

Automatisierte Prozessanalyse und Abhängigkeitsermittlung zwischen Fachdisziplinen im Flugzeugvorentwurf

Stefan König
Hochschule Landshut





HOCHSCHULE LANDSHUT

MASTERSTUDIENGANG SYSTEMS ENGINEERING

AUTOMATISIERTE PROZESSANALYSE UND
ABHÄNGIGKEITSERMITTLUNG ZWISCHEN FACHDISZIPLINEN
IM FLUGZEUGVORENTWURF

Master-Thesis

vorgelegt von

Stefan König

Urbacher Weg 72

51149 Köln

Eingereicht:

Erstgutachter und Betreuer : Prof. Dr.-Ing. A. Brinkmann
Zweitgutachter : Prof. Dr. rer. nat., Dipl.-Inf. W. Jürgensen
Betreuer vor Ort : A. Bachmann M.A.
Deutsches Zentrum für Luft- und
Raumfahrt e.V. (DLR), Köln

Köln, August 2010

„Wer hohe Türme bauen will, muss lange beim Fundament verweilen.“

Anton Bruckner

WIDMUNG

Meinen Eltern, Karin und Werner, für die fürsorgliche Unterstützung in allen Belangen,
meiner Schwester Simone und Herrn Robert Milla,
die sich als wertvolle Lektoren erwiesen haben,
meinen Betreuern Herrn Prof. Dr. Armin Brinkmann
von der Hochschule Landshut,
Herrn Arne Bachmann und Herrn Markus Litz
vom Deutschen Zentrum für Luft- und Raumfahrt (DLR) e. V.,
die stets eine fachliche Betreuung und
Unterstützung in allen Angelegenheiten geboten haben, -
vielen herzlichen Dank.

ABSTRACT

This master thesis analyzes the possibilities of integration of a validating system into an existing IT infrastructure. It shows how an aircraft modeling system can be extended by automatically executed process analysis mechanisms to achieve valid results out of experimental test cases. The paper first introduces the given systems, their architecture and the surrounding environment like the initiated project, the cooperation between the project teams and the common knowledge base. An exemplary test case is also outlined to show weak spots in aircraft modeling. Afterwards general and specific objectives are set up which are required to meet. Based on these objectives the author presents five concepts on how to deal with the lack of validation within the operating processes. Chapter 6 finally compares the described concepts with the criteria found before. The best-fitting result is then highlighted and recommendations for future work are given. The thesis ends with a conclusion of the author and gives a short outlook at further project work.

ZUSAMMENFASSUNG

Die vorliegende Masterarbeit untersucht die Möglichkeiten ein Validierungssystem in eine bestehende IT-Infrastruktur zu integrieren. Es wird gezeigt, wie ein bestehendes Flugzeugvorentwurfssystem durch Mechanismen der automatisierten Prozessanalyse erweitert werden kann, um valide Berechnungsdaten aus experimentellen Testkonfigurationen generieren zu können. Die Arbeit stellt zuerst die gegebenen Systeme hinsichtlich ihrer Architektur und ihrer Umgebung dar. Weiterhin präsentiert sie das Projekt, auf Basis dessen die Thematik dieser Arbeit initiiert wurde, die beteiligten Personen sowie deren Zusammenarbeit. Exemplarisch wird ein Simulationsvorgang aufgezeigt anhand dessen die Schwachpunkte des derzeitigen Systems erörtert werden und anschließend ein Zielkriterienkatalog erstellt wird. Auf der Grundlage dieses Zielkriterienkataloges stellt der Autor fünf Konzepte vor und gibt beispielhaft Lösungsansätze vor. Im Kapitel 6 wird eine umfassende Nutzwertanalyse präsentiert, die die zuvor gefundenen Kriterien mit den Lösungskonzepten bewertet. Die geeignetste Alternative wird anschließend hervorgehoben. Abschließend werden Empfehlungen ausgesprochen, die für die zukünftige Arbeit an dem Projekt nützlich sein könnten. Die Masterarbeit schließt mit einem kurzen Ausblick auf weitere Projektteilbereiche und deren Verbindung zu dem Themengebiet Validierung gegeben.

INHALTSVERZEICHNIS

Abbildungsverzeichnis.....	VI
Tabellenverzeichnis	VII
Kapitel 1 : Einführung.....	1
1.1 Motivation	1
1.2 Zielsetzung.....	2
1.3 Struktur und Vorgehensweise	4
Kapitel 2 : Problembeschreibung	6
2.1 Projekt VAMP.....	6
2.2 Kooperation der Fachdisziplinen	7
2.3 Wissensmanagement	9
Kapitel 3 : Zielfeldanalyse.....	11
3.1 Beschreibung des Ist-Zustandes	11
3.2 Strukturelle Modellierung	12
3.3 Prozessflussorientierte Modellierung	15
Kapitel 4 : Zielformulierung.....	17
4.1 Analyse des Lösungsraumes	17
4.2 Definition des Zielkriterienkataloges.....	18
4.3 Festlegung des Zielinhaltes, Zielausmaßes und Zeitrahmens	19
4.4 Verfeinerung der Zielsetzung	21
Kapitel 5 : Lösungsfeldanalyse	24
5.1 Bedeutung der Modellbildung.....	24
5.2 Beschreibung von Lösungskonzepten	24
5.2.1 Systemtranszendente Konzepte	26
5.2.1.1 Dokumentation	27
5.2.1.2 Wissensplattform	28
5.2.1.3 XML Validierung	35
5.2.2 Systemimmanente Konzepte	42
5.2.2.1 Toolbasierte Semantik.....	42
5.2.2.2 Expertensystem.....	49
Kapitel 6 : Lösungsbewertung	58

6.1 Bewertung alternativer Systeme nach Erfüllungskriterien	58
6.2 Risikobewertung	61
Kapitel 7 : Ausblick und Fazit.....	66
7.1 Ausblick.....	66
7.1.1 Provenienz.....	67
7.1.2 Konfigurationsmanagement	68
7.2 Fazit.....	69
Erklärung zur Masterarbeit	VIII
Literaturverzeichnis.....	IX
Abkürzungsverzeichnis.....	XI

ABBILDUNGSVERZEICHNIS

ABBILDUNG 1: HIERARCHIE DER MESSGENAUIGKEIT	3
ABBILDUNG 2: UNTERSCHIEDUNG VON PROBLEMLÖSUNG UND FEHLERBESEITIGUNG NACH [BRI09 S. 96]	4
ABBILDUNG 3: ZENTRALES UND DEZENTRALES ENTWURFSSYSTEM.....	7
ABBILDUNG 4: VAMP - PROJEKTSTRUKTUR	8
ABBILDUNG 5: WISSEN-KÖNNEN-TREPPE NACH ROUMOIS	10
ABBILDUNG 6: KONTEXTDIAGRAMM	13
ABBILDUNG 7: VERFEINERUNG DER INTEGRATIONS- UND BERECHNUNGSPLATTFORMEN	13
ABBILDUNG 8: VERFEINERUNG DES BERECHNUNGSWEKZEUGES	14
ABBILDUNG 9: PROZESSFLUSSDIAGRAMM EINER BEISPIELHAFTEN MODELLIERUNGSKONFIGURATION	15
ABBILDUNG 10: INFORMATIONSFUSSDIAGRAMM EINER BEISPIELHAFTEN MODELLIERUNGSKONFIGURATION ...	16
ABBILDUNG 11: ANALYSE DES LÖSUNGSRAUMES	18
ABBILDUNG 12: ZIELRELATIONEN-MATRIX FÜR DIE EINSCHÄTZUNG DER SYSTEMKRITERIEN	22
ABBILDUNG 13: ZIELRELATIONEN-MATRIX FÜR DIE EINSCHÄTZUNG DER SYSTEMEIGENSCHAFTEN	23
ABBILDUNG 14: ZIEL-MITTEL-HIERARCHIE	25
ABBILDUNG 15: PROZESS DES KNOWLEDGE ENGINEERING.....	30
ABBILDUNG 16: INFRASTRUKTUR DER WISSENSPLATTFORM	30
ABBILDUNG 17: DAS SINGLE-SIGN-ON KONZEPT IM DLR	32
ABBILDUNG 18: HAUPTSTRUKTUR DES CPACS-DATENSATZES	36
ABBILDUNG 19: KLASSIFIZIERUNG VON XML-RESTRIKTIONEN NACH [HUJ03 S. 10-13].....	37
ABBILDUNG 20: KLASSIFIZIERUNG VON XML SCHEMASPRACHEN NACH LEE UND CHU (2000).....	38
ABBILDUNG 21: SCHEMATRON-VALIDIERUNGSPROZESS NACH [DÜN09 S. 94].....	40
ABBILDUNG 22: INTEGRATION EINER BETRACHTUNGSKOMPONENTE.....	43
ABBILDUNG 23: INTEGRATION EINES CHECKPOINTSYSTEMS.....	44
ABBILDUNG 24: VALIDIERUNGSKOMPONENTE IN CONTAINER-EINHEIT	45
ABBILDUNG 25: VALIDIERUNGSKOMPONENTE IN AUSFÜHRUNGSEINHEIT	47
ABBILDUNG 26: VALIDIERUNGSKOMPONENTE IN BIBLIOTHEK.....	48
ABBILDUNG 27: AUFBAU EINES EXPERTENSYSTEMS	50
ABBILDUNG 28: ENTWICKLUNGSZYKLUS EINES EXPERTENSYSTEMS	50
ABBILDUNG 29: DARSTELLUNG EINES FRAME-BEISPIELES	52
ABBILDUNG 30: SEMANTISCHES BEISPIELMODELL	53
ABBILDUNG 31: ARCHITEKTUR REGELBASIERTER SYSTEME	53
ABBILDUNG 32: DEKLARATIVE UND PROZEDURALE WISSENSREPRÄSENTATION NACH [KUR92].....	54
ABBILDUNG 33: SCHALTUNG EINES PETRINETZES	55
ABBILDUNG 34: RISIKOANALYSE UND -BEWERTUNG NACH [ISO14971]	62
ABBILDUNG 35: URSACHE-WIRKUNGSDIAGRAMM DER PROJEKTRISIKEN	63
ABBILDUNG 36: RISIKODIAGRAMM	65
ABBILDUNG 37: ARCHITEKTUR DER PROVENIENZFUNKTIONALITÄT NACH [KUN08].....	67
ABBILDUNG 38: SCHNITTSTELLEN DES PROVENIENZSYSTEMS NACH AERO-GRID-SPEZIFIKATION	68

TABELLENVERZEICHNIS

TABELLE 1: MORPHOLOGISCHES SCHEMA DES VALIDIERUNGSKONZEPTES	19
TABELLE 2: ZIELKATALOG	21
TABELLE 3: AUFBAU DER DOKUMENTATION.....	27
TABELLE 4: VOR- UND NACHTEILE DER DOKUMENTATION.....	28
TABELLE 5: VOR- UND NACHTEILE DER WISSENSPLATTFORM.....	35
TABELLE 6: BEISPIEL EINES DSD-SCHEMAS	40
TABELLE 7: BEISPIEL EINES SCHEMATRON-SCHEMAS.....	41
TABELLE 8: VOR- UND NACHTEILE DER XML VALIDIERUNG.....	42
TABELLE 9: VOR- UND NACHTEILE DER VISUALISIERUNG DER SIMULATION	43
TABELLE 10: VOR- UND NACHTEILE DES EINSATZES EINES CHECKPOINTSYSTEMS	45
TABELLE 11: VOR- UND NACHTEILE DES EINSATZES DER VALIDIERUNGSINSTANZ IN DER CONTAINER-EINHEIT	46
TABELLE 12: VOR- UND NACHTEILE DES EINSATZES DER VALIDIERUNGSINSTANZ IN DER AUSFÜHRUNGSEINHEIT	47
TABELLE 13: VOR- UND NACHTEILE DES EINSATZES EINER VALIDIERUNGSKOMPONENTE IN BIBLIOTHEKEN.....	48
TABELLE 14: VOR- UND NACHTEILE VON METHODEN DER WISSENSREPRÄSENTATION.....	55
TABELLE 15: VOR- UND NACHTEILE DES EINSATZES EINES EXPERTENSYSTEMS.....	57
TABELLE 16: PAARWEISER VERGLEICH DER OBERKRITERIEN.....	59
TABELLE 17: NUTZWERTANALYSE	60

KAPITEL 1 : EINFÜHRUNG

Inhalt dieses einführenden Kapitels ist die Darstellung der Motivation, die zu dieser Ausarbeitung führte. Einleitend werden sowohl die Gegebenheiten, die das eigentliche Thema umfassen, als auch die Ziele, die zu Ende dieses Projektes in Verbindung mit der vorliegenden Master-Thesis erreicht werden sollen, erörtert. Abschließend werden die Struktur und die Vorgehensweise aufgezeigt, die den Leitfaden dieser Arbeit identifizieren.

1.1 MOTIVATION

Die Entwicklung der Planungs- und Durchführungsqualität von Großprojekten bedurfte zu ihrer Beherrschung im Fortschritt der Zeit neue Methoden und Techniken aufgrund von steigender Komplexität. Ende des zweiten Weltkrieges wurde der Einsatz von Systems Engineering nach mehreren gescheiterten Projekten unter anderem im Bereich der Luft- und Raumfahrt (vgl. Apollo Programm¹) kontinuierlich in die Entwicklungsstrategie eingebettet und verfestigt. Durch die stetige Förderung der NASA gewann die interdisziplinäre Projektplanung und Projektkoordination ebenso für den europäischen Raum an Bedeutung. Kristallisationspunkte des Systems Engineering entstanden unter anderem in den Bereichen Anlagenbau, Telekommunikation und Flugzeugvorentwurf.

Der breitgefächerte Begriff Systems Engineering wurde aus der Vereinigung von System- und Ingenieurdenken geboren. Das Denken in Systemen, die ihre Aufspaltung in Komponenten, die wiederum Systeme bilden, und Schnittstellen finden können, sowie die ingenieurmäßige Handlungsweise, deren Vorgehensweise zur Komplexitätsbeherrschung die Systemstrukturierung unter Anwendung einheitlicher Methoden zur Problemlösung darstellt, bilden das Grundmanifest dieser fachübergreifenden Disziplin. Die Definition des Terms Systems Engineering ist aufgrund der vielfältigen Einsatz- und Interpretationsweise schwierig.

¹ <http://history.nasa.gov/SP-4102/ch4.htm> [90], [Abrufdatum: 10.05.2010]

Systems Engineering is an interdisciplinary collaborative approach to derive, evolve and verify a life-cycle balanced system solution which satisfies customers expectations and meets public acceptability. The Systems Engineering Process provides a focused approach for product development which attempts to balance all factors associated with life-cycle viability and competitiveness in a global marketplace. This process provides a structured approach for considering alternative design and configurations. The systems engineering process is applied recursively one level of development at a time. Initially, it is applied to identify the best concept, or approach, to satisfy the market opportunity. This could be a concept for a totally new product / system or a concept for making an incremental improvement to an already established product. The second application of the process adds value to the concept by fully describing the products / systems definition and establishing a configuration baseline. This application provides the basis for accomplishing the more detailed engineering development of subsystems, components and elements of a total system or the appropriated parts of an established product undergoing incremental improvement during the next application.

[IEEE94]

Systems Engineering umfasst die wesentlichen Ingenieur Tätigkeiten, die zur Entwicklung komplexer Produkte notwendig sind. Dazu gehören auch Aufgaben wie

- *Systemanalyse (System Analysis)*
- *Systemarchitekturentwicklung (System Architecture Design)*
- *Systementwicklung (System Design)*
- *Anforderungsentwicklung (Requirements Engineering)*
- *Konfigurationsmanagement (Configuration Management)*
- *Technologieentwicklung und -management (Technology Management, Obsolescence Management)*
- *Verifikation und Validierung (V&V)*

[Sch10]

1.2 ZIELSETZUNG

Als Teilaspekt des Systems Engineering definiert die Gesellschaft für Systems Engineering (GfSE) die Verifikation und Validierung komplexer Systeme, die als Haupteigenschaften qualitativer Prozesse gekennzeichnet werden. Die Validität als Gültigkeit oder Belastbarkeit von Anforderungen in Projekten fördert die nachhaltige und langfristige Qualität von Erzeugnissen. Validität erreicht mittels Operationalisierung von nicht-messbaren Faktoren die Einstufung von subjektiven Kundenanforderungen in ein Kriterienschema, das als Indikatorhierarchie der Erfüllungsqualität dient. Das geringst gewichtete Gütekriterium bezeichnet die Objektivität, d.h. Messungen sind unabhängig von den Messenden. Reproduzierbarkeit als weiteres Kriterium identifiziert die Unabhängigkeit der verwendeten Messinstrumente von gleichen Objekten mit identischen Messwerten. Das dritte Kriterium ist die Validität, die die Übereinstimmung der gemessenen Daten mit den zu messenden Größen repräsentiert. Die zu messende Größe wird hierbei durch Expertenschätzungen ermittelt.

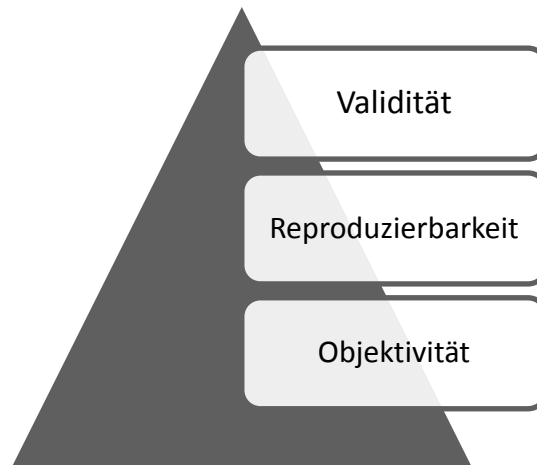


ABBILDUNG 1: HIERARCHIE DER MESSGENAUIGKEIT

Aufgabe dieser Arbeit ist die Ermittlung, Ausarbeitung und Evaluierung verschiedener Konzepte zur Validierung eines Flugzeugvorentwurfssystems. Es sollen Möglichkeiten aufgezeigt werden, die Berechnungen des Simulationswerkzeuges validieren können. Hierbei sollen keine Einschränkungen hinsichtlich der Systemgrenzen der zu evaluierenden Konzeptalternativen herrschen. Das Validierungssystem soll jedoch den Anforderungskriterienkatalog der an der Entwicklung und später der Durchführung partizipierenden Klienten verinnerlichen. Die Basis der Validierung soll ein Wissenspool darstellen, der in einer einfachen Form dargestellt, gepflegt und erweitert werden kann. Um eine langfristige und nachhaltige Wissensrepräsentationsgrundlage abbilden zu können, muss das Validierungssystem einfach in der Bedienung und pflegeleicht in der Instandhaltung sein, sodass unerfahrene Anwender nicht mangels programmiertechnischer Kenntnisse in der Einarbeitungsphase scheitern.

Die Validierung stellt ein Subsystem des Simulationsprojektes dar. Es unterliegt dabei keinerlei Eingrenzungen bezüglich der Integration. Die Architektur des Modellierungssystems wird in nachfolgenden Kapiteln umfassender erörtert. Zur Zeit der Erstellung dieser Master-Thesis ist der Vorentwurfssimulator im Stande, Flugzeugrumpf und -flügeln in verschiedenen Konfigurationen zu simulieren. Entsprechende Tools (Datenlade-, Datenextraktions-, Datenzusammenführungs-, Datenberechnungs- und Datenspeichervorgänge) stehen dem Anwender und Entwickler hierbei zur Verfügung. Die Modellierung beziehungsweise die Anordnung der Tools entscheidet über den Prozess- und Berechnungsfluss. Die Validierung soll in diesem Gesamtkontext eingreifen und den Modellierer zur Laufzeit unterstützen, indem sie ihm Einsicht über den Fehlerverlaufsprozess darstellt. Konzepte, die nicht zur Laufzeit interagieren, bedürfen jedoch nicht des Ausschlusses, sondern der Berücksichtigung unter Vorbehalt, da sie sich als wichtige Teile eines Konstrukts identifizieren lassen könnten.

Im Rahmen dieser Arbeit sollen die wesentlichen Konzeptionsaspekte vorgestellt und mit den ermittelten Anforderungskriterien abgeglichen werden. Es werden prototypisch Implementierungen und beispielhafte Wissensdaten aufgezeigt. Die vorliegende Arbeit grenzt sich von der praktischen Entwicklung der konzipierten Systeme ab. Kernpunkt sind die detaillierte Beschreibung und Evaluierung der Konzepte mittels Systems Engineering Methoden. Abschließend wird eine Empfehlung zu weiteren Verfahrensschritten ausgesprochen.

1.3 STRUKTUR UND VORGEHENSWEISE

Die Struktur und die Vorgehensweise dieser Master-Thesis bedienen sich der im Studiengang Systems Engineering an der Hochschule Landshut gelehrtene Konzepte zur Problemlösung und deren Methoden, die abschnittsweise ihre Anwendung finden, um Ergebnisse schrittweise zu sichern und Entscheidungen im Fortschritt des Projektes argumentativ zu begründen. Die Lehre des Systems Engineering bietet zwei unterschiedliche Lösungswege.

Der Problemlösungsprozess stellt eine Vorgehensweise dar, um Ziele zu identifizieren, die Formulierungen über zu erreichende oder zu vermeidende Eigenschaften eines zukünftigen Systems postulieren. Das Problem als Abweichung eines angestrebten Soll-Zustandes vom Ist-Zustand ist definiert als Betrachtungsgegenstand zweier unterschiedlicher Zeitpunkte (diachrone Betrachtung).

Im Gegensatz zur Vorhabenplanung und -ausführung ist der Fehlerbeseitigungsprozess eine Maßnahme zur Behebung einer unerwünschten Abweichung des Ist vom Soll zum derzeitigen Zeitpunkt der Betrachtung (synchron). Diesem Prozess stand keine vorherige Planung hinsichtlich Durchführung, Personal und Budget voran. Die Hauptaufgabe des Fehlerbeseitigungsprozesses ist das Auffinden von Fehlerursachen unter Berücksichtigung der vorübergehenden Behandlung von Symptomen.

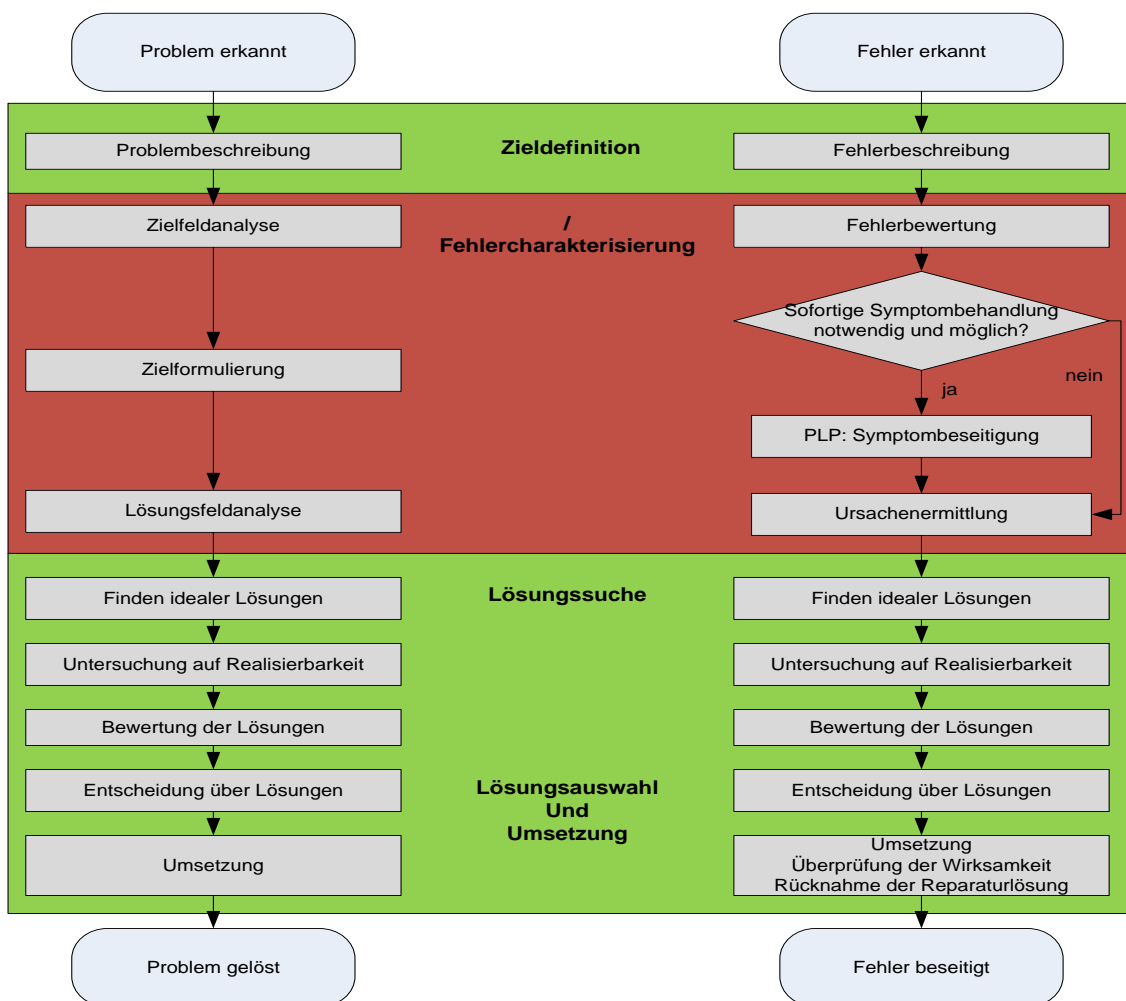


ABBILDUNG 2: UNTERSCHIEDUNG VON PROBLEMLÖSUNG UND FEHLERBESEITIGUNG NACH [BRI09 S. 96]

Die vorliegende Arbeit wird sich – gemäß der Definition – an dem Prozess der Problemlösung orientieren, da die Zieldefinition, die Lösungssuche und -auswahl Teil eines gedanklich vorweggenommenen Systems sind. Gemäß der Makrologik wird das Projekt als Problemlösungsprozess aufgefasst und realisiert. Die Struktur der Arbeit folgt deshalb der des hier aufgezeigten Lösungsweges unter Einbeziehung aller für sinnvoll und notwendig befundenen Schlüsselmethoden, die zur Ergebnissicherung sowie zur Entscheidungsbegründung dienen.

KAPITEL 2 : PROBLEMBESCHREIBUNG

In folgendem Kapitel wird das DLR-Projekt VAMP, der Initiator der vorliegenden Arbeit, kurz umrissen und die darin enthaltenen Abhängigkeiten und Kooperationen zwischen den Fachdisziplinen des DLR geschildert. Ferner wird eine Einführung in das Thema Wissensmanagement im Bezug auf das Projekt gegeben.

2.1 PROJEKT VAMP

Das Deutsche Zentrum für Luft- und Raumfahrt e.V. (DLR) mit Hauptsitz in Köln/Bonn steht im Forschungsauftrag der Bundesrepublik Deutschland. Das Institut entwickelt derzeit in Kooperation mit 16 weiteren Einrichtungen ein virtuelles Flugzeugvorentwurfssystem. Unter der Projektkennung VAMP („Virtual Aircraft Multidisciplinary Analysis and Design Processes“) wurde im Januar 2010 die Planung eines numerischen Gesamtentwurfssystems für Flugzeuge auf Vorentwurfsniveau initiiert, dessen Laufzeit auf zwei Jahre begrenzt ist.

Die Motivation, die dem Projekt zugrunde liegt, ist gekennzeichnet von den stetig steigenden Anforderungen und dem Wachstum der Luftverkehrsbranche unter Berücksichtigung der im Jahre 2008 definierten, umweltpolitischen ACARE-Ziele (Januar 2001), die unter anderem die fünfzigprozentige Reduktion der CO₂-Emission, die Reduzierung von Stickoxiden um 80 Prozent, die Halbierung (minus 10dB(A)) der Geräuschemission von Flugzeugen und Verkehrsflughäfen sowie umweltfreundlichere Industriestandards bezüglich Herstellung, Wartung und Entsorgung von Flugzeugen [ACA08 S. 26] fordern. Diese Proklamation wurde im Oktober 2004 bestätigt und durch einen breiter gefassten, systematischen Denkansatz erweitert, der die globalen Effekte der Luftfahrt sowie die Luftqualität und Geräuschemission an lokalen Standorten hinsichtlich neuartiger Technologien, alternativer Rohstoffe und Geschäftsmodelle inkludieren [ACA08 S. 27-30]. Die Verwendung kürzlich erforschter Technologien begünstigt zwar die Erreichung der ACARE-Forderungen, kann sie jedoch ohne synergetische Nutzung mit neuen Lufttransportsystemen nicht gänzlich erfüllen. Aufgrund fehlender empirischer Methoden und Erfahrungswerte stellt die Simulation des zu entwerfenden Luftfahr-

zeuges in diesem Zusammenhang die Schnittstellenkomponente zur Bewertung von erzielbaren und sinnvollen Konfigurationen durch Variation von Entwurfsparametern dar.

Die Einbettung von Analyse- und Entwurfswerkzeugen in eine Integrationsplattform sowie die Ausformulierung einer einheitlichen Flugzeugparametrisierung im Datenformat CPACS („Common Parametric Aircraft Configuration Scheme“) [Böh09 S. 36] in abgeschlossenen Vorprojekten TIVA I/II („Technology Integration for the Virtual Aircraft“) schuf die funktionelle Basis zur Verkettung von Simulationswerkzeugen, die im Projekt VAMP zur Erstellung des Gesamtentwurfs dienlich sind. Die nachfolgende Abbildung stellt das Konzept der zentralen und dezentralen Analyseverfahren mit einzelnen beziehungsweise mehreren Anwendern („Experten“) dar. Die Softwarewerkzeuge sind in einer vernetzten Systemarchitektur integriert, die es ermöglicht interdisziplinäre Aufgabenfelder und deren Resultate gemeinsam an unterschiedlichen Orten durchzuführen und zu interpretieren.

Die Thematik der Zusammenarbeit und Nutzung von Synergieeffekten spiegelt sich in der Zusammenführung von Einzeldisziplinen im Flugzeugentwurf wider. Die Integration der Kompetenzen spezialisierter Institute und Einrichtungen im DLR bildet den Kernprozess des vorliegenden Projektes.

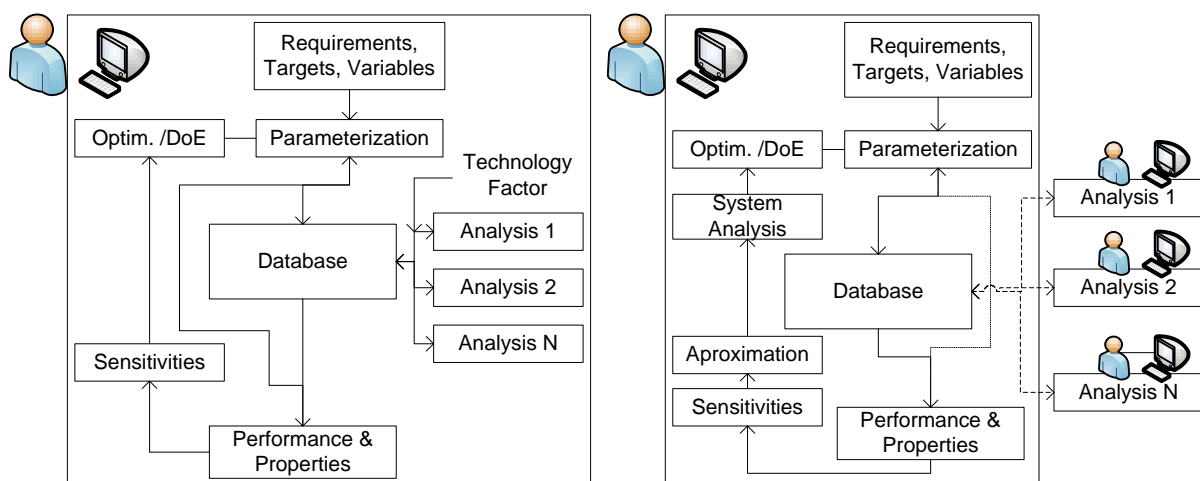


ABBILDUNG 3: ZENTRALES UND DEZENTRALES ENTWURFSSYSTEM

nach [Nag09 S. 6]

2.2 KOOPERATION DER FACHDISZIPLINEN

Die Kooperation der Fachdisziplinen bietet im Vorentwurfsprozess die Chance, Analysen und Optimierung von Flugzeugen und Testszenarien empirisch zu verfestigen und neues Wissen zu aggregieren. Im Folgenden werden die an dem Projekt VAMP beteiligten Institute und Einrichtungen des DLR aufgezählt:

- Institut für Aerodynamik und Strömungstechnik, Braunschweig (AS)
- Institut für Aeroelastik, Göttingen (AE)
- Institut für Antriebstechnik, Köln-Porz (AT)
- Institut für Bauweisen und Konstruktionsforschung, Stuttgart (BK)

- Institut für Faserverbundleichtbau und Adaptronik, Braunschweig (FA)
- Institut für Flugsystemtechnik, Braunschweig (FT)
- Institut für Lufttransportkonzepte und Technologiebewertung, Hamburg (LK)
- Institut für Robotik und Mechatronik, Oberpfaffenhofen (RM)
- Einrichtung Simulations- und Softwaretechnik, Köln-Porz (SC)

Die Aufteilung der partizipierenden Parteien gemäß der Projektstruktur (nachfolgende Abbildung) wurde im Projektplan beschlossen. Hinsichtlich der Kooperation während des Projektes wurden vierteljährliche Konferenzen vereinbart, um den Fortschritt der einzelnen Abteilungen in Berichterstattungen festhalten und eventuelle Umstrukturierungen möglichst zeitnah umsetzen zu können.

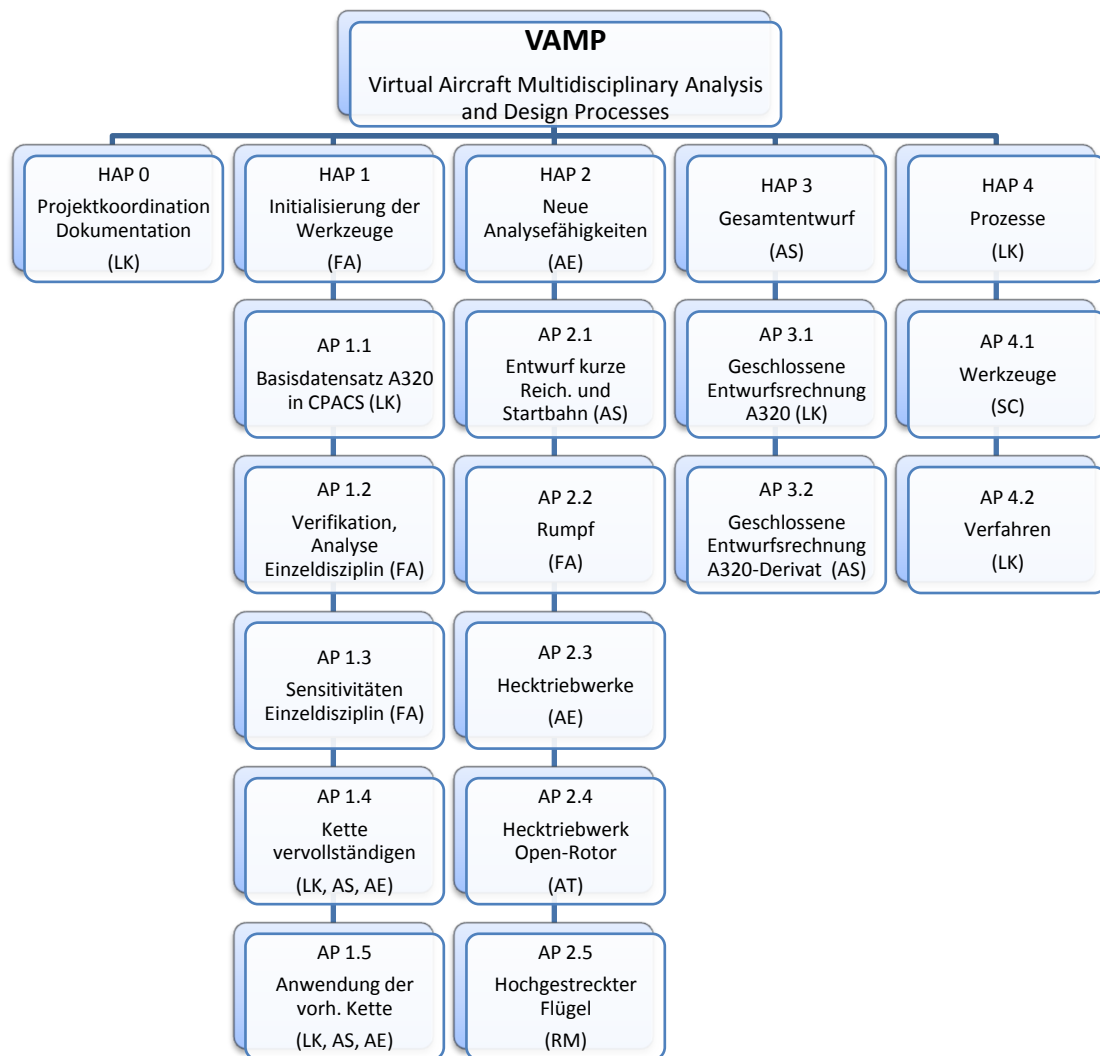


ABBILDUNG 4: VAMP - PROJEKTSTRUKTUR

nach [Nag09 S. 22]

Das Aufgabenpaket 4.1 unter der Leitung der Einrichtung SC („Simulations- und Softwaretechnik“) beschreibt als Kernprozess die Etablierung von Softwarewerkzeugen und deren interspezifische Kommunikation in den Entwurfsprozess. Zur Erreichung dieses Ziels wurden weitere Unterpakete definiert.

- UAP 4.1.1: Handhabung großer Datensätze (SC)
- UAP 4.1.2: Provenienz (Änderungsverfolgung) (SC)
- UAP 4.1.3: Plausibilität von Ketten (SC, Alle)
- UAP 4.1.4: Verwaltung von Konfigurationen (SC, LK)
- UAP 4.1.5: Pre-/Postprocessing, Visualisierung (SC, LK, Alle)
- UAP 4.1.6: Design of Experiments (DoE), Approximation, Optimierung (SC, LK, AE, AT)
- UAP 4.1.7: Weiterentwicklung CPACS und Geometriebibliotheken (SC, LK, Alle)

Die vorliegende Arbeit befasst sich mit dem Themengebiet „Plausibilität von Ketten“ (UAP 4.1.3), das zur Validierung der Kompatibilität der Werkzeuge beim Aufbau von Simulationsketten herangezogen werden soll. Hinsichtlich der Anwendbarkeit soll die Nachvollziehbarkeit von Fehlerketten, die das Verständnis des Ergebnisses sowie die anschauliche Verfolgung im Fehlerfall unterstützen, im Vordergrund stehen. Diese Vorgehensweise ist bei der Sammlung von empirischen Daten aus Simulationsmodellen von hoher Wichtigkeit, um eine Wissensbasis herzustellen, die wiederum bei der Erstellung neuer Modellketten Verwendung finden soll.

2.3 WISSENSMANAGEMENT

Seit Anfang der 90er Jahre erlangte das Management von Wissen nicht nur in Unternehmen, sondern auch in der Gesellschaft zunehmend größere Bedeutung. Verschiedene Phänomene sind seither zu betrachten: Die Gesellschaft verbringt einen größeren Teil ihrer Zeit – sei es beruflich oder privat – mit medialer Informationsverarbeitung. Wissen paart sich im Unternehmen als emporsteigender Produktionsfaktor mit den herkömmlichen Ressourcen, wie Rohstoffe, Arbeit und Kapital [Jas10]. Wissensbasierte Innovationen werden entwickelt, die auf intelligente Produkte und Dienstleistungen basieren und durch eingebettetes Wissen vervollständigt werden. Ein weiteres Phänomen ist die wachsende Zahl der Erwerbstätigen, deren Hauptberufstätigkeit darin besteht, Daten und Informationen zu generieren, zu verarbeiten und anschließend produktiv in Gewinn umzusetzen [Rou07 S. 15]. Aufgrund dieses markanten Wandels in allen Schichten des menschlichen Handelns ist zu definieren, was das Wissensmanagement beinhaltet.

„Die zentralen Grundbegriffe des Wissensmanagement (WM) sind Zeichen, Daten, Informationen und Wissen. Diese Begriffe sind in einem hierarchischen Zusammenhang zu sehen.“

[Mei10 S. 6]

Der von Meixner und Haas (Mei10) diskutierte, hierarchische Zusammenhang besteht durch steigende Komplexität und Umfang hinsichtlich der Semantik, die die Begriffe implizieren. Zeichen werden durch Syntaxregeln zu Daten, Daten werden durch Kontext und Assoziationen im Kopf des Empfängers zu Informationen oder Kompetenzen, wenn sie praktische Anwendung finden, und diese wiederum durch Vernetzung oder Vermittlung zu Wissen und zur Expertise des Empfängers. Das Wissen repräsentiert in dieser Taxonomie die Fähigkeit zur

Nutzung von Informationen in einem bestimmten Situationsumfeld (vgl. Abbildung 5: Wissen-Können-Treppe nach Roumois).

In der Literatur finden sich zwei Paradigmen des Wissensmanagements wieder: Das Wissen als Objekt (technologischer Ansatz) und das Wissen als Prozess (humanorientierter Ansatz) [Mei10 S. 8-10]. Der technologische Ansatz verfolgt den Einsatz von Informationstechnologien (Datenbanken, Expertensysteme und anderer Software) zur Speicherung, Aufbereitung, Verwaltung und Distribution von Wissen mit dem Ziel der Entscheidungsunterstützung und Prozessoptimierung. Das Business Process Reengineering (BPR), das im Zuge des Fortschrittes und der Umsetzung der Theorie des Wissensmanagements geboren wurde, bedient sich eben dieses Ansatzes, indem es Systeme aus der Informationstechnologie in neugestaltete Unternehmensprozesse integriert, um, gemäß der Definition von Hammer und Champy, das Ziel des BPR zu erreichen: „*Fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service, and speed.*“ [Ham94 S. 32] Der technologische Ansatz, auch bezeichnet als das „*Management von Wissen*“, exerziert die Akquisition von Wissen durch technische Dokumentenverwaltungssysteme. In der Regel wird bei Anwendung dieser Idee das zweite Paradigma vernachlässigt.

Wissen als Prozess beschreibt als humanorientierter Ansatz die Hegemonie des Menschen in Mitten des Wissensmanagements. Das zweite Paradigma des „*Managements für Wissen*“ konzentriert sich auf die Unterstützung der Kommunikationsprozesse zwischen MitarbeiterInnen. Die Aneignung und Akkumulation von implizitem Wissen durch Aufbau und Pflege digitaler Informationsplattformen wird häufig in ihrer Wichtigkeit in Bezug auf Nachhaltigkeit und Prækognition von Fehlern in nachfolgenden Entwicklungsprozessen unterschätzt.

Das Ziel sollte ein ganzheitlicher Ansatz des Wissensmanagements sein, der sowohl das objektorientierte als auch das prozessorientierte Wissen integrativ zur Schaffung einer neuen kognitiven Ebene des Unternehmens hinsichtlich automatisierter Prozessanalysen, Abhängigkeitsermittlungen zwischen Fachdisziplinen und Optimierung von Prozessstrukturen einbezieht.

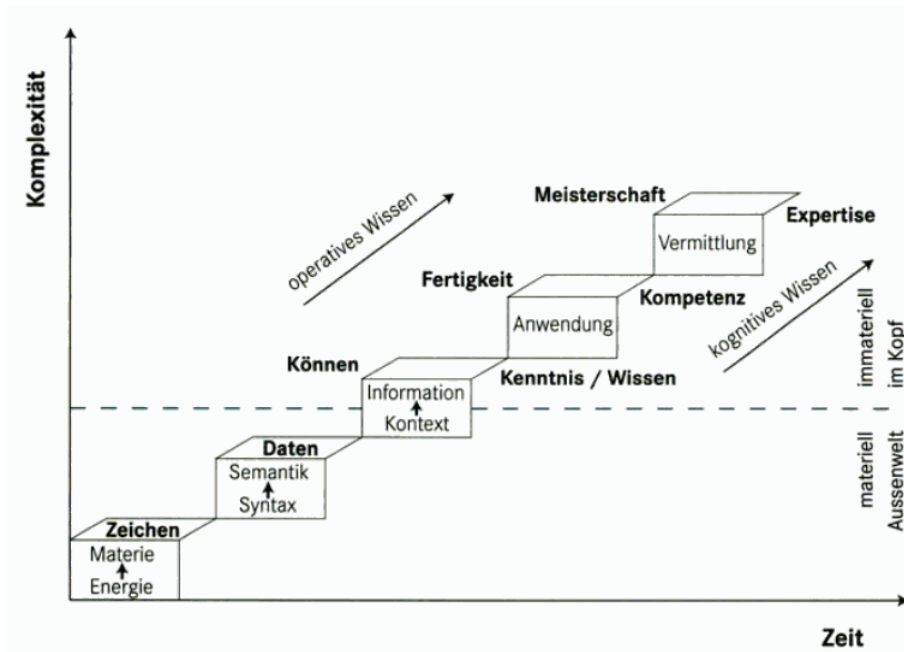


ABBILDUNG 5: WISSEN-KÖNNEN-TREPPE NACH ROUMOIS

[Rou07 S. 37]

KAPITEL 3 : ZIELFELDANALYSE

Das Kapitel Zielfeldanalyse beschreibt eingehend den aktuellen Zustand des zu analysierenden Systems. Es werden unter Zuhilfenahme studienbegleitender Methoden die strukturorientierten Systemgrenzen sowie die ablauforientierten Prozess- und Informationsflüsse aufgezeigt. Unter Berücksichtigung der vorherrschenden Gegebenheiten werden Anforderungen zusammengefasst in einem Kriterienkatalog präsentiert. Auf Basis dieser Kriterien wird in nachfolgenden Kapiteln ein Bewertungsschema gebildet, das die bis dahin gefundenen Konzepte in einer vertikalen Taxonomie abbilden wird.

3.1 BESCHREIBUNG DES IST-ZUSTANDES

Im Zuge des Projektes VAMP, das ein Gesamtvorentwurfssystem von Flugzeugen zum Ziel hat, wurde ein Datenformat ins Leben gerufen, welches die flugzeugspezifischen Parameter in prägnanter Weise repräsentiert. Die Idee einer zentralen Domäne, die als schematisierte Struktur Informationen aus Einzeldisziplinen zu einer ganzheitlichen Datenbasis modelliert, entstand aus dem ersten TIVA-Projekt („Technology Integration for the Virtual Aircraft“) des Deutschen Zentrums für Luft- und Raumfahrt (DLR). Das Datenformat dient zur Informationsspeicherung und -übertragung. Verschiedene Programme, die sich unter anderem derzeit noch in der Entwicklung befinden, nutzen das CPACS-Format. Das Ziel ist, eine einheitliche Zugriffsebene zu installieren, die Anwendern und Entwicklern jeden Fachbereiches entspricht, um die Komplexität und die Vielfalt von Tools weiter zu verringern.

Die Modellierung von Entwurfsszenarien erfolgt derzeit auf Basis einer modifizierten Version der Integrationsplattform „ModelCenter[®]“² der Firma Phoenix Integration. ModelCenter[®] ist ein Werkzeug, das Ingenieuren die Möglichkeit bietet, Entwurfsstudien automatisiert zu kreieren. Es bedient sich hierbei Konzepte verteilter Netzwerke, um entfernt operierende

² http://www.phoenix-int.com/software/phx_modelcenter.php [Abrufdatum: 12.05.2010]

Analyseprogramme einzubeziehen. Diese Programme können durch Verknüpfung der Ein- und Ausgangsparameter in einer Prozesskette angeordnet werden.

In vorliegender Arbeit wird ein beispielhafter Prozessablauf betrachtet und erläutert. Der CPACS-Datensatz wird als Zeichenkette in den jeweiligen Komponenten geladen, Teildaten werden dort modifiziert und der vollständige Datensatz zurückgesendet. Dieser Vorgang erfolgt für jede Komponente im Entwurfsprozess sequentiell. Der Prozessablauf selbst kann iterativ bezüglich eines zu optimierenden Parameters gestaltet werden, der durch ein Extrahierungswerkzeug aus dem Berechnungsvorgang entnommen und in Schritten zum Optimierungsziel hin modifiziert wird und daraufhin den Prozess wieder initiiert.

Die Hauptanforderung dieser Arbeit besteht in der Evaluierung einer Plausibilitätsprüfung der Prozessabfolge. Das System sollte – soweit wie möglich – selbstständig erkennen, ob eine geladene CPACS-Konfiguration alle zur Berechnung benötigten Daten zur Verfügung stellt und ob auf Basis dieser Daten eine Ausführung des Simulationsprogramms sinnvoll ist. Die Prüfung soll den Entwurfsingenieur bei der Modellierung der Prozessketten unterstützen.

3.2 STRUKTURELLE MODELLIERUNG

Zur Veranschaulichung wird mit Hilfe von Werkzeugen aus dem Bereich der Graphentheorie die kausale Struktur des zu analysierenden Systems dargestellt. Systemmodellierende Strukturen sind hierbei vorrangig Petrinetze. Sie stellen Kontexte aus miteinander in Verbindung stehenden Elementen her.

Definition:

Ein Petrinetz ist ein Tripel $N = (S, T, F)$ mit

- S (Stellen), T (Transitionen) sind endliche Mengen
- $S \cap T = \emptyset$ und $S \cup T \neq \emptyset$ (disjunkte Mengen)
- $F \subseteq (S \times T) \cup (T \times S)$ eine binäre Relation über $S \cup T$ (Menge der Bögen)

Ein Petrinetz wird definiert als ein bipartiter gerichteter Graph mit den zwei verschiedenen Knotenmengen S und T .

Eine Modellierungsart eines Petrinetzes ist das Kanal-Instanzen-Netz, das Systeme in hierarchischen Ebenen einteilt. Jede Verfeinerungsebene wird dabei durch ein weiteres System aus Instanzen (passive Behälter; Stellen oder Bedingungen) und Kanälen (aktive Transitionen; Aktionen) repräsentiert, die durch gerichtete Pfeile verbunden sind. Eine Marke durchläuft das Netz und stellt den Zustand einer Bedingung dar.

Die Methodik des Modellierens verfolgt den Top-Down Ansatz, der die Verfeinerung vom Groben ins Detaillierte darstellt. Das zu modellierende System wird hierbei mittels Diagrammen analysiert. Die höchste Ebene stellt das Kontextdiagramm dar, das einen Überblick über die systemspezifischen Komponenten und Schnittstellen bietet, die die intraspezifische Kommunikation realisieren. Anschließend werden die Instanzen, in separater Betrachtung und unter Berücksichtigung angrenzender Schnittstellen, in weiteren Unterdiagrammen verfeinert. Ist die feinste Stufe eines Systems erreicht und eine weitere Verfeinerung er-

scheint nicht mehr sinnvoll, wird die Modellierung der Verlaufsstrukturen angestoßen. Die Schnittstellen werden nicht weiter verfeinert.

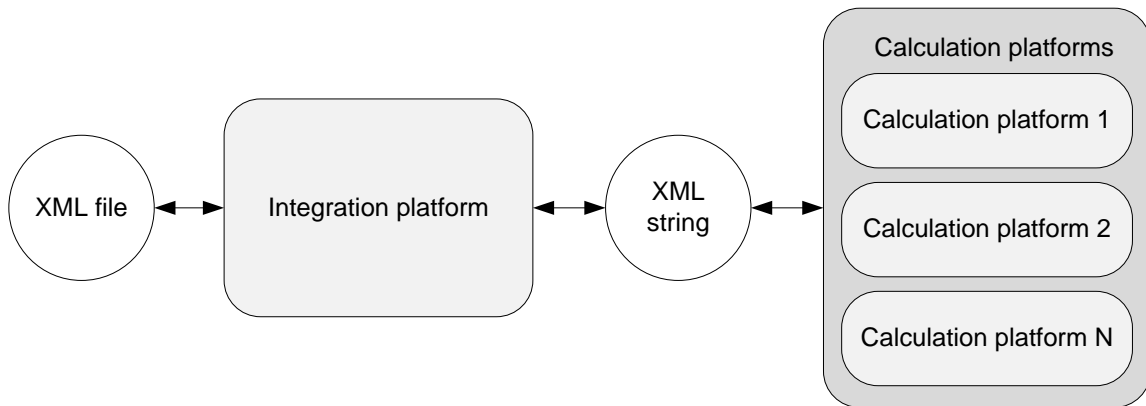


ABBILDUNG 6: KONTEXTDIAGRAMM

Die Komponente „Integration platform“ wird nachfolgend einzeln dargestellt und in einer weiteren Ebene verfeinert analysiert. Die Schnittstelle „Daten“, die von der Komponente lesend und schreibend genutzt wird, wird in diese Ebene übernommen. Da die Konfiguration von Berechnungswerkzeugen eine große Anzahl an Möglichkeiten bietet, wurde eine beispielhafte Anordnung gewählt, die iterativ drei Berechnungen und einen Optimierungsvorgang vollzieht, der wiederum den Kalkulationsvorgang starten kann.

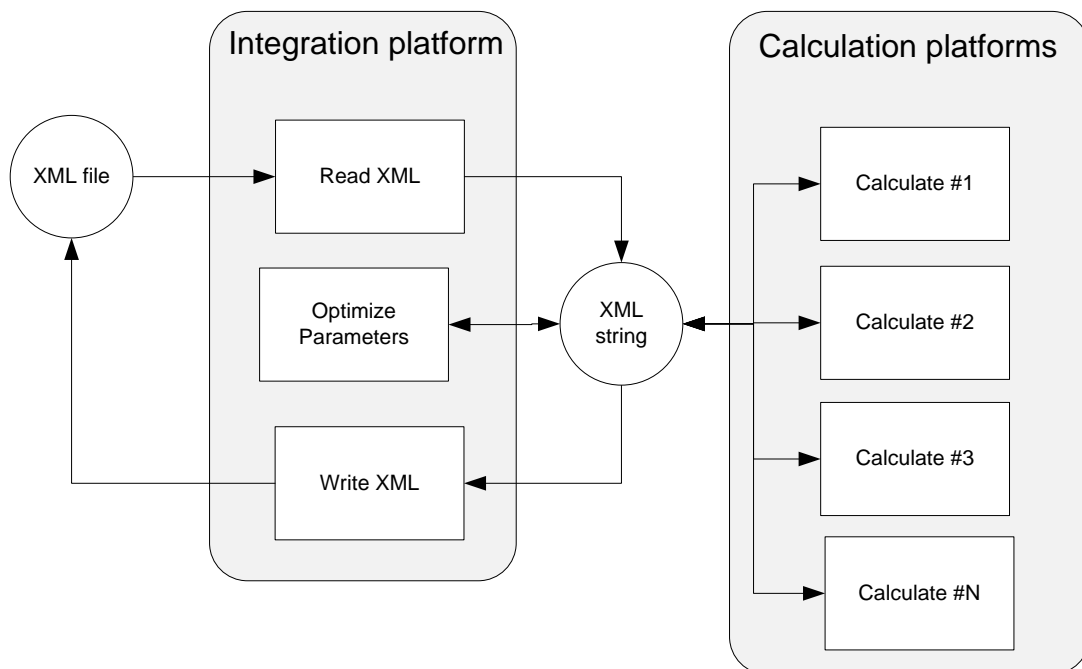


ABBILDUNG 7: VERFEINERUNG DER INTEGRATIONS- UND BERECHNUNGSPLATTFORMEN

Das Verfeinerungsdiagramm der Integrationsplattform zeigt den statischen Aufbau. In dieser findet man eingebettete Komponenten, die modular programmiert, integriert und zur Berechnung von verschiedenen Flugzeugvorentwurfsbereichen herangezogen werden können.

Die Anordnung der Komponenten bestimmt den prozessorientierten Ablauf der Vorentwurfsberechnung. Die Plattform bietet Optimierungskomponenten, die in verschiedenen Algorithmen gewählte Parameter aus dem Berechnungsvorgang extrahieren, schrittweise den vom Experten eingegebenen Grenzbereichen oder Zielfunktionen annähern und anschließend den Prozess an verknüpfter Stelle erneut initiieren.

Um die Struktur ganzheitlich betrachten zu können, müssen zudem die Komponenten der Berechnungsplattformen detaillierter analysiert werden. In einer weiteren Verfeinerungsebene werden nachfolgend die Strukturen der Berechnungswerkzeuge allgemein definiert. Aufgrund der inhaltlichen Differenz der Werkzeuge, wird der Fokus auf die der Komponenten gemeinsamen Strukturen und Schnittstellen gelegt.

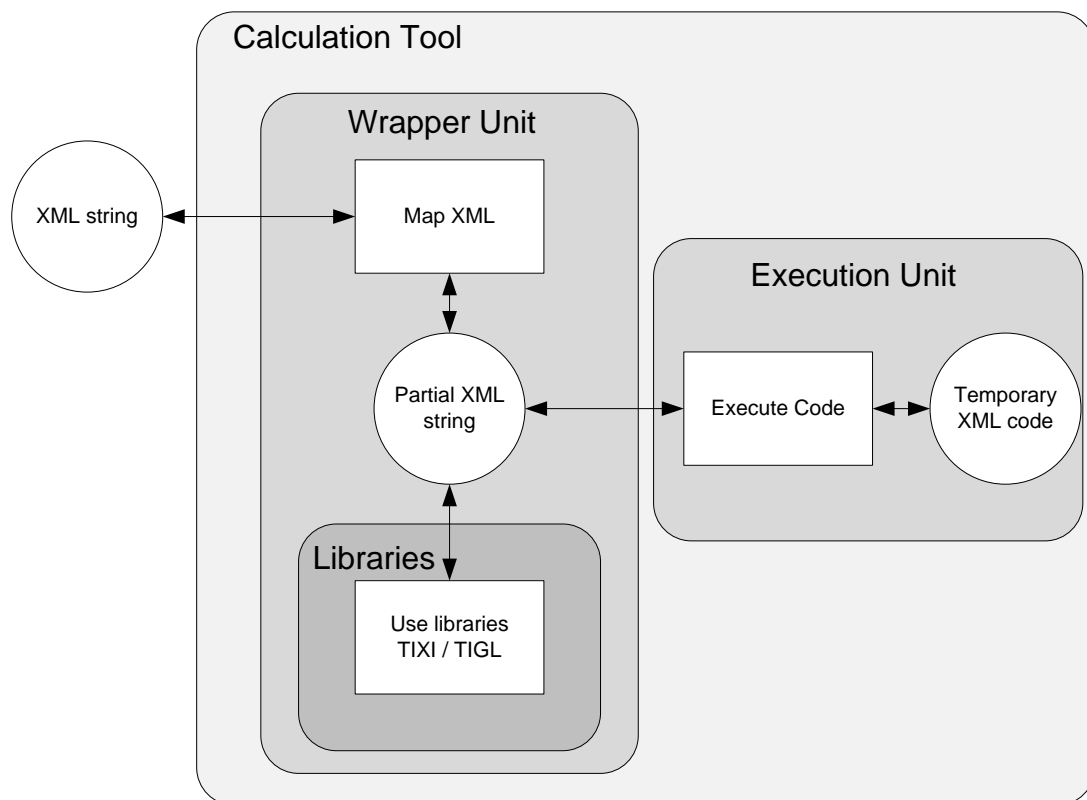


ABBILDUNG 8: VERFEINERUNG DES BERECHNUNGSWEKZEUGES

Die Berechnungskomponenten sind in Datenvorbereitungs- und Datenverarbeitungsbereiche separiert. Die Datenvorbereitungseinheit dient zur Extraktion von benötigten Daten aus dem Gesamtkontext. Die extrahierten und modifizierten Daten werden in einem der Ausführungseinheit zur Verfügung stehenden, temporären Speicherbereich zwischengespeichert. Die Bibliotheksdateien TIGL („TIVA Geometric Library“) und TIXI („TIVA XML Interface“), die sowohl Zugriffoperationen für das CPACS Datenformat sowie geometrische Berechnungs- und Exportfunktionen bereithalten, unterstützen den Datenverarbeitungsvorgang in der Ausführungsebene. Diese Bibliotheken stammen aus den zeitlich vorangegangenen Projekten TIVA I/II.

Nachdem der statische Aufbau des Modellierungssystems beleuchtet wurde, ist es notwendig, prozessorientierte Flüsse aufzuzeigen, um die Verhaltensweisen von Akteuren sowie Effekte zusammenwirkender Einheiten der strukturellen Ebenen zu untersuchen.

3.3 PROZESSFLUSSORIENTIERTE MODELLIERUNG

Die Modellierung des Prozesses, der auf eben dargestellter Struktur basiert, minimiert die Komplexität des zu analysierenden Systems, indem sie den Übergang von Eingaben zu Ausgaben über Verarbeitungssysteme identifiziert. Der Prozessfluss beginnt bei der Modellierung von Komponentenketten, die die Anwender in der Entwurfsplattform vornehmen. Anschließend wird der Simulationsvorgang durch das Laden der Parameter aus der CPACS Datei initiiert. Die Entwurfsplattform steuert gemäß der Eingabe des Benutzers die Daten und die von den Analysis Komponenten zurückgelieferten XML Datenströme.

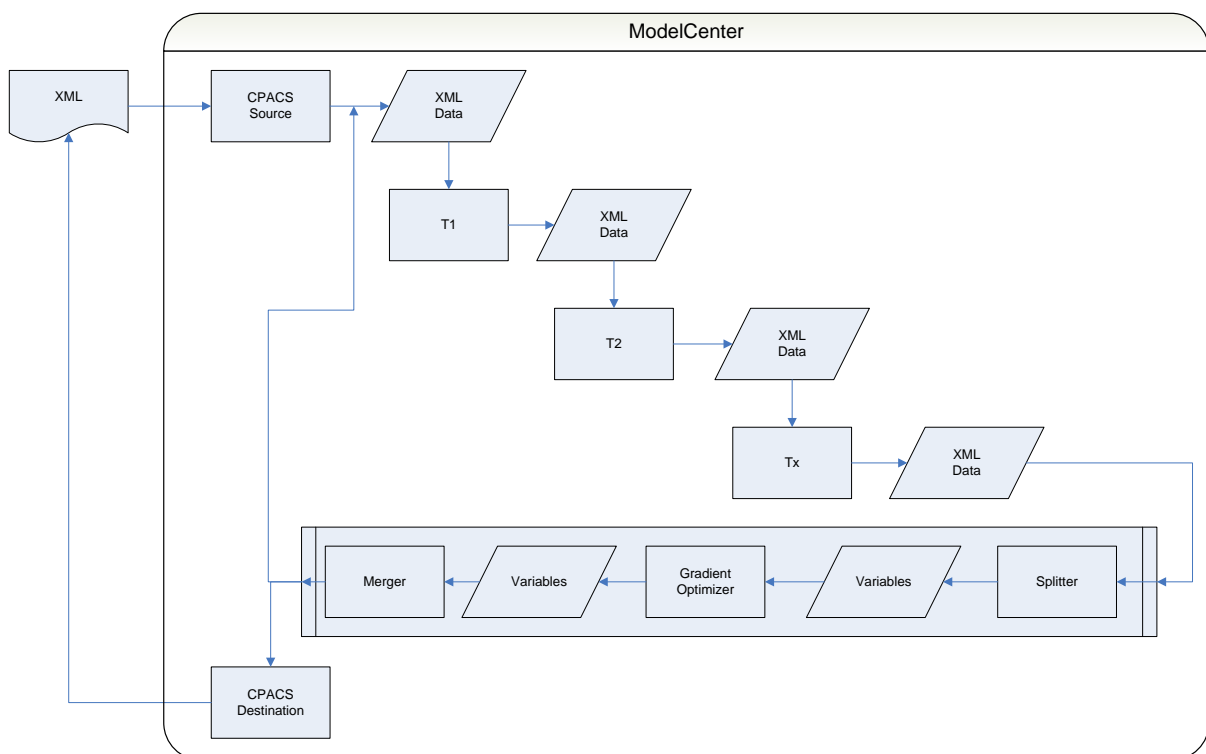


ABBILDUNG 9: PROZESSFLUSSDIAGRAMM EINER BEISPIELHAFTEN MODELLIERUNGSKONFIGURATION

Diese Komponentenanzordnung optimiert Berechnungen hinsichtlich einer gegebenen Grenze oder Funktion. Sie wird solange durchlaufen bis die Optimierungseinheit das Abbruchsignal sendet. Um den Ablauf zu verdeutlichen, wird die Darstellung in ein Informationsflussdiagramm übertragen.

Zusammenfassend lassen sich bereits Interventionsbereiche hervorheben, die als Ansatzpunkte für mögliche Lösungswege interessant sind. Darunter fallen unter anderem die Schnittmenge, die die Modellierungs- und Ausführungsinstanzen bilden, sowie der Datensatz als Kommunikator zwischen den Instanzen. Die extrahierten Schnittstellen werden in dem Themengebiet Lösungsfeldanalyse detaillierter betrachtet. Um relevante Lösungen zu erhalten, müssen zuerst Randbedingungen definiert und überprüft werden.

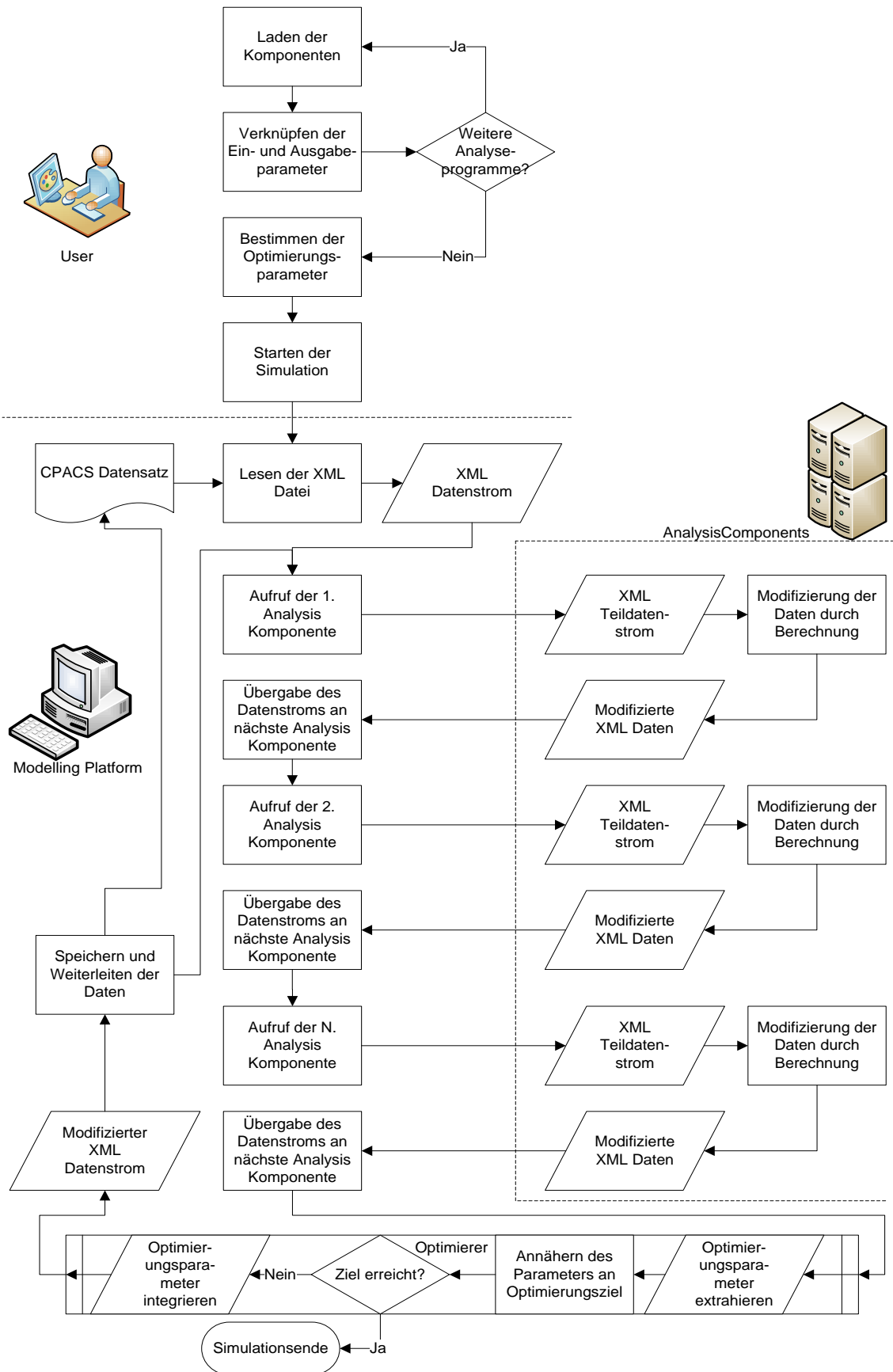


ABBILDUNG 10: INFORMATIONSFLUSSDIAGRAMM EINER BEISPIELHAFTEN MODELLIERUNGSKONFIGURATION

KAPITEL 4 : ZIELFORMULIERUNG

Die Zielformulierung beschäftigt sich mit der Thematik des zu erreichenden Zustandes, indem sie Eigenschaften deklariert, die entweder initiiert oder vermieden werden sollen. Eine Gefahren- und Chancenanalyse wird gestaltet, die a priori den Einsatz eines Validierungssystems wertschätzt. Unter Berücksichtigung vorangegangener Situationserkenntnisse folgt anschließend die Konkretisierung der Zielsetzung sowie der Kriterien, die eine spätere Bewertung der vorgestellten Konzepte ermöglichen.

4.1 ANALYSE DES LÖSUNGSRRAUMES

Das Ergebnis der Zielformulierung ist die Definition der Zielinhalte, die Abgrenzung der Ziele (Zielausmaß) und der Zeitrahmen, in dem die zu erreichenden Ziele umzusetzen sind. Waren es in der Problembeschreibung vage Aussagen und Wünsche, so sind diese in der Zielformulierung durch Operationalisierung zu konkreten Grundsätzen umgeformt. Haberfellner formuliert dies wie folgt: „Zielvorstellungen ergeben sich meistens bereits im Zusammenhang mit dem Anstoß und sind deshalb auch schon grob und global in einem Auftrag enthalten. Wichtige Impulse ergeben sich vor allem während der Situationsanalyse: Mit erkannten Mängeln, Schwierigkeiten oder Chancen entstehen bei den Beteiligten auch präzise Vorstellungen über mögliche und gewünschte Veränderungen. Außerdem liefert eine lösungsorientierte Betrachtung in der Situationsanalyse Hinweise auf Vorbilder bzw. den Stand der Technik, welche die Zielinhalte beeinflussen können und auch sollen.“ [Hab99]

Die Konkretisierung der geforderten Ziele geht mit der verfeinerten Analyse des Lösungsraumes einher, in dem sich die auszuformulierenden Konzepte bewegen können. Der Lösungsraum umfasst Akteure, graphische Benutzerflächensysteme sowie Werkzeuge, welche im Hintergrund passiv auf Befehle der Benutzerebene reagieren. Ein Validierungssystem kann als Schnittstelle zwischen zwei Systemen oder Akteuren gestaltet oder in einem System – zumeist das Kernsystem im Betrachtungsszenario – eingebettet werden.

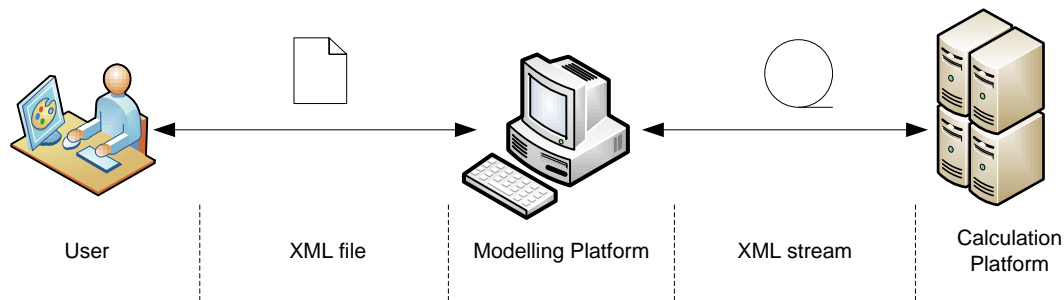


ABBILDUNG 11: ANALYSE DES LÖSUNGSRAUMES

In vorliegendem Fall werden die Systeme Benutzer, Modellierungsplattform und Berechnungsplattform ebenso wie die Schnittstellen XML-Datei und XML-Datenstrom inspiziert. Die Kategorisierung des Gesamtsystems in Kommunikations- und Bearbeitungsebenen dient der Reduktion von Komplexität in der Analyse des Lösungsraumes. Konzepte zur Validierung erstrecken sich zum Teil bereichsübergreifend auf Systeme und Schnittstellen. Das Resultat einer Validierung wird übertragen auf das hierarchisch nächstgelegene System. Das Hauptziel einer Validierung ist die Information des Benutzers und die darauffolgende Interpretation. Nachfolgend wird eine Ideensammlung dargestellt, die den Rahmen des Lösungsraumes definieren wird. Weiterhin wird ein, dem Zielkonzept entgegen gerichteter Sollkriterienkatalog erstellt, der durch System Engineering Methoden hinsichtlich folgender Aspekte beurteilt: Zielkonflikt (Widerspruch), Zielkonkurrenz (Gegenläufigkeit), gegenseitige Unterstützung und Indifferenz (Unabhängigkeit).

4.2 DEFINITION DES ZIELKRITERIENKATALOGES

Die Anforderungen an das Zielsystem wurden anhand von Projektmanagementmethoden, die fester Bestandteil des Systems Engineering Konzeptes sind, mit den an dem Projekt VAMP beteiligten Personen der Einrichtung Simulations- und Softwaretechnik erarbeitet. Nachfolgend wird die Ideensammlung für ein Validierungskonzept dargestellt:

- Der Ausführungsort kann entweder in der Integrationsplattform selbst, in einer eigenen Applikation, die beispielsweise als Komponente in die Integrationsplattform eingebunden wird (Checkpoint-System), oder als Bibliotheksdatei vorhanden sein.
- Das Konzept könnte auf eine Wissensdatenbank zurückgreifen, die sowohl Regeln für Berechnungen als auch Kenntnis über bereits verarbeitete Berechnungen und deren Resultate beinhaltet.
- Der Ausführungszeitpunkt der Validierung könnte bei Start des Simulationsablaufes (a priori), zur Laufzeit selbst (Echtzeit), bei der Erstellung des Workflows oder zur Analyse (a posteriori) gewählt sein.
- Das Resultat der Validierung könnten gespeicherte Log-Einträge sein, die entweder zur Laufzeit oder im Nachhinein ausgewertet werden. Ebenso besteht die Möglichkeit einer interaktiven Wirkung, die den Anwender Entscheidungen im Zweifelsfall präsentiert.

- Der Schweregrad der Meldungen könnte zwischen Empfehlung, Information, Warnung und Fehler variieren.
- Die Einbindungsform der Validierung könnte als externes System, Bibliothek oder intern stattfinden.
- Die Möglichkeit zur Nutzung synergetischer Effekte in Bezug auf weitere Anforderungen, wie zum Beispiel die Änderungsverfolgung (Provenienz), das Pre- und Post-processing sowie die Visualisierung von Datenströmen könnte ebenso Einfluss auf das Zielkonzept haben.

Ausführungsort	Integrationsplattform		eigene Applikation		Bibliothek	
Provenienz	interspezifisch			intraspezifisch		
Fähigkeit zum Lernen	Ja			Nein		
Ausführungszeitpunkt	bei Start	zur Laufzeit	bei Erstellung des Workflows		Interaktiv	
Wirkung	Logging			Interaktiv		
Schweregrad der Fehler	Information	Empfehlung	Warnung		Fehler	
Einbindungsform	Externes System		Bibliothek		Intern	

TABELLE 1: MORPHOLOGISCHES SCHEMA DES VALIDIERUNGSKONZEPTES

4.3 FESTLEGUNG DES ZIELINHALTES, ZIELAUSMAßES UND ZEITRAHMENS

Die Ziele, die im Zielformulierungsprozess mit den verschiedenen Interessenskreisen diskutiert wurden, beinhalten nicht nur die systemspezifischen Anforderungen, sondern betreffen darüber hinaus Teilbereiche des Unternehmens. Der Katalog klassifiziert Ziele in Haupt- und Unterzielkategorien. Die Zielunterklassen definieren Maßstäbe (operationalisierte Aussagen) und Bedingungen sowie Prioritäten, die jeweilige Ziele in Muss-, Soll- und Wunschziele einstufen.

- **Muss-Ziel (M):** Die Bedingung (Restriktion) muss zwingend umgesetzt werden.
- **Soll-Ziel (S):** Dieses Ziel ist wichtig. Die Bedingung soll eingehalten werden.
- **Wunschziel (W):** Das Ziel ist zu beachten. Die Einhaltung ist erwünscht.

Die Einteilung der Zielklassen erfolgt in Finanzziele, Funktionsziele und Personalziele.

DLR	ZIELKATALOG	
	Projekt/Systemkomponente: Plausibilität von Ketten	Datum: 14.05.2010
	Phase: Detailstudien	Zieldatum: 08.08.2010
	Zielobjekt: Flugzeugvorentwurf	
	Allgemeine Zielformulierung: Plausibilität von Vorentwurfketten	

ZIELKLASSE	Zielformulierung unter Verwendung der Maßstäbe	Bedingung/ Restriktion	Priorität
Zielunterklasse			
<i>Finanzziele:</i>			
Wirtschaftlichkeit	Kosteneinsparung durch Vermeidung von nicht sinnvollen Berechnungen		S
Belastung der Liquidität	möglichst geringe Anschaffungskosten in Person Jahren (PJ) geringe Lizenzkosten	< 0,5 PJ	S
			W
			W
		< 500 €	W
<i>Funktionsziele:</i>			
Leistung/ Funktionalität	Reduktion der Fehlkalkulationen		M
	Geringe Leistungsanforderungen		W
	Sicherung des Validierungsergebnisses		M
	Hohe Genauigkeit der Fehlerbeschreibung		M
	Fehlerkategorisierung		W
	Erkennung von Fehlkonfigurationen		S
	Präventive Fehlererkennung		W
	Reduzierung der Modellierungsschritte		W
	Rückwirkende Fehlerauswertung		M
	Hohe Interaktivität / niedrige Fehlerreaktionszeit		W
	Lernfähigkeit des Systems	< 60s	W
Sicherheit/ Zuverlässigkeit	Ausfallwahrscheinlichkeit möglichst gering	< 0,01%	S
	IT-Sicherheit soll möglichst hoch sein		W
Belastbarkeit/ Flexibilität	Anpassungsfähigkeit des Validierungskonzept		W
	Gleichzeitige Ausführung von Simulationen		W
Erweiterbarkeit	Erweiterbarkeit des Konzeptes soll möglich sein bzgl. anderen Add-Ons		W
Schnittstellen- erfordernisse	Integrierbar in vorhandene Plattformen		S
Wartbarkeit	kurze Instandsetzungszeit bei Hard- und Softwarefehlern	< 24h	S
	Möglichst geringer Administrationsaufwand	< 5h/w	W
Benutzerfreundlichkeit	kurze Einführungsphase		S
<i>Personalziele:</i>			

Anforderungen an Personalqualifikation	keine speziellen Anforderungen an das Bedienungspersonal		W
--	--	--	---

TABELLE 2: ZIELKATALOG

4.4 VERFEINERUNG DER ZIELSETZUNG

Das Kapitel Zielformulierung schließt mit der Gegenüberstellung von Zielen und Mitteln ab. Nachdem nun die Ziele konkretisiert worden sind, werden sie nun in hierarchisierter Form analysiert. Diese Vorgehensweise begünstigt die Findung von Maßnahmen, die zur Erreichung von nicht-operationalisierten Zielen notwendig sind. Eine Zielrelationen-Matrix stellt dar, ob sich Ziele gegenseitig unterstützend, unabhängig voneinander, gegenläufig oder im Widerspruch zueinander verhalten.

In wiederkehrenden Diskursen mit Verantwortlichen des Projektes VAMP in der Einrichtung Simulations- und Softwaretechnik des DLR sind Kriterien gefunden worden, die sowohl die Eigenschaften für ein Validierungssystem als auch dessen Funktionsumfang beschreiben. Das System Engineering bietet eine Methode, um Relationen zwischen Zielkriterien subjektiv abzuwägen. Zu unterscheiden sind hierbei vier Eigenschaften, die Kriterien in Beziehung miteinander innehaben können: Bei gegenseitiger Unterstützung (u) und Unabhängigkeit oder Indifferenz (i) sind keine Probleme in der Zusammenarbeit erkennbar. Ist die Korrelation zweier Kriterien konkurrierend/gegenläufig (g), sind Kompromisse in diesem Bereich anzustreben. Stehen Kriterium A und Kriterium B in Widerspruch (w) zueinander, sind drei Methoden der Konfliktlösung abzuwägen. Erstens besteht die Möglichkeit der Prioritätensetzung, das heißt ein Muss-Ziel wird zu einem Soll- oder Wunschziel verändert. Zweitens ist die Einführung von Mindest- und Höchstwerten in Betracht zu ziehen: Hohe Kosten für ein Produkt, dafür aber auch möglichst viel Leistung. Sind diese aufgezeigten Konfliktlösungen nicht durchsetzbar, muss die Vereinbarung der Streichung eines Konfliktverursachers getroffen werden.

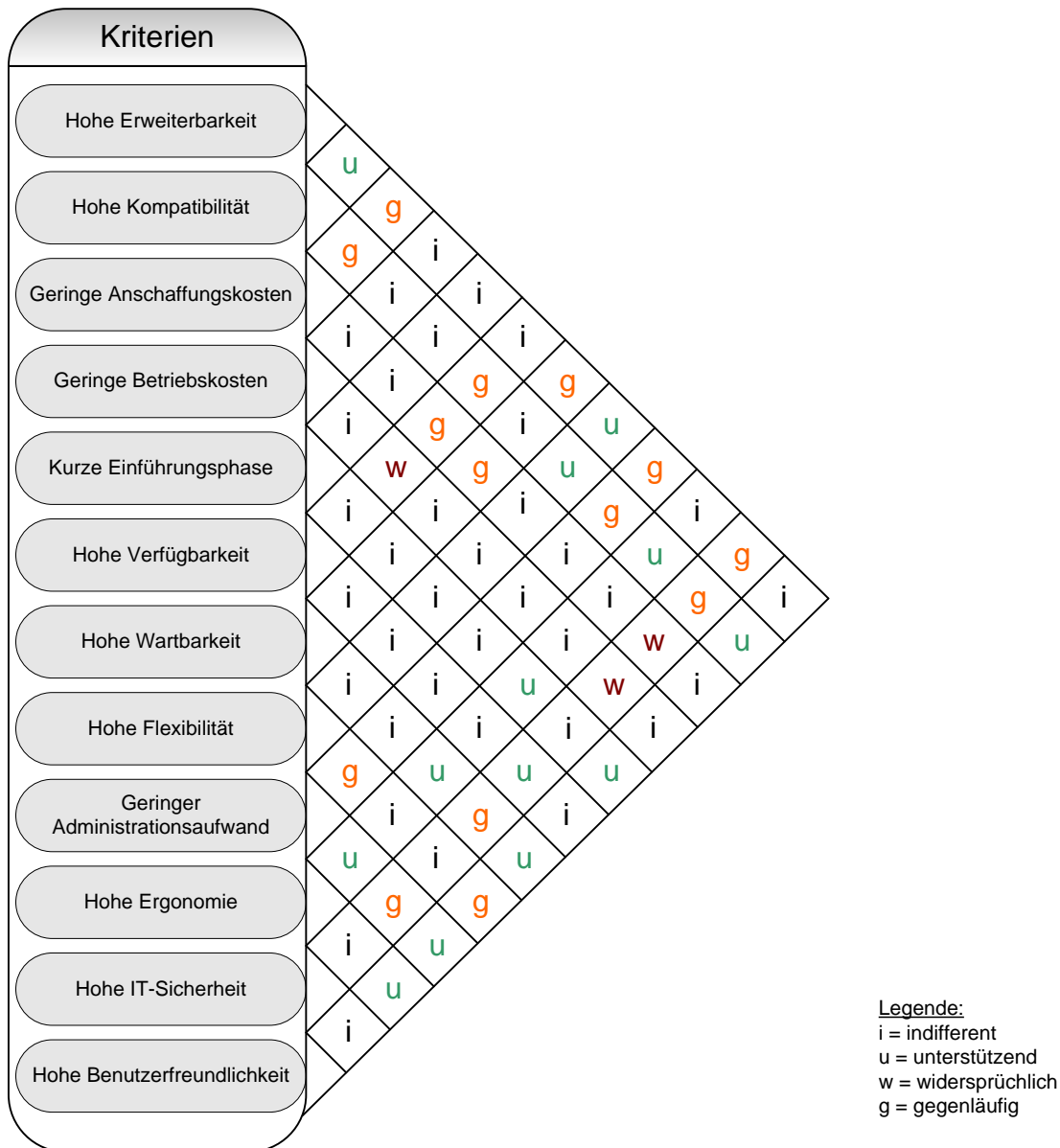


ABBILDUNG 12: ZIELRELATIONEN-MATRIX FÜR DIE EINSCHÄTZUNG DER SYSTEMKRITERIEN

In obiger Abbildung sind drei widersprüchliche Beziehungen zu diskutieren. Die hohe Verfügbarkeit des Systems steht in negativer Beziehung zu den geringen Betriebskosten. Hier wurde die Methode der Prioritätensetzung angewandt, indem das Betriebskostenziel auf Wunsch-Niveau gestuft wurde. Ebenso widerspricht die gewünschte, hohe IT-Sicherheit den zu erzielenden Anschaffungs- und Betriebskostenzielen. Auch hier wurde der Ansatz der Prioritätensetzung angewandt und die IT-Sicherheit hinter den Kostenzielen gesetzt.

Nach äquivalenter Vorgehensweise werden die Kriterien analysiert, die den Funktions- und Leistungsumfang des Validierungssystems darstellen sollen.

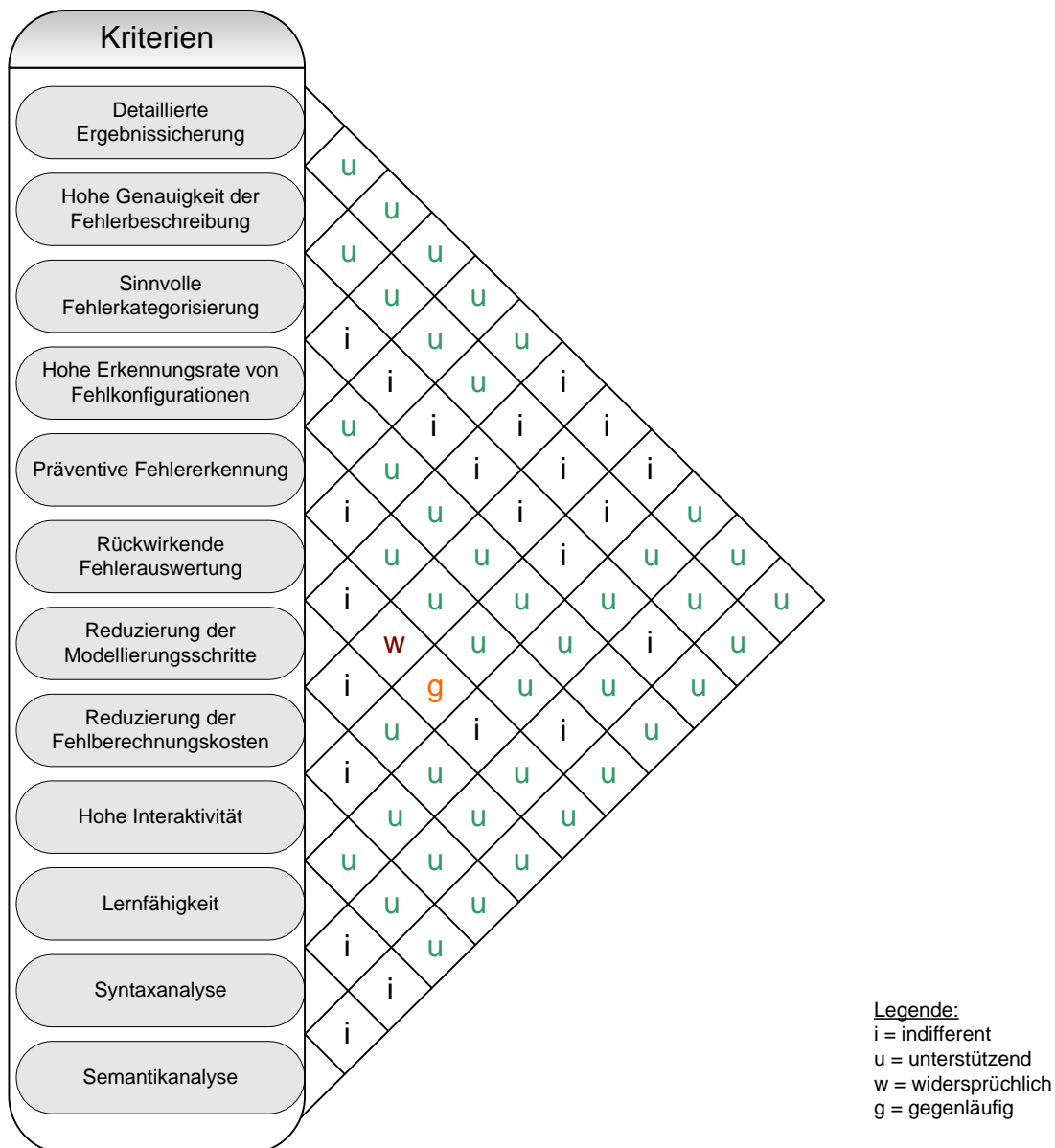


ABBILDUNG 13: ZIELRELATIONEN-MATRIX FÜR DIE EINSCHÄTZUNG DER SYSTEMEIGENSCHAFTEN

Die Widerspruchsfreiheit ist in dieser Konstellation nicht gänzlich gewährleistet. Die Relation zwischen der Forderung nach einer rückwirkenden Fehlerauswertung widerspricht dem Prinzip der Reduzierung der Fehlberechnungskosten in der Art, dass eine bereits durchgeführte Berechnung und die damit verursachten Kosten mittels einer nachfolgenden Fehlerauswertung nicht mehr rückgängig gemacht werden kann. Hier ist einstimmige Rücksicht auf die Fehlberechnungskosten zu geben. Die rückwirkende Fehlerauswertung kann als zusätzlicher Unterstützungsprozess in einem Validierungssystem dienlich sein.

KAPITEL 5 : LÖSUNGSFELDDANALYSE

Dieses Kapitel widmet sich der Synthese von Situationskenntnissen und Zielformulierungen, die aus vorangegangenen Analysen gewonnen wurden. Die Lösungsfeldanalyse verinnerlicht die wirkungsbezogene, zielorientierte und die strukturbezogene, lösungsorientierte Betrachtung und destilliert die möglichen Konfliktlösungen aus den gegebenen Problemstellungen. Es werden Konzepte und Lösungsvarianten dargestellt, die die beschriebenen Kernprozesse optimieren sollen. Dabei wird eine Einteilung in systemtranszendente, das heißt außerhalb des Systems agierende Konzepte, und systemimmanente, das heißt innerhalb des Systems agierende Konzepte, durchgeführt und Lösungsalternativen beschrieben.

5.1 BEDEUTUNG DER MODELLBILDUNG

„Die Behandlung komplexer Probleme erfordert die Abstraktion von konkreten Sachverhalten. Diese Abstraktion findet vorerst im Problemfeld statt und führt häufig von graphisch-anschaulichen Abbildungen der Realität zu quantifizierten graphischen Strukturmodellen und ggf. bis zur mathematischen Abstraktion.“ [Hab99 S. 159] Die Lösungsfeldanalyse bedingt nach Haberfellner et al. drei Dinge. Erstens findet eine „Erahnung“ oder „Erführung“ eines Konzeptes des späteren Systemkomplexes statt. Zweitens begibt man sich auf die Suche nach den erforderlichen Elementen, die zur Problemlösung beitragen und fügt im letzten Schritt die Teile gedanklich und modellhaft zu einem Ganzen zusammen. Diese Synthese erfolgt in Konkretisierungsstufen, die zunehmend die ersuchten Ergebnisse herauskristallisieren. Es gilt, eine Vielzahl von Lösungsvarianten zu erstellen, die anhand der vorweggenommenen Kriterien zu messen sind.

5.2 BESCHREIBUNG VON LÖSUNGSKONZEPTEN

Abweichend von der bisherigen Betrachtung des Ist-Zustandes stehen nun Kreativitätstechniken im Vordergrund, die zur Findung von Lösungsvarianten eingesetzt werden. Es können

Matrizen zur Top-Down Analyse herangezogen werden, die Ziele durch Ziel-Mittel-Zwischenebenen bis zu konkreten Mitteln verfeinern. Dies wurde in Kapitel 4.4 VERFEINERUNG DER ZIELSETZUNG bereits angewandt und wird nachfolgend im Detail erläutert.

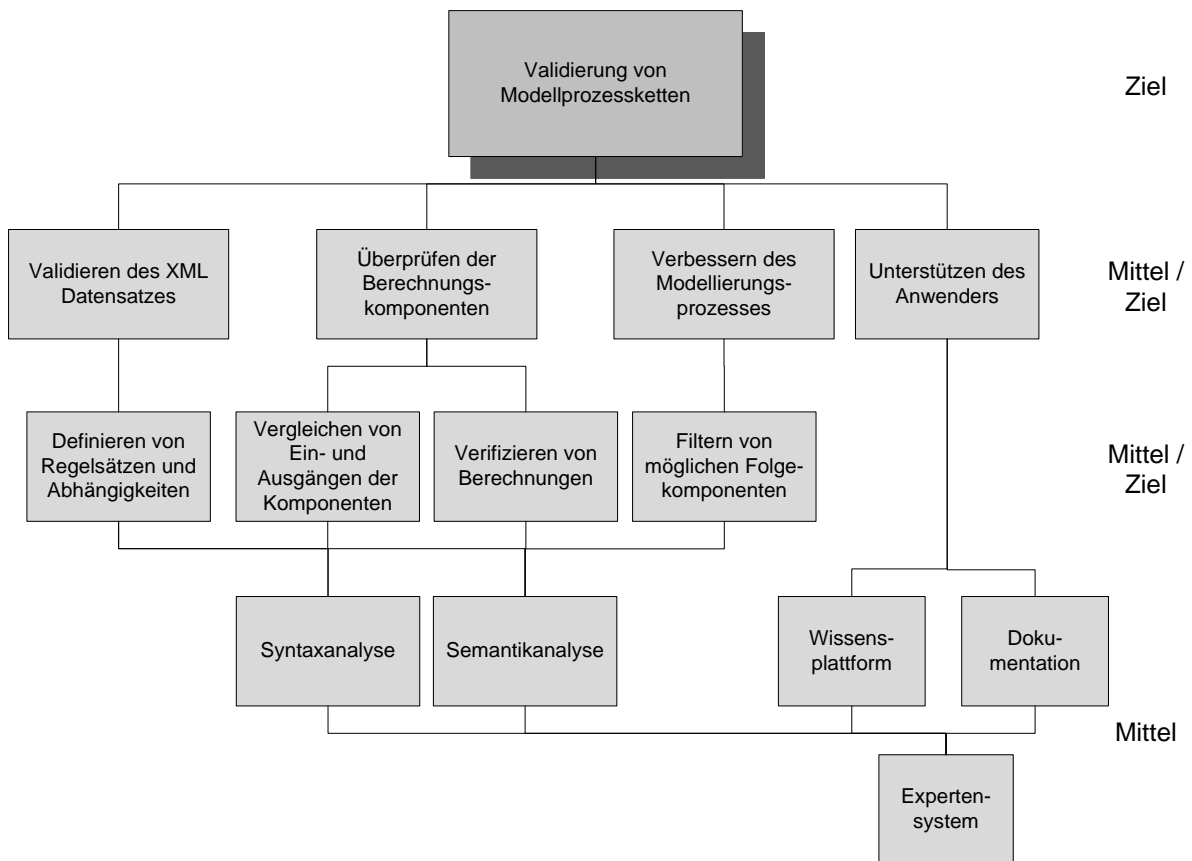


ABBILDUNG 14: ZIEL-MITTEL-HIERARCHIE

Das Hauptziel der Validierung von Modellierungsketten kann – wie bereits in Kapitel 4.1 ANALYSE DES LÖSUNGSRRAUMES geschildert – in die lokalen Aktionsbereiche (Benutzer, Modellierungsplattform, Berechnungsplattformen) und ihren Schnittstellenobjekten (XML Datei und XML Datenstrom) aufgeteilt werden. Diese Zwischenebene wird wiederum durch allgemeine Forderungen soweit konkretisiert, bis sich Endlösungen finden. In vorliegendem Projekt sind den vier Teilbereichen drei Lösungsansätze zugeordnet. Die Ergebnisse der Untersuchung zielen auf drei verschiedene Methoden zur Lösung der autarken Anwendungsbereiche ab. Sie unterscheiden sich in den Techniken sowie im Einsatzzweck.

- Syntaxanalyse
- Semantikanalyse
- Dokumentation
- Wissensplattform
- Expertensystem

Die Syntaxanalyse dient der Überprüfung des XML Datensatzes und der Komponenten der Modellierungsplattform. Ihre Einsatzmöglichkeit ist eingeschränkt in der Validierung von elementaren Attributen und der Auswertung der Informationsrepräsentation. Sie bestätigt die Richtigkeit der Eingabe hinsichtlich des Typs und Wertes.

Eine Abstraktionsebene höher stellt sich die Semantikanalyse auf. Sie betrachtet nicht nur Werte, sondern auch ihre Abhängigkeiten zueinander und damit verbunden die Bedeutung von Daten über vordefinierte Regelsätze und eine beschreibende Logik. Die Semantik überprüft Aussagen auf ihren Wahrheitsgehalt und interveniert im Ausnahmefall.

Die Dokumentation kann als weiteres Hilfsmittel zur Validierung von berechneten Daten dem Anwender zur Verfügung gestellt werden. Mittels Aufzeichnung und Vergleich vorangegangener Simulationen und Daten dient sie beispielsweise der Unterstützung bei Eingaben und Konfiguration von Berechnungspfaden im Vorfeld der Simulation und der Auswertung von Ergebnissen sowie der nachhaltigen Verfestigung von Erkenntnissen im Umgang mit dem virtuellen Flugzeugvorentwurf.

Eine weitere Lösungsalternative ist der Einsatz einer Wissensplattform. Ähnlich der Dokumentation steht die Wissensplattform jedoch allen zu jeder Zeit zur Verfügung und sollte gemeinsam gepflegt werden. Zu beachten ist die Einhaltung von strukturellen und inhaltlichen Regeln, um Übersichtlichkeit und Transparenz beizubehalten.

Das Expertensystem bietet als ganzheitlicher Ansatz die größtmögliche Unterstützung in der computergestützten Vorentwurfsplanung. Ein Expertensystem enthält eine Wissensbasis, die Fachwissen repräsentiert, und eine Beschreibungslogik, die Zusammenhänge zwischen Informationen darstellt. Es wird bevorzugt dort eingesetzt, wo eine hohe Komplexität der Problemstellung herrscht und empirisches Wissen fehlt.

Bei der Suche von Lösungskonzepten ergab sich die Unterscheidung von systemtranszendenten und systemimmanenten Konzepten. Systemtranszendente Konzepte beziehen sich auf Problemstellungen außerhalb von computergestützten oder -gesteuerten Anwendungsfällen. Sie interagieren entweder mit Anwendern oder mit Schnittstellen von Computersystemen und beauftragen diese mit der Beachtung der Validierung. Dagegen greifen systemimmanente Konzepte innerhalb von Programmen und Plattformen, die dem Vorentwurf zur Verfügung stehen. Sie optimieren oder erweitern Systemkomponenten hinsichtlich der Validierungsfähigkeit.

5.2.1 SYSTEMTRANSZENDENTE KONZEPTE

Nachfolgend beschriebene Konzepte betreffen Anwendungsfälle, die nicht Teil eines computerisierten Systems sind. Sie beziehen sich auf Akteure und Schnittstellenobjekte, die von außen auf die Modellierungs- und Berechnungsplattformen zugreifen und gehören damit hauptsächlich zu den präventiven Werkzeugen, die primär die Fehlererkennung im Vorfeld der Simulation fördern. Hierunter fallen die Erstellung und Anwendung einer Dokumentation, die Installation und Integration einer Wissensplattform sowie der Einsatz der XML-Validierung zur Überprüfung der Struktur und des Inhaltes.

5.2.1.1 DOKUMENTATION

Eines der Hauptkriterien für die Validierung ist die Sicherung des reibungslosen Simulationsablaufs und die Vermeidung von nicht sinnvollen Berechnungen. Die Dokumentation soll den Anwendern die Möglichkeit geben, vorzeitig und mit Hilfe von beispielhaften Konfigurationen Fehler zu vermeiden. Um dies gewährleisten zu können, müssen formal folgende Bereiche abgedeckt sein:

Allgemeine Informationen	<ul style="list-style-type: none"> • Herausgeber • Erstellungsdatum • Ansprechpartner • Kontaktdaten • Revisionen, Änderungshistorie
Inhaltsverzeichnis	<ul style="list-style-type: none"> • Themengebiete • Kapitel • Abbildungsverzeichnis • Tabellenverzeichnis
Einführung	<ul style="list-style-type: none"> • Initiatoren des Projekts • Beschreibung der Thematik • Darstellung des Beispieldatensatzes der Flugzeugklasse D150
Modellierungshinweise	<ul style="list-style-type: none"> • Sensibilisierung auf Problemsituationen während der Modellierung <ul style="list-style-type: none"> ○ Wiederkehrende Berechnungsvorgänge ○ Überwachung der wiederholt berechneten Parameter, Caching
Berechnungswerkzeuge	<ul style="list-style-type: none"> • Benennung der Werkzeuge • Detailinformationen über Werkzeuge <ul style="list-style-type: none"> ○ Funktionsbeschreibung ○ Ladekonfiguration ○ Eingangsparameter ○ Ausgangsparameter ○ Kontrollparameter • Kontaktinformationen der Entwickler <ul style="list-style-type: none"> ○ Email ○ Telefonnummer
Schnittstellen	<ul style="list-style-type: none"> • Einzuhaltende Wertebereiche von Eingangs- und Ausgangsparametern verschiedener Funktionen
Gesetzliche Vorschriften	<ul style="list-style-type: none"> • Knappe Einleitung in die gesetzlichen Standards für den Vorentwurf, zum Beispiel durch die Deutsche Flugsicherung, DIN Normen etc.
Problembehandlung	<ul style="list-style-type: none"> • Sammlung von Fragen und Antworten • Katalog von häufig auftretenden Problemstellungen und deren Lösung • Entscheidungsbäume zur Fehlererkennung • Best Practices
Indizes	<ul style="list-style-type: none"> • Abkürzungsverzeichnis • Glossar • Referenz

TABELLE 3: AUFBAU DER DOKUMENTATION

Die Dokumentation ist konzipiert für Experten. Der Fokus richtet sich auf den Modellierungsprozess der Entwurfsplattform ModelCenter[®]. Inhaltlich sind neben Detailinformationen zu eingesetzten Softwaresystemen und -werkzeugen auch Verfahrensanweisungen, Best Practices, gesetzliche Bedingungen und bereits behandelte Problemstellungen festgehalten. Das Dokument muss deswegen regelmäßig aktualisiert (Änderungshistorie, Revisionsnummer) und in Umlauf gebracht werden. Aufgrund der Kompatibilität ist es wünschenswert, dass das Dokument allen Anwendern online in den Formaten PDF, HTML oder Word zur Verfügung steht. Die Aktualisierung des Dokuments erfolgt nach Informationseingang bei:

- Einführung oder Wegfall einer zusätzlichen Software- oder Systemkomponente
- Erweiterung oder Minimierung des Funktionsumfangs
- neuen Best Practice-Erkenntnissen
- Änderungen der Umwelt
 - Gesetze
 - Nationale Normen (DIN)
 - Technologien & Infrastruktur
 - Industriestandards (ISO, IEC, IEEE)
- Erschließung von neuen Fehlerquellen
- Änderung von Kontaktinformationen der Entwickler

Dieses Konzept beinhaltet Vor- und Nachteile:

Vorteile	Nachteile
<ul style="list-style-type: none"> • Übersichtliches und einheitliches Layout • Schneller Zugriff auf spezielle Themengebiete (Wortsuche, Indizes, Referenzen) • Zentrale Verwaltung • Gute Komprimierbarkeit (PDF-Format) • Fehlerprävention 	<ul style="list-style-type: none"> • Hoher Verwaltungsaufwand bei Änderung der Dokumentation • Keine Fehleranalyse • Abschreckung des Anwenders durch Seitenumfang • Hohe Anforderung an Verfasser

TABELLE 4: VOR- UND NACHTEILE DER DOKUMENTATION

5.2.1.2 WISSENSPLATTFORM

Der interdisziplinäre Ansatz des zugrundeliegenden Projektes kann als unterstützendes Element eines Konzeptes herangezogen werden. Die Zusammenarbeit von verschiedenen Abteilungen konzentriert das spezifische Wissen der beteiligten Disziplinen auf einen Zielpunkt.

Die Akquisition dieses Wissens benötigt eine Plattform, die den jeweiligen Kommunikationsstilen gerecht wird. Das Konzept der Dokumentation setzt voraus, dass eine mittelnde Person, der Dokumentationsverfasser, jede Abteilungssprache erfassen und so übertragen kann, dass sie von anderen verstanden wird.

Das zweite Konzept, dass sich ebenso als systemtranszendentes Konzept versteht, ersetzt die zentrale Position des Verfassers und verlagert die Kompetenz des Vermittelns auf die Abteilungen. An Stelle des schriftlich verfassten Dokumentes tritt eine digitale Wissensplattform. Wissensplattformen speichern Expertenwissen, das frei in Struktur und Inhalt jedem Anwender zur Verfügung steht. Voraussetzende Faktoren zur Einführung eines Wissenssystems sind die Installation eines zentralen Zugriffsystems mit integrierter Rechteverwaltung und aktuellen Sicherheitsmechanismen, die Einrichtung von clientseitiger Zugriffsoftware und die Errichtung einer zeitgemäßen Infrastruktur, die eine möglichst ausfallsichere und schnelle Verbindung garantiert.

Die Sammlung von Wissen innerhalb einer Plattform kann wesentlich umfangreicher gestaltet werden, da sie in der Darstellungsebene selektiv vorgeht. Inhalte können strukturiert und über Sucheingaben hervorgehoben werden, ohne dass die Übersichtlichkeit leidet. Die Vorgehensweise bei der Erstellung einer Wissensplattform unterscheidet zwischen drei Faktoren:

- **Wissenserwerb:** Sammlung von Wissen unter medialer Zuhilfenahme von Büchern, Experten, Dokumenten und Dateien. Das Wissen kann hierbei spezifisch über einen bestimmten Sachverhalt sein. Ebenso bedarf es der Integration von „Metawissen“, das heißt Wissen über Wissen, das sich in der Niederschrift von Lösungsmechanismen zu Problemstellungen manifestiert: „Wie nutzen Experten ihr Wissen, um Konflikte zu lösen?“ Aufgrund des hohen Aufwands stellt sich der Wissenserwerb meist als Flaschenhals in der Entwicklung von Wissensplattformen dar.
- **Wissensrepräsentation:** Organisation von Wissen. Diese Disziplin erfordert den Sachverstand des Verfassers, indem sie Anforderungen an die Verständlichkeit, Übersichtlichkeit und Struktur des Wissens stellt. Die Informationen müssen so geordnet sein, dass sie in leichter Weise in eine Wissensplattform technisch abgebildet sowie von Anwendern verständlich dekodiert werden können.
- **Wissensvalidierung:** Die Informationen, die in einer Wissensplattform abgebildet sind, müssen qualitativ hochwertig sein, das heißt sie wurden durch Testfälle validiert und verifiziert. Die Testfälle und Testergebnisse müssen von Experten auf Richtigkeit hin geprüft werden.

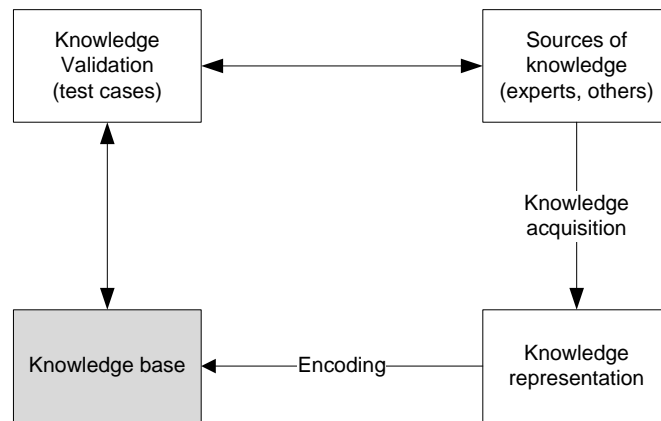


ABBILDUNG 15: PROZESS DES KNOWLEDGE ENGINEERING

nach [Tur05 S. 578]

Aufgrund des langwierigen Erwerbs von Wissen müssen Wissensplattformen für die Dauer der Benutzung zeitgemäß aufgebaut sein. Die zugrundeliegende Infrastruktur ist wie folgt dargestellt:

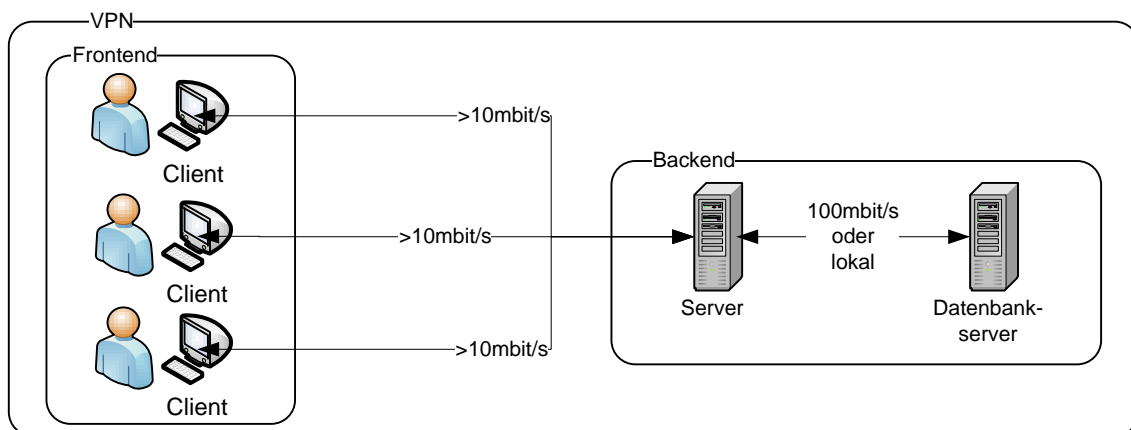


ABBILDUNG 16: INFRASTRUKTUR DER WISSENSPLATTFORM

Die Auswahl der Software hat Einfluss auf die Gestaltung der Client-Server-Beziehung, indem sie die Anforderungen an die Systemspezifikationen des Servers und die dahinterliegende Datenbank definiert. Der Client benötigt in diesem Konzept einen browserfähigen Rechner mit zeitgemäßer Bandbreite im lokalen (LAN, local area network) oder weltweiten (WAN, wide area network) Netzwerk. Empfohlen wird eine gesicherte, das heißt verschlüsselte, Verbindung (VPN), wenn sich Clients außerhalb von lokalen Netzwerken verbinden wollen.

Für den Informationsaustausch prädestiniert sind Websysteme zur gemeinschaftlichen Erstellung und Bearbeitung von Texten und Dokumenten, Content-Management-Systeme (CMS). Das Wiki („wiki“ ist hawaiianisch und bedeutet: „schnell“³) ist eine Unterklasse der CMS. Ein Wiki ist ein serverseitiges Content-Management-System, das angebunden ist an einer lokalen oder im selben Netzwerk befindlichen Datenbank. Wikis beinhalten Rechte-

³ <http://wehewehe.org/gsd/2.5/cgi-bin/hdict?e=q-0hdict--00-0-0--010---4----den--0-000lp0--1en-Zz-1---Zz-1-home-wiki--00031-0010utfZz-8-00&a=q&d=D21021> [Abrufdatum: 22.06.2010]

verwaltungssysteme, die Schreib-, Lese- und Zugriffsrechte auf verschiedenen Ebenen abbilden.

Technische Anforderungen eines Wikis:

- Backendverbund mit Server und Datenbank, die auf dem lokalen Server oder auf einem weiteren Server läuft
 - Betriebssystem
 - Installation und Konfiguration
 - Ausfallsichere Verbindung zwischen Webserver und Datenbankserver
- Clientsystem mit einem integrierten und zeitgemäßen Webbrowser zum Aufruf und zur Bearbeitung von Wiki-Seiten
- Authentifizierung: Benutzer müssen sich vor Erstellung oder Bearbeitung einer Seite am System anmelden. Die Einbindung in ein Single-Sign-On Konzept befreit den Benutzer von der mehrmaligen Identifizierung.
- Versionskontrolle: Bearbeitungen von Entwürfen werden identifizierbar. Die Versionskontrolle speichert den Zeitpunkt sowie den Urheber der Änderung. Versehentliche Änderungen können zurückgenommen werden. Diese Funktion schafft eine hohe Transparenz in der Modellierung von Wissen.
- Gruppenrechte: bestimmte Seiten sollen nur für ausgewählte Benutzergruppen zugreifbar gemacht werden können.
- Tagging: Indizierung von Inhalten
- Unicode-Unterstützung (beispielsweise utf-8) zur einheitlichen Kodierung von Informationen
- Vorlagen: Erleichtern die Erstellung von neuen Wiki-Seiten
- Bearbeitungssperren: Sperren von Seiten zur Vermeidung von Konflikten
 - Seitenspernung oder Konfliktbehandlung
- Backup: Sicherung des gesamten Wiki-Systems, um das Risiko von Ausfällen zu minimieren.
 - DBMS (Database Management System)
 - Dateisicherung, -archivierung
- Dateiupload: Upload-Funktionen sind notwendig, um Dokumente und Multimediadateien im Wiki zur Ansicht oder zum Download bereitzustellen.
 - Video / Audio
 - Grafik
- Plugin / Update System: Updates und Erweiterungen sollen mit möglichst geringem Aufwand integrierbar sein.

Navigation:

- Kategorien: Vereinheitlichung von Struktur und Ordnung des Inhaltes.
- Indizierungsschemata: für Titel, Worte, Seiten
- Back links: Funktion zur Navigation auf vorherig besichtigte Seiten. Eine übersichtliche Kategorienleiste sorgt für eine schnelle Identifizierung der Informationstiefe.
- Suchfunktion: Suche nach Überschriften, Anhängen, Revisionen und Volltext mittels Suchbegriffen und booleschen Ausdrücken.

Benutzerfreundliche Eigenschaften:

- WYSIWYG („What you see is what you get“) Benutzeroberfläche

- Voransicht (Preview) von Artikeln vor dem Absenden und Kommentarfunktion
- Änderungshistorie: Zusammenfassung von letzten Artikeländerungen
- Email-Benachrichtigung
- Highlighting: Formatierung von verschiedenen Abschnitten:
 - Quelltexte (nicht-proportionale Texte)
 - XML
 - HTML
 - Tabellen
 - Zitate
 - Fußnoten
- Integration von mathematischen Formeln (z.B.: LaTeX in MediaWiki)
- Hilfefunktion
- Evtl. Analysefunktionen zur statistischen Auswertung der Nutzung und des Benutzer-
verhaltens in der Bedienung des Wikis
- Exportfunktionen: PDF, HTML, XML, DocBook
- RSS-Feed abonnieren, Email

Die im DLR gegebenen, technischen Infrastrukturen sind prädestiniert für den Einsatz eines abteilungsübergreifenden Wikis. Das DLR setzt in dem Bereich Authentifizierung auf das Single-Sign-On Konzept: Der Verzeichnisdienst Active Directory[®]. Hierbei meldet sich der Benutzer nur einmal am DLR-System an. Weitere Anmeldungen zu internen Systemen, wie zum Beispiel dem Intranet oder dem Emaildienst, entfallen. Die Nutzer haben durch ein Authentifizierungsticket automatisch Zugang zu den gebotenen, internen Diensten.

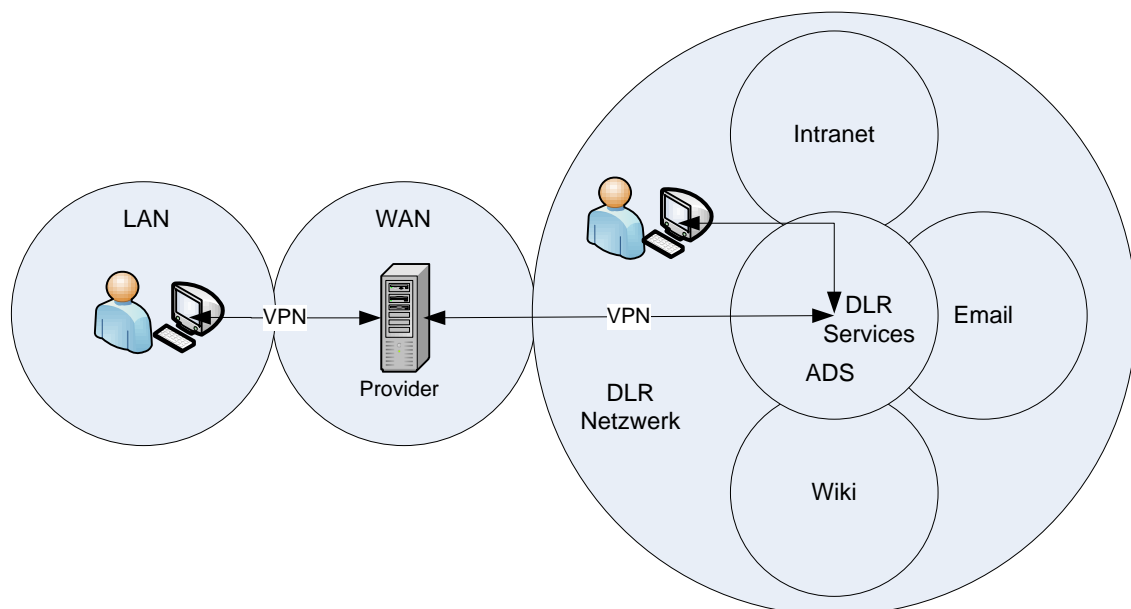


ABBILDUNG 17: DAS SINGLE-SIGN-ON KONZEPT IM DLR

Um die Konstruktion des Wissens zu fördern, müssen enge Rahmenbedingungen geschaffen werden, die den Anwendern als Schnittstellen für die Weitergabe ihres Wissens dienen. Die Struktur eines Wikis ist ausschlaggebend für den schnellen Zugriff und das Einbringen von neuem Wissen. Bei der Einführung muss die Übersichtlichkeit gewahrt werden. Eine grobe Vorstrukturierung der Seiten hilft gegen die auf dem Wiki-Prinzip – der gemeinschaftlichen Zusammenführung von Wissen – gründende nicht-hierarchische Unordnung. Die Abbildung

von bereits bekannten Formen und Strukturen unterstützt den Prozess des Wissenserwerbes, indem sie die Suche und die Erstellung von Inhalten beschleunigt. Andererseits ist darauf zu achten, dass eine zu starre Struktur gegen Innovation und Motivation der Nutzer spricht. Ein gewisser Grad an Flexibilität muss erhalten bleiben, um Änderungen durchsetzen zu können.

Zur Anregung der Partizipation gehören nicht nur richtige Rahmenbedingungen, sondern auch positive Eigenschaften eines Wikis. Im Folgenden wird eine Liste von Vorschlägen unterbreitet, die bei der Einrichtung eines Wikis beachtet werden sollten:

- Leere Wiki-Seiten können die Nutzer in der Einführungsphase als Leitfaden zur Wissensgenerierung dienen.
- Eine Liste von noch nicht ausgefüllten Wiki-Seiten erleichtert die Ideenfindung und regt zu neuen Wiki-Seiten an.
- Ebenso deuten farblich markierte Stichworte, für die noch keine Wiki-Seite existiert, das Fehlen von Wiki-Seiten an.
- Eine Wunschliste kann als Ideenaustausch zur Generierung oder Änderung von Themen und Inhalten erstellt werden.
- Aufgrund der antihierarchischen Struktur eines Wikis ist eine integrierte Suchfunktion notwendig, um eine effektive Nutzung des Wikis gewährleisten zu können. Neben einer Volltextsuche ist bei größeren Wikis der Einsatz von ergebniseinschränkenden, booleschen Operatoren, wie „UND“ und „NICHT“ zu empfehlen. Auch empfiehlt sich der Einsatz von Suchkriterien, die beispielsweise das Filtern nach Autor, Kategorie, Datum oder Namensräumen zulassen.
- Großzügiger Einsatz von Benutzerrechten ermöglicht das Entstehen von neuem Wissen. Die Anwender sollten weitestgehend die Möglichkeit haben, jeden Wiki-Artikel zu ändern.

nach [Mos08 S. 83-112]

Die Struktur und der thematische Aufbau haben maßgebende Bedeutung für die Akzeptanz der Nutzer und die zukünftige Entwicklung des Wikis. Deshalb ist die zu erstellende Hierarchie wichtig. Im Folgenden wird eine beispielhafte Verzeichnisstruktur eines möglichen Wikis dargestellt. Sie gründet auf der Informationsbasis des VAMP-Projektplanes und enthält die darin aufgezählten Fachbereiche und deren Werkzeuge für den Flugzeugvorentwurf:

Hauptseite

- Allgemeine Informationen
 - Projektkurzinformation, -status
 - Projektleitung
 - Beteiligte Institutionen und Einrichtungen
 - Kontaktinformationen
- Werkzeuge
 - Aerodynamik
 - Struktur
 - Aerolastik
 - Triebwerk

- Systeme
- Fahrwerk
- Missionssimulation
- Start/Landung
- Flugdynamik
- Handlingqualities
- Lasten
- Lastminderungssystem
- Ditching
- Kabinengeometrie
- Boarding
- Kosten
- Außenlärm
- Klimawirkung
- Höhenstrahlung
- IR Signatur
- Radar Signatur
- Flugverkehrssimulation
- Schnittstellen
 - ModelCenter[®]
 - CPACS
 - Analysis Server
- Sonstige Dokumentation
 - Modellierungstipps
 - ACARE-Ziele
 - FAQ
- Ideenliste
 - Verbesserungsvorschläge
 - Anmerkungen

Eine Erweiterung zu der konventionellen Wissensplattform stellen semantische Wikis dar. Durch das Hinzufügen von weiteren Informationen (Attributen) können Beziehungen zwischen Subjekten modelliert werden. Die Beziehungen sind charakterisiert durch „hat ein“- oder „ist ein“-Verknüpfungen. Das damit gesponnene Netz stärkt die Suchfunktion mit deutlich verbesserten Suchkriterien und Filtern. Beschreibende Suchbegriffe können verwendet werden, um flexibler auf explizites Wissen zuzugreifen.

Das Semantic MediaWiki+[®] der ontoprise GmbH⁴ baut auf diesem Prinzip auf. Es bietet Annotationen (Eigenschaften), um dem Inhalt beschreibende Fakten zuzuordnen und eine Ontologie aufzubauen. Die Abfragesprache erlaubt Auswertungen über Fakten, Kategorien und ihre Eigenschaften. Die Ontologie kann im Standardformat für SemanticWeb-Anwendungen, OWL, exportiert und in anderen Wissensmanagementsystemen wiederverwendet werden [Han09 S. 2]. Der Nutzen des Mehraufwandes in der Erstellung der Wiki-Seiten kann sich durch die Integration der Wissensplattform in die Modellierungsplattform in kürzester Zeit amortisieren. Mit Unterstützung der Semantik ist es möglich, eine Online-Hilfe bei Probleme-

⁴ http://wiki.ontoprise.de/smwforum/index.php/Main_Page [Abrufdatum: 07.06.2010]

men in der Modellierungsphase einzurichten und Fehlern entgegenzuwirken. Die Vorteile der größeren Flexibilität, das Explizieren von Wissen, der offenen Architektur und der verbesserten Suchfunktion sind gegen den Nachteil des deutlich gestiegenen Zeitaufwandes in der Erstellung und eventuellen Nachbearbeitung von Artikeln aufzuwiegen. Das Konzept der Semantik wird in späteren Lösungsvorschlägen nochmalig aufgegriffen und, in eine neue Umgebung integriert, betrachtet werden.

Zunächst sind die Vor- und Nachteile einer konventionellen Wissensplattform im Folgenden aufgezeigt:

Vorteile	Nachteile
<ul style="list-style-type: none"> • Übersichtliches und einheitliches Layout • Schneller Zugriff durch Suchfunktion • Dezentrale Verwaltung • Schneller Aufbau des Wissensnetzwerkes • Hohe Erweiterbarkeit • Mögliche Anbindung in das Modellierungssystem (Hyperlinks) • Exportierbarkeit in andere Datenformate • Fehlerprävention • Hohe Verfügbarkeit • Schneller Wissenserwerb • Geringer Verwaltungsaufwand nach Integrationsphase 	<ul style="list-style-type: none"> • Keine Fehlerauswertung • Hoher Installationsaufwand • Mehraufwand für Anwender • Keine Fehleranalyse

TABELLE 5: VOR- UND NACHTEILE DER WISSENSPLATTFORM

5.2.1.3 XML VALIDIERUNG

Ein weiteres Konzept der systemtranszendenten Validierung stellt die Überprüfung der CPACS-Datei dar. Die CPACS-Datei beinhaltet Informationen über den CPACS Datensatz selbst, Beschreibungen der Flugzeuge und deren Komponenten sowie diverse Informationen über Missionen, Flughäfen, Flotten, und spezifische Daten für die jeweiligen Analysewerkzeuge. Diese sind in XML Form hierarchisch gespeichert.

Im Sinne der Wohlgeformtheit, die per Definition eine Schemadatei zusätzlich zur XML Datei fordert, wird der CPACS Datensatz mittels der Modellierungskomponente „CPACSSource“ im Ladevorgang optional auf Validität hin analysiert [Bac10 S. 16]. Diese Vorgehensweise wird im späteren Diskurs differenziert betrachtet und die Notwendigkeit einer weiteren Überprüfungsinstantz erörtert.

Nachfolgend ist ein kurzer Ausschnitt der XML Datei aufgelistet. Der Datensatz unterteilt sich in die Hauptbereiche: Dateiinformation (header), Fahrzeuge (vehicles) und toolspezifische Informationen (toolspecific) eingeteilt.

```

- <cpacs xsi:noNamespaceSchemaLocation="cpacs_schema_v1.0.xsd">
  - <header>
    <name>Dataset with VFW-614 - ATTAS data</name>
    + <description></description>
    <creator>Martin Hepperle, DLR-AS</creator>
    <timestamp>2005-10-01T09:30:47</timestamp>
    <version>1</version>
    + <updates></updates>
  </header>
  - <vehicles>
    + <aircraft></aircraft>
    + <engines></engines>
    + <profiles></profiles>
  </vehicles>
  - <toolspecific>
    + <airportFees></airportFees>
    + <panam></panam>
    + <liftingLine></liftingLine>
    + <tWDat></tWDat>
    + <cabin></cabin>
  </toolspecific>
</cpacs>

```

ABBILDUNG 18: HAUPTSTRUKTUR DES CPACS-DATENSATZES

Die Validierung des Datensatzes erfordert die Kenntnis über die Abhängigkeiten der Flugzeugkomponenten, um einen Katalog an Regelsätzen definieren zu können. Nach diesen Regelsätzen sollen hinzukommende, berechnete Teildatensätze auf Validität hin überprüft werden, um die Gesamtheit des Datensatzes konsistent halten zu können. Aufgrund der Definition des XML-Schemas, das zwischen Attributen und Elementen unterscheidet, müssen zwei Analyseverfahren eingesetzt werden:

Die Syntaxanalyse überprüft in diesem Kontext die Richtigkeit der Werte. Sie ist nicht im Stande die Verifikation der Berechnung durchzuführen, kann aber zur Typisierung und Verifizierung des Wertebereichs beitragen. Ebenso dient sie der Prüfung der Existenz von Werten, die für nachfolgende Berechnungen notwendig sind. Beispielweise ist eine Berechnung des Fluglärms ohne Leistungsangabe des Triebwerkes nicht durchführbar.

Die Semantikanalyse beschreibt die Zusammenhänge der Elemente und Attribute. In Kombination mit der Syntaxanalyse vervollständigt sie die Validierung des XML-Datensatzes, indem sie die Beschreibungslogik überprüft. Sie stellt die Beziehungen zwischen intraspezifischen und interspezifischen Attributen und Elementen in Frage und präsentiert dem Anwender das Resultat der Auswertung. Ein anschauliches Beispiel hierfür ist die Berechnung von Flugzeugvorentwürfen, deren Anzahl von Tragflächen kleiner als zwei ist. Die semantische Validierung kann auf verschiedene Arten durchgeführt werden:

- Option 1: Unterstützung des XML Schemas mit einer XML Restriktionssprache
- Option 2: Überprüfung der Restriktionen auf Programmiererebene
- Option 3: Entwicklung von Regelsätzen mit XSLT oder XPath stylesheet

Die Starrheit der Umsetzung von Option 2 und die Komplexität in der Erstellung eines XSLT/XPath-Stylesheet von Option 3 begünstigt den Gebrauch von Option 1. Die Schwierigkeit hierbei ist, regelbasierte sowie dynamische Einschränkungen in einer Sprache zu vereinen. Die Mächtigkeit der Restriktionssprachen ist ein entscheidender Faktor, der vor dem Einsatz untersucht werden muss. Zur Auswahl stehen die aktuellen Vertreter der Klasse der Schemasprachen:

- DTD (Document Type Definition)
- W3C XML Schema
- XDR (XML Data Reduced)
- SOX (Schema for object oriented XML)
- Schematron
- DSD (Document Structure Description)

Zu unterscheiden sind regelbasierte und aussagenbasierte Einschränkungen, die jeweils für die Beziehung zwischen Elementen und Attributen sowie für das Werteformat, den Wertebereich, das Muster eines Elementes oder Attributes oder die Präsenz eines Elementes oder Attributes stehen. Ein Versuch eine allgemeingültige Klassifizierung von XML-Restriktionen zu identifizieren, kann folgendermaßen dargestellt werden:

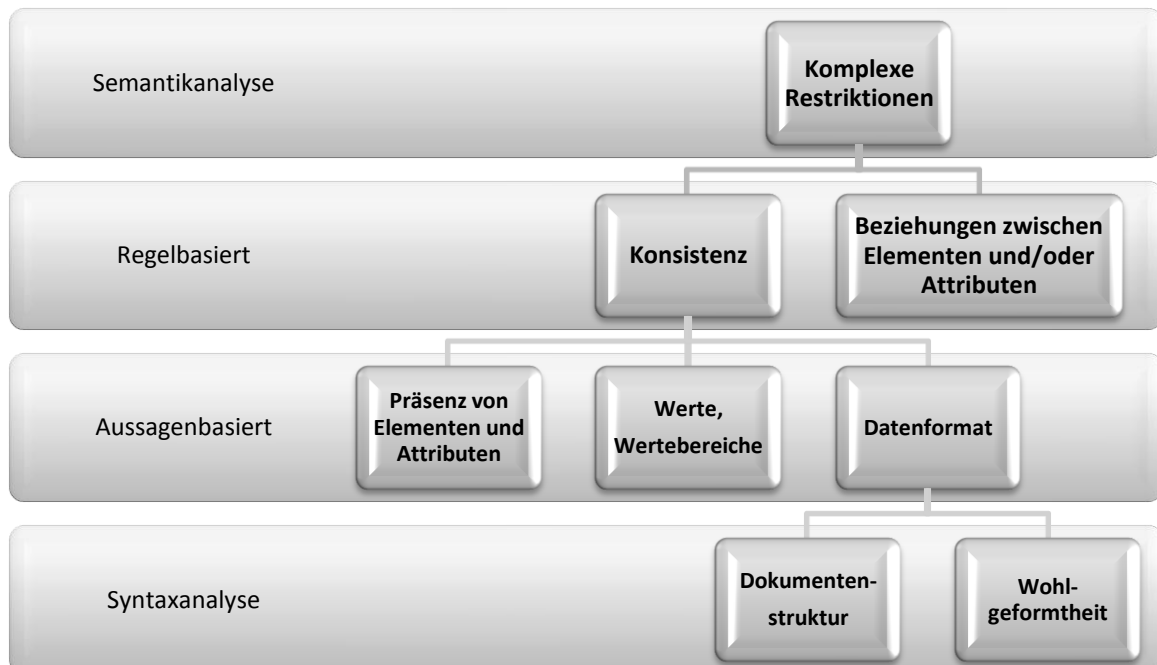


ABBILDUNG 19: KLASSIFIZIERUNG VON XML-RESTRIKTIONEN NACH [HUJ03 S. 10-13]

Wohlgeformtheit ist dann gegeben, wenn das Dokument den einfachen XML Syntaxregeln genügt. Gültig ist ein XML-Dokument dann, wenn zudem eine zugehörige Typdefinitionsdatei (DTD, Document Type Definition) existiert. Die Gültigkeit wird mit Hilfe eines XML-Prozessors

überprüft. Die DTD beschreibt schematisch die Struktur des XML-Dokuments und kann den zulässigen Wertebereich von Attributen einschränken. Die Bedeutung der Typdefinitionsdatei spiegelt sich in drei Aspekten wider:

- Für Anwendungen, die auf diese XML-Datei zugreifen, ist die Typdefinitionsdatei die Grundlage für die gegebene Struktur (Garantieaspekt).
- Mittels eingefügter Kommentare und anderen DTD Informationen über die Bedeutung der Struktur können Entwickler Rückschlüsse ziehen, die sie für die Programmierung ihrer Anwendung berücksichtigen (Informationsaspekt).
- Aufgrund der Unabhängigkeit zwischen DTD und XML-Dokument kann eine DTD sowohl innerhalb einer XML-Datei mehrfach zugewiesen, als auch von mehreren XML-Dateien gemeinsam genutzt werden (Modularisierungsaspekt).

nach [Sch03 S. 6]

Durch Einsatz einer DTD kann demnach die Syntaxanalyse vollzogen werden. Sie beschreibt Elementtypen, Attribute von Elementen, Entitäten und Notationen. Aufgrund der eingeschränkten Vorgabe von Attributlisten und -werten können nicht alle, im CPACS Datensatz enthaltenen Attributtypen abgedeckt werden. Beispielweise fehlen Unter- und Obergrenzen für Wertebereiche und Nachkommastellen für Dezimalzahlen. Dieses Defizit des DTD-Ansatzes stieß in der Vergangenheit auf Unzufriedenheit in der XML Gemeinschaft. Andere Lösungsmöglichkeiten reiften heran und lösten DTD als Mittel der Schemadefinition ab. Die Vielzahl der neu entstandenen Schemasprachen erforderte eine Standardisierung. Die Antwort darauf ist der W3C XML Schema Standard, kurz XML Schema, dessen Entwurfsziel es ist, auf den Fähigkeiten von DTD aufbauend, ein ausgefeiltes Typkonzept zu erstellen [Sch03 S. 32]. Unterschieden wird zwischen vordefinierten (einfachen) und neu definierten (komplexen) Typen, die durch Einschränkung oder Erweiterung von einfachen Typen erzeugt werden können.

Aufgrund des langwierigen Einführungsverfahrens der Standardisierung von W3C XML Schema haben sich weitere Schema- und Restriktionssprachen herauskristallisiert, die wegen ihrer geringeren Komplexität im Gegensatz zu W3C XML Schema große Beliebtheit erlangten. Analytische Vergleiche von diversen Schemasprachen wurden bereits auf informativen Webseiten behandelt (vgl. [Lee00] und [Vli01]). In diesen Studien ragen zwei Schemasprachen hervor, die das Lager der grammatikbasierten Sprachen verlassen und sich in das Gebiet der regel- oder musterbasierten Sprachen begeben. Vertreter dieser Sparte sind Schematron und DSD.

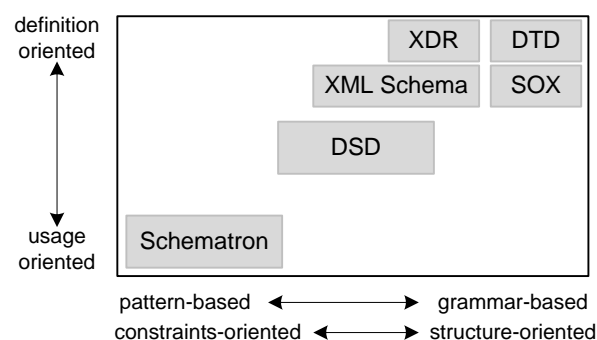


ABBILDUNG 20: KLASSIFIZIERUNG VON XML SCHEMASPRACHEN NACH LEE UND CHU (2000)

DSD schließt den Zwiespalt der rein grammatik- und strukturorientierten Sprachen und der musterbasierten und restriktionsorientierten Sprachen und integriert sie in einer programmiersprachenähnlichen Syntax. Während Schematron auf XSLT und XPath setzt, um das Regelwerk aufzubauen, bedient sich DSD einer eigenen Grammatik. Im Hinblick auf den Einsatz von benutzerdefinierten Regeln innerhalb einer XML-Datei sind DSD jedoch Grenzen gesetzt. DSD definiert die Struktur und die Beziehung von Elementen und Attributen top-down und bietet einschränkende Funktionen. Die Möglichkeit der Überprüfung von vorhandenen XML-Datensätzen a posteriori fällt demnach schwieriger.

Ein Anwendungsbeispiel der DSD zeigt folgende Tabelle:

XML	DSD
<pre> <collection xmlns=http://recipes.org xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xsi:schemaLocation=http://recipes.org/recipes.xsd> <description>Beispiel einer XML Datei</description> <recipe> <title>Gericht 1</title> <ingredient name="Steak" amount="1.5" unit="Pfund" /> <ingredient name="Zwiebel" amount="1" /> <ingredient name="Pfeffer" amount="1" unit="Tasse" /> ... <preparation> <step>Heize den Ofen auf 175°C vor.</step> ... </preparation> </recipe> </pre>	<pre> <dsd xmlns=http://www.brics.dk/DSD/2.0 xmlns:r=http://recipes.org root="r:collection"> <if><element name="r:collection" /> <declare><contents> <element name="r:description" /> <repeat><element name="r:recipe" /></repeat> </contents></declare> </if> <if><element name="r:description" /> <rule ref="r:ANYCONTENT" /> </if> <if><element name="r:recipe" /> <declare><contents> <sequence> <element name="r:title" /> <repeat><element name="r:ingredient" /></repeat> <element name="r:preparation" /> </sequence> <optional><element name="r:comment" /></optional> </contents></declare> </if> <if><element name="r:ingredient" /> <declare> <required><attribute name="name" /></required> <attribute name="amount"> <union> <string value="*" /> <stringtype ref="r:NUMBER" /> </union> </attribute> <attribute name="unit" /> </declare> <if><not><attribute name="amount" /></not> <require><not><attribute name="unit" /></not></require> <declare><contents> <repeat min="1"><element name="r:ingredient" /></repeat> <element name="r:preparation" /> </contents></declare> </if> </if> <if><element name="r:preparation" /> <declare><contents> <repeat><element name="r:step" /></repeat> </contents></declare> </if> <if> <or> <element name="r:step" /> </pre>

	<pre> <element name="r:comment" /> <element name="r:title" /> </or> <declare><contents> <string/> </contents></declare> </if> <stringtype id="r:DIGITS"> <repeat min="1"> <char min="0" max="9" /> </repeat> </stringtype> <stringtype id="r:NUMBER"> <sequence> <stringtype ref="r:DIGITS" /> <optional> <sequence> <string value="." /> <stringtype ref="r:DIGITS" /> </sequence> </optional> </sequence> </stringtype> <rule id="r:ANYCONTENT"> <declare><contents> <repeat><union><element><string/></union></repeat> </contents></repeat> </rule> </dsd> </pre>
--	---

TABELLE 6: BEISPIEL EINES DSD-SCHEMAS

Schematron stellt Musterprüfungen durch Zusicherungen bereit, die mittels XSLT im XML Dokument gesucht und durch „assertions“ überprüft werden. Weichen die geprüften Kontexte von den Vorgaben ab, werden eigens definierte Fehlermeldungen ausgegeben. Schematron erfordert eine tiefere Kenntnis der Programmiersprachen XSLT und XPath, die den Funktionsumfang zur Erstellung des Regelwerkes erheblich steigern. Die Schematron-Validierung erfolgt zum Beispiel mit Hilfe von Skeleton-Skripten in einer zweistufigen XSLT-Transformation.

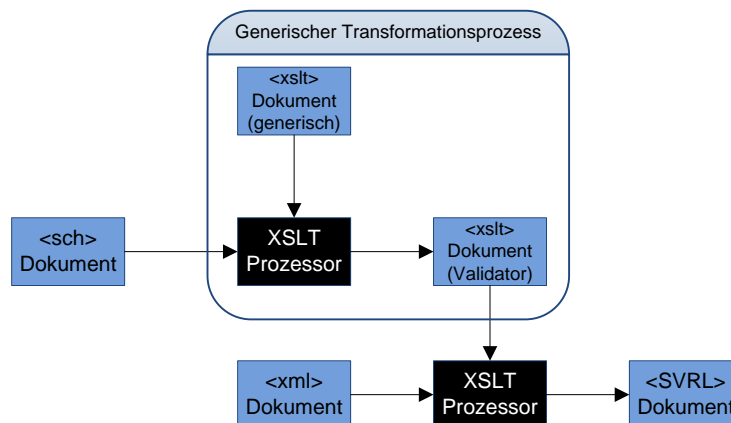


ABBILDUNG 21: SCHEMATRON-VALIDIERUNGSPROZESS NACH [DÜN09 S. 94]

Zuerst wird das Schematron-Schema durch ein generisches XSLT Stylesheet in ein XSLT-Dokument transformiert, das auf die zu prüfende XML-Instanz angewandt wird. Das Ergebnis ist ein Fehlerreport in der standardisierten Reporting-Sprache SVRL (Schema Validation Report Language). Die Anbindung eines Schematron-Schemas erfolgt durch Angabe entspre-

chender Processing Instructions. Ebenso kann in einer Applikation die Anbindung für einen Dokumenttyp konfiguriert werden, die neben der Validierung zusätzlich die Schematron-Prüfung initiiert.

XML	Schematron
<pre> <collection xmlns=http://recipes.org xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xsi:schemaLocation=http://recipes.org/recipes.xsd <description>Beispiel einer XML Datei</description> <recipe> <title>Gericht 1</title> <ingredient name="Steak" amount="1.5" unit="Pfund" /> <ingredient name="Zwiebel" amount="1" /> <ingredient name="Pfeffer" amount="1" unit="Tasse" /> ... <preparation> <step>Heize den Ofen auf 175°C vor.</step> ... </preparation> </recipe> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <sch:schema xmlns:sch=http://purl.oclc.org/dsdl/schematron> <sch:pattern name="Schematron Tests"> <sch:rule context="recipe"> <sch:assert test="ancestor::recipe"> Innerhalb von Rezepten sind keine weiteren Rezepte erlaubt! </sch:assert> </sch:rule> <sch:rule context="ingredient"> <sch:assert test="sum(amount) > 0"> Mengenangaben kleiner als 0 sind nicht erlaubt! </sch:assert> <sch:assert test="@name"> Element Ingredient muss ein Attribut Name haben. </sch:assert> </sch:rule> <sch:rule context="preparation"> <sch:report test="fett"> Zubereitungsschritte dürfen keine Auszeichnungen enthalten! </sch:report> </sch:rule> </sch:pattern> </sch:schema> </pre>

TABELLE 7: BEISPIEL EINES SCHEMATRON-SCHEMAS

Schematron bedient sich der XSLT-Sprache, um Abhängigkeiten zu modellieren und zu überprüfen. „Assertion Tests“ beurteilen Sachverhalte nach ihrem Wahrheitsgehalt und liefern eine Meldung zurück, wenn die Bedingung nicht zutrifft. Dagegen liefern „Report Tests“ Resultate zurück, wenn deren Bedingung zutreffend ist.

Die Kombination der zwei kontextorientierten Schemasprachen erfüllt die Anforderung der Syntax- und Semantikanalyse in vorliegendem Projekt (vgl. [Lee00]). Der Aufwand, der durch die Entwicklung einer XML Validierungslösung entsteht, darf jedoch nicht unterschätzt werden. Es entsteht eine Ontologie, die die Grenzen des Datensatzes skizzieren soll. Je nach Bedarf kann dies jedoch für wissenschaftliche Zwecke ungeeignet sein, da die Forschungseinrichtung des DLR nicht nach bereits vorhandenem Wissen sucht, sondern nach neuen Erkenntnissen in Luft- und Raumfahrt forscht. Der Einsatz dieses Instrumentariums muss daher sorgfältig abgewogen werden, um nicht zu restriktiv in den weiteren Fortschritt des Projektes einzugreifen.

Dem Konzept der XML Validierung liegen ebenso Vor- und Nachteile zugrunde:

Vorteile	Nachteile
<ul style="list-style-type: none"> • Entwicklung systemexterner Tools zur Validitätsprüfung außerhalb des Simulationsvorganges oder während des Einlesens des CPACS Datensatzes • Entstehung einer Beschreibungslogik 	<ul style="list-style-type: none"> • Keine Fehlerprävention: Fehler werden erst a posteriori erkannt. • Langwierige Reifephase • Verschiedene Schemasprachen haben ihr Vorzüge und Grenzen

<p>(Ontologie)</p> <ul style="list-style-type: none"> • Standardformat der Beschreibungslogik sichert die Wiederverwendung in anderen Systemen • Einfachere Schemasprachen aufgrund der Abkehr von XML Schemas Komplexität • Großflächige Unterstützung von XSLT/XPath 	<ul style="list-style-type: none"> • Fragwürdige Langzeitentwicklung von Schemasprachen durch Entwickler (meist nur ein/wenige Entwickler) • Zwei Dokumente vorhanden (XML Datei und Schemadefinition), die synchronisiert werden müssen
---	--

TABELLE 8: VOR- UND NACHTEILE DER XML VALIDIERUNG

5.2.2 SYSTEMIMMANENTE KONZEPTE

Der zweite Teil der Konzeptfindung beschäftigt sich mit Anwendungsfällen, die sich innerhalb eines Computersystems beziehungsweise während der Simulation abspielen. Es werden Lösungen vorgestellt, die einerseits Erweiterungen von vorhandenen Systemkomponenten, andererseits eigenständige Softwarekonzepte darstellen. Sie lassen sich als reaktive Analyserwerkzeuge einsetzen und dienen hierbei der Fehlerauswertung vor allem während der Simulation. Darunter befinden sich die toolbasierte Semantik in verschiedenen Ausprägungen sowie die Installation eines Expertensystems.

5.2.2.1 TOOLBASIERTE SEMANTIK

Das Konzept der toolbasierten Semantik agiert innerhalb der Modellierungsplattform und erweitert Komponenten, die essentiell für den Simulationsablauf sind. Wie in Kapitel 3.2 STRUKTURELLE MODELLIERUNG bereits veranschaulicht, unterteilt sich das Modellierungssystem in zwei Untersysteme: die Integrationsplattform und die Kalkulationsplattformen. Die Kalkulationsplattformen beinhalten wiederum zwei verschiedene Ausführungseinheiten: die Containereinheit (wrapper unit) sowie die Ausführungseinheit (execution unit).

Durch die Betrachtung des Simulationsablaufs werden verschiedene Integrationspunkte eines Validierungssystems aufgezeigt:

1. Visualisierung des Ergebnisses
2. Checkpointsystem in der Modellierungsplattform
3. Validierungskomponente in der Containereinheit
4. Validierungskomponente in der Ausführungseinheit
5. Validierungskomponente in der Bibliothek

Die aufgezählten Verfahren zur Validierung der Simulation werden im Folgenden der Reihenfolge entsprechend aufgezeigt sowie die Vor- und Nachteile der jeweiligen Integrationsmöglichkeit erörtert.

Zu 1: Visualisierung des Ergebnisses

Die Visualisierung des Ergebnisses basiert auf der interaktiven Kommunikation mit dem Anwender. Zwischen- sowie Endergebnisse sollen pro Zwischenschritt aktualisiert und grafisch in einer Betrachtungskomponente (viewer) dargestellt werden. Der Anwender soll die Möglichkeit haben, die dargestellte Grafik durch Maussteuerung in der dreidimensionalen Ebene bewegen zu können. Ebenso müssen Zoomfunktionen integriert werden, um die Geometrie in jedem Winkel begutachten zu können. Die Integration einer Betrachtungskomponente ist in folgender Abbildung beispielhaft in einer Optimierungsschleife dargestellt.

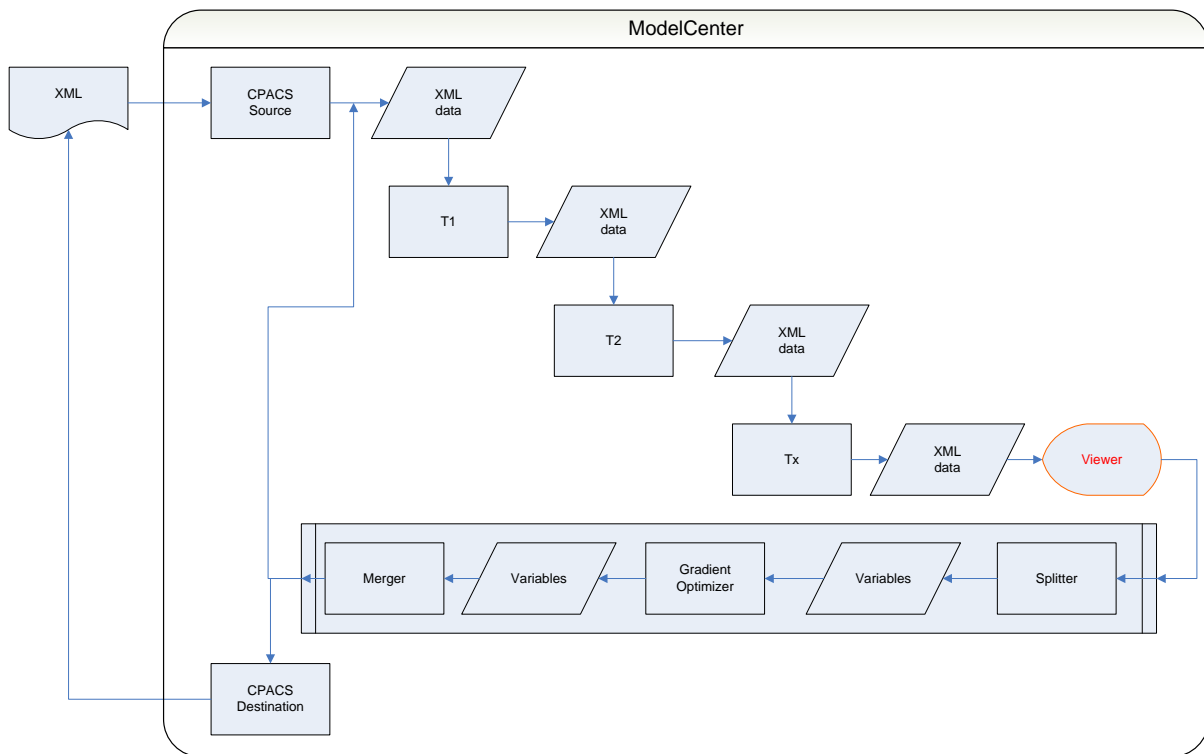


ABBILDUNG 22: INTEGRATION EINER BETRACHTUNGSKOMPONENTE

Der Anwender soll weiterhin die Möglichkeit haben, die Komponente an jede beliebige Stelle des Simulationsflusses setzen zu können. Der Einsatz eines Visualisierungswerkzeuges bringt verschiedene Vor- und Nachteile mit sich:

Vorteile	Nachteile
<ul style="list-style-type: none"> • Echtzeitdarstellung von Berechnungen ermöglicht den vorzeitigen Abbruch durch Benutzerinteraktion • Geringer Aufwand für Anwender und Entwickler • Kurzfristige Reaktion des Anwenders kann Kosten reduzieren und lässt Raum für Experimente 	<ul style="list-style-type: none"> • Fehlererkennung erst nach ganzem Berechnungszyklus • Keine Fehlerbeschreibung • Ursachen eines Fehler sind nicht ersichtlich

TABELLE 9: VOR- UND NACHTEILE DER VISUALISIERUNG DER SIMULATION

Zu 2: Checkpointsystem in der Modellierungsplattform

Das Checkpointsystem soll dem Anwender die Möglichkeit geben, Marken in dem Simulationsablauf zu setzen. Die Marken speichern eine Kopie des Datensatzes in eine temporäre Datei. Ähnlich einem Versionierungssystem soll der Anwender Zugriff auf die letzten Speicherzustände des CPACS Datensatzes haben und sie bei Bedarf zurücksetzen können. Dies erfordert das Pausieren des Berechnungsvorganges an Stellen, die Checkpointkomponenten darstellen.

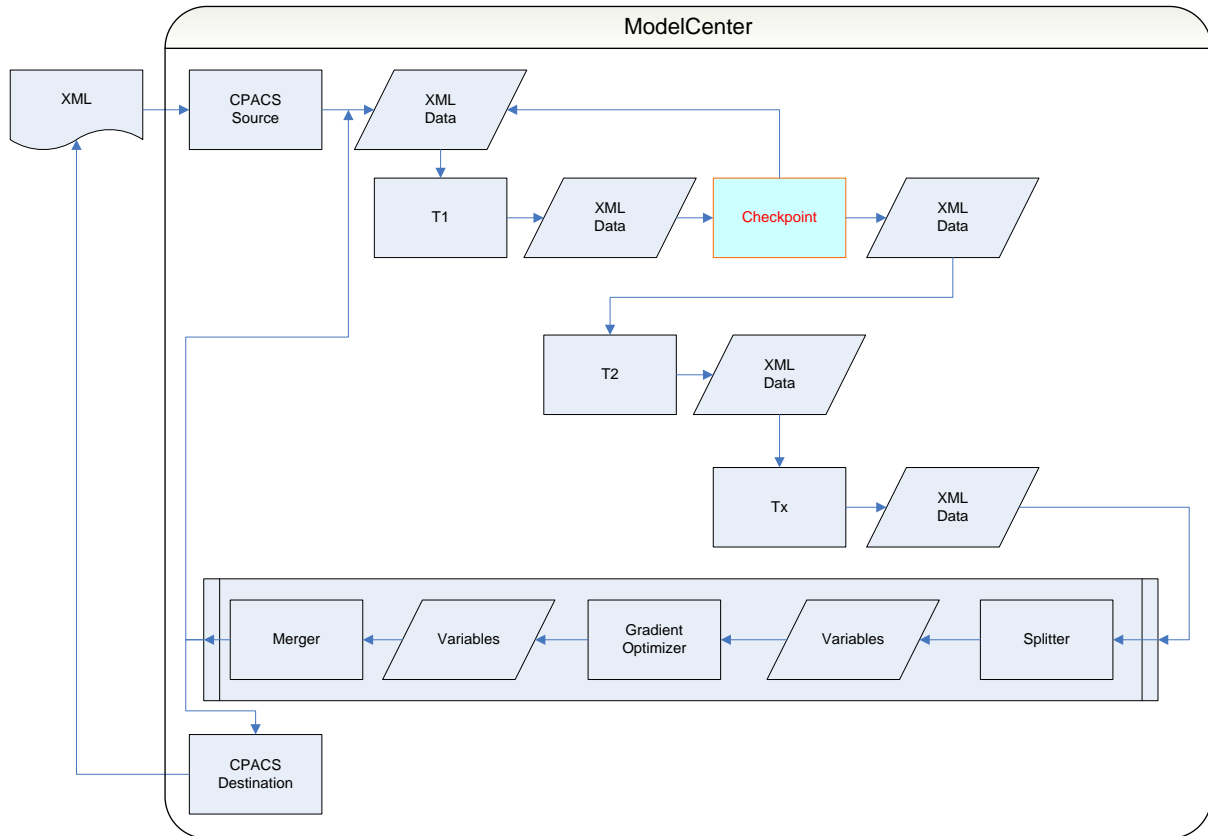


ABBILDUNG 23: INTEGRATION EINES CHECKPOINTSYSTEMS

Die gespeicherten Datensätze sollen in einem verständlichen Format gesichert werden. Beispielsweise müssen folgende Basisattribute festgehalten werden:

Datum & Uhrzeit	Datensatzstring	ModelCenter® Variablen
-----------------	-----------------	------------------------

Sollte der Wunsch bestehen, die Versionsdaten in einer Datenbank zu fixieren, ist darauf zu achten, dass der Datensatzstring gegebenenfalls ein sehr großes Datenfeld benötigt. Es ist zu untersuchen welches Datenbankmanagementsystem dies zulässt. Empfehlenswert ist daher die Sicherung der Daten in einer Datei innerhalb eines Betriebssystems mit entsprechender Dateigrößenunterstützung.

Da das vorgestellte Checkpointsystem keine Validierung enthält, ist es zudem ratsam eine zusätzliche Form der Datenvalidierung einzusetzen, die dem Anwender bei Datensicherung auf etwaige Abweichungen in der Berechnung hinweist und die Möglichkeit zur Rücksicherung vorheriger Berechnungsstände anbietet. Die Rücksicherung ersetzt den aktuellen CPACS-Datensatz mit dem gespeicherten Datensatz und führt die Berechnung fort. Ebenso

verfährt die Modellierungsplattform. ModelCenter[®] bietet einen Datensammeldienst an, der zur Laufzeit berechnete Werte speichern kann. Dieser Dienst ist in der Lage, Daten einzulesen (vgl. Data Explorer [Pho08 S. 549-550]), um eine komplette Rücksicherung zu gewährleisten.

Vorteile	Nachteile
<ul style="list-style-type: none"> • Früher Abbruch einer fehlerhaften Berechnung; kein kompletter Zyklus nötig • Geringer Aufwand für Anwender • Versionierung des Berechnungsverlaufes 	<ul style="list-style-type: none"> • evtl. Implementierungsaufwand bei Anpassung an besondere, gewünschte Funktionalitäten • Fehlererkennung erst nach zyklischer Berechnungsphase • Ungewissheit in welcher Berechnungsphase ein Fehler eingetreten ist

TABELLE 10: VOR- UND NACHTEILE DES EINSATZES EINES CHECKPOINTSYSTEMS

Zu 3,4,5: Validierungskomponente in der Containereinheit / Ausführungseinheit / Bibliothek

Aufgrund der äquivalenten Vorgehensweise in allen drei Bereichen werden diese Lösungsmöglichkeiten zusammengefasst. Der Einsatzort der Validierungskomponente variiert, der Effekt bleibt gleich. Die Validierung findet in diesen Varianten in der Kalkulationsplattform statt.

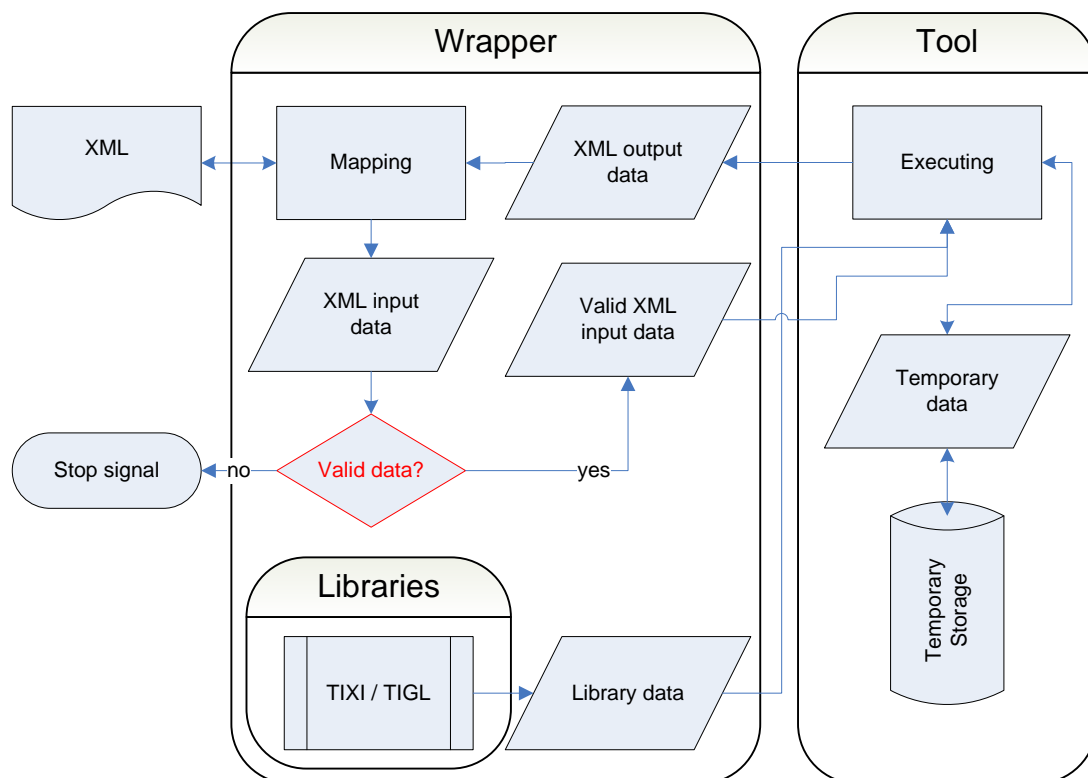


ABBILDUNG 24: VALIDIERUNGSKOMPONENTE IN CONTAINEREINHEIT

In dieser Konstellation steht die Validierungskomponente zwischen der Integrationseinheit (mapping unit) und der Ausführungseinheit (executing unit). Der XML Datenstrom wird ausschließlich vor dem Berechnungsvorgang validiert. Es wird davon ausgegangen, dass die Berechnungsprogramme korrekt arbeiten. Der Validierungsvorgang ist demnach vorbeugend (preemptive), sodass bereits vor der eigentlichen Codeausführung festgestellt werden kann, ob die Eingangsdaten für den vorliegenden Berechnungsvorgang valide sind.

Stellt die Validierung fest, dass der XML Datenstrom nicht valide ist, so sendet die Containereinheit ein Abbruchsignal (exception) an die Modellierungsplattform. Es muss sowohl verhindert werden, dass die Berechnung initiiert, als auch der Prozessablauf weitergeführt wird. Dem Anwender müssen dementsprechend Informationen über die Details des Abbruchs dargestellt werden: Abbruchhinweis, Datum und Uhrzeit des Abbruchs, Grund des Abbruchs: Fehler in Zeile X wegen Regelverletzung Y, Weiterführende Menüoptionen: ignorieren / fortsetzen, abbrechen.

Durch den Einsatz der Validierung innerhalb des Containersystems ergeben sich folgende Vor- und Nachteile in der Anwendung:

<i>Vorteile</i>	<i>Nachteile</i>
<ul style="list-style-type: none"> • Zentrale Verwaltung des Stylesheets in der Containereinheit (zuständig für alle Berechnungsprogramme) • Entlastung des Anwenders • Fehlerprävention: keine Berechnungsausführung 	<ul style="list-style-type: none"> • Synchronisierung der Schemadefinition mit der XML Datei • Langwierige Definition und Umsetzung des Regelwerkes

TABELLE 11: VOR- UND NACHTEILE DES EINSATZES DER VALIDIERUNGSINSTANZ IN DER CONTAINEREINHEIT

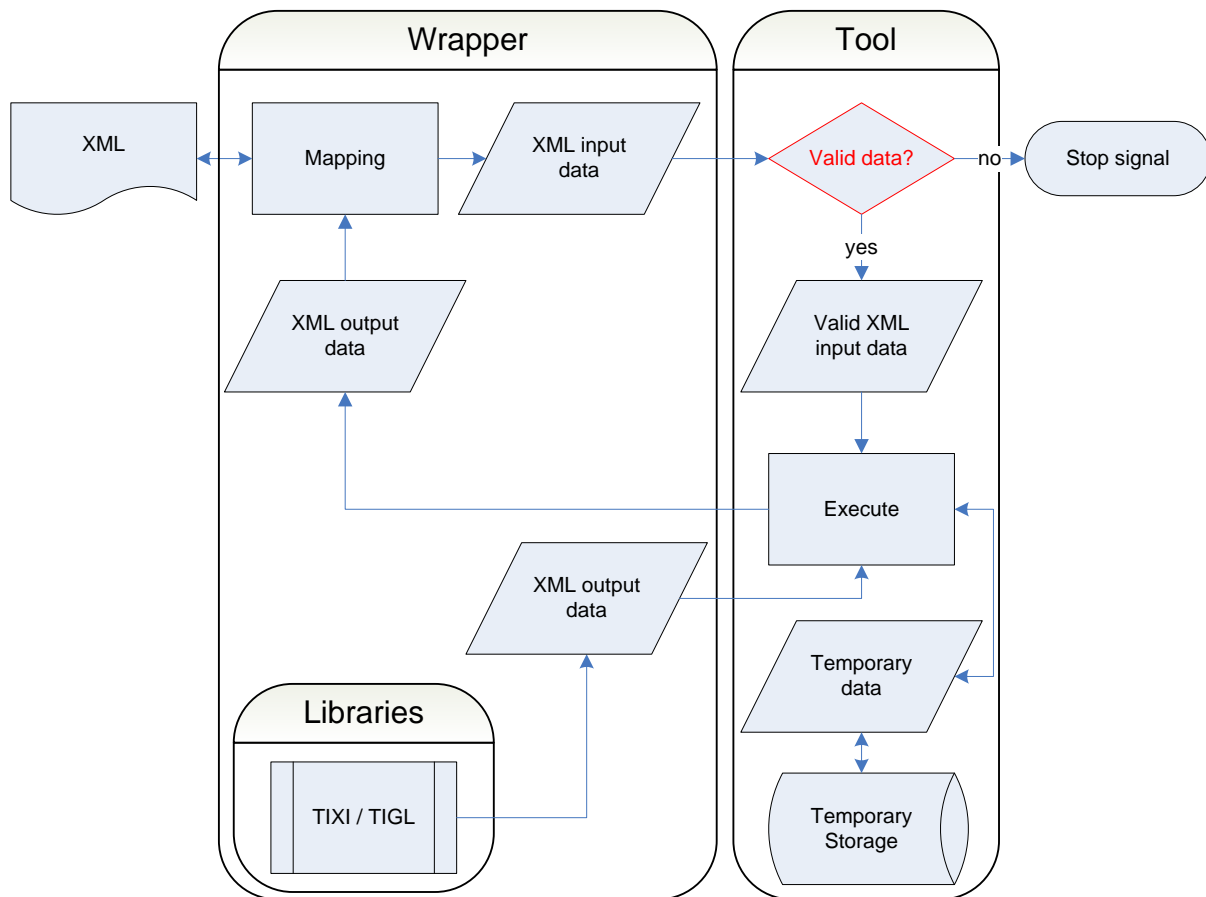


ABBILDUNG 25: VALIDIERUNGSKOMPONENTE IN AUSFÜHRUNGSEINHEIT

Die Integration der Validierung in das Berechnungssystem leitet den XML Datenstrom unangetastet über die Containereinheit weiter zu den Berechnungswerkzeugen, die diesen anschließend überprüfen soll, bevor die Berechnung initiiert wird. Hierbei kann werkzeugspezifische Validität eingesetzt werden, die die Prüfung bezüglich Umfang und Zeit verkürzt. Jedoch erfordert die Verwaltung der vielen Validierungsschemata einen höheren Aufwand und eine intensivere Wartung. Weitere Vor- und Nachteile dieser Variante:

Vorteile	Nachteile
<ul style="list-style-type: none"> • Dezentrale Verwaltung ermöglicht die Reduzierung des Regelwerkes • Verkürzung der Validierungszeit • Aufwandreduzierung in der Implementierung 	<ul style="list-style-type: none"> • Abbildung werkzeugübergreifender Regelwerke bei Parallelisierung von Berechnungsschritten • Höherer Verwaltungs- und Wartungsaufwand vieler verteilter Stylesheets

TABELLE 12: VOR- UND NACHTEILE DES EINSATZES DER VALIDIERUNGSEINHEIT IN DER AUSFÜHRUNGSEINHEIT

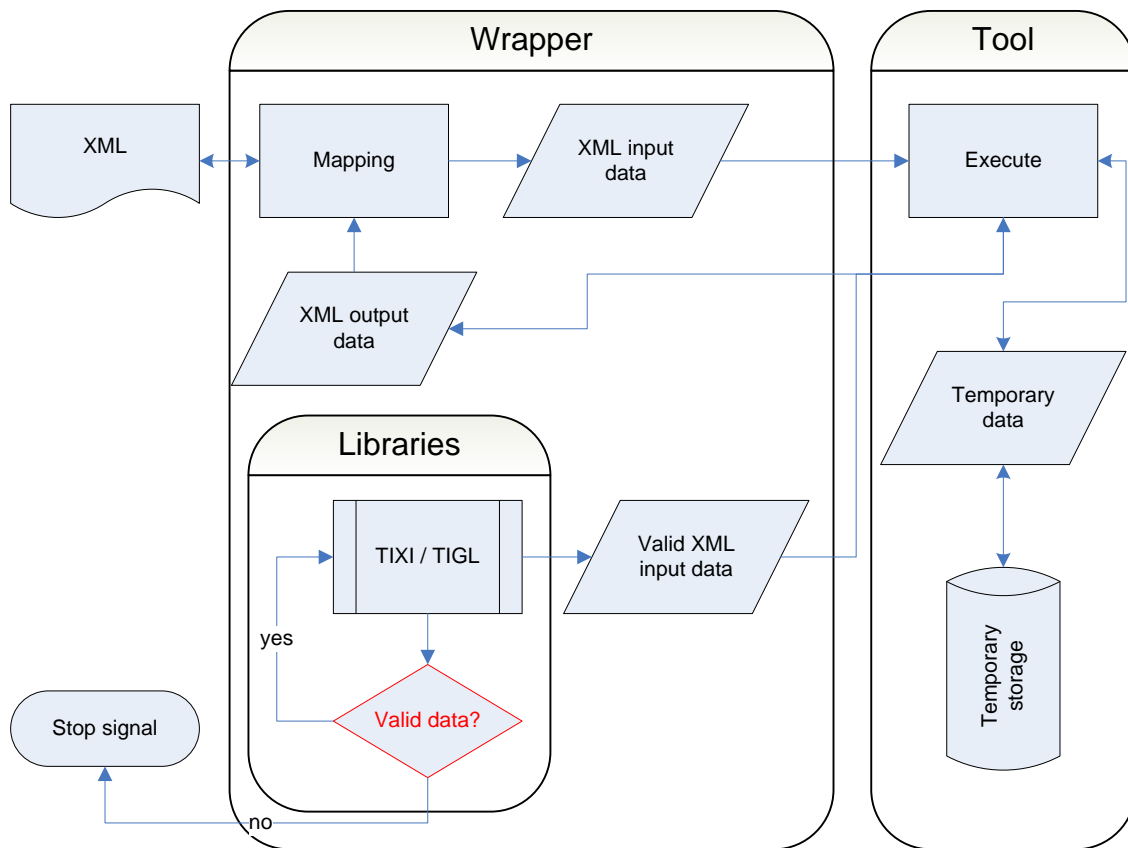


ABBILDUNG 26: VALIDIERUNGSKOMPONENTE IN BIBLIOTHEK

Die dritte Einsatzvariante zielt auf die Integration der Validierung in den Bibliotheken ab, die für die Berechnungsvorgänge hinzugezogen werden. Auf diese Weise können Berechnungen, die Bibliotheksfunktionen verwenden, validiert werden. Auch hier muss die Validierungskomponente entsprechende Priorität besitzen, um den vorzeitigen Abbruch der interaktiven Kommunikation mit dem Anwender einzuleiten.

Weiterhin müssen die Berechnungsprogramme sensibler auf Interaktionen ausgelegt werden, da die Unterbrechung der Ausführung erfolgen kann. Besondere Beachtung muss im Fall der Parallelisierung von Berechnungsvorgängen geschenkt werden. Es bedarf einer übergeordneten Instanz in der Modellierungsplattform, die die Administration der Werkzeuge übernimmt.

Vorteile	Nachteile
<ul style="list-style-type: none"> • Zentrale Verwaltung des Stylesheets in der Containereinheit (zuständig für alle Berechnungsprogramme) • Fehleranalyse auf Berechnungsebene (während der Ausführung) 	<ul style="list-style-type: none"> • Synchronisierung der Schemadefinition mit der XML Datei • Langwierige Definition und Umsetzung des Regelwerkes • Höherer Aufwand in der Entwicklung von Berechnungsprogrammen (Abbruchfunktionen, Fehlermeldung, Interaktion, Garbage Collection)

TABELLE 13: VOR- UND NACHTEILE DES EINSATZES EINER VALIDIERUNGSKOMPONENTE IN BIBLIOTHEKEN

„ An *expert system* is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice. “

Peter Jackson

Expertensysteme erfüllen Aufgaben, die normalerweise nur mit menschlicher Expertise wahrgenommen werden können. Sie assistieren in Belangen der Entscheidungsfindung oder in beratender Funktion, indem sie die Qualität der Entscheidung oder die Produktivität steigern. Die Technologie, auf der Expertensysteme beruhen, stammt aus dem Forschungsbereich der künstlichen Intelligenz. Dieser Zweig der Computerwissenschaft befasst sich mit dem Design und der Implementierung von Programmen, die im Stande sind, kognitive Denkweisen, wie das Lösen von Problemen, die visuelle Wahrnehmung und das Sprachverständnis, erlernen zu können. Typische Anwendungsfälle für Expertensysteme sind:

- Interpretation von Daten
- Fehlerdiagnose
- Analyse von komplexen Objekten
- Konfiguration von komplexen Objekten
- Planung von Vorgehensweisen

Aufbau eines Expertensystems

Der Aufbau von Expertensystemen ist gekennzeichnet durch drei autonome Komponenten, die sich gegenseitig beeinflussen und gemeinsam auf eine Wissensbasis (Wissensdatenbank) zugreifen. Gesteuert wird ein Expertensystem mittels Benutzerschnittstellen, die zur Eingabe und Abfrage von Wissen genutzt werden. Die zentrale Rolle im System spielt die Inferenzmaschine, die als Problemlösungskomponente aus Problemstellungen Daten extrahiert, diese mit dem internen Regelwerk abgleicht und so neue Schlussfolgerungen erzielt. Neben der Inferenzkomponente existiert eine Erklärungskomponente, die dem Anwender Informationen über die Beweisführung der Lösung liefert. Die dritte Komponente ist die Wissenserwerbskomponente, die den Aufbau und die Erweiterung der Wissensdatenbank verwaltet. Sie ist zudem für die Konsistenz des Datensatzes zuständig. Die Wissenserwerbskomponente kommuniziert über die Benutzerschnittstelle anfänglich mit dem „Knowledge Engineer“, der das Fachwissen interpretiert und in Maschinensprache übersetzt. Im Zuge der Installation geht die Haupttätigkeit der Benutzerschnittstelle über zur sprachlichen und grafischen Lösungspräsentation.

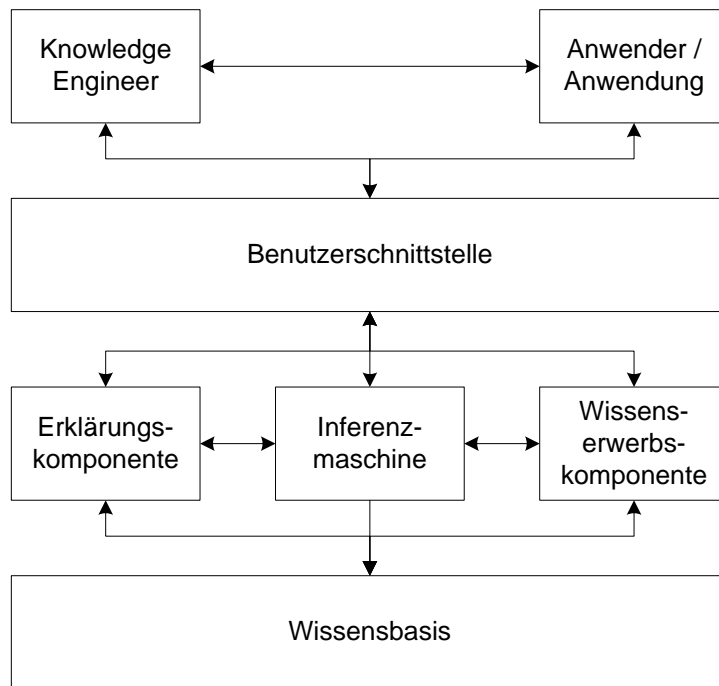


ABBILDUNG 27: AUFBAU EINES EXPERTENSYSTEMS

Entwicklungsphasen eines Expertensystems

Um eine möglichst kurze Integrationszeit zu erreichen, wird in der Implementierungsphase die Vorgehensweise des Rapid Prototyping eingesetzt. Hierbei entsteht in kurzen Zyklen ein Prototyp des Zielsystems.

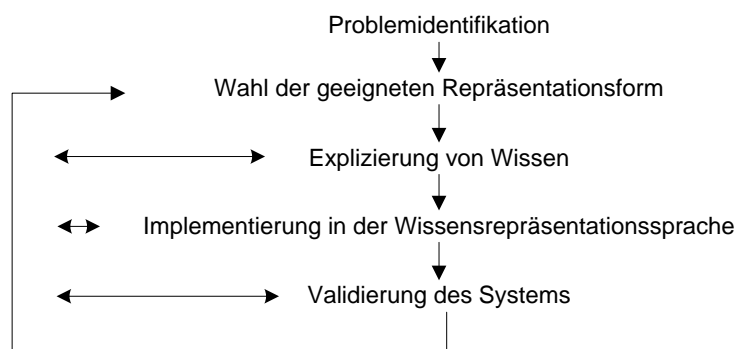


ABBILDUNG 28: ENTWICKLUNGSZYKLUS EINES EXPERTENSYSTEMS

Die Entwicklung des Expertensystems erfolgt in schnellen, kurzen Iterationen mit Einbezug der derzeitigen und zukünftigen Anwender. Eine frühe Integration der Benutzer ermöglicht eine kürzere Einführungszeit bei Inbetriebnahme des Systems. Die weiteren Entwicklungsschritte werden anschließend detaillierter betrachtet:

1. Prototypisches Design der Wissensbasis
 - a. Problemidentifikation: Zieldefinition, Randbedingungen, Ressourcen, Projektteam, Terminplanung
 - b. Formalisierung von Wissen: Wahl des Mediums der Wissensrepräsentation. Technische Fragen sollen geklärt werden. Eine Maschinensprache soll festgelegt werden. Grobe Implementierungsaspekte sind festzuhalten.

2. Implementierung und Test eines Prototyps: Sobald eine repräsentative Version des Prototyps produziert ist und diese Version akzeptable Resultate erzielt, wird die Verfeinerungsstufe der nächsten Iteration erhöht. Gleichzeitig wird die aktuelle Iteration mittels Testdaten validiert.
3. Verfeinerung beziehungsweise Generalisierung der Wissensbasis: Die zeitintensivste Tätigkeit besteht darin, den Stand der Wissensbasis auf das Niveau der beteiligten Experten zu bringen.
4. Implementierung der Benutzerschnittstelle: Die nachhaltige Entwicklung einer ansprechenden und ergonomischen Benutzeroberfläche ist der Garant für die Langlebigkeit einer Anwendung. Sie dient nicht nur dem Wissensingenieur im Vorfeld, sondern auch den Benutzern, die das System mehrere Jahre lang pflegen und warten.
5. Dokumentation: Die Erstellung von Dokumentations-, Wartungs- und Schulungsunterlagen dient ebenso der Nachhaltigkeit des Expertensystems und ermöglicht nachfolgenden Generationen die Erneuerung und Erweiterung von Systemkomponenten und Software.
6. Einführung in das Unternehmen: In dieser Phase wird das Expertensystem in die Infrastruktur des Unternehmens integriert. Zu den Aufgaben zählen hierbei die Installation der Hardware und Software sowie die Schulung des Personals. Die frühzeitige Miteinbeziehung von späteren Anwendern ist daher von Vorteil.

Wissensgenerierung

Die Phasen der Generierung eines Expertensystems sind aufgeteilt in:

- | | |
|---------------------------|--|
| 1. Wissensakquirierung | Transfer von Wissen von einer Person in das Programm |
| 2. Wissensrepräsentation | Kodierung von Wissen in Maschinensprache |
| 3. Kontrolle des Denkens | Entscheidung, wann welches Wissen angewandt wird |
| 4. Erklärung der Lösungen | Transfer von Resultaten von dem Programm zur Person |

nach [Jac98 S. 2-12]

Die Themengebiete Akquirierung und Repräsentation von Wissen wurden in vorhergehenden Kapiteln bereits angesprochen. Der Transfer und die Kodierung von Wissen in die Sprache von Maschinen erfolgt mittels semantischer Modellierung von Beziehungen zwischen attributierten Elementen. Hierfür existieren verschiedene Methoden.

- Deduktive Shells: In dieser Vorgehensweise wird Wissen allgemein in Wenn-Dann-Regeln mit arithmetischen Ausdrücken oder natürlicher Sprache dargestellt.

wenn das wetter gut ist

und die hitze erträglich

dann eis essen ist sehr wahrscheinlich

und putzen ist verschoben

- Induktive Shells: Aus einer Menge von Fallbeispielen, die tabellarisch abgelegt sind, werden selbstständig Regeln produziert. Die Reduktion auf das Wesentliche erfolgt mittels Formung eines Regelbaumes, der als „general rule“ für die Beratung herangezogen wird.
- Frames: Sie bezeichnen eine Darstellungsform, die eine übermäßige Aufblähung des Regelwerkes vermeiden kann. Hinter diesem Konzept verbirgt sich die physiologische Erkenntnis, dass auf unerwartete oder unbekannte Situationen mit bekannten Problemlösungsverfahren reagiert wird. Die Grundstruktur wird nicht erneuert, sondern es werden vereinzelt Parameter ausgetauscht und mit dem Lösungsweg wird experimentiert. Wissen wird hierbei in Karten („frames“) abgespeichert. Um die Organisation von Karten beizubehalten, werden sie in Felder („slots“) – ähnlich einer Karteikarte oder Felder einer Datenbank – eingetragen. Jeder Slot beinhaltet desweiteren eine Anzahl von freien Feldern, um Zusatzinformationen speichern zu können („subslots“, „Facetten“).

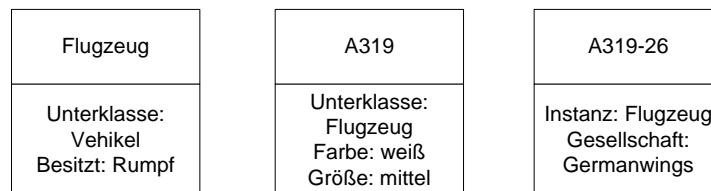


ABBILDUNG 29: DARSTELLUNG EINES FRAME-BEISPIELES

- Eine weitere Form der Wissensrepräsentation ist die Wissensbeschreibung in Termini der Aussagenlogik, wie sie in höheren Programmiersprachen durch Einsatz von logischen (boolsche) Operatoren angewandt wird.

Art(Flugzeug, Vehikel)

Aktivität(Transport, Flugzeug)

Bestandteil(Tragfläche, Flugzeug)

Dies wird folgendermaßen interpretiert:

Flugzeug ist eine Art von Vehikel

Transport ist die Aktivität von Flugzeug

Tragfläche ist Bestandteil von Flugzeug

- Semantische Netze: Sie sind eine Form der Darstellung und Interpretation von aussagenlogischen Informationen.

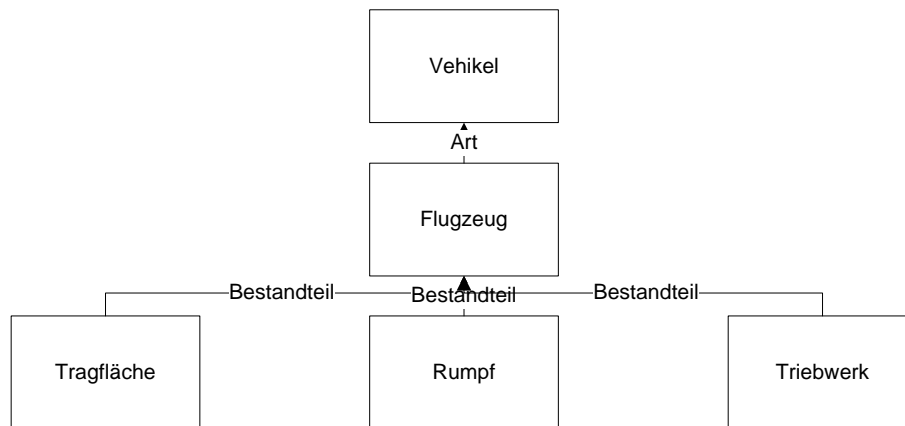


ABBILDUNG 30: SEMANTISCHES BEISPIELMODELL

- Regelbasierte Systeme: Sie modellieren eine Sammlung von Anweisungen, die dem Anwender in bestimmten Situationen beraten und Entscheidungsmöglichkeiten offenlegen. Die Architektur eines regelbasierten Systems besteht aus einem Katalog von Wenn-Dann-Regeln, einer Datenbank, die Fakten über den Fachbereich enthält, und einem Interpreter, der die Regeln administriert.

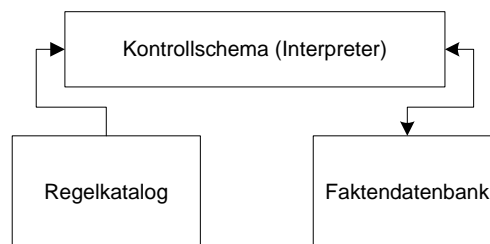


ABBILDUNG 31: ARCHITEKTUR REGELBASIERTER SYSTEME

Die Funktionsweise des Interpreters (Inferenzmaschine) basiert auf Transformationsregeln, die es erlaubt, bekannte Ausdrücke in einer formalen Sprache umzuformen, sodass daraus neue Ausdrücke resultieren. Der Interpreter kann mithilfe von fünf „klassischen“ Schlussregeln diese neuen Ausdrücke beweisen, beziehungsweise schlussfolgern:

1. Modus ponens (lat. bejahende Schlussweise): direkter Beweis

$$\frac{p \rightarrow q \quad p}{q}$$

Wenn p eine hinreichende Bedingung für q ist und p wahr ist, dann ist auch q wahr.

2. Modus tollens (lat. verneinende Schlussweise): indirekter Beweis

$$\frac{p \rightarrow q \quad \neg q}{\neg p}$$

Wenn p eine hinreichende Bedingung für q ist und q nicht wahr ist, dann ist auch p nicht wahr.

3. Kettenschluss

$$\frac{p \rightarrow q \quad q \rightarrow r}{p \rightarrow r}$$

Wenn *p* eine hinreichende Bedingung für *q* ist und *q* eine hinreichende Bedingung für *r* ist, dann ist *p* eine hinreichende Bedingung für *r*.

4. Modus tollendo ponens (lat. durch Verneinung bejahende Schlussweise)

$$\frac{p \cup q \quad \neg p}{q}$$

Wenn *p* oder *q* gilt und *p* nicht wahr ist, dann ist *q* wahr.

5. Widerspruch

$$\frac{\neg p \rightarrow (q \cap \neg q)}{p}$$

Wenn nicht-*p* eine hinreichende Bedingung dafür ist, dass ein Widerspruch wahr ist, dann ist *p* wahr.

nach [Bie08 S. 38-40]

Zusammenfassung

Die dargestellten Methoden unterscheiden sich in der Anwendung. Während die Aussagenlogik und semantische Netze in den Bereich der deklarativen Wissensrepräsentation fallen, werden Produktionsregelsysteme sowie die objektorientierte Programmierung in den Bereich der prozeduralen Wissensrepräsentation eingestuft. Die Frames-Methode stellt sich als Mittler zwischen den Teilbereichen dar, es können beide Seiten abgedeckt werden.

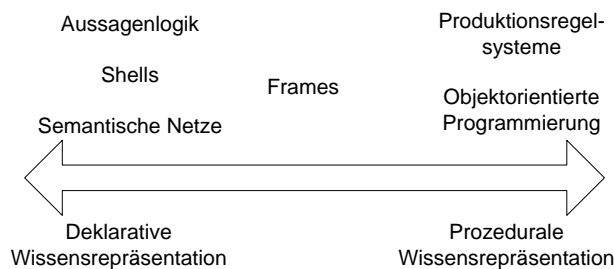


ABBILDUNG 32: DEKLARATIVE UND PROZEDURALE WISSENSREPRÄSENTATION NACH [KUR92]

Um einen Überblick über die aufgezählten Methoden der Wissensrepräsentation und Wissenskontrolle zu erlangen, werden nachfolgend die Vor- und Nachteile aufgeführt.

Methode	Vorteile	Nachteile
Regelbasierte Systeme	<ul style="list-style-type: none"> - Wissensdatenbank enthält Fakten und Regeln - Die Ausführung der Regeln, Bestimmung der Variablenwerte, der Zugriff auf das Da- 	<ul style="list-style-type: none"> - Schwierigkeiten bei Modellierung von vagen Informationen - schwerfällig bei nicht-prozeduralem Wissen

	tensystem sind der Inferenzmaschine (Schlußfolgerungssystem) überlassen	
Deduktive Shells	- klassische Wenn-Dann-Regeln auch in natürlicher Sprache möglich	- Definition eines Ermittlungszieles bei Regelketten notwendig
Induktive Shells	- einfache Form der Wissensaufbereitung - Redundante, ungeordnete Eingabemenge wird analysiert	- Beispielmenge kann unverhältnismäßig hoch werden (mehrere Bedingungen, eine oder mehrere Lösungen)
Frames	- Vermeidung von aufgeblähten Wissensbasen durch Kategorisierung	
Aussagenlogik	- bewährter Einsatz - behandelt Funktionen boolescher Wahrheitswerte	- schwierige Modellierung von Unschärfe. Ausgleich durch Fuzzy-Logik
Semantische Netze	- grafische, anschauliche Darstellung von Wissen	- Geringe Transparenz bei zunehmender Komplexität - kein einheitlicher Standard

TABELLE 14: VOR- UND NACHTEILE VON METHODEN DER WISSENSREPRÄSENTATION

Aufgrund der bestehenden Modellierung von Flugzeugdaten in XML und ansatzweise UML, ist im vorliegenden Projekt der Einsatz von regelbasierten Produktionssystemen zu empfehlen, da nicht nur Regeln erstellt, sondern auch Fakten und Variablenwerte mit einbezogen werden können. Das Regelwerk sollte aufgrund der experimentellen Natur restriktive Schranken weitestgehend offen lassen.

Mechanismen in Expertensystemen

Das Regelwerk eines Expertensystems besteht aus Bedingungen und Aktionen. Ähnlich der Systematik eines Petrinetzes „feuert“ eine Regel/Transition erst, wenn alle Bedingungen/Stellen erfüllt/besetzt sind. Erst dann kann die Aktion folgen.

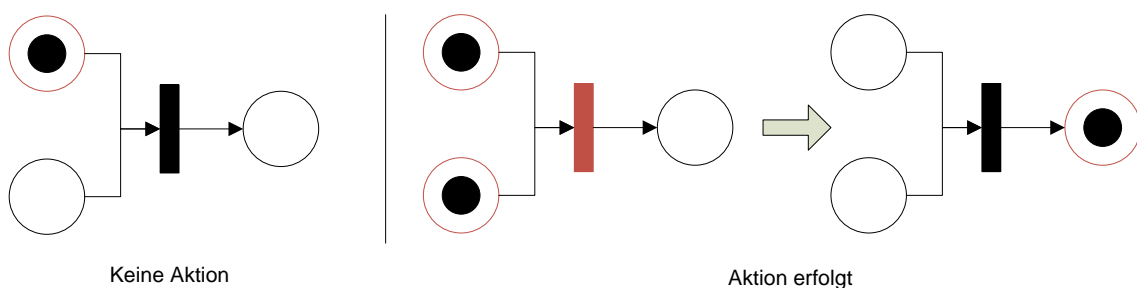


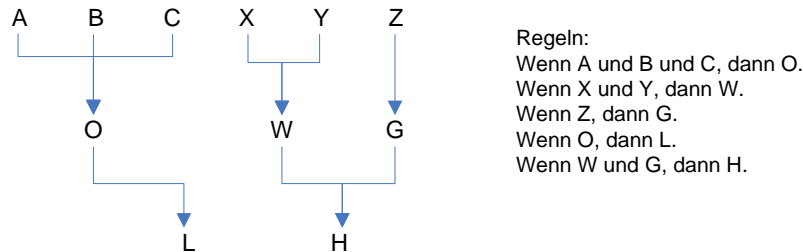
ABBILDUNG 33: SCHALTUNG EINES PETRINETZES

Für die Abarbeitung des integrierten Regelwerkes sind in der Regel zwei Methoden verantwortlich.

- Vorwärtsverkettung (forward chaining): Bei der Vorwärtsverkettung werden alle Lösungen hergeleitet, die der Inferenzmaschine aus den Ausgangsinformationen ersichtlich sind. Das System prüft iterativ alle Regeln deren Vorbedingungen erfüllt sind und wechselt in einen neuen Zustand. In diesem Zustand prüft das System wiederum alle Regeln, deren Bedingungen erfüllt sind.

Dieses System soll an einem Beispiel erläutert werden:

Gegeben sind Fakten X und Y. Das System ermittelt nun die ableitbaren Lösungen aus dem Regelbaum.



Im vorliegenden System kann aufgrund der gegebenen Fakten X und Y nur eine Regel „feuern“. Die präsentierte Lösung ist W.

- Rückwärtsverkettung (backward chaining): Das Pendant zur Vorwärtsverkettung ist die Rückwärtsverkettung. Diese Methode erfragt die Lösung, indem sie den Aktions- teil betrachtet, in dem das Ziel enthalten ist. Sind unbekannte Parameter in gefunde- nen Aktionsteilen enthalten, werden diese Unbekannten als Unterziele deklariert und zusätzlich rekursiv bestimmt. Das Resultat ist dann gefunden, wenn für ein Ziel alle Unterziele bestimmt wurden.

Betrachtet man obiges Beispiel noch einmal und wendet die Rückwärtsverkettung an, so ergibt sich folgende Ergebnismenge.

Gegebenes Faktum: H

Soll L bewiesen werden, so führen zwei Pfade zu möglichen Zielen:

$H \rightarrow W$ und $H \rightarrow G$ und damit $W \rightarrow X, Y$ und $G \rightarrow Z$

Angenommen der Interpreter nähme den ersten Weg, dann ist W ein unbe- kannter Parameter, der als Unterziel hinzugefügt wird. Da eine Regel existiert, die W beweisen kann („Wenn X und Y, dann W.“), wird die Rückwärtsverket- tung des Interpreters auf W „gesetzt“. Nun verfolgt der Mechanismus wieder- um den Beweis des aktuellen Zustandes, nämlich W. Dies erfolgt solange, bis keine Regeln für gefundene Unbekannte vorhanden sind. In diesem Zustand muss der Interpreter Fakten vom Benutzer erfragen (hier: X und Y). Bestätigt der Benutzer, dass X und Y wahr sind, ist die Beweisführung abgeschlossen. Bestätigt der Benutzer, dass X und Y nicht wahr sind, kann W nicht bewiesen werden. Der Interpreter muss den nächsten Lösungsweg überprüfen (hier: G).

Die Entscheidung zwischen Vorwärts- und Rückwärtsverkettung geschieht auf Basis der Fak- ten- und Regelmenge. Sind wenige Regeln sehr vielen Fakten gegenübergestellt, ist die Ten- denz zur Vorwärtsverkettung gegeben. Diese Methode wird ebenfalls favorisiert, wenn alle ableitbaren Lösungen betrachtet werden sollen. [Win93]

Die Anwendung des Expertensystemkonzepts bringt folgende Vor- und Nachteile mit sich:

Vorteile	Nachteile
<ul style="list-style-type: none"> • Lernfähiges Unterstützungssystem • Gleichbleibende Urteilsfähigkeit: Das 	<ul style="list-style-type: none"> • Begrenzter Wissensumfang des Ex- pertensystems: Randgebiete oder

<p>System wird nicht beeinflusst durch Stress oder Emotionen -> Einsatz unter Extrembedingungen möglich</p> <ul style="list-style-type: none">• Wissenstransfer: Expertenwissen wird expliziert und dokumentiert. Lerneffekt für andere Mitarbeiter	<p>andere Fachgebiete werden im Problemlösungsprozess vernachlässigt</p> <ul style="list-style-type: none">• Problemverständnis durch streng formalisierte Eingabe eingeschränkt• Soziale Gründe der Verweigerung der Weitergabe von Wissen
--	--

TABELLE 15: VOR- UND NACHTEILE DES EINSATZES EINES EXPERTENSYSTEMS

KAPITEL 6 : LÖSUNGSBEWERTUNG

Nachdem die verschiedenen Konzepte dargelegt wurden, müssen diese in einem anforderungsorientierten Vergleich miteinander abgewogen werden. Hierbei werden sowohl technische als auch wirtschaftliche Interessen herangezogen und in einem Wertesystem eingeordnet. Ziel dieses Kapitels ist die Analyse der Lösungsmöglichkeiten, die Abschätzung der bestehenden Chancen und Risiken.

6.1 BEWERTUNG ALTERNATIVER SYSTEME NACH ERFÜLLUNGSKRITERIEN

Die Schwierigkeit des Abwägens zwischen verschiedenen Lösungsmöglichkeiten und deren Konsequenzen wächst mit der Zahl der Anforderungskriterien. Die Disziplin des Systems Engineering bietet zu diesem Zweck einige Methoden, die unterstützend in der Lösungsfindung sein können. Die Anwendung von Fehlerbäumen, Wirkungsgitter oder –netze, Kosten-Wirksamkeitsanalysen und Beeinflussungsmatrizen sowie Nutzwertanalysen zeigen in subjektiver Weise auf, welche Aspekte einer Lösungsalternative förderlich für das gesetzte Ziel sind. In vorliegender Arbeit wurden bereits allgemeine Anforderungskriterien festgesetzt (vgl. KAPITEL 4 : ZIELFORMULIERUNG).

„Nutzwertanalyse ist die Analyse einer Menge komplexer Handlungsalternativen mit dem Zweck, die Elemente dieser Menge entsprechend den Präferenzen des Entscheidungsträgers bezüglich eines multidimensionalen Zielsystems zu ordnen.“

[Zan76 S. 45]

Auf dieser Basis entsteht nachfolgend ein Wertesystem aus hierarchisierten und gewichteten Kriterien sowie Alternativen zugeordneten Erfüllungsgraden. Das Resultat dieser Analyse ist die Findung einer bestmöglichen Lösung – in diesem Fall zeichnet sich die geeignetste Lösungsalternative durch die höchste Punktzahl aus, die aus dem Produkt von Gewichten und Erfüllungsgraden sowie der Addition von Kriterienpunkten errechnet wird.

$$N_i = \sum_{z=1}^r n_{iz} g_z \quad \text{mit} \quad \sum_{z=1}^r g_z = 1$$

N_i = Nutzwert der Alternative i

n_{iz} = Teilnutzwert der Alternative i in Bezug auf das Kriterium z

g_z = Gewicht des Kriteriums z

Zur Ermittlung der Kriteriumsgewichtung wird eine Skala von 0 bis 4 festgesetzt.

Paarweiser Vergleich: Gewichtung 0-4	Installationsaufwand	Vorbereitungsaufwand	Bedienung	Funktionalität	Interaktivität	Erweiterbarkeit	Integrierbarkeit	Summe	Prozent
Installationsaufwand		1	1	1	1	1	1	6	7%
Vorbereitungsaufwand	3		1	1	2	3	3	13	15%
Bedienung	3	3		2	3	3	3	17	20%
Funktionalität	3	3	2		2	2	3	15	18%
Interaktivität	3	2	1	2		2	3	13	15%
Erweiterbarkeit	3	1	1	2	2		3	12	14%
Integrierbarkeit	3	1	1	1	1	1		8	10%
								84	100%

TABELLE 16: PAARWEISER VERGLEICH DER OBERKRITERIEN

Anschließend werden die Lösungsalternativen mittels Erfüllungsgrad bezüglich den gefundenen Ober- und Unterkriterien ausgewertet.

Nutzwertanalyse: Erfüllungsgrad EG: 1-100		Dokumentation	Wissensplattform	XML Validierung	Toolbasierte Semantik	Expertensystem
Kriterien\Alternativen	Gewichtung	EG	EG	EG	EG	EG
Installationsaufwand	6					
Technische Installation einfach	3	95	40	100	95	75
Know-How Anforderung niedrig	2	100	70	80	80	70
Anschaffungskosten niedrig	1	100	70	95	95	90
<i>Teilsumme</i>	6	585	330	555	540	455

Vorbereitungsaufwand	13					
Administrationsaufwand möglichst gering	4	95	60	80	90	80
Schulung der Teilnehmer schnell/einfach	3	100	90	70	85	60
Aufwand der Regelwerksinitiiierung gering	6	90	80	70	75	65
<i>Teilsumme</i>	13	1220	990	950	1065	890
Bedienung	17					
Intuitivität hoch	6	95	85	80	85	80
Änderungsaufwand niedrig	5	60	70	70	75	70
Know-How Anforderung gering	3	100	85	90	95	85
(Gefühlte) Geschwindigkeit hoch	3	70	60	100	100	90
<i>Teilsumme</i>	17	1380	1295	1400	1470	1355
Funktionalität	15					
Fehlerprävention funktioniert gut	3	30	35	45	55	90
Fehleranalysefähigkeit hoch	5	10	35	55	55	80
Genauigkeit der Fehlerbeschreibung/-lokalisierung	3	5	10	50	60	80
Manifestation von Wissen vorhanden	2	50	60	40	40	70
Änderbarkeit (z.B. deaktivieren) einfach/dynamisch	2	100	100	40	35	30
<i>Teilsumme</i>	15	455	630	720	770	1110
Interaktivität	13					
gemeinsamer Zugriff möglich	3	100	100	100	100	95
Einbindung der Abteilungen möglich	3	80	95	80	75	90
Zusammenarbeit der Abteilungen forciert	4	50	80	90	85	90
Möglichkeiten zum Experimentieren/Debug	3	5	5	30	30	40
<i>Teilsumme</i>	13	755	920	990	955	1035
Erweiterbarkeit	12					
Plattformübergreifende Formate genutzt	3	95	95	90	80	75
Interne Schnittstellen vorhanden	2	70	75	60	60	65
Ist Technologie zukunftssicher	2	95	80	70	70	85
Wartung einfach/günstig	4	90	80	70	75	75
Know-How Anforderung gering	1	90	95	70	75	70
<i>Teilsumme</i>	12	1065	1010	880	875	895
Integrierbarkeit	8					
Externe Schnittstellen vorhanden	3	80	85	75	65	70
Systemabhängigkeit gering	2	100	80	70	80	70
Aufwand gering	3	95	75	70	70	70
<i>Teilsumme</i>	8	725	640	575	565	560
Summe	84	6185	5815	6070	6240	6300

TABELLE 17: NUTZWERTANALYSE

Die Auswertung der Nutzwertanalyse zeigt, dass sich die Alternative „Expertensystem“ etwas von den übrigen abhebt. Das System besticht hinsichtlich der Integrierbarkeit, des Funktionsumfangs sowie der Erweiterbarkeit. Im Mittelfeld befinden sich die Alternativen „Toolbasierte Semantik“, „Dokumentation“ und „XML Validierung“, die allesamt eine gute Bewertung erhielten, jedoch in den Eckpunkten beispielsweise der Bedienung oder der Funktionalität nicht das Optimum erreichen konnten. Das Schlusslicht bildet die Alternative „Wissensplattform“, die vor allem hinsichtlich des Installationsaufwandes sowie in der Integrierbarkeit Schwierigkeiten aufweist.

Die Empfehlung fällt daher auf das Expertensystem, das als zukunftsichere Investition für vorliegendes Langzeitprojekt dienlich sein wird.

6.2 RISIKOBEWERTUNG

Allen genannten Alternativen liegen elementare Risiken zugrunde, die besonderer Betrachtung bedürfen, weil sie die Umsetzung der vorliegenden Untersuchung wesentlich behindern können.

Die Risikobewertung folgt der Risikoanalyse, die die Ziele festlegt, die Gefährdungen identifiziert und das Risiko einzuschätzen versucht. Die Bewertung entscheidet über die Vertretbarkeit der gefundenen Risiken. Diese beiden Methoden bilden die Komponente der Risikobeurteilung. Sind die Risiken erkannt und bewertet, folgt deren Management (von lat. „manus“ bedeutet „Hand“). Das Management besteht aus der Risikokontrolle, die die Handlungsoptionen analysiert und Umsetzungsstrategien anbietet. Außerdem wird das Restrisiko sowie das Gesamtrisiko in einem Bewertungsschema interpretiert. Zuletzt folgt die Produktbetrachtung, die aus der Beobachtung der Marktsituation gewonnenen Erfahrungen wiederum analysiert, bewertet und kontrolliert. Da es sich im vorliegenden Projekt nicht um ein kommerzielles Produkt handelt, wird die Vorgehensweise auf die interne Ausführung übertragen und gleichbedeutend behandelt.

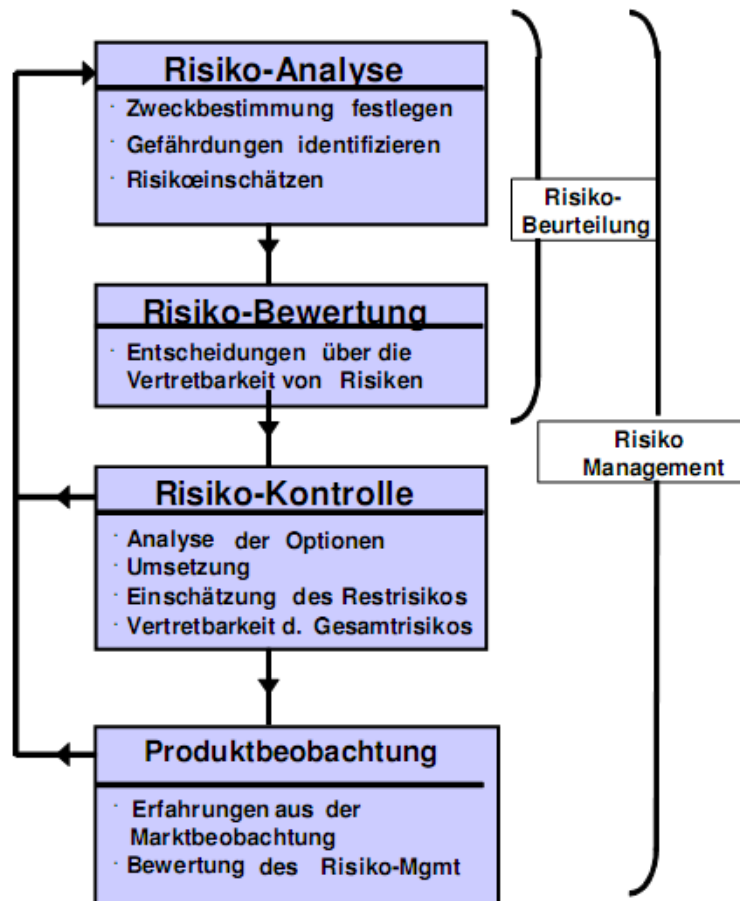


ABBILDUNG 34: RISIKOANALYSE UND -BEWERTUNG NACH [ISO14971]

Risikoanalyse

Die Einschätzung der Risiken erfolgt auf Grundlage von Untersuchungen, die den Prozess sowie das Umfeld des Prozesses analysieren. Hierbei werden verschiedene Betrachtungsweisen herangezogen, die das Spektrum der Analyse erweitern sollen.

Unter Zuhilfenahme eines Ursache-Wirkungsdiagrammes (auch Ishikawa-Diagramm oder Fischgrätendiagramm genannt), das Anfang der 1940er Jahre von Kaoru Ishikawa entwickelt wurde und Kausalitätsbeziehungen darstellt, werden Risiken der Umsetzung der Validierung ermittelt. Die vier Kategorien „Mensch“, „Maschine“, „Methoden“, „Management“ stellen verschiedene Blickwinkel auf das vorgegebene Negativziel „Projektrisiko“ dar. Ursprünglich existierten nur vier Haupteinflussgrößen (4M: Material, Mensch, Maschine, Methode). Diese wurden im Laufe der Zeit um weitere drei Größen (7M: Management, Mitwelt, Messung) ergänzt. Heutzutage werden auch andere Einflussgrößen, wie Prozesse und Umfeld, hinzugezogen. Das Ziel der Darstellungsform ist die Gruppierung von Ursachen in immer feinere Strukturen. Es ist eine leicht anwendbare Methode zur Förderung des Verständnisses von Ursache und Wirkung hinsichtlich eines gesetzten Zieles. Nachteilig sind jedoch die steigende Unübersichtlichkeit bei umfangreichen und komplexen Problemen, die Missachtung der Wechselwirkungen sowie die Nichteinbeziehung von zeitlichen Abhängigkeiten.

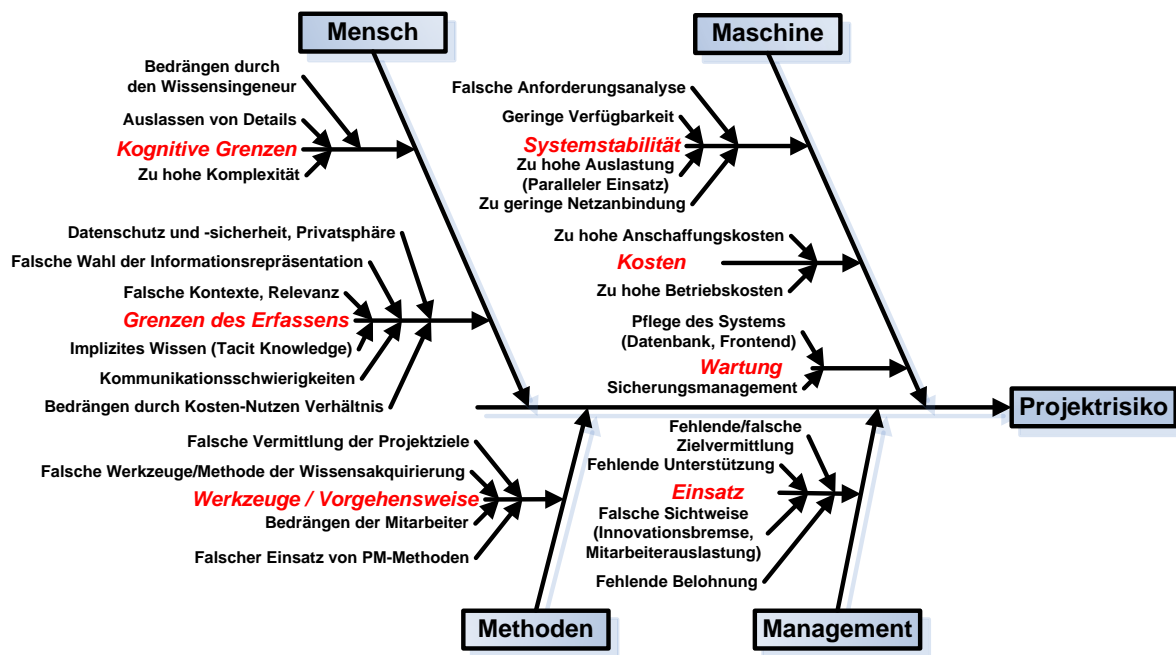


ABBILDUNG 35: URSACHE-WIRKUNGSDIAGRAMM DER PROJEKTRISIKEN

Risiken in der Kategorie „Mensch“ stellen vor allem die menschlichen Grenzen im Bereich des Wissenserwerbs dar. Kognitive Grenzen zeigen sich durch die Leistung des menschlichen Gehirns in Bezug auf den Erfassungsgrad hochkomplexer Zusammenhänge und kontextueller Überschreitungen. Dem Wissensingenieur steht nur ein begrenzter Zeitraum zur Akquise von Wissen bereit. In diesem Zeitraum muss er die feinen Wissensstrukturen der Mitarbeiter herauskristallisieren und dabei die verschiedenen Kontexte ansprechen. So kann sich beispielsweise die Bedrängung der Mitarbeiter negativ auf deren Wohlbefinden und damit deren Anteilnahme am Projekt auswirken. Dies manifestiert sich im Auslassen von Details, das bewusst, aber auch unbewusst geschehen kann. Bei hoher Komplexität des Sachverhaltes können Anzeichen von Resignation eintreten. Neben den kognitiven Fähigkeiten gibt es zudem weitere Grenzen, die Risiken bergen, wie zum Beispiel der betriebliche Datenschutz oder Datensicherheit, Kommunikationsschwierigkeiten sowie das Nichtvermitteln von implizitem Wissen (engl. „tacit knowledge“). Die falsche Wahl der Repräsentationsplattform inklusive der gemeinsamen Symbolbasis sowie die falsche Kommunikation des Kontext und der Informationsrelevanz können den Wissenstransfer verhindern. Ein ernstzunehmendes Problem ist weiterhin die Privatsphäre der Mitarbeiter, die bewusst Wissensvermittlung aus sozialen Gründen verweigern. Soziale Gründe wären die Angst vor Ersetzung des Arbeitsplatzes, der Vorbehalt von Kniffen und Tricks zur Beibehaltung einer unternehmenswichtigen Position oder eines bestimmten Status [Dut06 S. 78-85].

Die nächste Kategorie erörtert die Projektrisiken im Bezugssystem „Maschine“. Hierbei können falsche Anforderungsanalysen zu Systeminstabilitäten führen, wenn die parallele Berechnung von Simulationen eine zu hohe Auslastung mit sich bringt. Weitere Risiken sind eine zu gering dimensionierte Netzanbindung oder –infrastruktur, die zu geringer Systemverfügbarkeit führen. Die nachhaltige Wartung des Systems erfordert die Pflege der Wissensdatenbank sowie der weiteren Komponenten, wie die Benutzeroberfläche, sowie ein zukunfts-

sicheres Backupmanagement. Neben diesen Argumenten stehen zudem die Projektkosten im Raum, die sich unterteilen in Anschaffungs- und Betriebskosten.

In der Kategorie „Methode“ befinden sich Werkzeuge und Vorgehensweisen des Wissensingenieurs, die, in unzutreffender Weise angewendet, kontraproduktiv hinsichtlich des Wissenserwerbs sein können. Die Wahl der Werkzeuge und Methoden ist von hoher Bedeutung, da sie in der fragilsten Phase der Umsetzung eingesetzt werden und Grund eines Projektabbruches sein können.

Die entscheidende Stütze des Projektes bildet das „Management“. Ohne diese Kategorie würde das Projekt nicht existieren. Fehlt die notwendige Unterstützung des Managements gegenüber dem Projekt, ist es zum Scheitern verurteilt. Das Management ist verantwortlich für die geeignete Kommunikation des Projektes inklusive der zu verfolgenden Ziele sowie die projektbegleitende Aufrechterhaltung des Kontakts mit den designierten Projektleitern.

Nachfolgend werden den im Ursache-Wirkungsdiagramm gefundenen Projektrisiken Zahlen zugeordnet, die sodann in einem Risikodiagramm kategorisiert werden.

- ① Bedrängen durch den Wissensingenieur
- ② Auslassen von Details
- ③ Zu hohe Komplexität
- ④ Datenschutz und -sicherheit, Privatsphäre
- ⑤ Falsche Wahl der Informationsrepräsentation
- ⑥ Falsche Kontexte, Relevanz
- ⑦ Implizites Wissen (Tacit Knowledge)
- ⑧ Kommunikationsschwierigkeiten
- ⑨ Bedrängen durch Kosten-Nutzen-Verhältnis
- ⑩ Falsche Vermittlung der Projektziele
- ⑪ Falsche Werkzeuge/Methode der Wissensakquirierung
- ⑫ Bedrängen der Mitarbeiter
- ⑬ Falscher Einsatz von PM-Methoden
- ⑭ Falsche Anforderungsanalyse
- ⑮ Geringe Verfügbarkeit
- ⑯ Zu hohe Auslastung (Paralleler Einsatz)
- ⑰ Zu geringe Netzanbindung
- ⑱ Zu hohe Anschaffungskosten
- ⑲ Zu hohe Betriebskosten
- ⑳ Pflege des Systems (Datenbank, Frontend)
- ㉑ Sicherungsmanagement
- ㉒ Fehlende/falsche Zielvermittlung
- ㉓ Fehlende Unterstützung
- ㉔ Falsche Sichtweise (Innovationsbremse, Mitarbeiterauslastung)
- ㉕ Fehlende Belohnung

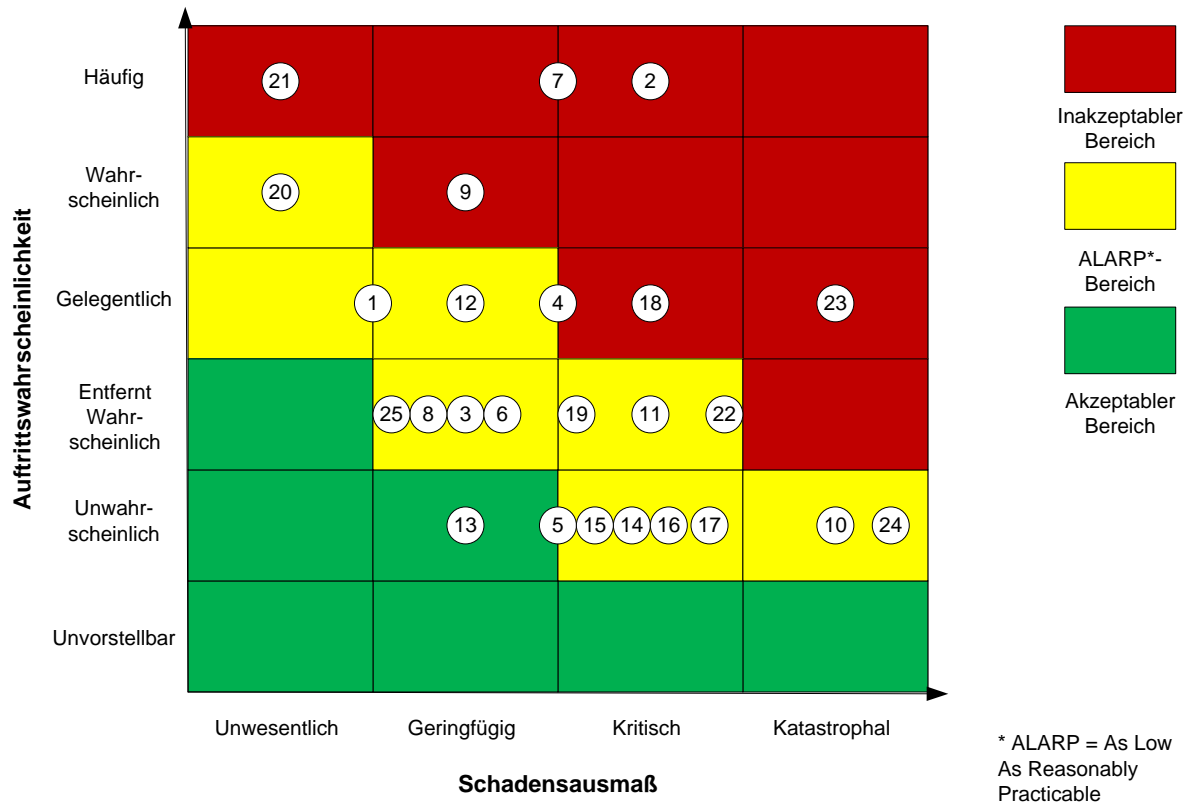


ABBILDUNG 36: RISIKODIAGRAMM

Die Auswertung zeigt, dass sich die Mehrzahl der gefundenen Risiken im zulässigen Bereich befindet. Ein Projekterfolg ist unter gegebenen Umständen möglich, wenn die Risikofaktoren, die sich im inakzeptablen Bereich befinden, vermieden werden können. Hervorzuheben sind die notwendige, zur Motivierung führende Unterstützung des Managements sowie die Vorarbeit des Wissensingenieurs, der interdisziplinäre Kenntnisse und gute soziale Kompetenzen aufweisen sollte. Projektspezifische sowie soziale Konflikte können durch entsprechende Gegenmaßnahmen oder mittels vorausschauender Planung gelöst werden. Dies ist jedoch nicht Teil der vorliegenden Arbeit und fällt demnach in den Aufgabenbereich der verantwortlichen, beteiligten Personen.

KAPITEL 7 : AUSBLICK UND FAZIT

Im letzten Kapitel der vorliegenden Arbeit wird ein zusammenfassender Überblick über die Methoden und Vorgehensweisen sowie über Schwierigkeiten im Verlauf der Ausarbeitung gegeben. Weiterhin werden kurz Konzepte erläutert, die sich thematisch in naher Umgebung zu dieser Studie befinden und somit einen Einfluss auf das tatsächliche System haben könnten.

7.1 AUSBLICK

Das Projekt VAMP ist für einen Zeitraum von zwei Jahren ausgelegt. Die Validierung der berechneten Daten erfolgt zum aktuellen Zeitpunkt durch den Abgleich mit realen Konfigurationen sowie aus Ergebnissen anderer Vorentwurfverfahren, wie PrADO (Preliminary Aircraft Design and Optimization program, Flugzeugentwurfssystem der TU Braunschweig) und FLOPS (Flight Optimization System, Flugzeugentwurfssystem der NASA) [Nag09 S. 6]. Neben diesem Vergleichsverfahren müssen Experten weitere Einschätzungen abgeben sowie Richtlinien der Deutschen Flugsicherung hinzugezogen werden, um ein umfassendes Regelwerk erstellen zu können. Dies wird eine der umfangreicheren Aufgaben sein, die das Projektteam in naher Zukunft zu bewältigen hat.

Die Verknüpfung des Themas Validierung mit anderen Arbeitspaketen ist denkbar. Die Projektplanung sieht in dem Arbeitspaket 4.1, der die Einrichtung Simulations- und Softwaretechnik betrifft, neben der Validierung von Simulationsketten auch die Bearbeitung der Unterarbeitspakete Provenienz und Konfigurationsmanagement vor. Aufgrund des projektübergreifenden Umfangs eines Validierungssystems können mögliche Anknüpfungspunkte zu diesen Themen diskutiert werden.

7.1.1 PROVENIENZ

Provenienz bezeichnet in softwaretechnischer Hinsicht die Verfolgung von Veränderungen. Während des Ablaufes des Entwurfsprozesses wird der CPACS-Datensatz iterativ von den Werkzeugen gelesen und in neue Datensätzen gespeichert. Im Fehlerfall soll eine nachvollziehbare Fehlerkette ersichtlich sein, die das Verständnis des Ergebnisses sowie dessen Herkunft (auf lat. provenire bedeutet herkommen) fördert. Es wird ein Konzept gefordert, das die benötigten Daten der Prozesskette sicherstellt und übersichtlich aufbereitet.

Die Funktionalität eines Provenienzsystems liegt in der Aufzeichnung einer Kette aus einer oder mehreren Ursachen, einer Abfolge von Zwischenschritten und einem Endergebnis. Dies entspricht in softwaretechnischer Sichtweise einem Protokollierungsdienst, der versteckt in das Zielsystem interagiert. Nachfolgend ist die allgemeine Struktur der Provenienzfunktionalität dargestellt.

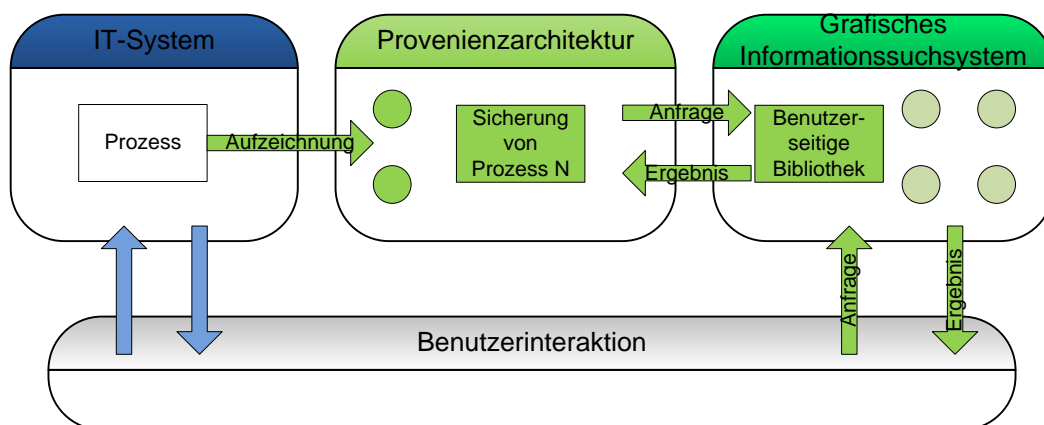


ABBILDUNG 37: ARCHITEKTUR DER PROVENIENZFUNKTIONALITÄT NACH [KUN08]

Die Provenienzarchitektur (hellgrün) sammelt Daten über ausgeführte Prozesse im IT-System (blau) und speichert diese Vorgänge in sogenannten „Provenance stores“ (hellgrüne Kreise). Das IT-System stößt die Aufzeichnung von sich aus an. Die Provenienzfunktionalität erlaubt neben der Datensicherung den Zugriff auf gespeicherte Daten mittels einer Abfragesprache. Dieses Informationssuchsystem (dunkelgrün) hält „Provenance data“ (dunkelgrüne Kreise) bereit, die zur Beantwortung der Benutzerfragen herangezogen werden. Die Unterscheidung zwischen „Provenance stores“ und „Provenance data“ dient in obiger Darstellung nur der Veranschaulichung. Der Benutzer hat sowohl Zugriff auf das IT-System als auch auf das Informationssuchsystem. Die Aufzeichnung der Daten geschieht aus Benutzersicht jedoch transparent.

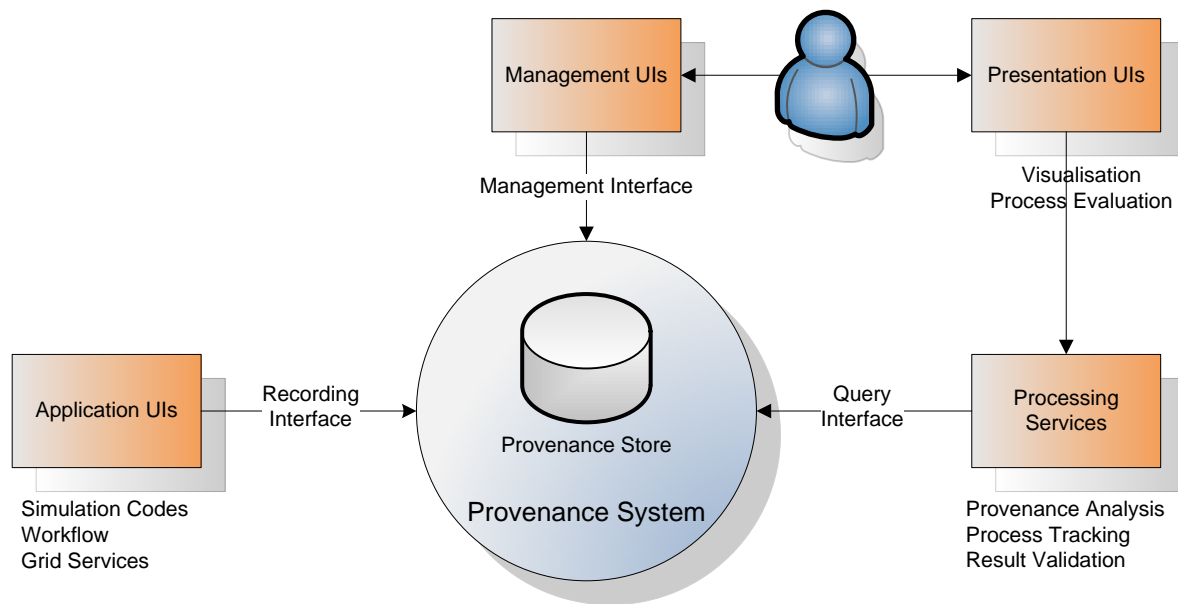


ABBILDUNG 38: SCHNITTSTELLEN DES PROVENIENZSYSTEMS NACH AERO-GRID-SPEZIFIKATION⁵

Die Sicherung der Prozesskette kann innerhalb der aufgezeigten Konzepte erfolgen, deren Systemarchitektur eine Datenbank beinhalten, da sie eine große Datenmenge bewältigen müssen. Eine weitere Forderung ist die integrative Anbindung des Konzeptsystems in die Modellierungsplattform, um zu gewährleisten, dass die Daten mittels Schnittstellen erfasst und aufbereitet werden können.

Das Validierungskonzept könnte hierbei als Zwischeninstanz eingefügt werden. Prozessschritte, die von dem IT-System aufgezeichnet werden, könnten auf Validität überprüft werden und die Prozesskette im Fehlerfall abgebrochen werden. Mit den vorherigen Aufzeichnungen könnte dem Benutzer eine anschauliche Prozesskette mit Fehlerfolge präsentiert werden.

7.1.2 KONFIGURATIONSMANAGEMENT

„Airlines drängen auf sparsamere Triebwerke. Die hohen Klimaziele der Luftlinien setzen die Flugzeugbauer unter Druck: Neue Triebwerke sollen einen deutlich niedrigeren Verbrauch haben, die Kunden erwarten einen Quantensprung. [...]“, betitelt das Handelsblatt [Big10] einen kürzlich erschienenen Artikel in der Kategorie Luftfahrt. Die Branche strebt nach höherer Effizienz im Flugzeugentwurf. Die große Anzahl möglicher Konfigurationen im Vorentwurf erzeugt einen hohen Komplexitätsgrad.

In der Modellierungsphase können verschiedene Konfigurationen untereinander betrachtet werden. Unter anderem differenziert man den Flugzeugtyp, den Triebwerkstyp, die Beschaffenheit der Flugstrecke, die Bauart der Flügel. Je nach Zusammenstellung birgt eine bestimmte Konfiguration Vor- und Nachteile. Das Ziel des Experimentierens ist die Schaffung eines neuen Flugzeugtyps mit gewinnbringenden Marktvorteilen durch Einsparungen von Kraftstoff und Entwicklungskosten sowie der Reduzierung von Fluglärm.

⁵ <http://www.aero-grid.de/projektbeschreibung/provenance> [Abrufdatum: 19.07.2010]

In der Modellierungsplattform sollen, dem Projektplan folgend [Nag09 S. 14], Konfigurationen zur Verfügung stehen, die es den Anwendern erleichtern sollen, neue Zusammenstellungen generieren und berechnen zu können. Im Zusammenhang mit dem Thema der vorliegenden Arbeit könnte man die Komplexität der Validierung auf die Einzelkonfigurationen verlagern. Jede Konfiguration erhält danach beschränkende Parameterbereiche und ein übergreifendes Regelwerk, das Abhängigkeiten zwischen Konfigurationen kontrolliert. Das Netz aus Regeln könnte während des Ladevorganges überprüft werden. Eine übergeordnete Instanz unterscheidet zwischen lokalen und globalen Regeln. Die lokalen Regeln beschränken sich auf die Einzelkonfiguration. Sie könnte bereits beim Laden der Konfiguration ausgeführt werden. Hinsichtlich globaler Regeln kann die Validierung erst erfolgen, wenn alle Abhängigkeitsbedingungen erfüllt sind, das heißt alle von der aktuellen Konfiguration abhängigen Einzelkonfigurationen geladen sind. Eine dem Modellierungsprogramm zugehörige Instanz muss in diesem Fall sowohl die Menge der Konfigurationen als auch die Menge der Bedingungen verwalten und die Validierung initiieren.

7.2 FAZIT

Im Vorfeld dieses Projektes stand die Motivation zur Entwicklung eines virtuellen Modellierungswerkzeugs für Flugzeuge. Die interdisziplinären Projekte TIVA I und II sowie nachfolgend VAMP wurden initiiert, um schrittweise die Zielerfordernungen umzusetzen. Dabei waren Ingenieure aller Bereiche involviert. Die Kommunikation zwischen den Fachdisziplinen fand vierteljährlich in ein- bis zweitägigen Meetings statt.

Die vorliegende Arbeit begann zum Zeitpunkt des Kick-Off-Meetings des Projektes VAMP. Es folgte die Einarbeitungsphase, in der vor allem die Beschaffung von Information im Vordergrund stand. Aufgrund von Vorprojekten existierten bereits eine geringe Anzahl von Dokumentationen über einzelne Werkzeuge, die in dem Modellierungsprogramm ModelCenter[®] eingefügt worden sind. Andere Quellen der Informationsbeschaffung stellten am Projekt beteiligte Mitarbeiter dar. Durch Befragungen wurden Zielerfordernungen gefunden, die die Analyse des Lösungsraumes und die Definition eines Zielkriterienkataloges inklusive Zielinhalt, Zielausmaß sowie Zeitrahmen ermöglichten. Schrittweise kristallisierten sich so mögliche Lösungsalternativen heraus, die mit den beteiligten Personen auf Sinnhaftigkeit sowie mögliche Vor- und Nachteile hin erörtert wurden.

Die Schwierigkeit dieser Arbeit bestand darin, eine einheitliche systemübergreifende Lösung zu finden, die von jeder Fachdisziplin verstanden und angewendet werden kann. Dabei spielte die Systemtiefe eine übergeordnete Rolle, indem sie die Komplexität der Ausgestaltung von Lösungen erhöhte. Die verschiedenen horizontalen und vertikalen Systemebenen boten Angriffspunkte für die automatisierte Prozessanalyse und Fehlerermittlung. Das Spektrum der gefundenen Lösungen zeigt bereits in welchen Metaebenen sich Ansätze entwickeln lassen. So stellt die Einteilung in systemtranszendente, das heißt außerhalb des eigentlichen Systems, und systemimmanente, das heißt innerhalb des zu betrachtenden Systems, agierende Konzepte per se eine Differenzierung von Alternativen dar. Die Konzepte wurden anschließend erörtert und beispielhafte Integrationsmöglichkeiten in das bestehende System aufgezeigt. Die Bewertung der Konzepte erfolgte mittels der Systems Engineering-Methode Nutzwertanalyse, die durch gefundene und gewichtete Ober- und Unterkriterien sowie durch gewählte Erfüllungsgrade den höchstmöglichen Nutzen der dargestellten Alternativen

ermittelte. Eine anschließende Risikobewertung analysierte mögliche Projektrisiken und stellte sie in einem Auftrittswahrscheinlichkeits-Schadensausmaß-Diagramm grafisch dar.

Die Entwicklung von virtuellen Modellierungssystemen ist ein bestehendes Bestreben im Bereich der Forschung und Entwicklung. Die Beherrschung der Komplexität, der Transfer von Wissen hin zu IT-Systemen sowie die Entwicklung von intelligenten Algorithmen in Beratungssystemen werden weiterhin große Thematiken bleiben, die die Menschheit beschäftigen werden. Das Fundament hierfür ist bereits gelegt. Die Expertise in diesen Gebieten kann nur gelingen, wenn die Grundsäulen festen Bestand haben.

ERKLÄRUNG ZUR MASTERARBEIT

Name, Vorname

Hochschule Landshut

Masterstudiengang Systems Engineering

Hiermit erkläre ich, dass ich die Masterarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benützt sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

.....

(Datum)

.....

(Unterschrift)

LITERATURVERZEICHNIS

- [ACA08] ACARE. 2008.** Advisory Council for Aeronautics Research in Europe. *2008 Addendum to the Strategic Research Agenda*. [Online] 2008. [Cited: April 28, 2010.] http://www.acare4europe.com/docs/ACARE_2008_Addendum.pdf.
- [Bac10] Bachmann, Arne, Kunde, Markus and Litz, Markus. 2010.** *ModelCenter Integration Components*. Köln : DLR e.V., 2010.
- [Bie08] Bierle, Christoph and Kern-Isberner, Gabriele. 2008.** *Methoden wissensbasierter Systeme - Grundlagen, Algorithmen, Anwendungen*. 4. Auflage. Wiesbaden : Vieweg+Teubner Verlag, 2008. ISBN.
- [Big10] Bigalke, Silke. 2010.** Airlines drängen auf sparsamere Triebwerke. *Handelsblatt*. Juni 08, 2010.
- [Böh09] Böhnke, Daniel. 2009.** *Data integration in preliminary airplane design*. Köln : DLR, 2009.
- [Bri09] Brinkmann, Prof. Dr. Armin. 2009.** *Systems Engineering Script*. 2009.
- [Dün09] Dünckel, Björn, Hedler, Marko and Pineda, Manuel Montero. 2009.** Über die Eignung von Schema-Sprachen zur Prüfung von XML-Dokumenten. *entwickler magazin*. Juli/August 2009, 4.09, pp. 90-94.
- [Dut06] Dutoit, Allen H., et al. 2006.** *Rationale Management in Software Engineering*. Berlin Heidelberg : Springer-Verlag, 2006. ISBN 978-3-540-30997-0.
- [Hab99] Habermehl, et al. 1999.** *Systems Engineering - Methodik und Praxis*. [ed.] W. F. Daenzer and F. Huber. Zürich : Verlag Industrielle Organisation, 1999. 10. Auflage.
- [Ham94] Hammer, M. and Champy, J. 1994.** *Business Reengineering*. NY : Campus-Verlag, 1994.
- [Han09] Hansch, Daniel, Hans-Peter, Schnurr and Pissierssens, Peter. 2009.** Semantic MediaWiki+ als Wissensplattform für Unternehmen. *WM 2009: 5th Conference on Professional Knowledge Management*. März 25, 2009.
- [HuJ03] Hu, Jingkun. 2003.** *Visual Modeling of XML Constraints Based on A New Extensible Constraint Markup Language*. Pace University : School of Computer Science and Information Systems, 2003.
- [IEEE94] 1994.** IEEE P1220 Standard. *Standard for Application and Management of the Systems Engineering Process*. NY : IEEE Standards Dept., Dezember 26, 1994.
- [ISO14971] ISO. 2009.** DIN EN ISO 14971. *ISO 14971*. Oktober 2009. 10.2009.
- [Jac98] Jackson, Peter. 1998.** *Introduction to Expert Systems*. 3. Edition. England : Pearson Addison-Wesley, 1998. ISBN 0-201-87686-8.

- [Jas10] Jaspers, Wolfgang Prof. Dr. 2010.** business-wissen.de. [Online] Februar 26, 2010. [Cited: Juni 07, 2010.] <http://www.business-wissen.de/organisation/wissensmanagement-faktor-wissen-in-der-heutigen-zeit-immer-wichtiger/>.
- [Kun08] Kunde, Markus. 2008.** *Visualization-Panel for Provenance Data*. [ed.] DLR e.V. Köln : s.n., 2008.
- [Kur92] Kurbel, Karl. 1992.** *Entwicklung und Einsatz von Expertensysteme - Eine anwendungsorientierte Einführung in wissensbasierte Systeme*. 2. Auflage. s.l. : Springer-Verlag, 1992. ISBN 3-540-55237-5.
- [Lee00] Lee, Dongwon and Chu, Wesley W. 2000.** Comparative Analysis of Six XML Schema Languages. *Comparative Analysis of Six XML Schema Languages*. [Online] Juni 2000. [Cited: Juni 21, 2010.] <http://www.cobase.cs.ucla.edu/tech-docs/dongwon/ucla-200008.html>.
- [Mei10] Meixner, Oliver and Haas, Rainer. 2010.** *Wissensmanagement und Entscheidungstheorie*. Österreich : Facultas Verlag, 2010. ISBN 978-3-7089-0587-7.
- [Mos08] Moskaliuk, Johannes. 2008.** *Konstruktion und Kommunikation von Wissen mit Wikis*. [ed.] Johannes Moskaliuk. Boizenburg : Verlag Werner Hülsbusch (VWH), 2008. ISBN 978-3-940317-29-2.
- [Nag09] Nagel, Björn et al. 2009.** VAMP Projektplan. Dezember 14, 2009. Version 0.99.
- [Pho08] Phoenix Integration, INC. 2008.** *ModelCenter 80. Help*. 2008.
- [Rou07] Roumois, Ursula Hasler. 2007.** *Studienbuch Wissensmanagement*. Zürich : Orell Füssli, 2007. ISBN 978-3-8252-2954-2.
- [Sch03] Schöning, Harald. 2003.** *XML und Datenbanken - Konzepte und Systeme*. s.l. : Carl Hanser Verlag München Wien, 2003. ISBN 3-446-22008-9.
- [Sch10] Schulze, Sven-Olaf.** Gesellschaft für Systems Engineering e.V. *Gesellschaft für Systems Engineering e.V.* [Online] [Cited: April 26, 2010.] <http://www.gfse.de/home-mainmenu-1.html>.
- [Tur05] Turban, Efraim, Aronson, Jay E. and Liang, Ting-Peng. 2005.** *Decision Support Systems and Intelligent Systems*. 7th Ed. New Jersey : Prentice-Hall Inc., 2005. 978-81-203-2961-4.
- [Vli01] Vlist, Eric van der. 2001.** XML.com. *Comparing XML Schema Languages*. [Online] Dezember 12, 2001. [Cited: Juni 21, 2010.] <http://www.xml.com/lpt/a/2001/12/12/schemacompare.html>.
- [Win93] Winston, Patrick Henry. 1993.** *Artificial intelligence*. Reading : Addison-Wesley Verlag, 1993.
- [Zan76] Zangemeister, Christof. 1976.** *Nutzwertanalyse in der Systemtechnik*. 4. Auflage. München : Zangemeister & Partner, 1976. ISBN 978-3923264001.

ABKÜRZUNGSVERZEICHNIS

AE	DLR Institut für Aeroelastik
AS	DLR Institut für Aerodynamik und Strömungstechnik
AT	DLR Institut für Antriebstechnik
BK	DLR Institut für Bauweisen und Konstruktionsforschung
FA	DLR Institut für Faserverbundleichtbau und Adaptronik
FT	DLR Institut für Flugsystemtechnik
LK	DLR Lufttransportkonzepte und Technologiebewertung
RM	DLR Institut für Robotik und Mechatronik
SC	DLR Simulations- und Softwaretechnik
ACARE	Advisory Council for Aeronautics Research in Europe
AD	Active Directory; Verzeichnisdienst
ADS	Active Directory Service; siehe AD
BPR	Business Process Reengineering; Geschäftsprozessneugestaltung
CMS	Content Management System; Inhaltsverwaltungssystem
CPACS	Common Parametric Aircraft Configuration Schema; Datensatz
DBMS	Database Management System
DIN	Deutsches Institut für Normung
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DTD	Document Type Definition; Beschreibungsdatei einer XML-Datei
DSD	Document Structure Document; Regelbeschreibung einer XML-Datei
FLOPS	Flight Optimization System, Flugzeugentwurfssystem der NASA
GfSE	Gesellschaft für Systems Engineering (German Chapter of INCOSE)
HTML	Hypertext Markup Language
IEC	Internationale Elektrotechnische Kommission (International Electrotechnical Commission)
IEEE	Institute of Electrical and Electronics Engineers
ISO	Internationale Organisation für Normung (International Organization for Standardization)
IT	Informationstechnik (-technologie)
NASA	Nationale Luft- und Raumfahrtbehörde (National Aeronautics and Space Administration)
PDF	Portable Document Format ((trans)portables Dokumentenformat)
PrADO	Preliminary Aircraft Design and Optimization program, Flugzeugentwurfssystem der TU Braunschweig
RSS	Rich Site Summary in den RSS-Versionen 0.9x RDF Site Summary in den RSS-Versionen 0.9 und 1.0 Really Simple Syndication in RSS 2.0

SOX	Schema for object oriented XML; Schemasprache
TIGL	TIVA Geometric Library
TIVA	Technology Integration for the Virtual Aircraft, DLR Projekt
TIXI	TIVA XML Interface
TU	Technische Universität
UAP	Unterarbeitspaket
VAMP	Virtual Aircraft Multidisciplinary Analysis and Design Processes, DLR Projekt
VPN	Virtual Private Network
WAN	World Area Network
WM	Wissensmanagement
WYSIWYG	"What you see is what you get"; Vereinfachter Editor
XDR	XML Data Reduced; Schemasprache
XML	Extensible Markup Language
XPath	XML Path Language; XML Abfragesprache
XSL	Extensible Stylesheet Language; Transformationsprache für XML
XSLT	XSL Transformation; Programmiersprache zur Transformation von XML

FREIGABEERKLÄRUNG

Name, Vorname.....

Hiermit erkläre ich, dass die vorliegende Masterarbeit in den Bestand der Hochschulbibliothek aufgenommen werden kann und über sie öffentlich zugänglich gemacht werden darf.

.....
(Datum)

.....
(Unterschrift der/des Studierenden)