ITEM führt für jede Bestellung zwei Überprüfungen durch. Im ersten Schritt REPAIR CYCLE wird kontrolliert, ob das Ersatzteil reparierbar ist und zu einem Repair Shop geschickt werden muss. Konnte das Ersatzteil repariert werden, so

werden die Kosten für den Transport (hin und zurück), die Reparatur und die Zeit für den Transport und die Reparatur ermittelt. Mit den Transportkosten und der Gesamtzeit wird ein neuer Transport zur aktuellen Station im Modell erzeugt. Dies bildet ab, wann das Ersatzteil an dieser Station wieder für die Ersatzteillogistik verfügbar ist und welche Kosten dabei entstanden sind. Die Reparaturkosten werden gesondert gespeichert. Konnte das Teil nicht repariert werden, so entstehen nur die Kosten und Transportzeit für einen einfachen Transport und die Kosten für die versuchte Reparatur. Aus Sicht der Ersatzteillogistik verschwindet das Teil an dieser Stelle aus dem eigenen Bestand.

Im zweiten Schritt REORDER überprüft die Station, ob das Ersatzteil vorrätig ist oder nachbestellt werden muss. Dabei wird auch berücksichtigt, ob die schon getätigten Nachbestellungen ausreichen, um die aktuelle Nachfrage zu decken. Ist ein Ersatzteil vorrätig, so wird als möglicher Lieferzeitpunkt ‚sofort‘ angenommen. Andernfalls wird der nächstmögliche Lieferant bestimmt und der erwartete Lieferzeitpunkt berechnet. Die Ersatzteillogistik erstellt mit diesen Daten dann eine neue Bestellung an die entsprechende Station. Abschließend wird der Status der Bestellung auf „in Bearbeitung“ gesetzt und der erwartete Lieferzeitpunkt an das Flottenmodell gesendet.

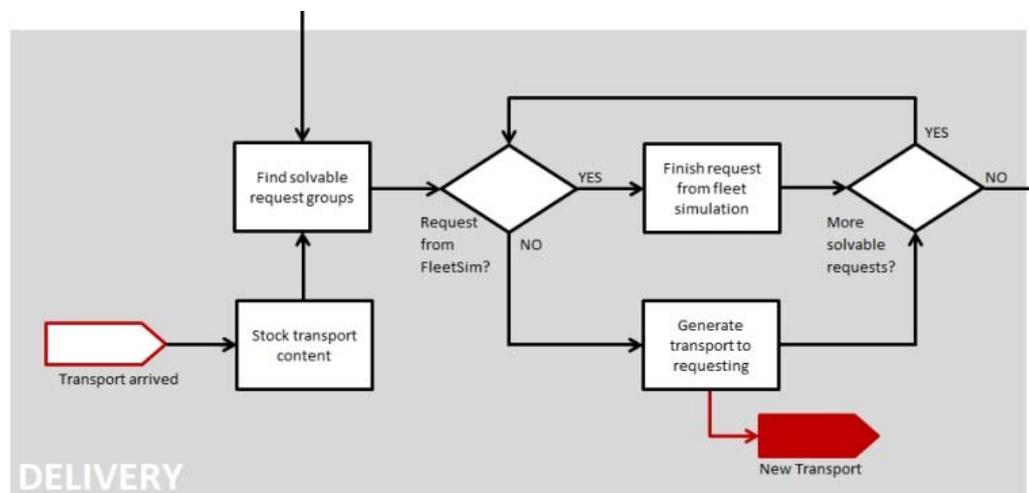


Abbildung 34: Modul DELIVERY

Das Modul DELIVERY (siehe Abbildung 34) wird durch zwei mögliche Events aufgerufen: Entweder durch das Eintreffen eines Transports an der Station (TransportArrived), oder durch das Modul ITEM SELECTION nachdem alle Teile einer Bestellung bearbeitet wurden. Bei einem eingehenden Transport wird das eingetroffene Ersatzteil zum Lagerbestand hinzugefügt. Anschließend wird (genau wie beim Aufruf durch das Modul ITEM SELECTION) überprüft, ob es an dieser Station gegenwärtig Bestellungen mit Status „in Bearbeitung“ vorliegen, die abgeschlossen werden könnten. Handelt es sich um eine Bestellung durch das Flottenmodell, so

wird diese abgeschlossen und der Lagerbestand angepasst. Die Flottensimulation wird über den Abschluss der Bestellung mit Hilfe des Events [NoticeSpareArrival] informiert. Kam die Bestellung von einer anderen Station, werden Transportzeit und -kosten berechnet und ein entsprechender Transport erzeugt. Auch hier wird der aktuelle Lagerbestand angepasst. Das Modul BALANCING wird nach Abschluss möglicher Liefertätigkeiten von DELIVERY aufgerufen. Es ist im Kontext des gesamten Logistikmodells in Abbildung 35 dargestellt. Im Modul wird überprüft, ob die Lagerbestände für alle Ersatzteile innerhalb der minimalen bzw. maximalen Grenzwerte liegen. Ist der minimale Lagerbestand für ein Ersatzteil unterschritten, so wird dieser auf den Ausgangswert aufgestockt. Es wird überprüft, welche anderen Stationen die zur Aufstockung benötigten Ersatzteile liefern können. An diese entsprechenden Stationen werden Bestellungen geschickt. Ist der maximale Lagerbestand überschritten, wird überprüft, welche Station Bedarf an dem entsprechenden Ersatzteil hat. Die Simulation erzeugt ggf. einen Transport zu dieser Station.

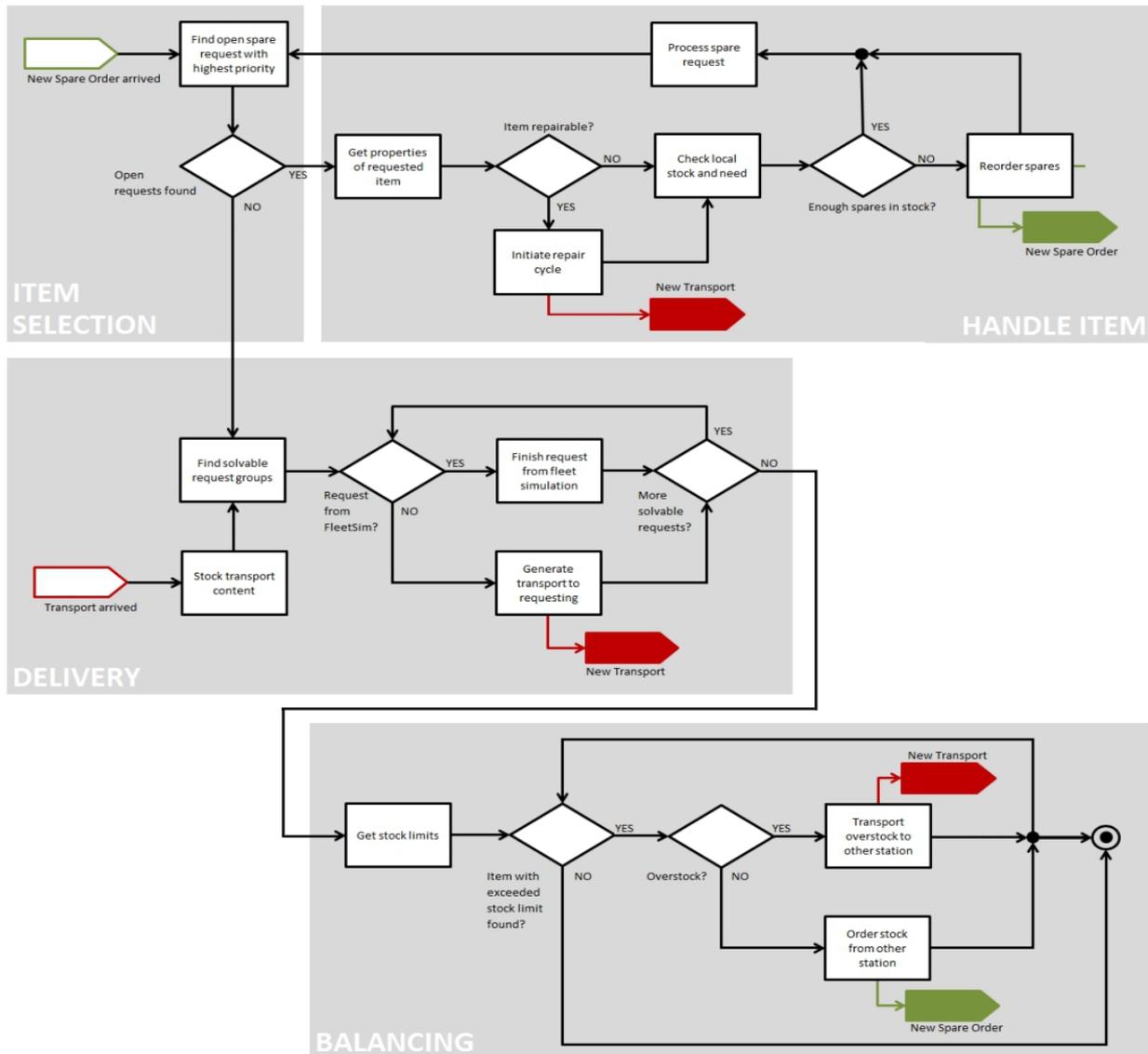


Abbildung 35: Ablaufdiagramm des gesamten Logistikmodells

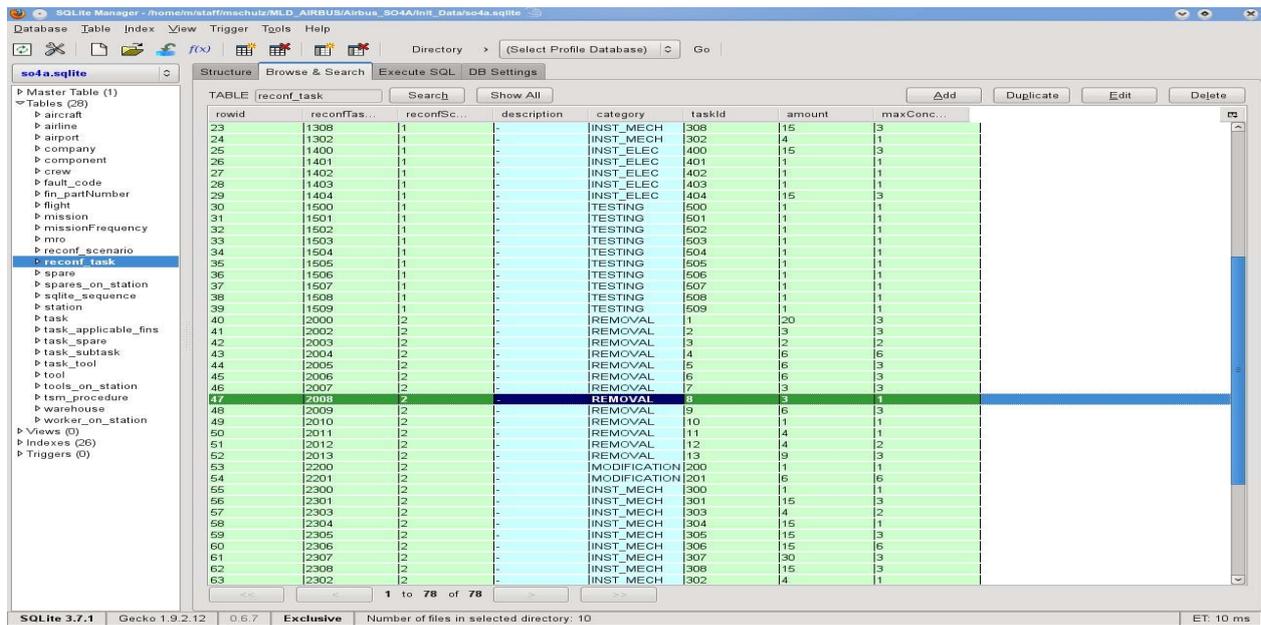
2.1.6 Sonstige Zuarbeiten

Neben den weiter oben berichteten Arbeitspaketen war die TU Ilmenau mit ihren Unterauftragnehmern auch ergänzend in anderen Arbeitspaketen tätig.

Im Bereich der funktionalen Definition (AP130) wurden die bestehenden Modelle der Fluggesellschaften tiefer detailliert. Der Grad der Detaillierung in der funktionalen Hierarchie musste hierbei so weit vorangetrieben werden, dass eine sinnvolle Kopplung von Flugbetrieb, Infrastruktur und Logistik möglich wurde.

Für die Gesamtarchitektur ist es von großer Wichtigkeit, den Einfluss einer Flugzeugkomponente auf die Wartungsplanung und damit auf den Flugplan abzuschätzen. Dies gilt im besonderen Maße auch für die Ersatzteilversorgung und deren Logistik. Damit beschäftigten sich Zuarbeiten für das Arbeitspaket AP133 (Service Level Agreement). Um quantitative Aussagen machen zu können und somit die Bewertung der Dienstgüte vorzubereiten war es nötig, echte Flugzeugdaten auf der Ebene der Wartungsarbeiten in das Modell einzuführen. Diese Daten stellen den Zusammenhang zwischen den Fehlerereignissen und den benötigten Ressourcen zur Instandsetzung her. Diese Daten umfassen z.B.: Ersatzteilart und –menge, Werkzeugart und –menge, Verbrauchsmaterialbedarf und Mechanikeranzahl. Zu diesem Zweck wurde das erstellte ausführbare Modell um eine entsprechende Datenbank ergänzt. Ein beispielhafter Screenshot ist in Abbildung 36 dargestellt.

Ziel ist es, eine Simulation über das gesamte System durchführen zu können, d.h. Flugbetrieb im Simulationsmodell korrekt mit der Ersatzteilbevorratung und der Logistik zu verbinden und beides als Einheit auszuführen. Des Weiteren sind mit dieser Erweiterung detaillierte Untersuchungen in Bezug auf die Nachverfolgbarkeit von Einzelereignissen und deren Folgen möglich geworden. Ein Beispiel hierfür ist die Analyse der Verspätungshistorie von Flugzeugen. Diese werden mit dem Simulationsmodell nun berechnet und benutzungsfreundlich grafisch dargestellt, wie beispielhaft in Abbildung 37 für einen Beispiel-Flugbetrieb im europäischen Luftraum zu sehen.



rowid	reconfTask	reconfSc...	description	category	taskid	amount	maxConc...
23	1308	1	INST_MECH	INST_MECH	308	15	3
24	1302	1	INST_MECH	INST_MECH	302	4	1
25	1400	1	INST_ELEC	INST_ELEC	400	15	3
26	1401	1	INST_ELEC	INST_ELEC	401	1	1
27	1402	1	INST_ELEC	INST_ELEC	402	1	1
28	1403	1	INST_ELEC	INST_ELEC	403	1	1
29	1404	1	INST_ELEC	INST_ELEC	404	15	3
30	1500	1	TESTING	TESTING	500	1	1
31	1501	1	TESTING	TESTING	501	1	1
32	1502	1	TESTING	TESTING	502	1	1
33	1503	1	TESTING	TESTING	503	1	1
34	1504	1	TESTING	TESTING	504	1	1
35	1505	1	TESTING	TESTING	505	1	1
36	1506	1	TESTING	TESTING	506	1	1
37	1507	1	TESTING	TESTING	507	1	1
38	1508	1	TESTING	TESTING	508	1	1
39	1509	1	TESTING	TESTING	509	1	1
40	2000	2	REMOVAL	REMOVAL	1	20	3
41	2002	2	REMOVAL	REMOVAL	2	3	3
42	2003	2	REMOVAL	REMOVAL	3	2	2
43	2004	2	REMOVAL	REMOVAL	4	6	6
44	2005	2	REMOVAL	REMOVAL	5	6	3
45	2006	2	REMOVAL	REMOVAL	6	6	3
46	2007	2	REMOVAL	REMOVAL	7	3	3
47	2008	2	REMOVAL	REMOVAL	8	3	1
48	2009	2	REMOVAL	REMOVAL	9	6	3
49	2010	2	REMOVAL	REMOVAL	10	1	1
50	2011	2	REMOVAL	REMOVAL	11	4	1
51	2012	2	REMOVAL	REMOVAL	12	4	2
52	2013	2	REMOVAL	REMOVAL	13	3	3
53	2200	2	MODIFICATION	MODIFICATION	200	1	1
54	2201	2	MODIFICATION	MODIFICATION	201	6	6
55	2300	2	INST_MECH	INST_MECH	300	1	1
56	2301	2	INST_MECH	INST_MECH	301	15	3
57	2303	2	INST_MECH	INST_MECH	303	4	2
58	2304	2	INST_MECH	INST_MECH	304	15	3
59	2305	2	INST_MECH	INST_MECH	305	15	3
60	2306	2	INST_MECH	INST_MECH	306	15	6
61	2307	2	INST_MECH	INST_MECH	307	30	3
62	2308	2	INST_MECH	INST_MECH	308	15	3
63	2302	2	INST_MECH	INST_MECH	302	4	1

Abbildung 36: Wartungsdatenbank zur Integration in das ausführbare Modell

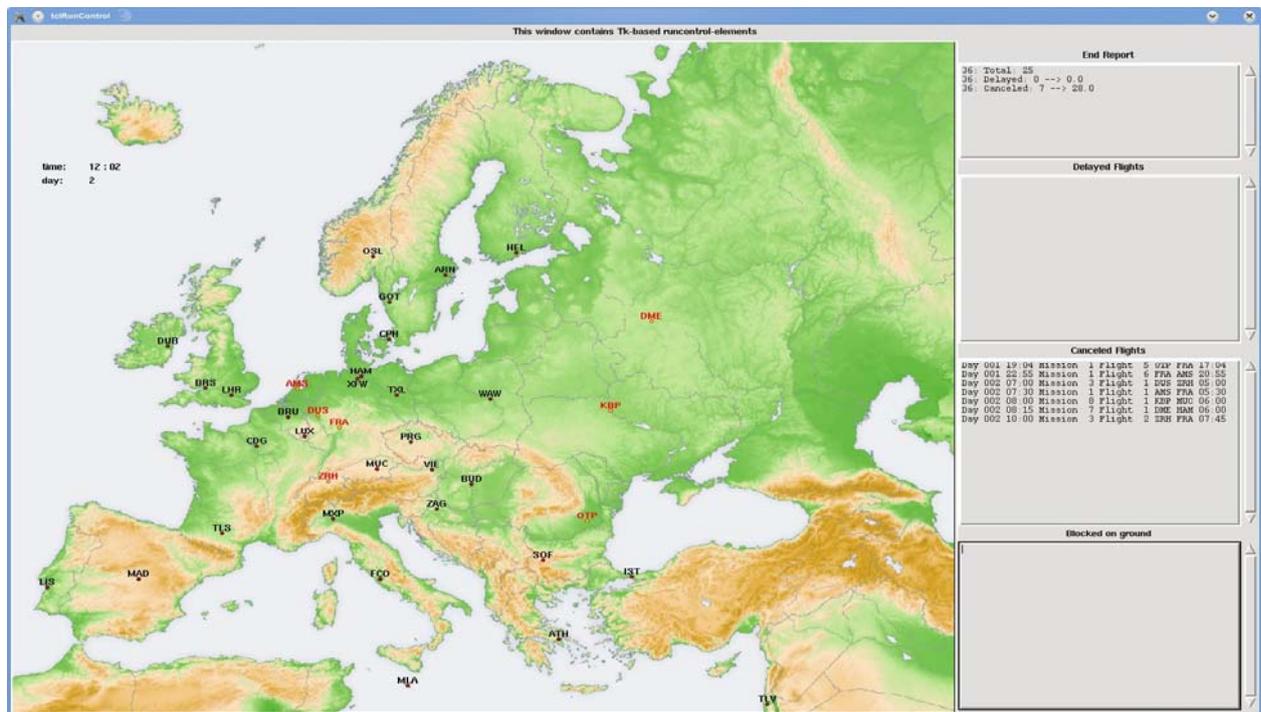


Abbildung 37: Anzeige: Verspätungen, gestrichene Flüge, Ressourcenverfügbarkeit

2.2 Verwendung der Zuwendung, Notwendigkeit und Angemessenheit der Arbeit

Ein geprüfter, detaillierter rechnerischer Verwendungsnachweis wird dem Projektträger durch die Verwaltung und das Rektorat der TU Ilmenau vorgelegt.

Die wichtigsten Positionen des zahlenmäßigen Nachweises für die Projektlaufzeit nach der Umstrukturierung und des Auftretens der TU Ilmenau als eigenständiger Projektpartner beinhalten:

1. Mittel für wissenschaftliche Mitarbeiter

Hauptteil der direkt bei der TU Ilmenau anfallenden Ausgaben waren Personalkosten, die für wissenschaftliche Mitarbeiter (Einstufung E13) eingesetzt wurden. Die Mittel wurden wie geplant für insgesamt 43 Personenmonate genutzt. Beschäftigt wurden, wie in den ergänzenden Informationen zur Projektumstrukturierung 2010 beantragt, die Mitarbeiter Herr Schulz, Herr Jäger und Frau Schulze. Aufgrund ihres Vorwissens auf dem Gebiet der Modellierung und Entwurfsverfahren für komplexe technische Systeme sowie Kenntnisse mit dem eingesetzten Softwarewerkzeug MLDesigner konnten Sie die Arbeiten effizient und erfolgreich durchführen.

Darüber hinaus wurden auch geringere Mittel für die Beschäftigung studentischer Hilfskräfte bewilligt. Diese wurden für die Lösung kleinerer nicht-konzeptioneller Projektarbeiten eingesetzt.

2. Mittel für Unteraufträge

Arbeiten an der TU Ilmenau mit Bezug auf das Softwarewerkzeug MLDesigner wurden gemeinsam mit unserem langjährigen Partner Mission Level Design GmbH, Ilmenau durchgeführt. Dafür waren bereits seit Anfang des Hauptprojekts die notwendigen Mittel geplant und bewilligt. Nach der Umstrukturierung und die damit verbundene Übertragung von Aufgaben von T-Systems auf den neuen Partner Philotech wurden dafür zusätzliche Mittel auf das Teilprojekt der TU Ilmenau übertragen, die wie geplant verwendet wurden. Insgesamt war daher der Anteil der Unteraufträge am Budget der TU Ilmenau verhältnismäßig hoch. Die Arbeiten der Unterauftragnehmer erfolgten wie bewilligt und in deren jeweiligen Angeboten beschrieben. Die Rechnungen wurden von der TU Ilmenau entsprechend den vereinbarten Meilenstein- und Zahlungsplänen abgewickelt.

3. Dienstreisen

Zur Koordination des Verbundprojekts und Abstimmung der gemeinsamen Arbeiten waren regelmäßige Projekttreffen und Arbeitsgespräche notwendig. Mehrere Mitarbeiter führten die Arbeiten durch, die dann an den Treffen teilnahmen. Dadurch wurden Dienstreisemittel verbraucht, jedoch letztendlich nicht im geplanten Umfang. Dadurch konnten nach einer in 2012 vom Projektträger bewilligten Umwidmung Restmittel für die längere Beschäftigung eines wissenschaftlichen Mitarbeiters gewonnen werden.

4. Gegenstände

Für die anfallenden Simulationen und Visualisierung der Ergebnisse auch bei Dienstreisen wurden zwei leistungsfähige Arbeitsplätze mit großem Monitor, Dockingstation und schnellem Laptop wie bewilligt angeschafft. Diese sind so nicht über die Grundausstattung finanzierbar. Da die umfangreichen Modelldaten und Ergebnisse der Simulationen vor Ausfällen und unbefugtem Zugriff auf die vertraulichen Daten der beteiligten Unternehmen sicher gespeichert werden mussten, wurde wie bewilligt ein Speichersystem angeschafft.

Eine Liste der Geräte mit einem Vorschlag zur weiteren Verwendung wird mit dem Bericht beim Projektträger eingereicht. Die Geräte sollen nach Projektende im hoheitlichen Aufgabenbereich des Zuwendungsempfängers für wissenschaftliche Zwecke genutzt werden.

Um die im Gesamtprojekt gewünschten Ziele zu erreichen, ist eine integrierte Betrachtung der Vorgänge in einer komplexen Airline wie in Abschnitt 2.1 dargestellt grundlegend. Dafür waren die im hier beschriebenen Teilprojekt durchgeführten Arbeiten notwendig. Da es sich hauptsächlich um die Entwicklung und Überprüfung von Ideen mit einem konzeptionellen Modell handelt, sind wissenschaftliche Mitarbeiter nötig. Breites Wissen über die betrachteten Logistikprozesse brachte der Unterauftragnehmer Philotech ein. Die Arbeiten und damit einhergehenden Kosten waren daher für das Projektziel notwendig und angemessen.

2.3 Nutzen des Projekts und Verwertung der Ergebnisse

Der Nutzen des Teilprojekts besteht vor allem in dem nun existierenden komplexen Airline-Modell, mit dessen Hilfe Einflüsse u.a. von innovativen Wartungs- und Logistikkonzepten auf einen effizienten Flottenbetrieb mit Simulationsverfahren bewertet werden können. Verbesserungspotentiale können so entdeckt, neue Lösungen entwickelt, und diese miteinander verglichen werden.

Als universitärer Forschungspartner zielt das Fachgebiet System- und Software-Engineering der TU Ilmenau nicht direkt auf eine Nutzung als industrielles Produkt ab. Unser Interesse liegt vor allem in der wissenschaftlichen Nutzung, dem Einsatz in Forschungs- und Entwicklungsprojekten für die deutsche Luftfahrtindustrie, der Veröffentlichung der Ergebnisse, der Qualifikation von wissenschaftlichem Nachwuchs und dem Einsatz in der universitären Lehre.

Mit der Durchführung des Verbundvorhabens wurde die Kompetenz des Fachgebietes im Bereich Avionik und modellbasierter Entwurf weiter gestärkt. Die Ergebnisse dienen als Grundlage für Veröffentlichungen (siehe Abschnitt 2.5) und damit der Darstellung des Know-hows in der Fachwelt.

Die mit der Bewilligung verbundenen Auflagen zur Verwertung wurden im Einzelnen wie folgt erfüllt:

1. Die im **Vorhaben entwickelte Technologien** (Modelle, Simulationsverfahren, Ergebnisauswertung, generische Modellbibliotheken) **werden in Forschung und Lehre zur Ausbildung qualifizierten Ingenieurnachwuchses eingesetzt**. Anwendungsbeispiele aus laufenden Projekten werden für weiterführende Lehrveranstaltungen im Master-Bereich der Studiengänge Informatik, Ingenieurinformatik, sowie Research in Computer and Systems Engineering verwendet. Über die Beteiligung von Studierenden als wissenschaftliche Hilfskraft oder in Masterarbeiten an den Forschungsarbeiten wird qualifizierter Nachwuchs ausgebildet. Absolventen des Fachgebiets sind mit ihren Kenntnissen gefragte Mitarbeiter unter anderem bei deutschen Automobilbauern, Airbus, und den Zulieferern. Dadurch erfolgt auch über die Projektzusammenarbeit hinaus ein Transfer aktueller Forschungsergebnisse in die industrielle Praxis. Die im Projekt tätigen drei Doktoranden haben durch die Arbeit wertvolles Hintergrundwissen über Avioniksysteme, Modellierung und komplexen Systementwurf gewonnen, der für ihre zukünftige Arbeit bzw. Dissertationsvorhaben sehr nützlich ist.

2. Die **erzielten Ergebnisse stehen den Verbundpartnern** vereinbarungsgemäß **zur Verfügung**. Insbesondere ist das Airline-Modell, an dem ja auch direkt bei Airbus gearbeitet wurde, auf dortigen Rechnern installiert und wird genutzt.
3. Die hier gemachten Aussagen zur Verwertung gelten für die von den zwei Unterauftragnehmern Mission Level Design und Philotech im Projekt erarbeiteten Ergebnisse ebenso. Anforderungsgemäß werden daher die **durch zwei FE-Unteraufträge entstehenden Ergebnisse ebenfalls verwertet**.
4. Entsprechend den Förderzielen werden **die im Projekt erzielten Ergebnisse an deutschen Standorten** (Ilmenau, Hamburg/Finkenwerder) **verwertet**.
5. Die Verwertung insbesondere zu Punkten 1) und 2) erfolgt während der Projektlaufzeit und begann aus Sicht der TU Ilmenau als eigenständiger Partner mit der Umstrukturierung ab 2011.

Auf zukünftige Verwertungsmöglichkeiten der Projektergebnisse geht der mit dem Schlussbericht beim Projektträger eingereichte Erfolgskontrollbericht ein.

2.4 Während des Vorhabens bekannt gewordene Ergebnisse Dritter

Auf dem speziellen Gebiet des Teilprojektes sind während der Laufzeit keine aktuellen Ergebnisse Dritter bekannt geworden. Die benutzten Grundlagentechnologien wie Modellierung und Simulation entwickeln sich natürlich parallel weiter, und wurden im Projekt verfolgt.

Beispielsweise wurden als mögliche Herangehensweise an die integrierte Modellierung von Struktur und Verhalten von komplexen Systemen neue Ergebnisse der Object-Process-Methode [ODS11] veröffentlicht. Es wurde Kontakt zum Autor dieser Methode [D02] aufgenommen, Prof. Dov Dori vom Technion Haifa (Industrial Engineering and Management). Prof. Dori ist ein international ausgewiesener Spezialist auf dem Gebiet des Systems Engineering. Der Kontakt wurde unter anderem bei einem Tutorial auf der Konferenz SMC2011 vertieft, und das gemeinsame Forschungsinteresse führte zu einem Antrag in der deutsch-israelischen VISIONAIR-Förderung. Für das Jahr 2013 ist ein Forschungsaufenthalt in Haifa geplant.

Vom Projektpartner und Unterauftragnehmer im Projekt, der Mission Level Design GmbH, wurde während der Projektlaufzeit das verwendete Softwarewerkzeug MLDesigner auch außerhalb des Projektkontextes weiterentwickelt. Die Ergebnisse (verbesserte Bedienung, unter Windows lauffähige Version) konnten im Projekt genutzt werden.

2.5 Veröffentlichung der Ergebnisse

Zwischenergebnisse des Projekts wurden in folgenden wissenschaftlichen Veröffentlichungen publiziert:

- [SZY11a] Mario Schulz, Volker Zerbe, Kai Yang und Armin Zimmermann: Optimization of Avionic System Architectures. CEAS Aeronautical Journal Vol. 2, (1-4) Dezember 2011, Springer, S. 289-294.¹
- [SZY11b] Mario Schulz, Volker Zerbe, Kai Yang und Armin Zimmermann: Optimization of Avionic System Architectures. Tagungsband 60. Deutscher Luft- und Raumfahrtkongress 2011, September 2011, Bremen.
- [ZSZ10] Volker Zerbe, Mario Schulz, Armin Zimmermann, und Stephan Marwedel: Model-Based Evaluation of Avionics Maintenance and Logistics Processes. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2010), Oktober 2010, Istanbul.
- [SZM10a] Mario Schulz, Volker Zerbe und Stephan Marwedel: Using the Object Process Method to build Simulation Models. 3rd International Conference on Model-Based Systems Engineering (MBSE 2010), September 27-28, 2010. George Mason University, Fairfax, Virginia.
- [SZM10b] Mario Schulz, Volker Zerbe und Stephan Marwedel: Analyse von Wartungs- und Logistikprozessen auf der Basis ausführbarer Modelle. Deutscher Luft- und Raumfahrtkongress 2010 (DLRK 2010), Hamburg, 31. August - 02. September 2010.

Die Ergebnisse des Projekts beeinflussten auch Arbeiten in anderen laufenden Forschungsvorhaben, deren Zwischenergebnisse während der Projektlaufzeit unter anderem in [SCB12, FS12, FS11a, FS11b, FS11c, MS11, MFS11, ZB11, ZS11, BLZ10, JWZ10, MFS10] veröffentlicht wurden.

¹ Als Zeitschriften-Beitrag ausgewählte Publikation von [SZY11b]

3. Referenzen

- [AIS08] Agarwal, C.-D. Iskander, R. Shankar, G. Hamza-Lup: System-Level Modeling Environment: MLDesigner. SysCon 2008, IEEE International Systems Conference, Montreal, Canada, April 2008.
- [BLZ10] Christoph Bodenstern, Frank Lohse und Armin Zimmermann: Executable Specifications for Model-Based Development of Automotive Software. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2010), Oktober 2010, Istanbul.
- [D02] Dov Dori: Object-Process Methodology: A Holistic Systems Paradigm. Springer Verlag, 2002.
- [F09] J. Friedman: Model-Based Design Helps Aerospace Engineers. EE Times Europe, Juni 2009.
- [FS12] Nils Fischer und Horst Salzwedel: Validating Avionics Conceptual Architectures with Executable Specifications. Journal on Systemics, Cybernetics and Informatics (JSCI), 10 (12), Pages: 46-55 , ISSN: 1690-4524
- [FS11a] Nils Fischer und Horst Salzwedel: Validating Avionics Conceptual Architectures with Executable Specifications. MMMse2011, 19-22 July 2011, Orlando, Florida.
- [FS11b] Nils Fischer und Horst Salzwedel: Complex Networked Avionics Systems Design at Early Conceptual Architecture Level. DASS 2011, 3-4 Mai 2011, Dresden.
- [FS11c] Nils Fischer und Horst Salzwedel: Overcoming the Generation Gap in Aircraft Design with Executable Specifications. Digital Avionics System Conference DASC2011, 16-20 October 2011, Seattle, Washington.
- [JWZ10] Tino Jungebloud, Johannes Werner, Volker Zerbe: Beiträge zur Erhöhung der Qualität komplexer Avionik-Systeme durch modellgetriebene anforderungsbasierte Entwicklungsmethoden. Deutscher Luft- und Raumfahrtkongress 2010 (DLRK 2010), Hamburg, 31. August - 02. September 2010, Proceedings.
- [MFS11] Stephan Marwedel, Nils Fischer und Horst Salzwedel: Improving Performance and Reliability Assessments of Avionics Systems. Digital Avionics System Conference DASC2011, 16-20 Oktober 2011, Seattle, Washington.
- [MFS10] Stephan Marwedel, Nils Fischer und Horst Salzwedel: Improving the Design Quality of Complex Networked Systems Using a Model-based Approach. 3rd International Conference on Model-based Systems Engineering, MBSE2010, 27-30 September 2010, Fairfax, Virginia.
- [MS11] Stephan Marwedel und Horst Salzwedel: Improving System Availability by Optimization of Conceptual Architectures. 6th International Conference on Systems Engineering (INCOSE-IL), 8-9. März 2011, Herzliya, Israel.
- [ODS11] Carlos A. Osorio, Dov Dori, Joseph Sussman: COIM: An object-process based method for analyzing architectures of complex, interconnected, large-scale socio-technical systems. Systems Engineering 14(4): 364-382 (2011)
- [PZ10] Paunovic und Zerbe: Modeling and Simulation of Digital Systems in different Domains. 3rd SSSS- Small Systems Simulation Symposium, Nis, Serbia, Februar 2010, S. 17-23.
- [S04] Horst Salzwedel: Mission Level Design of Avionics. AIAA-IEEE DASC 04 - The 23rd Digital Avionics Systems Conference 2004, 24.-28. October 2004, Salt Lake City, Utah, USA.

-
- [S00] Gunar Schorcht: Entwurf integrierter Mobilkommunikationssysteme auf Missionsebene. Dissertation, Juli 2000, TU Ilmenau.
- [SCB12] Karin Schulze, Miguel Caldeira, João Filipe Baptista und Armin Zimmermann: Model-Based Design and Evaluation of Fault-Tolerant Fibre-Optical Networks for Avionics. 11th Int. Probabilistic Safety Assessment & Management Conference / European Safety & Reliability Conference (PSAM11 & ESREL 2012), Helsinki, Finland, Juni 2012.
- [SF08] Horst Salzwedel, Nils Fischer: Aircraft Level Optimization of Avionics Architectures, 27th Digital Avionics Conference (DASC), Oktober, 2008, St. Paul, Minnesota.
- [SZM10a] Mario Schulz, Volker Zerbe und Stephan Marwedel: Using the Object Process Method to build Simulation Models. 3rd International Conference on Model-Based Systems Engineering (MBSE 2010), September 27-28, 2010. George Mason University, Fairfax, Virginia.
- [SZM10b] Mario Schulz, Volker Zerbe und Stephan Marwedel: Analyse von Wartungs- und Logistikprozessen auf der Basis ausführbarer Modelle. Deutscher Luft- und Raumfahrtkongress 2010 (DLRK 2010), Hamburg, 31. August - 02. September 2010.
- [SZY11a] Mario Schulz, Volker Zerbe, Kai Yang und Armin Zimmermann: Optimization of Avionic System Architectures. CEAS Aeronautical Journal Vol. 2, (1-4) Dezember 2011, Springer, S. 289-294.
- [SZY11b] Mario Schulz, Volker Zerbe, Kai Yang und Armin Zimmermann: Optimization of Avionic System Architectures. Tagungsband 60. Deutscher Luft- und Raumfahrtkongress 2011, September 2011, Bremen.
- [Z03] Volker Zerbe: Mission-Level Design of Complex Autonomous Systems. XL VII ETRAN 2003 – Proc. IEEE Conference, Herceg Novi, Serbien und Montenegro, Juni 2003, S. 55-59.
- [Z07] Armin Zimmermann: Stochastic Discrete Event Systems - Modeling, Evaluation, Applications. Springer, Berlin Heidelberg New York, 2007.
- [ZB11] Armin Zimmermann und Christoph Bodenstein: Towards Accuracy-Adaptive Simulation for Efficient Design-Space Optimization. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2011), Oktober 2011, Anchorage.
- [ZS11] Armin Zimmermann und Karin Schulze: Model-Based Architectural Design of Avionic Optical Networks in the DAPHNE Project. 37th European Conference and Exhibition on Optical Communication (ECOC 2011), Symposium on Optical Avionic Networks, Geneva, Switzerland, September 2011.
- [ZSZ10] Volker Zerbe, Mario Schulz, Armin Zimmermann, und Stephan Marwedel: Model-Based Evaluation of Avionics Maintenance and Logistics Processes. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC 2010), Oktober 2010, Istanbul.