



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# **Bachelorarbeit**

Bahadir Kocacan

## **Automatisierter Designentwurf hoch detaillierter Flugzeugkabinen durch einen wissensbasierten Ansatz**

*Fakultät Technik und Informatik  
Department Fahrzeugtechnik und Flugzeugbau*

*Faculty of Engineering and Computer Science  
Department of Automotive and  
Aeronautical Engineering*



# **Zusammenfassung**

**Bahadir Kocacan**

## **Thema der Bachelorthesis**

Automatisierter Designentwurf hoch detaillierter Flugzeugkabinen durch einen wissensbasierten Ansatz

## **Stichworte**

Knowledge Based Engineering, Kabinenentwurf, MOKA, CPACS, Wissen

## **Kurzzusammenfassung**

Diese Arbeit umfasst die Entwicklung einer KBE-Anwendung für einen automatisierten Designentwurf von Flugzeugkabinen. Dafür werden zunächst die grundlegenden Begriffe Knowledge Based Engineering, Wissen und Kabinenentwurf erläutert. Anschließend wird die MOKA-Methodik vorgestellt und das informelle und formelle Modell der Methodik aufgebaut. Auf diesen Modellen aufbauend wird die KBE-Anwendung in der Python-Schnittstelle von Blender implementiert. Im Anschluss daran werden die Ergebnisse des Programms präsentiert und mit bestehenden Kabinenlayouts verglichen. Zum Schluss folgt eine Bewertung der KBE-Anwendung und ein Ausblick.

**Bahadir Kocacan**

## **Title of the paper**

Automated design of highly detailed aircraft cabins through a knowledge-based approach

## **Keywords**

Knowledge Based Engineering, cabin design, MOKA, CPACS, Knowledge

## **Abstract**

This work comprises the development of a KBE application for an automated design of aircraft cabins. To this end, the basic terms knowledge-based engineering, knowledge and cabin design are explained first. Then the MOKA methodology is presented and the informal and formal model of the methodology is set up. Building on these models, the KBE application is implemented in the Python interface of Blender. The results of the program are then presented and compared with existing cabin layouts. Finally, there is an evaluation of the KBE application and an outlook.

# Inhaltsverzeichnis

Abbildungsverzeichnis .....	VI
Tabellenverzeichnis .....	VIII
Abkürzungsverzeichnis .....	IX
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Ziel und Aufbau dieser Arbeit.....	2
<b>2 Grundlagen.....</b>	<b>4</b>
2.1 Knowledge Based Engineering .....	4
2.1.1 Definition.....	5
2.1.2 Vor- und Nachteile .....	6
2.1.3 Aufbau von KBE Projekten.....	8
2.2 Wissen .....	10
2.2.1 Definition.....	11
2.2.2 Wissen im KBE .....	11
2.3 Kabinenentwurf.....	13
2.3.1 Richtlinien .....	14
2.3.2 CPACS.....	14
<b>3 MOKA-Methodik .....</b>	<b>15</b>
3.1 Was ist MOKA?.....	15
3.2 Der MOKA Lebenszyklus .....	15
3.2.1 IDENTIFY.....	16
3.2.2 JUSTIFY .....	19
3.2.3 CAPTURE.....	21
3.2.4 Das informelle Modell.....	22
3.2.5 ICARE-Formulare .....	23
3.2.6 FORMALIZE .....	28
3.2.7 PACKAGE.....	31
3.2.8 ACTIVATE .....	32
<b>4 Entwicklung der KBE Anwendung .....</b>	<b>34</b>
4.1 IDENTIFY.....	34
4.2 JUSTIFY .....	35
4.3 CAPTURE.....	35
4.3.1 Entity.....	38
4.3.2 Activity.....	40
4.3.3 Rule.....	43
4.3.4 Constraint.....	43
4.3.5 Illustration.....	44
4.4 FORMALIZE .....	45
4.5 PACKAGE.....	46
4.6 ACTIVATE .....	47

<b>5</b>	<b>Ergebnisse.....</b>	<b>49</b>
<b>6</b>	<b>Vergleich mit bestehenden Kabinenlayouts.....</b>	<b>51</b>
<b>7</b>	<b>Bewertung der Anwendung .....</b>	<b>53</b>
<b>8</b>	<b>Fazit und Ausblick .....</b>	<b>54</b>
	<b>Literaturverzeichnis .....</b>	<b>X</b>
	<b>Anhang .....</b>	<b>XII</b>
A.1	Dateien auf der beigefügten CD.....	XII

# Abbildungsverzeichnis

Abbildung 1.1: Air traffic evolution (International Civil Aviation Organization, 2019)	1
Abbildung 2.1: KBE - ein spezieller Typ eines KBS (Singer, 2014) .....	5
Abbildung 2.2: Vergleich von KBE und traditionellem CAD nach (Stokes, 2001) aus (Singer, 2014).....	8
Abbildung 2.3: Gesamtvorgehensweise im Phasenmodell eines KBE-Projekts (Verein Deutscher Ingenieure, 2017) .....	9
Abbildung 2.4: Wesentliche Rollen eines KBE-Projekts (Verein Deutscher Ingenieure, 2017)	9
Abbildung 2.5: Die Wissenspyramide (Singer, 2014) .....	12
Abbildung 3.1: Der IDENTIFY Schritt nach (Stokes, 2001) .....	17
Abbildung 3.2: Der JUSTIFY Schritt nach (Stokes, 2001) .....	19
Abbildung 3.3: Der CAPTURE Schritt nach (Stokes, 2001).....	21
Abbildung 3.4: Der FORMALIZE Schritt nach (Stokes, 2001) .....	29
Abbildung 3.5: Beziehungen zwischen dem formellen und dem informellen Modell nach (Stokes, 2001) .....	29
Abbildung 3.6: Der PACKAGE Schritt nach (Stokes, 2001) .....	31
Abbildung 3.7: Der ACTIVATE Schritt nach (Stokes, 2001) .....	33
Abbildung 4.1: Das Produktmodell als Klassendiagramm .....	46
Abbildung 4.2: Ordnerstruktur der KBE-Anwendung .....	47
Abbildung 4.3: Die Benutzeroberfläche der KBE-Anwendung.....	48
Abbildung 5.1: Auf CPACS basierender Kabinenentwurf mittels KBE, große Kabine, Ansicht Innen.....	49
Abbildung 5.2: Auf CPACS basierender Kabinenentwurf mittels KBE, große Kabine,, Draufsicht .....	49
Abbildung 5.3: Auf CPACS basierender Kabinenentwurf mittels KBE, kleine Kabine, Ansicht innen.....	50
Abbildung 5.4: Auf CPACS basierender Kabinenentwurf mittels KBE, kleine Kabine, Draufsicht .....	50
Abbildung 6.1: Kabinenlayout der Embraer E-195 der Lufthansa (TripAdvisor LLC, 2020)	51

Abbildung 6.2: Kabinenlayout der A330-300 der Lufthansa (TripAdvisor LLC, 2020)  
52

# Tabellenverzeichnis

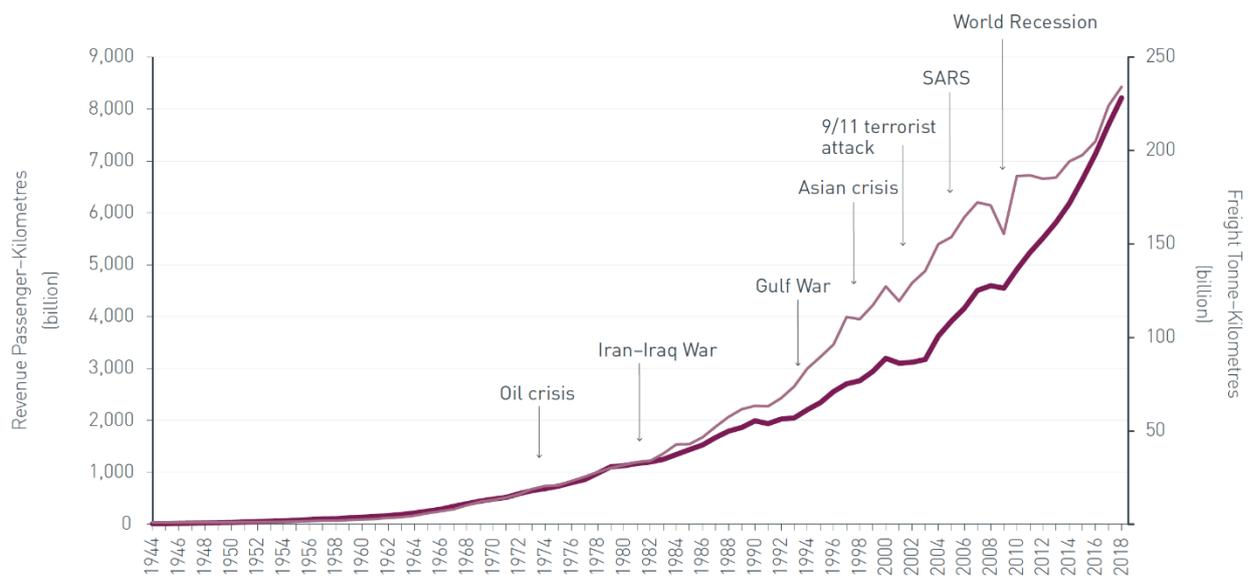
Tabelle 3-1: Entity-Formular nach (Stokes, 2001) .....	24
Tabelle 3-2: Activity-Formular nach (Stokes, 2001).....	25
Tabelle 3-3: Constraint-Formular nach (Stokes, 2001).....	26
Tabelle 3-4: Rule-Formular nach (Stokes, 2001).....	27
Tabelle 3-5: Illustration-Formular nach (Stokes, 2001).....	28
Tabelle 4-1: Anzahl an zulässigen Passagieren pro Notausgangstyp laut CS-25	36
Tabelle 4-2: Mindestanzahl an Türen pro Seite laut CS-25 (Gollnick, 2015) .....	37
Tabelle 4-3: Mindestabstand der Sitze an einem Notausgang laut CS-25 (Gollnick, 2015) 37	
Tabelle 4-4: Entity-Formular der Kabine.....	38
Tabelle 4-5: Entity-Formular des Datensatzes.....	39
Tabelle 4-6: Activity-Formular der Vorbereitung .....	40
Tabelle 4-7: Activity-Formular der Bestimmung der Objektdaten .....	41
Tabelle 4-8: Activity-Formular des Aufbaus der Kabine.....	42
Tabelle 4-9: Rule-Formular der Positionsbestimmung der Economy-Sitze .....	43
Tabelle 4-10: Constraint-Formular der Anzahl an Sitzen pro Sitzmodul .....	44
Tabelle 4-11: Illustration-Formular der Plätze der Flurelemente.....	45

# Abkürzungsverzeichnis

CAD .....	<i>Computer Aided Design</i>
KBE .....	<i>Knowledge Based Engineerinr, Knowledge Based Engineering</i>
KBS .....	<i>Knowledge Based Systems</i>
KI .....	<i>künstliche Intelligenz</i>
MOKA	<i>Methodology and software tools Oriented to Knowledge based engineering Applications</i>

# 1 Einleitung

Die Luftfahrtbranche ist stetig am Wachsen. Immer mehr Menschen wollen fliegen und immer mehr Güter werden über Luftwege transportiert. Im Jahr 2018 wurden beispielsweise laut der International Civil Aviation Organisation weltweit 4,3 Milliarden Passagiere befördert. Abbildung 1.1 zeigt die Entwicklung des Luftverkehrsaufkommens in einem stetigen Wachstum, trotz weltweiter negativer Ereignisse. Auch in den nächsten Jahrzehnten soll der Flugverkehr zunehmen. (International Civil Aviation Organization, 2019)



**8.3 trillion RPK**

↑+7.1% growth rate vs. 2017

**231 billion FTK**

↑+3.6% vs. 2017

Abbildung 1.1: Air traffic evolution (International Civil Aviation Organization, 2019)

Mit der zunehmenden Nachfrage an Flügen steigt der Bedarf an neuen Flugzeugen. Um dieser gerecht zu werden müssen Flugzeuge in höherer Stückzahl konzipiert, entworfen und gebaut werden. Alle drei Phasen nehmen viele Ressourcen in Anspruch. Besonders der Entwurf ist ein sehr zeit- und kostenintensiver Prozess und damit auch eines der größeren Hindernisse und möglichen Probleme der Luftfahrtindustrie im Ausbau der Produktion. Der Entwurf eines Flugzeugs richtet sich größtenteils nach den Bedürfnissen und Wünschen der Fluggesellschaften. Eine enge Zusammenarbeit der Airlines mit den Flugzeugherstellern ist demnach unerlässlich.

Ein für die Fluggesellschaften besonders interessanter Bereich des Entwurfs ist der Kabinenentwurf. Beispielsweise ist es ihnen möglich durch individueller Kabinenlayouts Alleinstellungsmerkmale zu definieren und somit Kunden für sich zu gewinnen oder durch eine enge Bestuhlung den Kabinenraum wirtschaftlich optimal auszunutzen. In welchem Fall auch immer durchlaufen Entwürfe von Kabinen trotz unterschiedlicher

Anforderungen häufig ähnliche Auslegungsprozesse aufgrund geltender Regularien und Zulassungsvorschriften. Dabei haben die Fluggesellschaften verschiedene Parameter, mit denen sie den Entwurf regulieren können. Viele Bestandteile des Entwurfs, wie z.B. die Platzierung von Sitzen, sind repetitiv und bilden somit eine optimale Grundlage für eine Automatisierung der Prozesse. Durch eine solche Automatisierung kann sowohl eine Vielzahl an Kombinationen von Entwurfparametern analysiert, als auch eine größere Spannweite an Parametern betrachtet und die optimale Lösung identifiziert werden. Neben der damit einhergehenden Zeitersparnis sinken auch die Kosten, die beispielsweise durch aufwendige Kabinenmockups verursacht wurden. Arbeitet man zudem noch mit digitalen 3D-Modellen, ergeben sich dadurch weitere Vorteile. Mithilfe einer virtuellen Lösung können auf Kundenwünsche schneller eingegangen und reagiert werden. Außerdem können dadurch Technologien wie z.B. Virtual Reality (VR) in den Entwurfsprozess mit eingebunden werden.

## **1.1 Ziel und Aufbau dieser Arbeit**

Diese Arbeit beschäftigt sich mit der Entwicklung einer Anwendung für einen automatisierten Entwurf von Kabinen. Dabei sollen Entwurfparameter und 3D-Modelle von Einzelteilen eingegeben und hochdetaillierte 3D-Modelle der Kabine ausgegeben werden. Der Entwurf soll durch den wissensbasierten Ansatz Knowledge Based Engineering erfolgen. Als Wissensbasis dienen dafür die bereits erwähnten Richtlinien und Zulassungsvorschriften und ein im März 2020 absolviertes Praktikum beim Deutschen Zentrum für Luft- und Raumfahrt. Das Datenschema CPACS soll dabei die geometrische Randbedingung bilden.

Zunächst werden in Kapitel 2 die Grundlagen behandelt. Es wird erklärt, was Knowledge Based Engineering ist und welche Möglichkeiten dieser wissensbasierte Ansatz bietet. Außerdem soll das Thema Wissen näher betrachtet werden, da es eine zentrale Rolle in dieser Arbeit spielt. Zu guter Letzt wird der Kabinenentwurf näher betrachtet.

Kapitel 3 widmet sich komplett der KBE Methodik MOKA, die in der späteren Entwicklung der Anwendung angewandt wurde. Dieses Kapitel fokussiert sich insbesondere auf den MOKA-Lebenszyklus, dem informellen und dem formellen Modell, die die Kernelemente der Methodik bilden.

In Kapitel 4 ist die Entwicklung der Anwendung beschrieben. Dabei folgt die Entwicklung den einzelnen Schritten des MOKA-Lebenszyklus, weshalb das Kapitel ähnlich wie Kapitel 3 aufgebaut ist.

Die aus dem entwickelten Programm resultierenden 3D-Kabinenmodelle werden in Kapitel 5 dargestellt. Es werden verschiedene Kabinenlayouts bei unterschiedlichen Sitzkonfigurationen und Kabinen-Geometrien gezeigt.

Kapitel 6 vergleicht die Resultate der entwickelten KBE-Anwendung mit bereits bestehenden Kabinenlayouts von bekannten Fluggesellschaften.

In Kapitel 7 wird die KBE-Anwendung kritisch betrachtet und diskutiert. Es werden die Stärken und Schwächen der Anwendung analysiert.

Kapitel 8 soll eine Zusammenfassung dieser Arbeit bieten und einen Ausblick auf zukünftige weiterführende Möglichkeiten zur Erweiterung der Anwendung geben.

## 2 Grundlagen

Im folgenden Kapitel werden die Themengebiete des Knowledge Based Engineerings und des Kabinenentwurfs behandelt und grundlegende Begriffe geklärt, um ein Verständnis der Thematik in dieser Arbeit zu gewährleisten. Dabei wird der Begriff „Wissen“ ausführlicher erklärt, da ein korrektes Verständnis dessen für diese Arbeit von Nöten ist. Außerdem wird der Aufbau von Projekten zur Entwicklung von KBE-Anwendungen näher erläutert, um eine Herangehensweise an das eigene Projekt auszuarbeiten.

### 2.1 Knowledge Based Engineering

Knowledge Based Engineering (KBE), zu Deutsch wissensbasierte Konstruktion, hat seine Ursprünge im Themengebiet der künstlichen Intelligenz (KI), einem Zweig der Informatik, der sich mit der Simulation des intelligenten Verhaltens von Menschen befasst, insbesondere der Fähigkeit zu lernen und Probleme zu lösen. Einer der größten Erfolge im Gebiet der computergestützten Problemlösung sind Knowledge Based Systems (KBS). (La Rocca, 2011)

Um zu verstehen, was KBE ist, ist das Verständnis von KBS essentiell. KBS, zu Deutsch wissensbasierte Systeme, sind problemspezifische Anwendungen, die darauf ausgelegt sind, Wissen zu speichern und dieses zur Lösung eines Problems zu nutzen. Das Wissen kann dabei aus unterschiedlichen Quellen stammen und ist im Gegensatz zu konventioneller Software unabhängig von der Verarbeitung in einer Datenbank hinterlegt. Das bringt verschiedene Vorteile mit sich. Zum einen beeinflussen sich Veränderungen in Prozessen oder Wissen gegenseitig nicht und zum anderen sind sie getrennt voneinander erweiterbar. (Tripathi, 2011)

Des Weiteren weisen KBS nach (Tripathi, 2011) folgende Charakteristiken auf:

- KBS können performant schwierige Probleme genauso gut oder sogar schneller als Experten lösen
- KBS können über eine große Menge an (detailreichem) Wissen verfügen
- KBS wenden Heuristiken an, um Argumentationen zu leiten und so den Suchbereich für eine Lösung einzugrenzen
- KBS verwenden Symbole, um verschiedene Arten von Wissen darzustellen

Mithilfe von KBS soll es Anwendern ermöglicht werden, Probleme größtenteils ohne die Notwendigkeit von Fachpersonal zu lösen.

KBE wiederum ist ein Bereich der KBS und nutzt ebenfalls Wissen, allerdings nicht zu Lösen eines spezifischen Problems, sondern, wie der Name schon sagt, um zu

konstruieren (La Rocca, 2011). Abbildung 1.1 veranschaulicht die drei Themengebiete KI, KBS und KBE noch einmal.

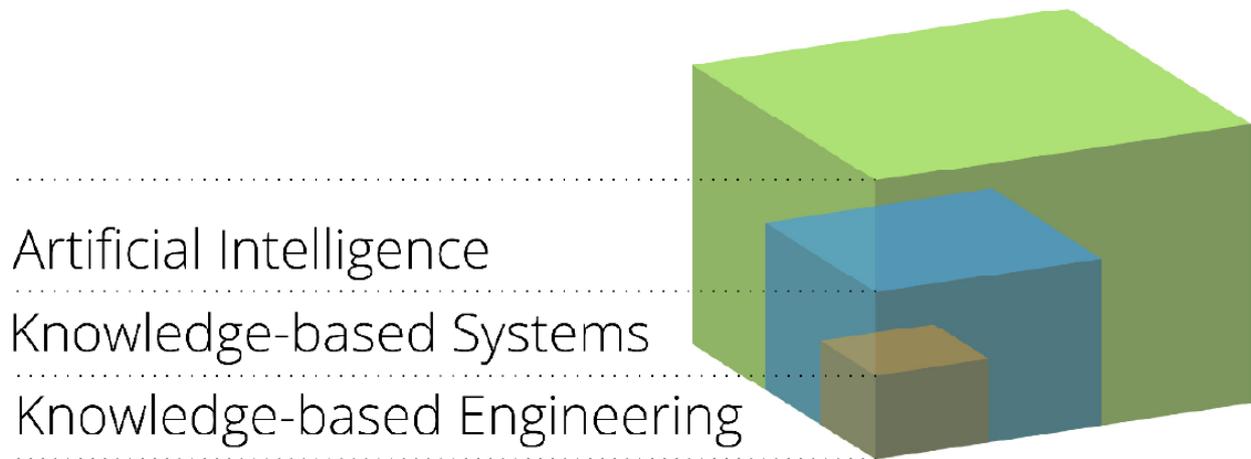


Abbildung 2.1: KBE - ein spezieller Typ eines KBS (Singer, 2014)

### 2.1.1 Definition

Auf die Frage, was KBE ist, gibt verschiedene Definitionsansätze, die vieles gemein haben, sich aber inhaltlich oder im Umfang geringfügig unterscheiden. Beispielsweise definiert Stokes KBE folgendermaßen:

*„The use of advanced software techniques to capture and re-use product and process knowledge in an integrated way.“ (Stokes, 2001)*

Er beschreibt KBE als den Gebrauch von Softwaretechniken zur Speicherung und Wiederverwendung von Produkt- und Prozesswissen. Eine weitere Definition macht La Rocca:

*„Knowledge based engineering (KBE) is a technology based on dedicated software tools called KBE systems, that are able to capture and reuse product and process engineering knowledge. The main objective of KBE is the reduction of time and cost of product development [...]“ (La Rocca, 2011)*

Im Gegensatz zu Stokes beschreibt La Rocca KBE als Technologie basierend auf Softwaretools, die ebenfalls in der Lage sind Produkt- und Prozesswissen zu speichern und wiederzuverwenden. Zudem soll das Hauptziel von KBE die Kosten- und Zeitreduktion in der Produktentwicklung sein. Eine Definition von Chapman will ich noch anführen:

*„KBE is an engineering method that represents a merging of object oriented programming (OOP), Artificial Intelligence (AI) techniques and computer-aided design technologies, giving benefit to customised or variant design automation solutions. The KBE systems aim to capture product and process information [...] to model engineering design*

*processes, and then use the model to automate all or part of the process.” (Chapman, et al., 1999)*

Von Chapman wird KBE als Konstruktionsmethode aufgefasst, die objektorientierte Programmierung, künstliche Intelligenz und CAD Technologien vereint um Automatisierungslösungen zu nutzen. Weiterführend sollen KBE Systeme darauf abzielen, Produkt- und Prozessinformationen zu speichern um Designprozesse zu modellieren und anschließend zu automatisieren.

Während sich die Definitionen von KBE unterscheiden, sei es als Gebrauch, Technologie oder Methode, weisen sie trotzdem in folgenden Begrifflichkeiten Gemeinsamkeiten auf:

- **Speicherung und Wiederverwendung von Wissen:** Die KBE-Anwendung muss Wissen in einer Form speichern können, die es ermöglicht, dieses Wissen wiederzuverwenden.
- **Produkt- und Prozesswissen:** Das Wissen, welches die KBE-Anwendung beinhaltet, ist Wissen über das zu entwickelnde Produkt und dessen Entwicklungsprozesse.
- **Produktentwicklung:** KBE-Anwendungen haben das Ziel, ein Produkt zu entwickeln.
- **Automatisierung:** Die Entwicklung von Produkten erfolgt automatisch mit minimalem Eingriff der Nutzer.

Aus diesen Begriffen lässt sich zusammenfassen, dass sich KBE im Themengebiet der Produktentwicklung einordnet lässt. Es ist ein wissensbasiertes System und soll bestehendes Wissen über Produkte und Prozesse speichern und wiederverwenden und die Entwicklung automatisieren. Aus diesen Gemeinsamkeiten kann KBE samt Eigenschaften und Ziele wie folgt definiert werden:

*„Knowledge Based Engineering ist ein wissensbasiertes System zur automatisierten Produktentwicklung durch Erfassung und Wiederverwendung von Produkt- und Prozesswissen.“*

## 2.1.2 Vor- und Nachteile

Der Gebrauch von KBE bringt viele Möglichkeiten, aber auch Risiken mit sich. Zudem ist es nicht in jeder Situation einsetzbar. Laut Stokes sollte KBE beispielsweise nicht genutzt werden, wenn:

- der Entwurfsprozess nicht klar definiert werden kann,
- die Technologien im Entwurf sich stetig ändern,

- der Entwurf in einem simplen Programm modellierbar ist,
- das Wissen für die geplante Anwendung nicht verfügbar ist, oder
- die nötigen Ressourcen ein KBE System einzuführen nicht vorhanden sind.

Sind die Voraussetzungen erfüllt birgt die Einführung von KBE immer noch Risiken, da mit der Einführung auch Veränderungen im Arbeitsablauf und der Organisation innerhalb des Unternehmens verbunden ist. Ob diese Veränderungen gut oder schlecht sind hängt von dem Umgang des Unternehmens mit diesen ab. (Stokes, 2001)

Ist die Situation den Voraussetzungen entsprechend und die Veränderungen im Unternehmen verkraftbar verspricht KBE enorme Zeit- und Kostenersparnisse. Beispielsweise konnte Jaguar durch KBE die Entwurfszeit einer inneren Motorhaube von 8 Wochen auf 20 Minuten senken. Ein weiteres Beispiel wäre die Zeitersparnis von British Aerospace im Neuentwurf der Wing-box in 10 Stunden mit KBE, statt 8000 Stunden ohne. (Stokes, 2001)

Was KBE von traditionellen CAD Anwendungen unterscheidet ist die Arbeit an Produkten auf unterschiedlichen Ebenen. Während CAD Anwendungen an einzelnen Produkten arbeiten und Geometrien generieren sind KBE Anwendungen in der Lage ganze Produktfamilien zu betrachten und sich sowohl mit Geometrien als auch mit Wissen zu befassen. Die enorme Zeitreduktion ermöglicht mehr Raum für kreative Arbeiten und bietet zudem die Möglichkeit aus dem Konventionellen herauszutreten und trotzdem noch einen Zeitgewinn zu haben. (Stokes, 2001) Abbildung 2.2 verdeutlicht dies.

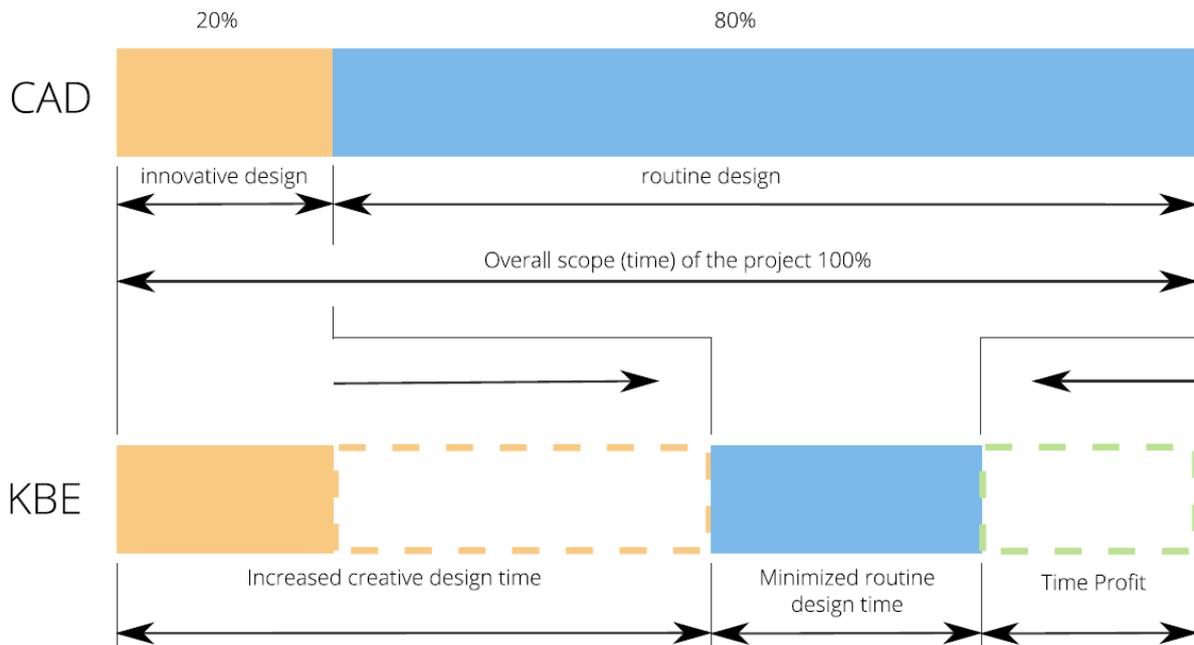


Abbildung 2.2: Vergleich von KBE und traditionellem CAD nach (Stokes, 2001) aus (Singer, 2014)

Der wissensbasierte Ansatz von KBE hat einen weiteren Vorteil. Die Speicherung von Wissen macht den Anwender unabhängig von Fachpersonal. Außerdem kann altes Wissen archiviert und für zukünftige Projekte, wenn die entsprechenden Experten nicht mehr vorhanden sind, genutzt werden.

### 2.1.3 Aufbau von KBE Projekten

Die Entwicklung von KBE Anwendungen erfolgt zumeist einem ähnlichen Muster. Der Verein Deutscher Ingenieure unterteilt KBE-Projekte in vier Phasen: Planung, Entwicklung, Test und Betrieb (vgl. Abbildung 2.3).

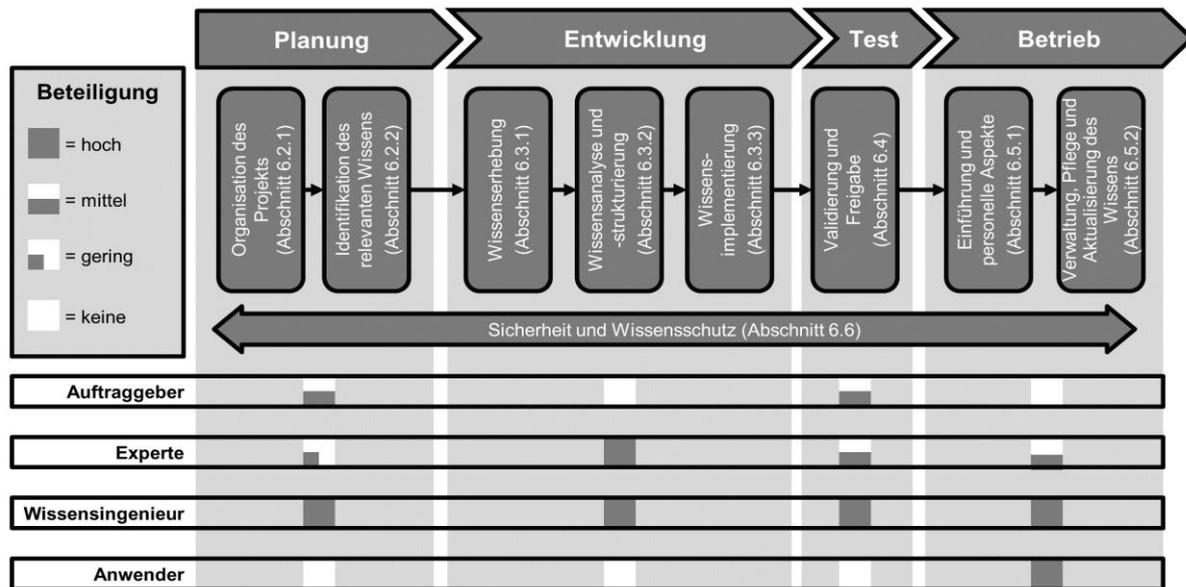


Abbildung 2.3: Gesamtvorgehensweise im Phasenmodell eines KBE-Projekts (Verein Deutscher Ingenieure, 2017)

Personen, die an dem Projekt beteiligt sind werden in die vier Rollen Auftraggeber, Experte, Wissensingenieur und Anwender unterteilt. Durch das Rollenmodell werden Aufgaben Rollen statt Personen zugewiesen. Das hat den Vorteil, dass Personen mehrere Rollen besetzen können und somit KBE-spezifische Aufgaben flexibler zugewiesen werden können. (Verein Deutscher Ingenieure, 2017) Einen Überblick über die Rollen ist in Abbildung 2.4 veranschaulicht.

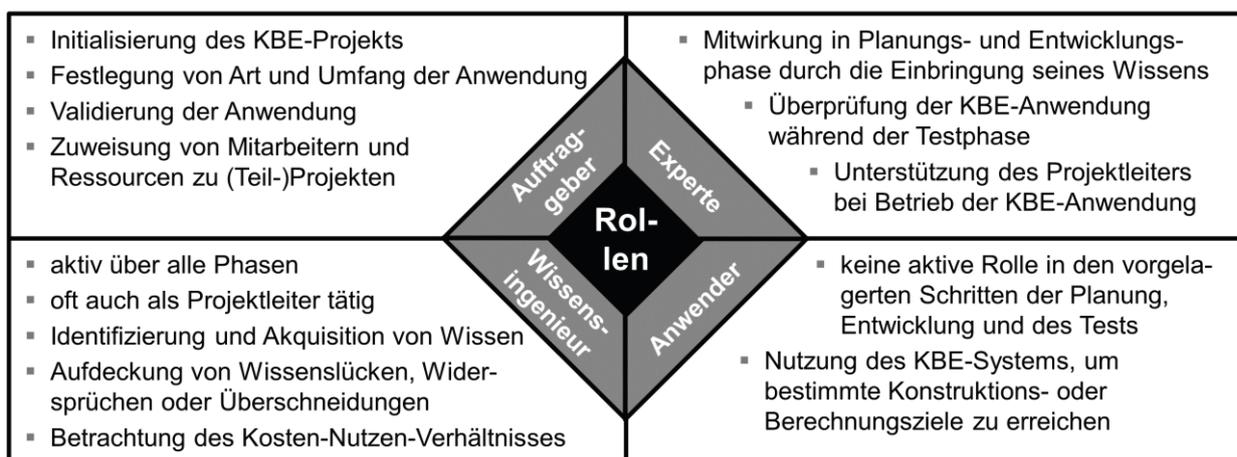


Abbildung 2.4: Wesentliche Rollen eines KBE-Projekts (Verein Deutscher Ingenieure, 2017)

Die erste Phase eines Projekts ist die Planungsphase. Sie ist die Vorphase, in der größtenteils projekt- und unternehmensseitige Organisationen stattfinden und beinhaltet vor allem die Beschaffung von Ressourcen. Die Identifikation von relevantem Wissen ist ebenfalls ein wichtiger Bestandteil dieser Phase, die sich in zwei Teile gliedern lässt.

Der erste ist die Problem- und Lösungsidentifikation. Dabei wird zunächst das anzugehende Problem und die betroffenen Prozesse detailliert beschrieben. Zudem wird der Wissensbestand innerhalb der Prozesse analysiert. Basierend darauf werden Lösungen identifiziert. Im zweiten Teil findet eine Untersuchung der Machbarkeit statt, in dem das Vorhaben wirtschaftlich, technisch und projektspezifisch analysiert wird. (Verein Deutscher Ingenieure, 2017)

Ist die Planungsphase abgeschlossen geht das Projekt in die Entwicklungsphase über. Die Entwicklungsphase ist die Hauptphase eines KBE-Projekts und Schwerpunkt vieler KBE-Methoden. Die Phase teilt sich in die Abschnitte Wissenserhebung, Wissensanalyse und -strukturierung und Wissensimplementierung auf. Bei der Wissenserhebung wird Wissen aus verschiedenen Wissensquellen gesammelt. Dabei werden verschiedene Erhebungstechniken wie z.B. Interviews, Textanalysen und Beobachtungstechniken eingesetzt. Im nächsten Abschnitt wird das gesammelte Wissen analysiert und strukturiert. „Dieser Schritt enthält Maßgaben zur Interpretation des Wissens. Es werden Strukturen erarbeitet und Grundlagen für [...] Schlussfolgerungsmechanismen gelegt.“ (Verein Deutscher Ingenieure, 2017) Hierfür werden Vorgehensweisen der KBE-Methoden verwendet. Ist das Wissen strukturiert und formalisiert erfolgt die Wissensimplementierung. In diesem Abschnitt der Entwicklungsphase findet „die Realisierung der Aufgabenstellung in Form einer KBE-Anwendung“ (Verein Deutscher Ingenieure, 2017) statt, sprich, in diesem Abschnitt wird das formalisierte Wissen digitalisiert und die KBE-Anwendung programmiert. (Verein Deutscher Ingenieure, 2017)

Die Testphase folgt auf die Programmierung in der Entwicklungsphase. Sie ist, wie in jedem Softwareprojekt, essentiell. Sie ist dazu da Fehler jeglicher Art offenzulegen und somit Folgeschäden durch fehlerhafte Konstruktionen zu vermeiden. Allerdings werden durch die Testphase gefundene Fehler nicht behoben, weshalb die Testphase in eine Iterationsschleife mit der Entwicklungsphase übergeht. (Verein Deutscher Ingenieure, 2017)

Sind die Test abgeschlossen und die KBE-Anwendung validiert geht das Projekt in die letzte Phase, die Betriebsphase, über. Die Anwendung wird im Unternehmen eingeführt und das Personal für den Einsatz geschult. In dieser Phase ist es besonders wichtig, Rückmeldungen der Anwender einzuholen, um mögliche Fehler, die im Einsatz der Anwendung auftreten, zu beseitigen. Zudem kann dadurch auf Vorschläge und Wünsche der Anwender eingegangen werden. Ein weiterer wichtiger Teil, den die Betriebsphase ausmacht, ist die Verwaltung, Pflege und Aktualisierung des Wissens. (Verein Deutscher Ingenieure, 2017)

## **2.2 Wissen**

Wissen ist das Zentrum einer jeden wissensbasierten Ansatzes und somit auch von Knowledge Based Engineering. Die Handhabung von Wissen ist unerlässlich in der Entwicklung von wissensbasierten Systemen.

## 2.2.1 Definition

Wissen kann nicht eindeutig definiert werden. Die Debatte über die Definition von Wissen reicht laut Reinmann-Rothmeier bis in die Zeit von Aristoteles (384 – 322 v.Chr.) und Platon (428 – 347 v.Chr.) zurück. (Reinmann-Rothmeier, et al., 2000) Jedoch kann, je nachdem, in welchem Kontext der Begriff „Wissen“ genutzt wird, er unterschiedlich interpretiert werden. Daher ist auch eine starre Definition nicht möglich. Dennoch soll im Folgenden versucht werden Wissen möglichst umfangreich zu definieren.

Das Wort „wissen“ stammt von dem altdeutschen Wort „wissan“, welches so viel bedeutet wie „ich habe gesehen“. Dies lässt zum einen den Schluss zu, dass Wissen auf Erkenntnisse und Erfahrungen basiert. ( (Cenker, 2012) zitiert nach (Singer, 2014)) Außerdem sind mit der Wissens- bzw. der Erkenntnisgewinnung Sinneswahrnehmungen verbunden. Das Bibliographische Institut definiert Wissen im Duden wie folgt:

*„Gesamtheit der Kenntnisse, die jemand [auf einem bestimmten Gebiet] hat“ (Bibliographisches Institut GmbH)*

Aus dieser Definition lassen sich ebenfalls einige Aussagen treffen. Wissen ist eine Ansammlung von Kenntnissen. Das heißt dem Wissen ist eine Menge an Kenntnissen zuzuordnen. Wie viele Kenntnisse dafür benötigt werden, lässt sich jedoch nicht eindeutig sagen, weil dies von Wissen zu Wissen variiert.

Eine weitere wichtige Aussage, die getroffen werden kann ist, dass die Generierung von Wissen durch den menschlichen Verstand erfolgt. Gemeinsam mit der Erkenntnis, dass Wissen auf Erfahrungen basiert, lässt sich außerdem schließen, dass Wissen personengebunden ist.

Zusammenfassend lässt sich sagen, dass Wissen eine Ansammlung von Erfahrungen und Erkenntnissen, die jeweils durch Sinneswahrnehmungen aufgenommen werden, ist. Die Generierung des Wissens basierend auf diesen Erkenntnissen erfolgt alleine durch den menschlichen Verstand und ist damit gleichzeitig personengebunden. Zudem lässt sich mit keiner Definition eindeutig unterscheiden, was Wissen ist und was nicht.

## 2.2.2 Wissen im KBE

In den Informations- und Kommunikationswissenschaften wird Wissen an der Spitze der sogenannten Wissenspyramide angeordnet. Dabei baut Wissen auf Informationen auf, für die wiederum Daten als Fundament dienen. (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**)

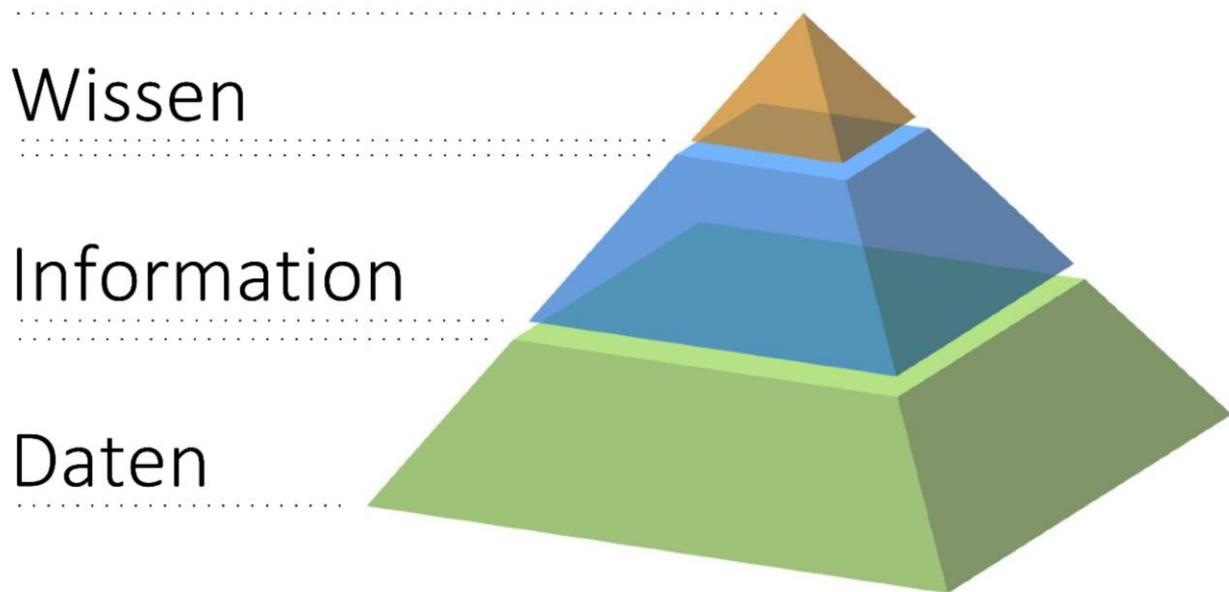


Abbildung 2.5: Die Wissenspyramide (Singer, 2014)

(Verein Deutscher Ingenieure, 2017) definiert die drei Begriffe wie folgt:

- **Daten** sind „objektive Fakten, die ohne Zusammenhang und weitere Hintergründe nicht deutbar und als „Rohmaterial“ zu verstehen sind“ (Verein Deutscher Ingenieure, 2017)
- **Informationen** sind „strukturierte Daten mit Relevanz und Zweck, die semantisch in einen Kontext gebracht, kategorisiert, kalkuliert und korrigiert werden können“ (Verein Deutscher Ingenieure, 2017)
- **Wissen** ist „vernetzte Information, die in die Lage versetzt, Vergleiche anzustellen, Verknüpfungen herzustellen und Entscheidungen zu treffen“ (Verein Deutscher Ingenieure, 2017)

Beispiele für Daten sind der Wert „30“, die Dimension „mm“ und die Würfelkantenlänge „l“, die in keinem Zusammenhang stehen. Informationen ergeben sich aus der Verknüpfung dieser Daten in der Form „Eine Würfel besitzt die Kantenlänge  $l=30\text{mm}$ “. Wissen wäre die Vernetzung dieser Information mit anderen Informationen wie bspw. die Formel zur Berechnung des Würfelvolumens:

$$V_{\text{Würfel}} = l^3$$

Neben der Wissenspyramide existieren noch weitere Möglichkeiten Wissen zu strukturieren. Bspw. kann Wissen laut dem Verein Deutscher Ingenieure folgendermaßen unterteilt werden:

- **Wissenstyp:** fachlicher/thematischer Bereich, den das Wissen abbildet. *Bsp.: Fachwissen, Produktwissen, Methodenwissen, usw.*
- **Wissensart:** charakteristische Eigenschaften, die das Wissen beschreiben. *Bsp.: implizit, explizit, prozedural, usw.*
- **Wissensform:** Vorliegeform des Wissens. *Bsp.: Text, Abbildung, Formel, usw.*
- **Wissensort:** Ort, an dem das Wissen auftritt. *Bsp.: Person, Datenbank, Abteilung, usw.* (Verein Deutscher Ingenieure, 2017)

Eine für KBE besonders interessante Klassifizierung ist die Unterscheidung zwischen den Wissensarten implizit und explizit.

Implizites Wissen ist Wissen, welches weder verbal noch schriftlich formulierbar ist. Es entsteht durch Erfahrungen und ist an Personen gebunden. Ein Beispiel für impliziertes Wissen wäre die Fähigkeit zu schwimmen. Die Implementierung von implizitem Wissen in wissensbasierte Systeme erweist sich dementsprechend auch als schwierig. (Singer, 2014)

Explizites Wissen ist Wissen, welches sich sowohl in Sprache als auch in Schrift formulieren lässt. Es lässt sich also leicht kommunizieren und ist daher leicht in Software zu implementieren. (Singer, 2014)

Eine weitere Unterscheidung die wichtig für KBE ist, ist die in prozedurales und deklaratives Wissen. Dabei geht es speziell um Wissen, welches in der Softwareentwicklung eingesetzt wird bzw. im Programm vorhanden ist.

Prozedurales Wissen ist Wissen, welches aus dem Programmcode nicht direkt ersichtlich ist und sich aus dem Verhalten des Programms ergibt. „Das Wissen ist „versteckt“ und damit nur schwer analysier- und kommunizierbar.“ (Singer, 2014)

Deklaratives Wissen hingegen erschließt sich direkt aus dem Programmcode und kann direkt herausgelesen werden. Diese Art von Wissen ist leicht analysier- und kommunizierbar. (Singer, 2014)

Wissen ist also ein komplexer Sachverhalt ist. Es kann sowohl klar ersichtlich und formulierbar als auch „versteckt“ und schwer analysier- und kommunizierbar sein. Die Implementierung von Wissen in KBE Software erweist sich dadurch als eine Herausforderung. KBE-Methodiken bieten dafür jedoch hilfreiche Werkzeuge.

## 2.3 Kabinentwurf

Der Kabinentwurf ist eine Teildisziplin des Flugzeugentwurfs. Er läuft parallel mit den Entwürfen der anderen Disziplinen, wodurch sich grundsätzlich zwei Herangehensweisen ergeben. Ist die Rumpfgeometrie durch einen Vorentwurf bereits gegeben, spricht man vom „outside-in“ Verfahren. Dabei wird die Kabine den

geometrischen Randbedingungen entsprechend entworfen. Kommt der Kabinenentwurf jedoch an erster Stelle, spricht man vom „inside-out“ Verfahren. Dabei wird zuerst die Kabine der Mission entsprechend entworfen und die Entwürfe anderer Disziplinen schließen daran an. Beim Entwurf einer Kabine können verschiedene Parameter, wie z.B. Anzahl der Sitze, Gangbreite, Deckenhöhe usw. herangezogen werden. Trotz der sich daraus ergebenden Anzahl an Möglichkeiten kann eine Kabine nicht beliebig entworfen werden. In dieser Arbeit wird das „outside-in“ Verfahren aufgrund von gegebener Kabinengeometrie in CPACS-Datensätzen (vgl. Kapitel 2.3.2) verfolgt.

### 2.3.1 Richtlinien

Die Gestaltung einer Kabine ist nicht direkt und in allen Aspekten an gesetzliche Richtlinien gebunden. Die einzelnen Fluggesellschaften können ihre Kabinen ihren Missionszielen entsprechend auslegen. Dabei verfolgen sie zumeist verschiedene Ansätze. Ein möglicher Ansatz ist die wirtschaftlich optimale Nutzung der Kabine. Dieser berücksichtigt eine möglichst hohe Passagieranzahl durch niedrige Sitzabstände und ermöglicht somit niedrige Ticketpreise. Ein anderer Ansatz ist die Erhöhung von Komfort und Kundenzufriedenheit durch höhere Sitzabstände und breitere Sitze. Das jedoch bringt höhere Ticketpreise mit sich. (Gobbin, 2015)

Neben den internen Richtlinien der Fluggesellschaften müssen die Kabinenentwürfe auch gesetzliche Richtlinien einhalten. Diese und weitere Richtlinien sind in der Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25, kurz CS-25, beschrieben. Eine der wichtigsten für den Kabinenentwurf relevanten Richtlinien sind die Vorschriften für die Notausgänge. Diese sollen gewährleisten, dass das Flugzeug im Falle eines Notfalls innerhalb von 90 Sekunden durch die Nutzung der Ausgänge auf nur einer Seite des Rumpfs evakuiert werden kann. Die CS-25 bildet somit eine Wissensgrundlage für die zu entwickelnde KBE-Anwendung.

### 2.3.2 CPACS

CPACS steht für „Common Parametric Aircraft Configuration Schema“ und ist ein auf XML basierendes Dateiformat für den Vorentwurf von Lufttransportsystemen, entwickelt vom Deutschen Zentrum für Luft- und Raumfahrt. Es soll allen Entwurfsdisziplinen eine einzige gemeinsame Plattform bieten und Ingenieuren ermöglichen, Daten und Informationen in einem einheitlichen Format zu speichern und auszutauschen. Dadurch soll die Grundlage eines multidisziplinären Flugzeugentwurfs geschaffen werden. (German Aerospace Center)

Neben Daten über Flügel, Fahrwerk, Rumpf, Triebwerke und vielem mehr beinhaltet CPACS auch Daten über die Kabine und deren Geometrie. Diese Daten sollen in der KBE-Anwendung und in ihrer Entwicklung mit einbezogen werden.

## 3 MOKA-Methodik

Die in dieser Arbeit entwickelte KBE-Anwendung basiert auf der MOKA-Methodik. Daher soll in diesem Kapitel diese Methodik vorgestellt und ausführlich behandelt werden. Es wird zunächst erklärt, was MOKA ist, wie es entstanden ist und aus was es besteht. Anschließend werden dessen Lebenszyklus und die darin enthaltenen Schritte näher erläutert. Ein besonderes Augenmerk wird dabei auf die Schritte CAPTURE und FORMALIZE gelegt.

### 3.1 Was ist MOKA?

MOKA steht für „Methodology and software tools Oriented to Knowledge based engineering Applications“ und war ein von der Europäischen Kommission finanziertes Forschungsprojekt. Das Projekt startete 1998 und dauerte 30 Monate. Partner des Projektes waren hauptsächlich die Automotiv- und Luftfahrtindustrie sowie die IT-Branche und die Coventry University KEM Centre. Ziel des Projekts war die Entwicklung einer Methodik zur kosten- und zeiteffizienten Entwicklung und Wartung von KBE-Anwendungen. Des Weiteren sollte MOKA einen konsistenten Weg zu deren Entwicklung und Wartung bieten. Außerdem sollte die Methodik als Basis für einen internationalen Standard dienen. (Stokes, 2001)

Die Kernelemente von MOKA sind folgende:

- **Lebenszyklus:** MOKA beschreibt einen Lebenszyklus, der die Methodik repräsentiert und den Anwendern die Nutzung der Konzepte erleichtert. Dieser teilt sich in unterschiedliche Phasen auf, die in Kapitel 3.2 näher erläutert werden. (Stokes, 2001)
- **Wissensrepräsentation:** Ein wesentlicher Punkt des Projekts war es, einen geeigneten Weg zu finden, Produkt- und Prozesswissen einheitlich zu repräsentieren. MOKA teilt die Wissensrepräsentation in zwei Ebenen: der informellen und der formellen Ebene. (Stokes, 2001)
- **Softwaretool:** Ein weiterer Bestandteil des Projekts war die Entwicklung eines Softwaretools, welches den Anwendern die Implementierung der Methodik ermöglicht. Jedoch sind diese Softwaretools in den Quellen, auf die verwiesen werden, nicht mehr aufzufinden. (Stokes, 2001)

### 3.2 Der MOKA Lebenszyklus

Der Lebenszyklus einer KBE-Anwendung ist im Prinzip dem einer jeden Softwareentwicklung ähnlich. Die MOKA Methodik beschreibt die Details der einzelnen Schritte des Lebenszyklus und zielt darauf ab einen Rahmen für effektives Management

bei der Entwicklung von KBE-Anwendungen zu bieten. Der Lebenszyklus ist in sechs Schritten beschrieben:

- **IDENTIFY:** In diesem Schritt werden Ziele und Umfang der KBE-Anwendung klar festgelegt und die Anwendung wird auf Konzeptebene spezifiziert.
- **JUSTIFY:** Dieser Schritt führt zu einem Projektplan, der einen Business Case enthält. Zudem wird in diesem Schritt das Projekt durch das Management genehmigt und die nächsten Schritte des Lebenszyklus werden ermöglicht.
- **CAPTURE:** Dieser Schritt umfasst das Sammeln von Rohwissen und dessen Strukturierung in ein informelles Modell.
- **FORMALIZE:** Dieser Schritt übersetzt das informelle Modell in eine formelle Version, welches ein Produkt- und Prozessmodell beinhaltet.
- **PACKAGE:** In diesem Schritt wird das informelle Modell in Code überführt und die KBE-Anwendung wird programmiert.
- **ACTIVATE:** Dieser Schritt umfasst die Verteilung, Einführung und Verwendung der KBE-Anwendung.

Der Lebenszyklus ist sehr iterativ und modular. Obwohl sie nur der Vollständigkeit halber in den Lebenszyklus mit einbezogen wurden sind die Schritte IDENTIFY und ACTIVATE für den Erfolg eines KBE-Projekts von großer Bedeutung. Der Fokus von MOKA jedoch richtet sich auf die Schritte CAPTURE und FORMALIZE. (Stokes, 2001)

### 3.2.1 IDENTIFY

Der MOKA-Lebenszyklus kann erst beginnen, wenn die entsprechenden Geschäftsmöglichkeiten dazu ermittelt wurden. Eine weitere Voraussetzung für den Start des Projekts ist die Verfügbarkeit von Ressourcen. Diese und weitere Punkte werden im IDENTIFY Schritt behandelt. In Abbildung 3.1 ist der IDENTIFY Schritt dargestellt. (Stokes, 2001)

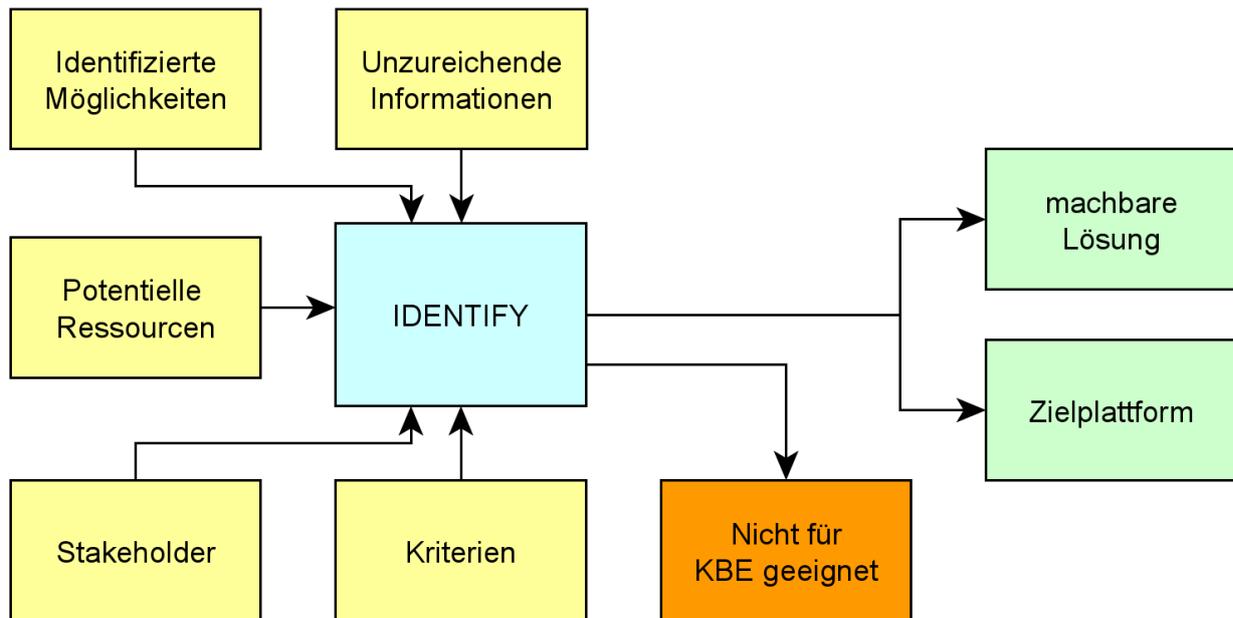


Abbildung 3.1: Der IDENTIFY Schritt nach (Stokes, 2001)

Ziel dieses Schrittes ist es:

- Klare Geschäftsziele für die mögliche KBE-Anwendung anzugeben,
- Meinungen darüber einzuholen, was die KBE-Anwendung tun muss oder könnte,
- Den Umfang der Anwendung zu definieren,
- Zu überprüfen, ob Wissen vorhanden ist und wie es erfasst werden kann,
- Zu überprüfen, welche Software und Hardware für die mögliche KBE-Anwendung vorhanden ist,
- Zu bestimmen, ob KBE das richtige Werkzeug ist und
- Eine technische Spezifikation auf Konzeptebene bereitzustellen.

Um diese Ziele zu erreichen teilt sich der IDENTIFY Schritt in sechs Unterschritte auf:

1. **Motivation und Anforderungen klären:** Im ersten Schritt werden die Bedürfnisse und Wünsche aller Parteien, die Interesse an dem Projekt pflegen, ermittelt. Dafür werden zunächst die Möglichkeiten des Unternehmens geklärt und mögliche Stakeholder werden identifiziert. Danach werden die Erwartungen, Bedürfnisse und Wünsche bestimmt und aus denen Ziele und Randbedingungen formuliert. Anschließend wird geprüft, ob Motivation und Ziele klar definiert wurden. Der erste Unterschritt ist in der Hinsicht wichtig, dass er ein Verständnis über die unternehmensseitige Bedeutung von KBE verschafft und einen Überblick bietet. Das Ergebnis dieses Schrittes sind klaren Aussagen über

Möglichkeiten und Ziele, sowie schon bekannte Randbedingungen (wie z.B. Zeit, Geld oder Personal). (Stokes, 2001)

2. **Umfang der möglichen Anwendung klären:** Dieser Schritt definiert den Umfang der KBE-Anwendung und präzisiert ihren Rahmen. Es wird geklärt, welche Teile oder wieviel des Produktentwurfs von der KBE-Anwendung übernommen werden soll. Um dies zu ermöglichen wird daher zunächst der bestehende Prozess untersucht. Danach wird ein zu erwartender verbesserter Prozess betrachtet. Darauf folgt die genaue Definition des Anwendungsumfangs. In diese Definition fließt zudem die Rolle der KBE-Anwendung im Gesamtprozess mit ein. Es ist wichtig zu klären, welche Aufgaben sie genau übernehmen und wie sie mit dem Anwender interagieren soll. Außerdem beinhaltet dieser Unterschnitt die Beurteilung, ob der Entwurfsprozess für eine KBE-Anwendung geeignet ist bzw. ob KBE das richtige Mittel ist. (Stokes, 2001)
3. **Wissensquellen identifizieren:** Dieser und der nächste Unterschnitt dient der Vorbereitung für den CAPTURE Schritt später im MOKA-Lebenszyklus. In diesem Schritt werden mögliche Wissensquellen innerhalb des vorher durch den Umfang definierten Bereichs untersucht. Ein Teil des Schrittes ist es, die Existenz von bereits verfügbaren Materialien festzustellen. Ein anderer Teil besteht darin, die Eigenschaften des Wissens zu untersuchen. Dazu werden zunächst Wissensquellen ermittelt und untersucht, worauf dann eine Charakterisierung derer, bspw. in menschlich, digital usw., folgt. Zuletzt wird beurteilt, ob das verfügbare Wissen für den Einsatz im KBE geeignet ist. (Stokes, 2001)
4. **Mittel zur Wissenserfassung identifizieren:** Dieser Schritt beschäftigt sich mit der Erhebung und Verarbeitung von Wissen aus den zuvor ermittelten Quellen. Für beide Kategorien existieren viele Methoden. Beispielsweise gibt es für die Wissenserhebung die Methoden Interview, Beobachtungen oder Protokollanalysen. Für die Wissensverarbeitung sind multidimensionale Techniken oder Zielzerlegungen Beispiele. In diesem Schritt sollen Ermittlungstechniken und Analysetechniken identifiziert und gesammelt werden. Außerdem ist zu prüfen, ob die nötigen Ressourcen zur Nutzung dieser Techniken vorhanden sind. (Stokes, 2001)
5. **Zielplattformen identifizieren:** Dieser Schritt ist entscheidend für die Schritte PACKAGE und ACTIVATE in der späten Phase des Lebenszyklus. Da MOKA unabhängig von jeglicher Plattform ist muss in diesem Schritt eine Plattform gewählt werden, in der KBE später zur Anwendung kommt. Zu beachten dabei sind die allgemeine Architektur des zukünftigen KBE-Tools und die mit einzubindenden Systemen und die dazugehörigen Schnittstellen. Die Benutzeroberfläche der Anwendung darf auch nicht außer Acht gelassen werden. (Stokes, 2001)

**6. Beurteilung der technischen Machbarkeit:** Der IDENTIFY Schritt wird mit diesem Unterschritt abgeschlossen. Alle zuvor gemachten technischen Überlegungen und Entscheidungen werden zusammengetragen und auf Konzeptebene spezifiziert. Gemeinsam mit allen Beteiligten wird bewertet, ob das skizzierte KBE-System den Zielen und Anforderungen entspricht und die Konzepte technisch machbar sind. Wenn eine grundlegende Übereinstimmung erreicht ist, geht der Prozess in den nächsten Schritt JUSTIFY des Lebenszyklus über. (Stokes, 2001)

### 3.2.2 JUSTIFY

Der Schritt JUSTIFY befasst sich mit den Aspekten der Risikobewertung, der Schätzung von Zeit und Kosten des Projekts sowie der Bestimmung einer Methode zur Beurteilung von Erfolg oder Misserfolg des Projekts. Er umfasst die Erstellung eines globalen Projektplans gemeinsam mit einem Business Case, um anschließend das Projekt genehmigen zu lassen. In Abbildung 3.2 ist der JUSTIFY Schritt veranschaulicht. (Stokes, 2001)

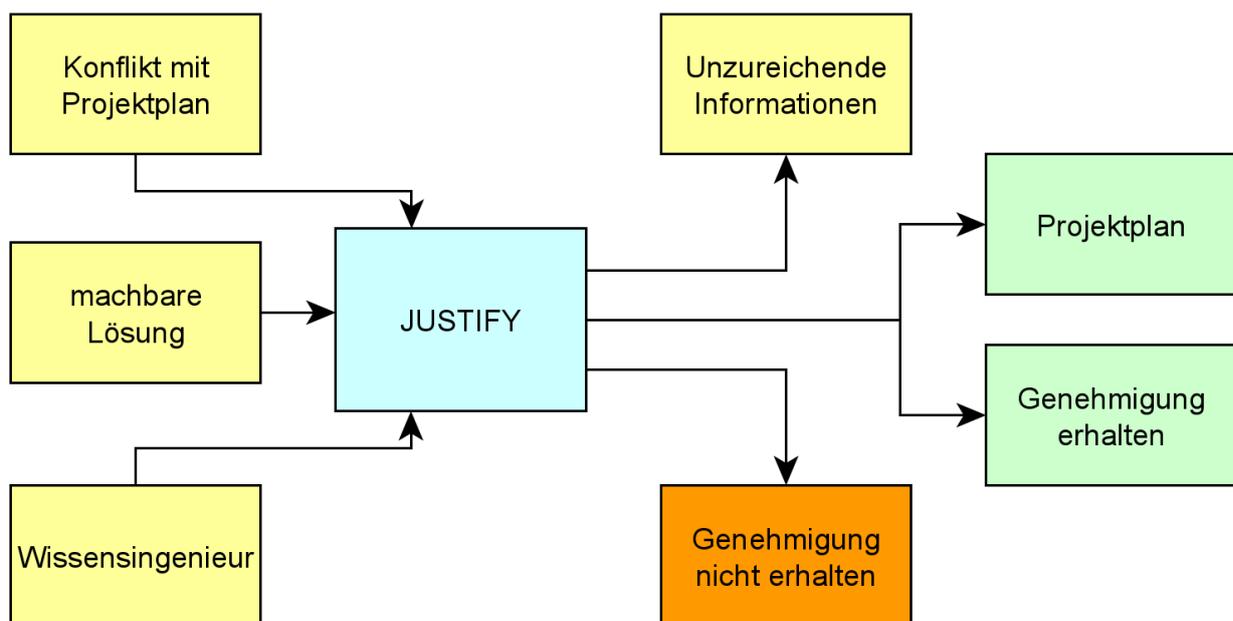


Abbildung 3.2: Der JUSTIFY Schritt nach (Stokes, 2001)

Die Hauptziele von JUSTIFY sind:

- Schätzung der benötigten Ressourcen und der damit verbundenen zeitlichen und kostentechnischen Auswirkungen,
- Bewertung der assoziierten technischen und kommerziellen Risiken,
- Definition der Kriterien für die Beurteilung des Erfolgs oder Misserfolgs der Anwendung,

- Erstellen eines Projektplans,
- Erstellen eines Business Cases,
- Genehmigung des Managements für das Projekt einholen

Der JUSTIFY Schritt teilt sich in folgende Unterschritte auf:

1. **Schätzen des Ressourcenbedarfs und der Kosten:** Die konzeptionellen Spezifikationen aus dem Schritt IDENTIFY werden in diesem Schritt verwendet, um die diskreten Arbeitsmodule für das Projekt zu bestimmen. Für jedes Modul werden die Ressourcen geschätzt, die in Bezug auf Personal, Software, Hardware usw. erforderlich sind. Mithilfe der Dauer des Einsatzes jeder Ressource werden die Kosten aller Module geschätzt. (Stokes, 2001)
2. **Bewertung technischer und kommerzieller Risiken:** Teile der Risiken wurden bereits im IDENTIFY Schritt untersucht. Dabei wurden Themen wie die Eignung von KBE im Unternehmen, die Gewinnung von Wissen und die Verfügbarkeit von Ressourcen behandelt. Dieser Schritt jedoch beschäftigt sich mit der Risikobewertung in organisatorischen, kommerziellen und wirtschaftlichen Bereichen. Dabei werden die organisatorischen Auswirkungen der KBE-Anwendung und die kommerziellen Chancen und Risiken bewertet. Außerdem wird ihr wirtschaftlicher Einfluss beurteilt. Schließlich wird eine Risikobewertung erstellt, die alle Aspekte miteinbezieht. (Stokes, 2001)
3. **Akzeptanzkriterien definieren:** In diesem Schritt wird beurteilt, ob das KBE-Projekt die erwarteten Geschäftsvorteile bietet. Dafür werden zunächst unter Berücksichtigung der vereinbarten Ziele und Spezifikationen Überprüfungsmöglichkeiten, die implementiert werden können, identifiziert. Anschließend werden Methoden, mit denen Erfolg oder Misserfolg beurteilt werden können, untersucht. Die Ausgabe dieses Schritts ist ein Bericht über Akzeptanzkriterien, der die Anforderungen für die Testphase des Projekts enthält. (Stokes, 2001)
4. **Projektplan erstellen:** In diesem Schritt werden die Kernelemente der Risikobewertung und die Meilensteine der Akzeptanzkriterien zusammengetragen und ein Projektplan wird erstellt. (Stokes, 2001)
5. **Business Case erstellen:** In diesem Schritt wird ein Business Case formuliert, der alle bisher behandelten Schritte miteinbezieht. Die Ziele, Spezifikationen, Risiken, Kosten und der Projektplan fließen in den Business Case, welcher wiederum für den Rest des Projekts als Leitfaden dienen kann. (Stokes, 2001)
6. **Genehmigung erhalten:** Dies ist die letzte Phase der JUSTIFY Schritts und ein Meilenstein im KBE-Projekt. Alle gesammelten Informationen werden allen Parteien präsentiert, um sicherzustellen, dass das Projekt eine gemeinsame Sichtweise darstellt. Es kann einige Iterationen um die Schritte in JUSTIFY oder

sogar zurück zu IDENTIFY geben, bis die Projektübersicht akzeptabel ist. Ist dieser Punkt erreicht, gibt das Unternehmen das Projekt frei und der MOKA-Lebenszyklus kann in die nächste Phase CAPTURE übergehen. (Stokes, 2001)

### 3.2.3 CAPTURE

Der CAPTURE Schritt ist der erste Schritt, in dem Wissen für KBE-Plattformen geeignet gemacht wird. Die Wissensquellen, -tools und Erhebungstechniken wurden schon in den früheren Schritten des Lebenszyklus identifiziert. In diesem Schritt werden diese genutzt, um Rohwissen zu sammeln und diese in ein einheitliches Format, dem informellen Modell, zu transformieren. Abbildung 3.3 bietet einen Überblick über CAPTURE. (Stokes, 2001)

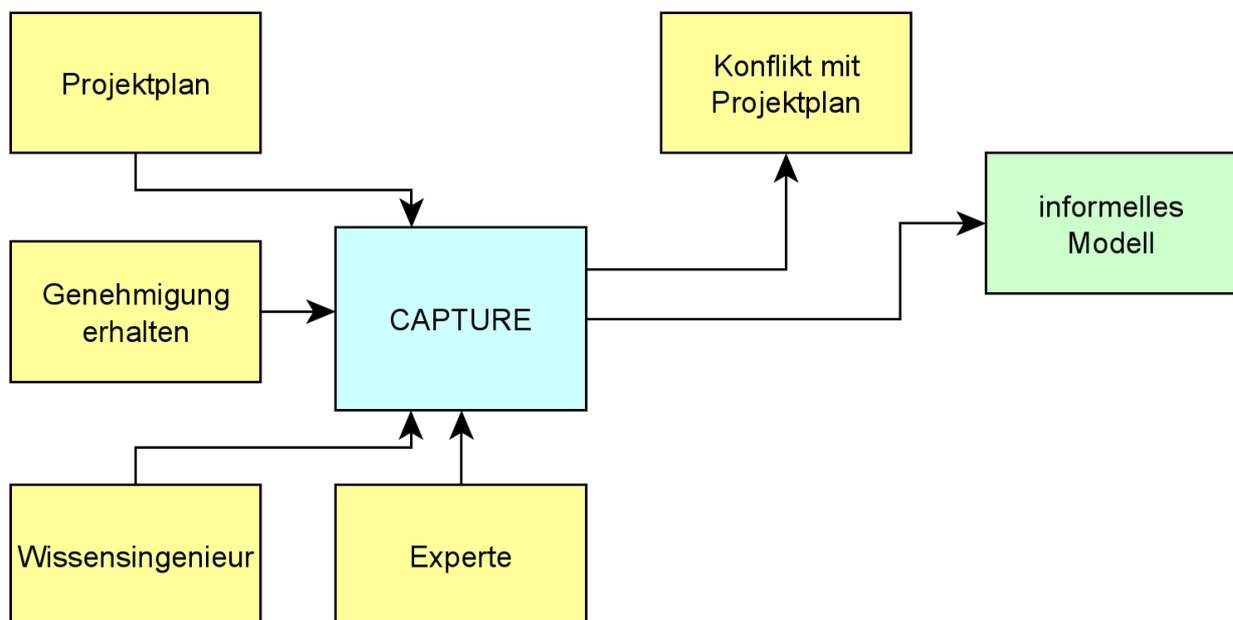


Abbildung 3.3: Der CAPTURE Schritt nach (Stokes, 2001)

Der CAPTURE-Schritt erfolgt in vier Unterschritten:

1. **Sammlung von Wissen vorbereiten:** Das Ziel dieses Schrittes ist es, alle Wissensquellen erneut zu betrachten und sie für die Sammlung des Wissens vorzubereiten. Während des IDENTIFY Schrittes wurden potenzielle Quellen identifiziert, bei denen es sich um Personen, Text, Datenbanken etc. handeln kann. Die Aufgabe besteht nun darin, sicherzustellen, dass diese noch relevant und verfügbar sind. Dasselbe wird mit den Methoden zur Wissenserhebung gemacht und geprüft, ob sie noch geeignet sind oder durch andere Techniken ersetzt werden müssen. Zudem wird in diesem Schritt der Lager- und Speicherort für das gesammelte Wissen vorbereitet. Das Ergebnis dieses Schrittes ist ein Plan zur Wissenssammlung. (Stokes, 2001)

2. **Rohwissen sammeln:** Ziel dieses Unterschritts ist es, alle für das informelle Modell erforderlichen Rohkenntnisse zu sammeln. Die verwendeten Methoden hängen dabei stark von der Art der Quelle ab – Experten, Dokumente oder Repositorien. Der Schritt ist Teil eines hochgradig iterativen Prozesses, bei dem sowohl beim Erstellen des informellen Modells gelegentlich zurück auf die Quellen als auch in den Schritten FORMALIZE und PACKAGE auf CAPTURE zurückgegriffen werden muss. (Stokes, 2001)
3. **Rohwissen strukturieren:** In diesem Schritt wird das gesammelte Wissen organisiert und strukturiert. Zunächst wird aus der Sammlung brauchbares Wissen identifiziert und fehlerhaftes oder fehlendes Wissen ersetzt bzw. ergänzt. Ist die Menge an Wissen komplett werden die einzelnen Wissenseinheiten klassifiziert. Diese Klassifizierung dient als Vorbereitung für das informelle Modell mittels der ICARE-Forms. Diese werden in den nachfolgenden Kapiteln näher behandelt. Das Ergebnis dieses Schritts ist das informelle Modell. (Stokes, 2001)
4. **Tauglichkeit überprüfen:** Dieser Schritt dient als Kontrolle und Korrektur des informellen Modells. Es werden der Wissensumfang und die Auflösung des Wissens geprüft, redundantes Wissen wird identifiziert und entfernt. Des Weiteren wird das Wissen auf seine Richtigkeit geprüft und ggf. korrigiert. (Stokes, 2001)

### 3.2.4 Das informelle Modell

Das informelle Modell ist eine Sammlung von Diagrammen, Terminologien und Wissenshandbüchern, die genutzt werden können, aber nicht müssen. Das Kernelement des informellen Modells bilden nämlich die ICARE-Formulare und sind ein fester Bestandteil dessen. ICARE ist ein Akronym gebildet aus den Worten Illustration, Constraint, Activity, Rule, und Entity, die das Wissen, welches in den Formularen enthalten ist, beschreiben. Diese werden in Kapitel 3.2.5 jedoch näher behandelt. (Stokes, 2001)

Es gibt drei Arten von Diagrammen, die in das informelle Modell mit einbezogen werden können: Hierarchiediagramme, Prozessdiagramme und benutzerdefinierte Diagramme. Hierarchiediagramme dienen zur Darstellung von Produktstrukturen und -funktionen und Entwurfsschritten. Prozessdiagramme hingegen stellen Prozesse in ihren einzelnen Schritten dar und bilden somit Programmabläufe und Aktivitäten ab. Letztere der drei Diagrammtypen kann genutzt werden, um Strukturen abzubilden, die mithilfe der anderen nicht dargestellt werden können. In MOKA bspw. sind Verlinkungen zwischen einzelnen ICARE-Formularen durch eigene Diagramme darstellbar. (Stokes, 2001)

Die Terminologie im informellen Modell ist ein Wörterlexikon, welches Begrifflichkeiten enthält und erklärt, die zum Verständnis des Modells und der Domain notwendig sind. Besonders bei sprachübergreifenden und unterschiedlichen Wissensquellen ist es wichtig, Einheitlichkeit in der Nutzung von Fachbegriffen zu gewährleisten, da diese

sonst Unklarheiten und Verwirrung verursachen können. Auch für eine zukünftige Erweiterung des Programms ist es empfehlenswert, eine entsprechende Terminologie anzulegen. (Stokes, 2001)

Ein Wissenshandbuch ist eine Zusammenfassung des informellen Modells. Es beinhaltet alle ICARE-Formulare, Diagramme und die Terminologie und beschreibt die einzelnen Schritte der Entstehung des Modells. Es dient als Nachschlagewerk für spätere Schritte des MOKA-Lebenszyklus und ist eine Dokumentation der Wissenserhebung in MOKA. Auch in das informelle Modell nicht miteinbezogenes Wissen kann in das Wissenshandbuch aufgenommen werden. Dies kann für eine spätere Wiederbetrachtung des Wissens sinnvoll sein. (Stokes, 2001)

### 3.2.5 ICARE-Formulare

Die ICARE-Formulare stellen ein Gerüst dar, in welchem Wissen für den Benutzer noch verständlich und in einer standardisierten Form repräsentiert wird. Die Formulare sind der Hauptbestandteil des informellen Modells und teilen sich in fünf Kategorien auf:

- **Illustrations:** Beschreibung von Fallstudien oder relevanten Beispielen
- **Constraints:** Einschränkungen für Entitäten
- **Activities:** Beschreibung der Elemente des Entwurfsprozesses
- **Rules:** Mittel zur Regulierung der Aktivitäten und zur Bereitstellung des „Know-hows“ oder der Strategie des Entwurfsprozesses
- **Entitis:** Objekte, die das Produkt beschreiben

Jedes dieser Kategorien hat sein eigenes Formular, die im Allgemeinen jedoch ähnlich aufgebaut sind. Das Grundgerüst bilden die Pflichtfelder bestehend aus dem Namen, einer universellen Referenz, und der Quelle des Wissens. Für die Organisation der Formulare werden in jedem Formular zudem noch Autor, Datum, Versionsnummer und Status eines Formulars angegeben. Im kommenden sollen die fünf Kategorien näher behandelt und deren Zwecke erläutert werden. (Stokes, 2001)

#### Entity

Entity-Formulare beschreiben die Bestandteile des Endprodukts. Diese Bestandteile können sowohl reale Objekte als auch Konzepte sein. Entity-Formulare können in drei Bereichen eingesetzt werden:

- **Strukturen:** Beschreibung der Produktfamilie bezüglich Baugruppen, Einzelteilen usw.
- **Funktionen:** Untersuchung der Funktionalität eines Produkts und mögliche Aufzeichnung möglicher Designlösungen

- **Verhalten:** Aufzeichnung, wie ein Entwurf eine Funktion erfüllt und wie sich das Verhalten unter verschiedenen Umständen ändert

Alle ICARE-Formulare werden tabellarisch dargestellt. In Tabelle 3-1 ist das Entity-Formular dargestellt. (Stokes, 2001)

Tabelle 3-1: Entity-Formular nach (Stokes, 2001)

MOKA ICARE FORMS:		ENTITY
Name		
Reference		
Entity Type		
Functions		
Behaviour		
Context, information, validity		
Description		
Related Activities		
Related Entities	Parent	
	Child	
	Undefined	
Related Constraints		
Related Rules		
Related Illustrations		
Information Origin		
Management	Author	
	Date	
	Version Number	
	Status	

Das Ausfüllen des Entity-Formulars erfolgt in drei Schritten. Zunächst werden die Pflichtfelder *Name*, *Reference*, *Information Origin* und die *Management* Felder ausgefüllt. Dabei sollte darauf geachtet werden, dass Name und Referenz sinnvoll und verständlich gewählt werden. Im nächsten Schritt kommen die Felder *Description* und nach Bedarf *Functions* und *Behaviour* dazu. Zuletzt werden die restlichen Felder ausgefüllt und damit Bezug auf andere ICARE-Formulare genommen. Durch den letzten Schritt werden die Entity-Formulare mit anderen Formularen verlinkt. (Stokes, 2001)

Will man mit dem Entity-Formular Strukturen beschreiben, so sollten Hauptattribute und -eigenschaften, Verbindungen zu anderen Elementen, Geometrien, Materialien, Montage- und Fertigungsdetails in der Beschreibung enthalten sein. Entity-Formulare hingegen, die Funktionen beschrieben, sollten Grundfunktionen, Lösungsprinzipien oder auch technische Lösungen beschreiben. Dabei kann jede Lösung in Unterfunktionen zerlegt werden. Um Verhalten mithilfe von Entity-Formularen beschreiben zu können

sollten diese eine dementsprechende Beschreibung, sowie die Bedingungen, bei denen das Verhalten auftritt und dessen Auswirkungen auf den Entwurf enthalten. (Stokes, 2001)

### Activity

Activity-Formulare beschreiben Entwurfsprozesse der KBE Anwendung. Sie können dazu genutzt werden, Programmabläufe durch eine Kette von Prozessen, generelle Prozesse in jeder Ebene oder in Bibliotheken gespeicherte Prozesspakete zu beschreiben. Eine Unterteilung in verschiedene Activity-Typen gibt es im informellen Modell nicht. Alle Aspekte eines Prozesses sind mit ein und demselben Formular beschreibbar. Das Activity-Formular ist in Tabelle 3-2 dargestellt. (Stokes, 2001)

Tabelle 3-2: Activity-Formular nach (Stokes, 2001)

MOKA ICARE FORMS:		ACTIVITY
Name		
Reference		
Trigger		
Input		
Output		
Potential failure modes		
Objective		
Input requirements		
Context, information, validity		
Description		
Related Activities	Parent activity	
	Sub-activities	
	Precending activities	
	Following activities	
Rules involved		
Entities involved		
Related Illustrations		
Information Origin		
Management	Author	
	Date	
	Version Number	
	Status	

Das Ausfüllen des Activity-Formulars erfolgt ebenfalls in 3 Ebenen. Zunächst werden die Pflichtfelder *Name*, *Reference*, *Information Origin* und *Management* ausgefüllt. Die zweite Ebene besteht aus den Feldern *Trigger*, *Input*, *Output*, *Potential Failure Modes*, *Objective*, *Input Requirements*, *Description* und *Context, Information, Valitation*. Auf

dieser Ebene werden Prozesse samt eingehende und ausgehende Größen, Auslöser, Voraussetzungen, Ziele und potentieller Fehler beschrieben. Auf der dritten Ebene kommen die Verlinkungen mit anderen ICARE-Formularen zustande. (Stokes, 2001)

### Constraint

Constraint-Formulare werden zur Erfassung von Limitierungen und Randbedingungen von Entitäten genutzt. Sie beschreiben Begrenzungen von Größen und Attributen von Entitäten und können sich dabei sowohl lokal auf Einzelteile als auch global auf ganze Baugruppen beziehen. In Tabelle 3-3 ist das Constraint-Formular abgebildet. (Stokes, 2001)

Tabelle 3-3: Constraint-Formular nach (Stokes, 2001)

MOKA ICARE FORMS:		CONSTRIANTS
Name		
Reference		
Objectives		
Context, information, validity		
Description		
Linked Rules		
Related Entities		
Related Illustrations		
Information Origin		
Management	Author	
	Date	
	Version Number	
	Status	

Das Ausfüllen des Constraint-Formulars erfolgt wie in den beiden vorangegangenen Formularen in drei Schritten. Den ersten Schritt bilden wieder die Pflichtfelder *Name*, *Reference*, *Information Origin* und *Management*. Schritt zwei beinhaltet das Füllen der Felder *Context*, *Information*, *Validity*, *Objectives* und *Description*. Neben der Beschreibung der Randbedingungen wird mittels des zweiten Schritts das Ziel und der Zweck sowie die Relevanz der Limitierung erfasst. In Schritt drei erfolgt die Verlinkung des Formulars mit anderen ICARE-Formularen. (Stokes, 2001)

### Rule

Rule-Formulare sind eng mit den Activity-Formularen verbunden. Sie beschreiben Regeln und können dabei unterschiedlichen Zwecken nachgehen, wie z.B. die Generierung von Ausgaben, die Prüfung von Eingaben oder Entscheidungsmechanismen. In Tabelle 3-4 ist das Rule-Formular dargestellt. (Stokes, 2001)

Tabelle 3-4: Rule-Formular nach (Stokes, 2001)

MOKA ICARE FORMS:		RULE
Name		
Reference		
Objective		
Context, information, validity		
Description		
Related Activities		
Linked Rules		
Linked Constraints		
Related Entities		
Related Illustrations		
Information Origin		
Management	Author	
	Date	
	Version Number	
	Status	

Wie in den restlichen Formularen auch füllt man das Rule-Formular in drei Schritten aus, beginnend bei den Pflichtfeldern *Name*, *Reference*, *Information Origin* und *Management*. In Schritt zwei werden die Felder *Context*, *Information*, *Validity*, *Objectives* und *Description* befüllt. Gemeinsam mit dem Zweck wird in diesem Schritt die Regel beschrieben. Dabei kann die Beschreibung in unterschiedliche Formen, wie z.B. Formeln oder wenn-dann-Abfragen auftreten. Zuletzt folgt die Verlinkung des Rule-Formulars mit anderen ICARE-Formularen in Schritt drei. (Stokes, 2001)

### Illustration

Das letzte Element der ICARE-Formulare ist das Illustration-Formular. Sie werden nicht in die KBE-Anwendung eingebunden, tragen aber zum Verständnis des in den restlichen Formularen enthaltenen Wissens bei. Die Nutzung der Illustration-Formulare ist in keiner Weise begrenzt und kann beliebig vom Nutzer eingesetzt werden. Beispiele für den Einsatz des Formulars wären Statistiken, Abbildungen oder Fallstudien. Das Formular ist in Tabelle 3-5 dargestellt. (Stokes, 2001)

Tabelle 3-5: Illustration-Formular nach (Stokes, 2001)

MOKA ICARE FORMS:		ILLUSTRATION
Name		
Reference		
Context, information, validity		
Description		
Related Entities		
Related Activities		
Related Rules		
Information Origin		
Management	Author	
	Date	
	Version Number	
	Status	

Nach dem Ausfüllen von *Name*, *Reference*, *Information Origin* und *Management* können im zweiten Schritt mithilfe der Felder *Context*, *Information*, *validity* und *Description* beliebige Sachverhalte dargestellt werden. Der Bezug zu anderen ICARE-Formularen wird im dritten Schritt mittels der Verlinkungen hergestellt.

### 3.2.6 FORMALIZE

Der FORMALIZE Schritt baut auf dem im CAPTURE Schritt erstellten informellen Modell und deren ICARE-Formularen auf. Während im informellen Modell das Wissen noch ausführlich und leserlich formuliert ist, soll in diesem Schritt das Wissen in einer präzisen Form ohne die Notwendigkeit von Programmierkenntnissen repräsentiert werden. Der FORMALIZE Schritt ist in Abbildung 3.4 abgebildet. (Stokes, 2001)

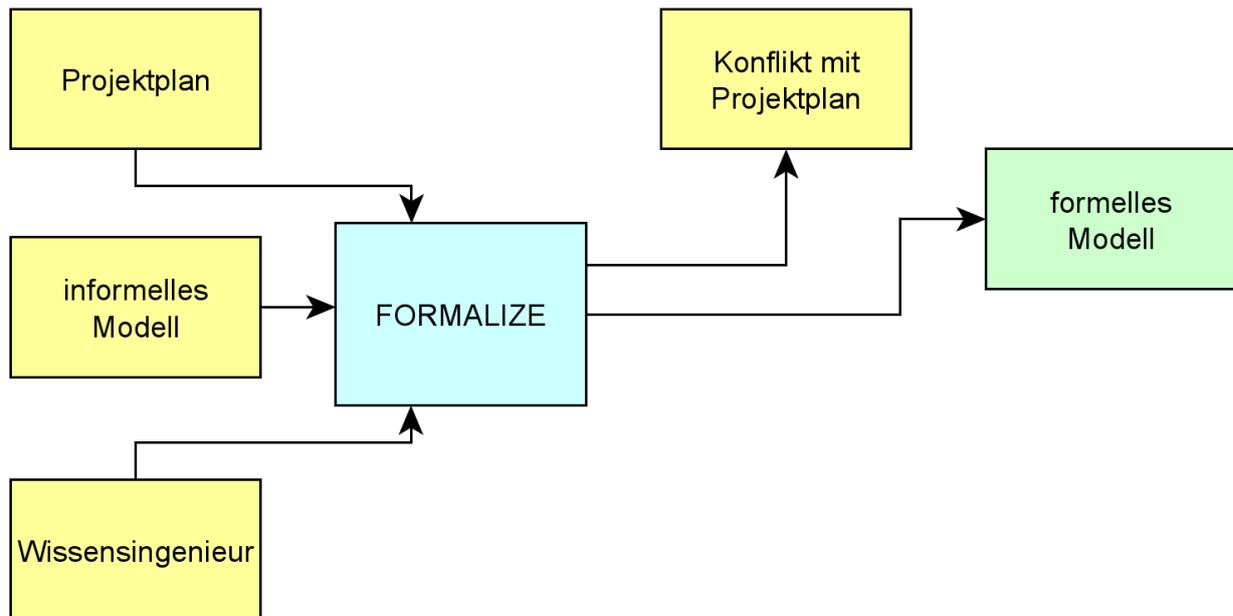


Abbildung 3.4: Der FORMALIZE Schritt nach (Stokes, 2001)

Ziel dieses Schritts besteht darin, das sogenannte formelle Modell aufzubauen. Dieser besteht aus dem Produktmodell und dem Prozessmodell. Beide Modelle stehen mit den ICARE-Formularen in Verbindung, wobei die Entity- und Constraint-Formulare das Produktmodell und die Activity- und Rule-Formulare das Prozessmodell beschreiben. In Abbildung 3.5 sind diese Beziehungen dargestellt. (Stokes, 2001)

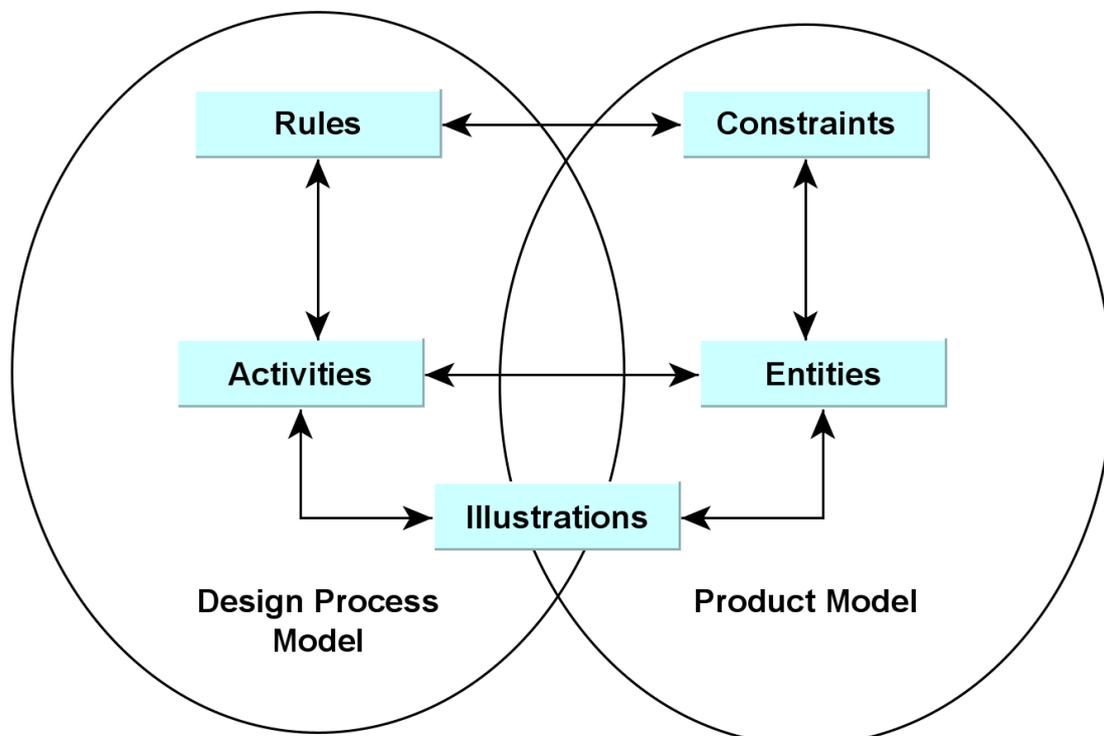


Abbildung 3.5: Beziehungen zwischen dem formellen und dem informellen Modell nach (Stokes, 2001)

Der FORMALIZE Schritt teilt sich in folgende Unterschritte auf:

1. **Vorbereitung für die Formalisierung:** Vor dem Beginn des Aufbaus des informellen Modells gibt es die Möglichkeit, den Projektplan erneut zu betrachten und ggf. Anpassungen vorzunehmen. In diesem Stadium des Projekts sollte der Umfang der KBE Anwendung bekannt sein. Dementsprechend sollte in diesem Schritt eine Betrachtung der Ressourcen auch nicht fehlen. (Stokes, 2001)
2. **Aufbau des Produktmodells:** Für den Aufbau des Produktmodells sollten alle vorherigen Schritte abgeschlossen sein. Das Modell wird aus den Entity- und Constraint-Formularen des informellen Modells aufgebaut. Dies kann entweder parallel oder sequenziell mit dem Aufbau des Prozessmodells erfolge. (Stokes, 2001)
3. **Aufbau des Prozessmodells:** Der nächste Schritt ist der Aufbau des Prozessmodells basierend auf den Activiity- und Rule-Formularen des informellen Modells. (Stokes, 2001)
4. **Zertifizierung des formellen Modells:** Die Zertifizierung des formellen Modells beinhaltet dessen Überprüfung und Validierung. Dabei werden nach fehlenden oder doppelten eingebauten ICARE-Modulen sowie fehlenden oder fehlerhaften Verlinkungen Ausschau gehalten. (Stokes, 2001)
5. **Integrierung des formellen Modells in einen Wissensspeicher:** Der letzte Schritt von FORMALIZE beinhaltet die Speicherung des formellen Modells gemeinsam mit Ressourcen und einer Dokumentation dessen. (Stokes, 2001)

Im Rahmen des MOKA Projekts wurde neben der Methodik auch ein Softwaretool entwickelt, welches die Transferierung des Wissens vom informellen zum formellen Modell vereinfachen soll. Zudem beinhaltet es eigens für den Aufbau des Produkt- und Prozessmodells entwickelte Modellierungssprache, die MOKA Modeling Language (MML). MML basiert auf der Unified Modeling Language (UML) und ist eine Erweiterung dessen. Mittels vorgefertigter Strukturen und Objekten bietet die MML einen Rahmen, mit dessen Hilfe Produkt- und Prozesswissen formell erfasst werden kann. Das Softwaretool steht unter den genannten Quellen nicht mehr zur Verfügung. (Stokes, 2001)

## Produktmodell

Das Produktmodell beschreibt das „Was“ im Produktentwurf. Es beinhaltet das Wissen, welches das Produkt selbst beschreibt, vom Aufbau über Funktionalitäten bis hin zu Verhaltensweisen. Das Produktmodell wird aus den Entity- und den Constraint-Formularen des informellen Modells aufgebaut und wird in der MML mittels vordefinierter Klassen, Attribute und sog. Ansichten beschrieben. Es gibt fünf solcher Ansichten: *Structure*, *Function*, *Behaviour*, *Technology* und *Representation*. Jede dieser Ansichten stellt eine andere Perspektive im Hinblick auf das Produkt dar. So sind beispielsweise in der *Structure* Ansicht der Zusammenbau des Produkts aus einzelnen

Baugruppen und Teilen hierarchisch dargestellt. Der Gebrauch und die Nutzung dieser Ansichten sind sehr stark vom Produkt abhängig. Es müssen nicht unbedingt alle Ansichten im Produktmodell vertreten sein. Das Ergebnis sind Klassendiagramme, die die verschiedenen Ansichten repräsentieren. (Stokes, 2001)

## Prozessmodell

Das Prozessmodell im formellen Modell beschreibt das „Wie“ und „Warum“ im Produktentwurf. Es beinhaltet das Wissen über Entwurfsprozesse und -abläufe. Das Prozessmodell wird aus den Activity- und Rule-Formularen des informellen Modells aufgebaut und mithilfe von Aktivitätsdiagrammen dargestellt. Allerdings ist diese Darstellung aufgrund der Komplexität der Entwurfsprozesse sehr schnell als schwierig und umfangreich erweisen. Dennoch sind Aktivitätsdiagramme die bevorzugte Art der Wissensrepräsentation und zumeist das Ergebnis des Prozessmodells. (Stokes, 2001)

### 3.2.7 PACKAGE

Der nächste Schritt im MOKA-Lebenszyklus ist der PACKAGE Schritt. Dieser Schritt nutzt das formelle Modell aus dem FORMALIZE Schritt um ein funktionsfähiges KBE-System zu programmieren. Die im IDENTIFY Schritt getroffenen Entscheidungen über die Zielplattform der KBE-Anwendung fließen hier mit ein. Eine Darstellung des Schritts ist in Abbildung 3.6 gegeben. (Stokes, 2001)

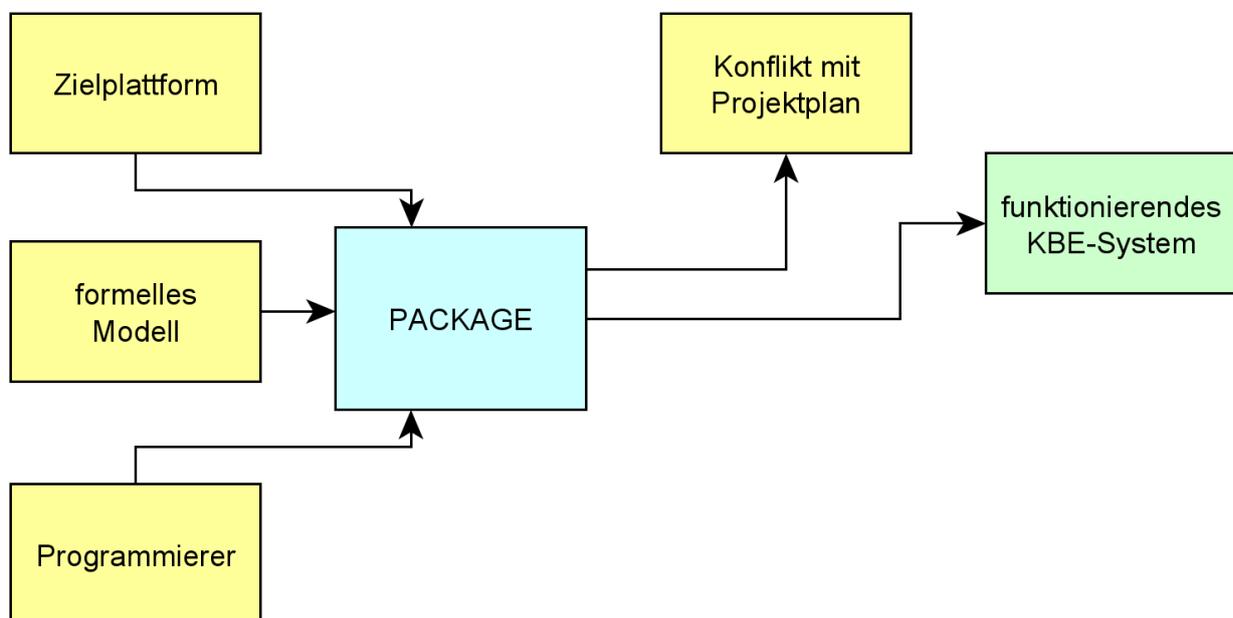


Abbildung 3.6: Der PACKAGE Schritt nach (Stokes, 2001)

Der PACKAGE Schritt ist die Übersetzung des formellen Modells in Programmcode der KBE-Anwendung und erfolgt größtenteils manuell. Der Prozess unterteilt sich in vier Unterschritte:

1. **Wissen abrufen:** In diesem Schritt werden die Produkt- und Prozessmodelle aus dem formellen Modell den Entwicklern übergeben und von denen überprüft. Im Optimalfall sollte das formelle Modell für sich sprechen, sodass der Entwickler unabhängig vom Wissensingenieur arbeiten kann. Jedoch kann es in kleineren Unternehmen durchaus vorkommen, dass der Wissensingenieur und der Entwickler ein und dieselbe Person sind. (Stokes, 2001)
2. **Wissen in geeigneter Form übersetzen:** Das formelle Modell ist eine unabhängige und neutrale Art der Wissensrepräsentation und muss in diesem Schritt daher erst in die Syntax der Zielplattform übersetzt werden. Dieser Vorgang erfolgt manuell und ist daher sehr zeitintensiv. Zudem können beim Transfer Probleme und Limitierungen seitens der Plattform auftreten, was zu einer erneuten Betrachtung des Projektplans führen kann. (Stokes, 2001)
3. **KBE-Anwendung aufbauen:** Der Aufbau der KBE-Anwendung ist stark von der gewählten Plattform abhängig. Jedoch sollte bei der Entwicklung auf den Einbau einer geeigneten Benutzeroberfläche geachtet werden. (Stokes, 2001)
4. **KBE-Anwendung testen:** Wie in jeder anderen Softwareentwicklung auch ist bei der Entwicklung der KBE-Anwendung das Testen der Anwendung unerlässlich. Neben der Überprüfung der Prozesse und Ergebnisse unter Standardbedingungen sollte auch das Verhalten der Anwendung unter Randbedingungen untersucht werden. (Stokes, 2001)

### 3.2.8 ACTIVATE

Der letzte Schritt im MOKA-Lebenszyklus ist der ACTIVATE Schritt. Er beinhaltet die Verteilung, die Installation und die Nutzung der KBE-Anwendung innerhalb des Unternehmens. Der Erfolg des gesamten KBE-Projekts ist abhängig von dem Erfolg dieses Schritts. Erweist sich die Anwendung als falsch oder fehlerhaft, müssen zu vorherigen Schritten zurückgekehrt werden. Abbildung 3.7 veranschaulicht den ACTIVATE Schritt. (Stokes, 2001)

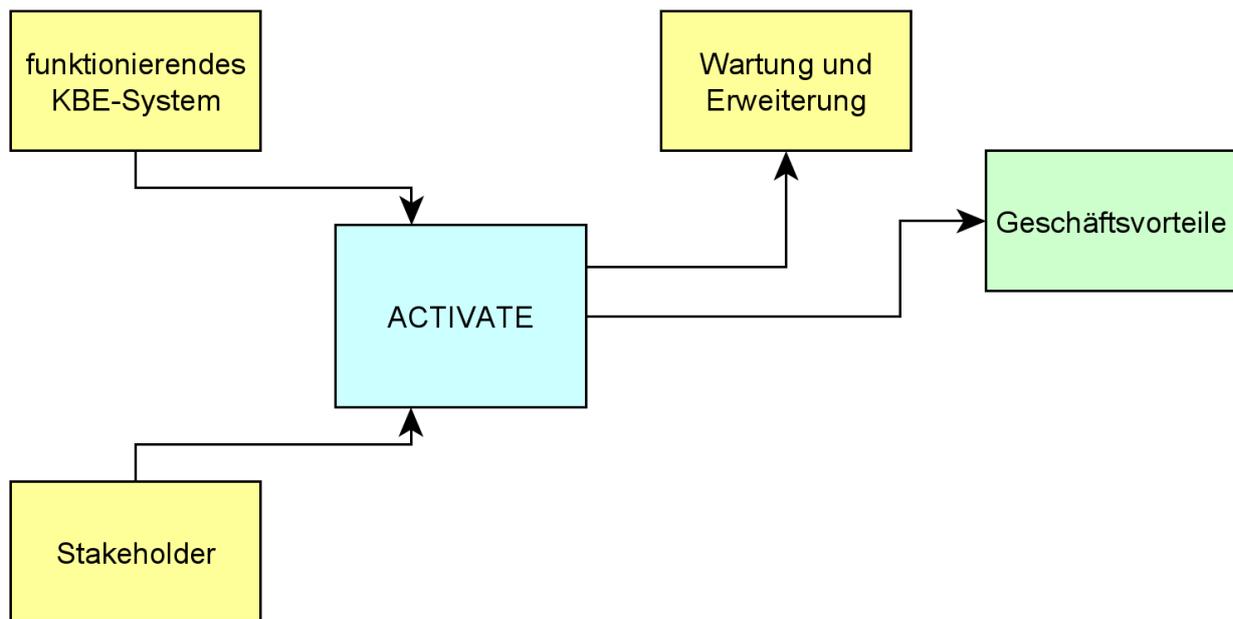


Abbildung 3.7: Der ACTIVATE Schritt nach (Stokes, 2001)

Der ACTIVATE Schritt lässt sich in folgende Unterpunkte aufteilen:

1. **Verteilung der KBE-Anwendung an Endnutzer:** Ist die Entwicklung der KBE-Anwendung abgeschlossen und durch die Entwickler verifiziert, wird sie an alle Endnutzer und Interessenten verteilt. (Stokes, 2001)
2. **Training der Endnutzer:** Alle Personen, die die KBE-Anwendung nutzen werden, sollten im Umgang mit der ihr geschult werden. Außerdem ist es empfehlenswert der Anwendung eine Bedienungsanleitung beizulegen. (Stokes, 2001)
3. **Nutzung der KBE-Anwendung:** Mit dem erfolgreichen Einsatz der KBE-Anwendung bleiben dem Unternehmen die Wartung der Anwendung und Bereitstellung eines Supports als Aufgaben. Des Weiteren ist das Einholen von Erfahrungsberichten der Nutzer ein wichtiger Bestandteil dieses Schritts. (Stokes, 2001)

## 4 Entwicklung der KBE Anwendung

In Kapitel 3 wurde die MOKA Methodik ausführlich behandelt. Nun soll die Methodik bei der Entwicklung eines Programms zum automatisierten Designentwurf hochdetaillierter Flugzeugkabinen genutzt werden. Dafür werden alle Schritte von MOKA einzeln abgearbeitet. Die Rollen des Wissensingenieurs und des Experten werden dabei von mir übernommen.

### 4.1 IDENTIFY

Der Kabinenentwurf ist ein aufwendiger Prozess bestehend aus größtenteils repetitiven Bestandteilen und kostenintensiven 3D-Mockups. Der Bedarf an einer automatisierten Lösung ist somit mehr als gegeben. Dabei pflegen vor allem Flugzeughersteller aber auch Forschungsinstitute wie das DLR Interesse an dem Projekt. Im Rahmen dieser Bachelorarbeit, die in Kooperation mit dem DLR entstanden ist, soll daher eine KBE-Anwendung entwickelt werden, die automatisiert detaillierte 3D-Modelle von Kabinen generiert. Die Flugzeughersteller versprechen sich durch den wissensbasierten Ansatz eine Kosten- und Zeitreduktion des Kabinenentwurfs, während das DLR das Ziel einer Digitalisierung dessen verfolgt.

Die KBE-Anwendung soll das Bindeglied zwischen einer Sammlung von Einzelteilen der Kabine und fertigen Kabinenlayouts bilden. Dabei soll der „outside-in“ Ansatz verfolgt und die Kabine einer vorhandenen Kabinengeometrie entsprechend entworfen werden. Dem Nutzer soll es ermöglicht werden, 3D-Modelle von Kabinen mit beliebiger Anzahl an Stützen jeder Klasse zu generieren. Die KBE-Anwendung soll dabei nur die Platzierung der Kabinenelemente übernehmen. Dafür müssen neben den gewünschten 3D-Modellen weitere Entwurfsparameter wie die Gangbreite, Anzahl der Gänge und Deckenhöhe eingegeben werden. Der Entwurfsprozess soll dabei den Regularien der CS-25 folgen.

Die CS-25 bilden somit eine der Wissensquellen für die KBE-Anwendung. Eine weitere Quelle sind die im Rahmen eines Praktikums beim DLR gesammelten Erfahrungen und Erkenntnisse in den Themengebieten Automatisierung und Kabinenlayoutgenerierung sowie im Umgang mit dem Entwurfsdatenmodell CPACS. Diese beiden sind die Hauptwissensquellen, wobei das Wissen aus der CS-25 als Text bereits in expliziter Form vorliegt. Die gesammelten Erfahrungen und Erkenntnisse sind noch in personengebunden und müssen erst einmal formuliert werden.

Die KBE-Anwendung soll als Erweiterung für die Open Source Software Blender entwickelt werden. Dafür soll die in Blender vorhandene Python API genutzt werden. Blender ist eine kostenlose 3D-Grafiksuite, mit der die Modellierung, Bearbeitung und Animation von Körpern möglich ist. Python wiederum ist eine objektorientierte Programmiersprache, die für seinen einfachen und gut lesbaren Syntax bekannt ist. Blender ermöglicht mit ihrer Python-Schnittstelle das Schreiben von Programmen zur Bearbeitung von 3D-Modellen und stellt dabei seine internen Funktionen und seine

Benutzeroberfläche zur Verfügung. Die Benutzeroberfläche der KBE-Anwendung soll ebenfalls in die von Blender integriert werden. Ein weiter Vorteil der Implementierung in Blender ist die Weiterverarbeitbarkeit der 3D-Kabinenmodelle in Blender selbst.

## 4.2 JUSTIFY

Da die KBE-Anwendung Teil dieser Bachelorarbeit ist, entfällt der Großteil des JUSTIFY-Schritts für dieses Projekt. Die dafür benötigten Ressourcen sind die Software Blender, sowie Rechner, auf dem die Software laufen kann. Blender, in welches die Programmiersprache Python bereits integriert ist, ist kostenlos und frei zugänglich. Die Rechner sind bereits vorhanden. Da die Kosten somit bei null liegen, sind auch kommerzielle und wirtschaftliche Risiken nicht vorhanden.

Aus den in IDENTIFY beschriebenen Anforderungen und dem Umfang der KBE-Anwendung lassen sich folgende Akzeptanzkriterien formulieren:

- Die 3D-Modelle müssen richtig platziert sein
- Die Anzahl der Sitze muss korrekt sein
- Die 3D-Modelle sollten sich nicht großartig überschneiden
- Die 3D-Modelle sollten nicht über die Kabinengeometrie hinausragen
- Die CS-25 Richtlinien müssen eingehalten werden.

Diese Kriterien sollen später in der Testphase der Anwendung herangezogen werden.

Durch das Nichtvorhandensein eines Unternehmens bedarf es keines Business Cases. Eine Genehmigung muss ebenfalls nicht eingeholt werden.

## 4.3 CAPTURE

Im IDENTIFY Schritt wurden zwei hauptsächliche Wissensquellen identifiziert: die Erfahrungen und Erkenntnisse aus dem Praktikum und die CS-25. Im Folgenden soll zunächst das Wissen gesammelt und zusammengefasst werden. Im nächsten Schritt wird das Wissen Mittels der ICARE-Formulare strukturiert. Dabei ist in den anschließenden Kapiteln nur eine Auswahl an Formularen dargestellt. Das gesamte informelle Modell ist im Anhang enthalten.

Flugzeugkabinen setzen sich grundsätzlich immer aus denselben Elementen zusammen. Zu diesen Elementen gehören:

- **Sitze:** Die Sitze in einer Kabine lassen sich in die drei Klassen First-class, Business und Economy unterteilen und werden auch in der Reihenfolge in der Kabine platziert. Sitze werden in Modulen installiert. Dabei hat jedes Modul eine

bestimmte Anzahl an Sitzen. Eine weitere Größe, die nicht direkt an ein Sitzmodul gebunden ist, ist der frontale Abstand zum nächsten Modul, auch Seat-Pitch genannt. Jede Klasse hat dabei einen eigenen Seat-Pitch. Sitze können in beliebiger Anzahl und Klassenkonfiguration platziert werden.

- **Flurelemente:** Unter Flurelementen versteht man Toiletten und Bordküchen. Sie sind zumeist an den Eingängen aufzufinden, in größeren Flugzeugen auch in mittleren Abschnitten der Kabine. Die Anzahl der installierten Toiletten und Bordküchen ist nicht gesetzlich gebunden und kann beliebig variieren. Dabei können sich die Elemente voneinander unterscheiden.
- **Gepäckablagen:** Als Gepäckablagen bezeichnet man den Stauraum für das Handgepäck. Sie sind über den Sitzen angebracht.
- **Seitenwände:** Die Seitenwände bilden die seitliche Begrenzung der Kabine. Sie grenzen das Innere der Kabine von der Struktur ab.
- **Deckenpaneele:** Die obere Grenze der Kabine bilden die Deckenpaneele. Sie sind über den Gängen installiert.

Abhängig von der Breite der Kabine kann zwischen einer Konfiguration mit einem oder mit zwei Gängen entschieden werden. Wählt man die Variante mit dem Doppelgang, unterscheidet man bei den Sitzen und Gepäckablagen zwischen seitlichen und mittleren Modulen. Dabei dürfen die Gänge die in der CS-25 festgelegten Mindestbreite nicht unterschreiten. Die maximale Anzahl an Sitzen pro Seite eines Gangs ist mit 3 vorgeschrieben.

In der CS-25 sind die Vorschriften für die Notausgänge besonders ausführlich beschrieben. Es gibt sieben verschiedene Typen von Notausgängen: A, B, C, I, II, III, IV. Für jeden Typ sind für den Fall einer Notevakuierung die Passagierzahlen, die ein Ausgang zulässt, festgelegt. In Tabelle 4-1 sind diese eingetragen. (European Union Aviation Safety Agency, 2020)

Tabelle 4-1: Anzahl an zulässigen Passagieren pro Notausgangstyp laut CS-25

Notausgang	Anzahl zulässige Passagiere
Typ A	110
Typ B	75
Typ C	55
Typ I	45
Typ II	40
Typ III	35
Typ IV	9

Abhängig von der Sitzanzahl sind die Mindestanzahlen der Türen vorgeschrieben. In Tabelle 4-2 sind diese Abhängigkeiten veranschaulicht. Sind in einer Kabine mehr als 300 Sitze eingebaut, müssen alle Notausgänge vom Typ A sein. (European Union Aviation Safety Agency, 2020)

Tabelle 4-2: Mindestanzahl an Türen pro Seite laut CS-25 (Gollnick, 2015)

Anzahl Sitze	Türtyp (Anzahl pro Rumpfseite)			
	Typ I	Typ II	Typ III	Typ IV
< 10				1
< 20			1	
< 40		1	1	
< 80	1		1	
< 110	1		2	
< 140	2		1	
< 180	2		2	

Sind mehr als zwei Ausgänge nötig, wirkt dies sich auf die Platzierung der Sitze aus. Je nach Typ des Notausgangs dürfen die beiden angrenzenden Sitzmodule einen Mindestabstand nicht unterschreiten. In Tabelle 4-3 sind diese Mindestabstände aufgelistet. (European Union Aviation Safety Agency, 2020)

Tabelle 4-3: Mindestabstand der Sitze an einem Notausgang laut CS-25 (Gollnick, 2015)

Türtyp	Breite /mm	Höhe/ mm	Freibereich-Tür $X_{FB}$ / mm
Typ I	610	1219	508
Typ II	508	1118	508
Typ III	508	914	508
Typ IV	483	660	508
Typ A <sup>2</sup>	1067	1829	914

Bei drei oder mehr Notausgängen werden die zu den Eingängen dazukommenden Ausgänge bevorzugt im Bereich der Tragflächen verbaut. Überschreitet jedoch der Abstand zwischen zwei Notausgängen 18,3 m muss dazwischen ein weiterer Ausgang eingebaut werden. (European Union Aviation Safety Agency, 2020)

In den folgenden Kapiteln wird das gesammelte Wissen mittels der ICARE-Formulare strukturiert und das informelle Modell erstellt.

### 4.3.1 Entity

Aus dem gesammelten Wissen lassen sich für den Kabinenentwurf die Bestandteile der Kabine ermitteln. Das 3D-Modell der Kabine ist ein Zusammenbau aus Boden, Sitzen, Flurelementen, Gepäckablagen, Seitenwänden, Deckenpanelen und Lichtern. Dabei sind alle Teile bis auf den Boden und die Lichter vorhandene 3D-Modelle. Der Boden wird mit Daten aus einem CPACS-Datensatz generiert. Die Lichter sind interne Objekte aus der Software Blender. Das Entity-Formular für die Kabine sieht dann so aus wie in Tabelle 4-4.

Tabelle 4-4: Entity-Formular der Kabine

MOKA ICARE FORMS:		ENTITY
Name	Kabine	
Reference	E1	
Entity Type	Struktur	
Functions		
Behaviour		
Context, information, validity		
Description	Das 3D-Modell einer Kabine ist das Ergebnis dieser KBE Anwendung. Es besteht aus einem Boden (E1.1), welcher aus CPACS-Daten geformt ist, Sitzen (E1.2), Flurelementen (E1.3), Gepäckablagen (E1.4), Seitenwänden (E1.5) und Deckenpanelen (E1.6), Lichtern (E1.7) und einer Kamera (E1.8). Die Objekte E1.2 bis E1.6 sind als einzelnes 3D-Modell vorhanden und werden nicht generiert, sondern nur platziert. E1.7 und E1.8 müssen ebenfalls nicht importiert werden und sind Blender-interne Objekte.	
Related Activities		
Related Entities	Parent	
	Child	E1.1, E1.2, E1.3, E1.4, E1.5, E1.6, E1.7
	Undefined	
Related Constraints		
Related Rules		
Related Illustrations		
Information Origin	Praktikum März 2020	
Management	Author	Bahadir Kocacan
	Date	02.09.2020
	Version Number	3
	Status	finished

Neben der Entität der Kabine wurde dem Entwurf eine weitere funktionelle Entität „Datensatz“ hinzugefügt. Diese dient dazu, Entwurfsdaten zu speichern und allen Prozessen zur Verfügung zu stellen. Dabei hat jeder Prozess die Möglichkeit, Daten in den Datensatz zu schreiben und vorhandene Daten auszulesen und zu modifizieren. Das Entity-Formular des Datensatzes ist in Tabelle 4-5 dargestellt.

Tabelle 4-5: Entity-Formular des Datensatzes

<b>MOKA ICARE FORMS:</b>		<b>ENTITY</b>
<b>Name</b>	Datensatz	
<b>Reference</b>	E2	
<b>Entity Type</b>	Funktion	
<b>Functions</b>	Der Datensatz hat die Funktion, Entwurfsdaten zu speichern und Prozessen zur Verfügung zu stellen.	
<b>Behaviour</b>		
<b>Context, information, validity</b>	Der Entwurf einer Kabine durchläuft verschiedene Prozesse, die alle jeweils Daten benötigen oder Daten produzieren. Dieser Datensatz dient als Zwischenspeicher für alle Entwurfsdaten	
<b>Description</b>	Der Datensatz ist ein Objekt, welches alle möglichen vordefinierten variablen Größen und Parameter enthält, die für den Kabinenentwurf relevant sind. Dabei haben alle Entwurfsprozesse Lese- und Schreibrechte am Datensatz.	
<b>Related Activities</b>	A1.1.2, A1.1.3, A1.1.4, A1.1.4.3, A1.1.5, A1.2.1, A1.2.2, A1.2.3, A1.2.4, A1.2.5, A1.2.6, A1.3.1, A1.3.2, A1.3.3	
<b>Related Entities</b>	<b>Parent</b>	
	<b>Child</b>	
	<b>Undefined</b>	
<b>Related Constraints</b>		
<b>Related Rules</b>	R2, R3, R4, R5, R8, R9, R11, R12, R13, R14, R15, R16, R18, R19, R20, R22, R23, R24, R26, R27, R28, R30, R31, R32, R34, R35, R36, R37, R38	
<b>Related Illustrations</b>		
<b>Information Origin</b>	Praktikum März 2020	
<b>Management</b>	<b>Author</b>	Bahadır Kocacan
	<b>Date</b>	02.09.2020
	<b>Version Number</b>	3
	<b>Status</b>	finished

### 4.3.2 Activity

Der Entwurf eines 3D-Modells einer Kabine wurde in 3 Hauptprozesse unterteilt: Vorbereitung, Bestimmung der Objektdaten und Aufbau des Kabinenmodells. In Tabelle 4-6 bis Tabelle 4-8 sind die Activity-Formulare der drei Prozesse dargestellt.

Tabelle 4-6: Activity-Formular der Vorbereitung

MOKA ICARE FORMS:		ACTIVITY
Name	Vorbereitung	
Reference	A1.1	
Trigger	Beginn des Programms	
Input	Eingaben vom Nutzer	
Output	Datensatz mit allen relevanten Daten zur Weiterverarbeitung	
Potential failure modes	Ungültige Eingaben, leere Geobetriebe, fehlende CPACS-Daten	
Objective	In dieser Aktivität sollen alle Vorbereitungen getroffen werden, um einen Datensatz aufzubauen, der alle relevanten Daten für die folgenden Prozesse beinhaltet.	
Input requirements	einzubauende 3D-Modelle, CPACS-Datei und einzugebende Parameter müssen vorhanden sein.	
Context, information, validity	Bei der Programmierung wird der Ansatz einer einmaligen Eingabe verfolgt. Daher widmet sich der erste Prozess des Kabinenentwurfs der Vorbereitung auf die restlichen folgenden Entwurfsprozesse	
Description	Die Vorbereitung unterteilt sich in vier Schritte: Eingaben (A1.1.1), Aufbau eines Datensatzes (A1.1.2), Zuweisung der Eingaben in den Datensatz (A1.1.3), Berechnung restlicher Größen (A1.1.4) und Prüfung der Daten (A1.1.5)	
Related Activities	Parent activity	A1
	Sub-activities	A1.1.1, A1.1.2, A1.1.3, A1.1.4, A1.1.5
	Preceding activities	
	Following activities	A1.2
Rules involved		
Entities involved	E1.2, E1.3, E1.4, E1.5, E1.6, E2	
Related Illustrations		
Information Origin	Praktikum März 2020	
Management	Author	Bahadir Kocacan
	Date	02.09.2020
	Version Number	3
	Status	finished

Tabelle 4-7: Activity-Formular der Bestimmung der Objektdaten

<b>MOKA ICARE FORMS:</b>		<b>ACTIVITY</b>
<b>Name</b>	Bestimmung der Objektdaten	
<b>Reference</b>	A1.2	
<b>Trigger</b>	Ende der Vorbereitung	
<b>Input</b>	Datensatz mit allen Entwurfsdaten und -parametern	
<b>Output</b>	Datensatz mit den Daten aller einzelnen Objekte in der Kabine	
<b>Potential failure modes</b>		
<b>Objective</b>	Name, Position und Rotation aller Objekte bestimmen.	
<b>Input requirements</b>	Vorbereitungen müssen abgeschlossen sein	
<b>Context, information, validity</b>		
<b>Description</b>	Die Bestimmung der Positionierungsdaten erfolgt für jedes 3D-Modell einzeln in folgender Reihenfolge: Frontmodule (A1.2.1), Sitze (A1.2.2), Gepäckablagen (A1.2.3), Seitenwände (A1.2.4), Deckenpaneele (A1.2.5), Heckmodule (A1.2.6)	
<b>Related Activities</b>	<b>Parent activity</b>	A1
	<b>Sub-activities</b>	
	<b>Precending activities</b>	A1.1
	<b>Following activities</b>	A1.3
<b>Rules involved</b>		
<b>Entities involved</b>	E1.2, E1.3, E1.4, E1.5, E1.6, E2	
<b>Related Illustrations</b>		
<b>Information Origin</b>	Praktikum März 2020	
<b>Management</b>	<b>Author</b>	Bahadir Kocacan
	<b>Date</b>	02.09.2020
	<b>Version Number</b>	3
	<b>Status</b>	finished

Tabelle 4-8: Activity-Formular des Aufbaus der Kabine

MOKA ICARE FORMS:		ACTIVITY
Name	Aufbau des Kabinenmodells	
Reference	A1.3	
Trigger	Ende der Bestimmung der Positionierungsdaten	
Input	Datensatz mit allen Eingabe- und Positionierungsdaten	
Output	3D-Modell der Kabine	
Potential failure modes		
Objective	3D-Modell der Kabine mithilfe des Datensatzes generieren	
Input requirements	Bestimmung der Positionierungsdaten muss beendet	
Context, information, validity		
Description	Der Aufbau erfolgt in folgenden Unterschritten: Bodengenerierung (A1.3.1), Platzierung der 3D-Modelle (A1.3.2), Platzierung der Lichter (A1.3.3)	
Related Activities	Parent activity	A1
	Sub-activities	A1.3.1, A1.3.2, A1.3.3
	Precending activities	A1.2
	Following activities	
Rules involved		
Entities involved	E1, E2	
Related Illustrations		
Information Origin	Praktikum März 2020	
Management	Author	Bahadir Kocacan
	Date	02.09.2020
	Version Number	3
	Status	finished

Im ersten Prozess sollen alle für den Kabinenentwurf notwendigen Parameter bestimmt werden. Dies erfolgt zum einen über Eingaben der Nutzer und zum anderen über Regeln, die in die Prozesse eingebunden sind. Im zweiten Prozess werden mithilfe der Parameter die Daten aller Bestandteile der Kabine bestimmt. Zu diesen Daten gehören Name, Position und Rotation der Objekte. Im dritten Prozess entsteht das 3D-Modell der Kabine. Der Boden wird mithilfe der Daten aus CPACS generiert und die einzelnen Objekte werden mit den ermittelten Daten aus dem zweiten Prozess platziert. Alle drei Prozesse laufen nacheinander ab und sind jeweils vom Ergebnis des vorherigen abhängig. Jeder der Prozesse lässt sich in Teilprozesse zerlegen und können damit hierarchisch strukturiert werden.

### 4.3.3 Rule

Das Wissen über die Regularien der CS-25 und die Regeln des Kabinentwurfs wurden mithilfe der Rule-Formulare formuliert und den entsprechenden Prozessen zugeordnet. Dabei stehen die Regeln in keinerlei Hierarchie oder Struktur zueinander. Einer der wichtigsten Regeln ist dabei die Positionsbestimmung der Sitze. In diese fließen nämlich die Regularien der Notausgänge mit ein. In Tabelle 4-9 ist diese Regel beispielhaft für die Economy-Sitze dargestellt.

Tabelle 4-9: Rule-Formular der Positionsbestimmung der Economy-Sitze

MOKA ICARE FORMS:		RULE
Name	Position Economy	
Reference	R22	
Objective	Position der Economy Sitze bestimmen	
Context, information, validity		
Description	<pre>for range(n_economy/n_economy_row): if row==1:   x=x_seat_start+1,12m, else: if exit_space &gt; 18.3:     x=x+seatpitch_economy +space_exit, elif x&gt;x_wingstart: x=x+seatpitch_economy+space_exit_ow,   else: x=x+seatpitch_economy,   y=aisle_width/2+width_economy/2, z=height/2,   row+=1, if passenger_count &gt; pax/2:   x+=2*Notausgangsbreite; x_seats_end=x+length/2</pre>	
Related Activities	A1.2.2.3	
Linked Rules		
Linked Constraints		
Related Entities	E1.2, E2	
Related Illustrations		
Information Origin	Praktikum März 2020	
Management	Author	Bahadir Kocacan
	Date	02.09.2020
	Version Number	3
	Status	finished

### 4.3.4 Constraint

Die Constraints fallen in dieser KBE-Anwendung nicht besonders stark aus. Einzig die Anzahl der Sitze pro Sitzmodul wird mithilfe eines Constraint-Formulars begrenzt. In Tabelle 4-10 ist dieses Formular dargestellt.

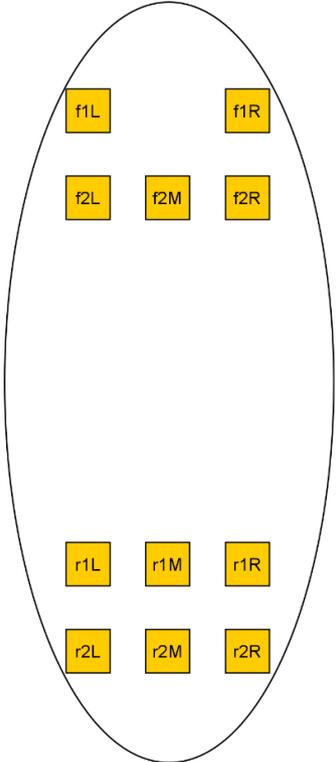
Tabelle 4-10: Constraint-Formular der Anzahl an Sitzen pro Sitzmodul

<b>MOKA ICARE FORMS:</b>		<b>CONSTRIANTS</b>
<b>Name</b>	Anzahl pro Sitzmodul	
<b>Reference</b>	C1	
<b>Objectives</b>	Begrenzung der Sitzanzahl pro Sitzmodul	
<b>Context, information, validity</b>		
<b>Description</b>	maximale Sitzanzahl an Sitzen pro Modul ist 3, wenn 1 Gang oder 6 in der Mitte und 3 außen, wenn 2 Gänge.	
<b>Linked Rules</b>	R1	
<b>Related Entities</b>	E1.2, E2	
<b>Related Illustrations</b>		
<b>Information Origin</b>	CS-25	
<b>Management</b>	<b>Author</b>	Bahadir Kocacan
	<b>Date</b>	02.09.2020
	<b>Version Number</b>	3
	<b>Status</b>	finished

#### 4.3.5 Illustration

Auch von den Illustration-Formularen gibt es in diesem Projekt nur eins. Dieser beinhaltet eine Veranschaulichung der möglichen Positionen der Flurelemente. Das Formular ist in Tabelle 4-11 dargestellt.

Tabelle 4-11: Illustration-Formular der Plätze der Flurelemente

MOKA ICARE FORMS:		ILLUSTRATION
Name	Plätze Flurelemente	
Reference	I1	
Context, information, validity		
Description		
Related Entities	E1.3	
Related Activities		
Related Rules		
Information Origin	Praktikum März 2020	
Management	Author	Bahadir Kocacan
	Date	20.08.2020
	Version Number	2
	Status	in progress

#### 4.4 FORMALIZE

Mit dem informellen Modell bestehend aus den ICARE-Formularen aus dem CAPTURE-Schritt wurden in diesem Schritt das Produkt- und das Prozessmodell erstellt.

Bei dem Produktmodell wurde hauptsächlich die strukturelle Ansicht verwendet. Die Gesamtstruktur der KBE-Anwendung besteht aus der Kabine als Produkt und aus dem Datensatz als funktionelles Element. Die Kabine wiederum besteht aus dem Boden, den

Lichtern und der Baugruppe der Kabinenelemente. Abbildung 4.1 stellt das Produktmodell in einem Klassendiagramm dar.

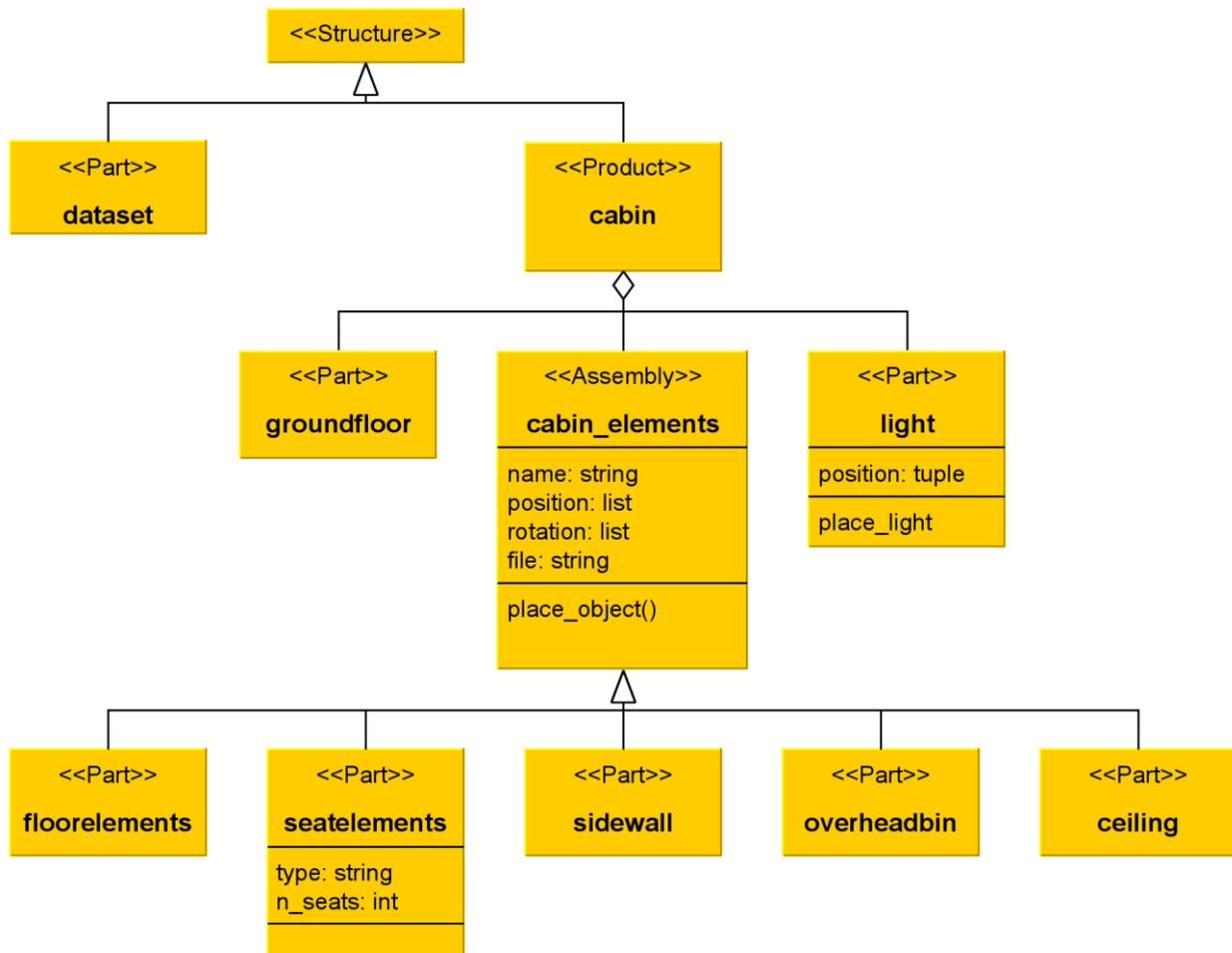


Abbildung 4.1: Das Produktmodell als Klassendiagramm

Das Prozessmodell dieses KBE-Projekts ist im Gegensatz zum Produktmodell um ein Vielfaches umfangreicher. Grundsätzlich lässt sich der Programmablauf in vier Ebenen einteilen. Dabei ist die oberste Ebene die allgemeinste und repräsentiert die drei im informellen Modell beschriebenen Hauptprozesse und mit jeder weiteren Ebene wächst der Detailgrad der Prozesse. Das Aktivitätsdiagramm des Prozessmodells ist aufgrund ihrer Größe im Anhang hinterlegt.

## 4.5 PACKAGE

Die Umsetzung des formellen Modells in eine KBE-Anwendung erfolgte in der Python-Schnittstelle des Programms Blender in der Version 2.83.0. Hierfür wurden Klassen und Module mit Methoden entsprechend dem Produkt- und Prozessmodell angelegt. Dabei wurde darauf geachtet, sowohl bei der Definition der Klassen als auch der Methoden die Strukturen aus dem formellen Modell beizubehalten. So wurden bspw. für die Kabinenelemente und für den Datensatz getrennte Dateien angelegt. Auch die Struktur

der Prozesshierarchie ist mittels getrennter Module und Ordnerstrukturen umgesetzt worden. Abbildung 4.2 zeigt die Ordnerstruktur der entwickelten KBE-Anwendung.

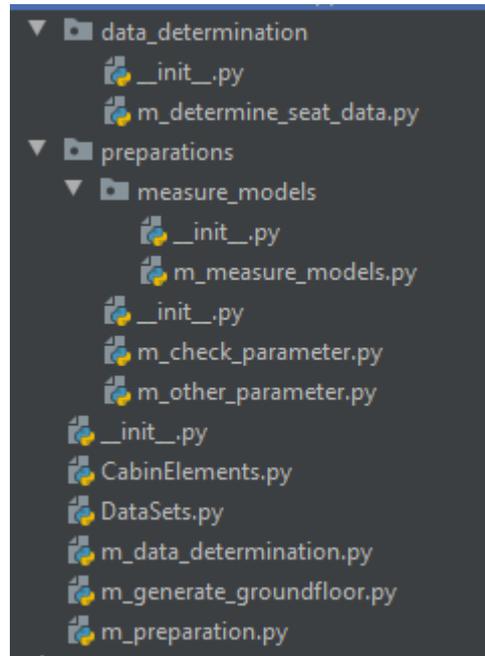


Abbildung 4.2: Ordnerstruktur der KBE-Anwendung

## 4.6 ACTIVATE

Der ACTIVATE-Schritt des KBE-Projekts wurde aufgrund der Tatsache, dass das Projekt Teil dieser Bachelorarbeit ist, noch nicht durchgeführt. Wenn die KBE-Anwendung zum Einsatz kommen sollte, würde sie als Blender-Erweiterung verteilt werden.

Die Installation der Erweiterung erfolgt über eine eigene Funktion in Blender, bei der ein Python-Skript oder eine ZIP-Datei ausgewählt werden kann. Wenn sie erfolgreich installiert worden ist, sollte sie als Tab auf der rechten Seite der 3D-Ansicht erscheinen, wie in Abbildung 4.3 dargestellt. Ist sie nicht zu sehen, kann es sein, dass die Erweiterung in den Einstellungen erst aktiviert werden muss.

Die Benutzeroberfläche der KBE-Anwendung bietet dem Nutzer die Möglichkeit der Auswahl, welche 3D-Modelle er einbauen will und welche nicht. Dabei muss darauf geachtet werden, dass von Modellen, die beidseitig eingebaut werden, das rechte eingegeben wird. Abhängig von der Auswahl öffnen sich neue Eingabefelder. Für einen erfolgreichen Kabinenentwurf müssen alle geforderten Eingaben erfolgen. Ansonsten gibt das Programm einen Fehler aus. Bei der Nutzung der KBE-Anwendung ist das Öffnen von Blenders System Konsole ratsam, da die Prozesse und Fehler in ihr ersichtlich sind.

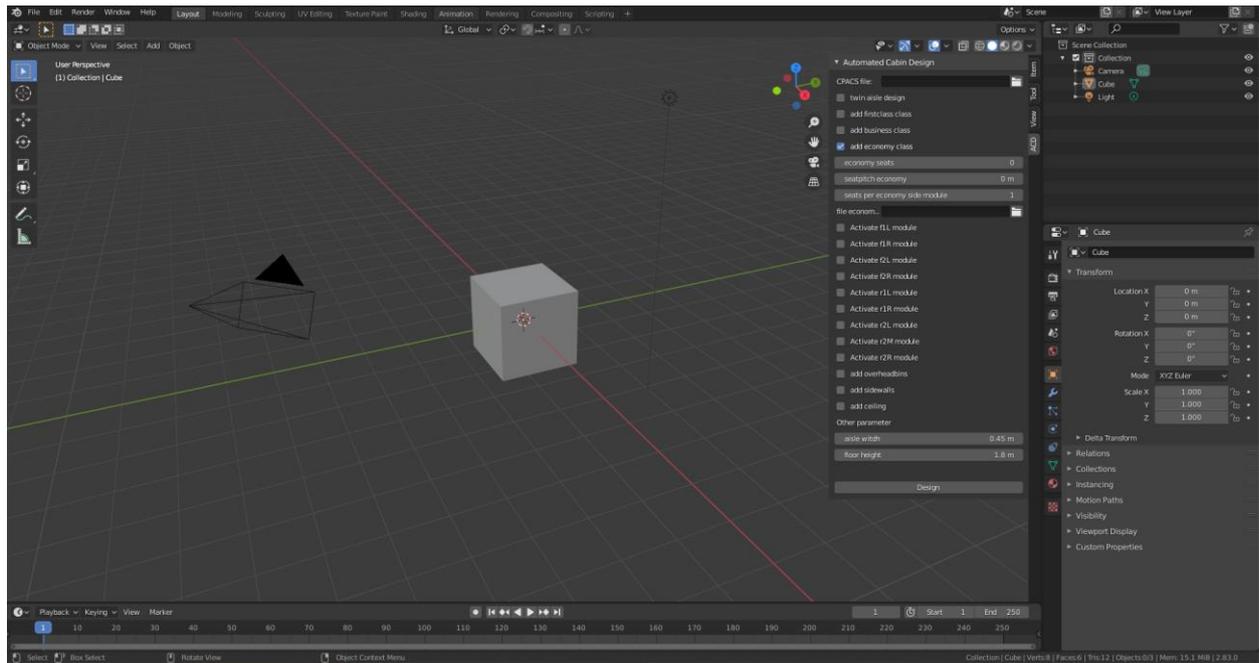


Abbildung 4.3: Die Benutzeroberfläche der KBE-Anwendung

## 5 Ergebnisse

Für den Kabinenentwurf wurden 3D-Modelle und CPACS-Datensätze seitens der DLR bereitgestellt. Mithilfe dieser wurde aus zwei CPACS-Datensätzen zwei Kabinen entworfen. Abbildung 5.1 und Abbildung 5.2 zeigen eine doppelgängige Konfiguration. Sie besteht aus insgesamt 28 Business- und 266 Economy-Sitzen. Der Seat-Pitch beider Klassen beträgt 30 Zoll. Seitens der KBE-Anwendung wurden insgesamt vier Notausgänge je Seite platziert.



Abbildung 5.1: Auf CPACS basierender Kabinenentwurf mittels KBE, große Kabine, Ansicht Innen

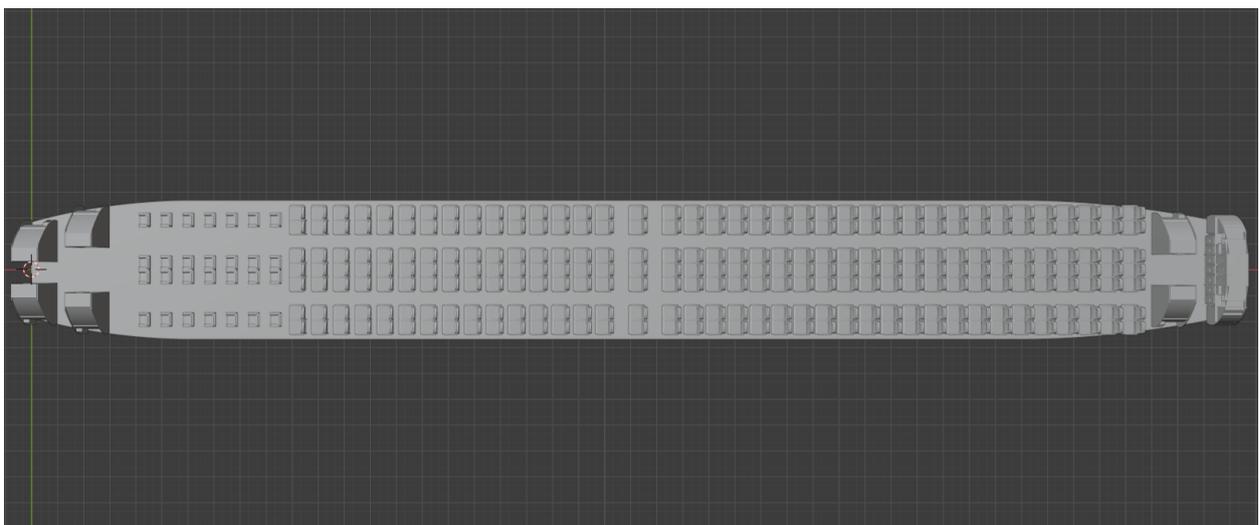


Abbildung 5.2: Auf CPACS basierender Kabinenentwurf mittels KBE, große Kabine,, Draufsicht

Abbildung 5.3 und Abbildung 5.4 zeigen eine eingängige Konfiguration. Sie besteht aus insgesamt 20 Business- und 120 Economy-Sitzen. Der Seat-Pitch beider Klassen beträgt 30 Zoll. Seitens der KBE-Anwendung wurden insgesamt drei Notausgänge je Seite platziert.



Abbildung 5.3: Auf CPACS basierender Kabinenentwurf mittels KBE, kleine Kabine, Ansicht innen

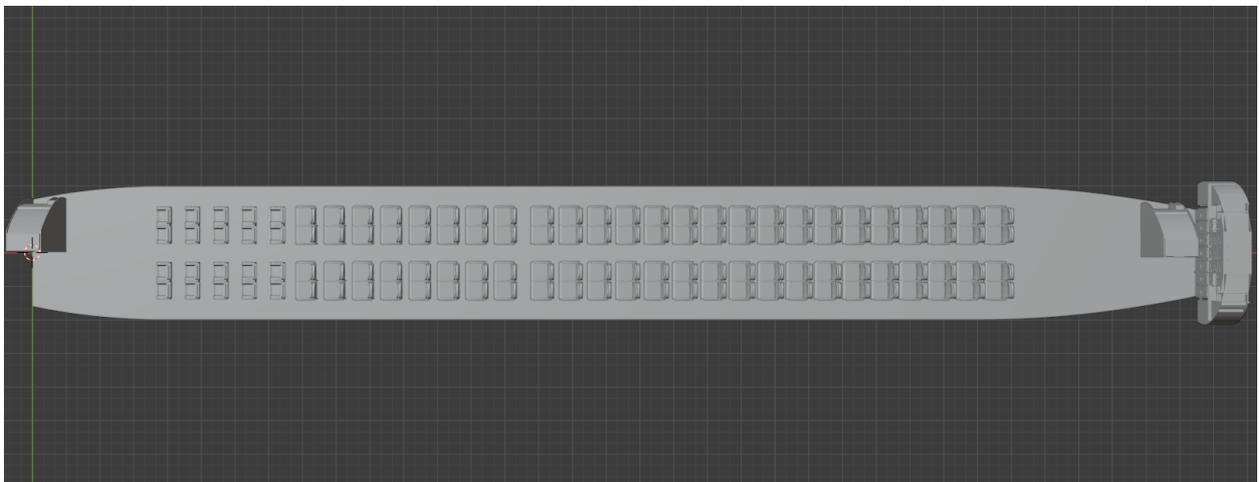


Abbildung 5.4: Auf CPACS basierender Kabinenentwurf mittels KBE, kleine Kabine, Draufsicht

In beiden Modellen wurden neben den Sitzen auch Toiletten, Bordküchen, Gepäckablagen, Seitenwände und Deckenpanele an zu erwarteten Positionen platziert.

## 6 Vergleich mit bestehenden Kabinenlayouts

Im Folgenden sollen die Ergebnisse des automatisierten Kabinenentwurfs aus Kapitel 5 mit bereits bestehenden Kabinenlayouts von etablierten Fluggesellschaften verglichen werden.

Zunächst soll die Einzelgangkonfiguration der mit der Embraer E-195 der Lufthansa (s. Abbildung 6.1) verglichen werden.

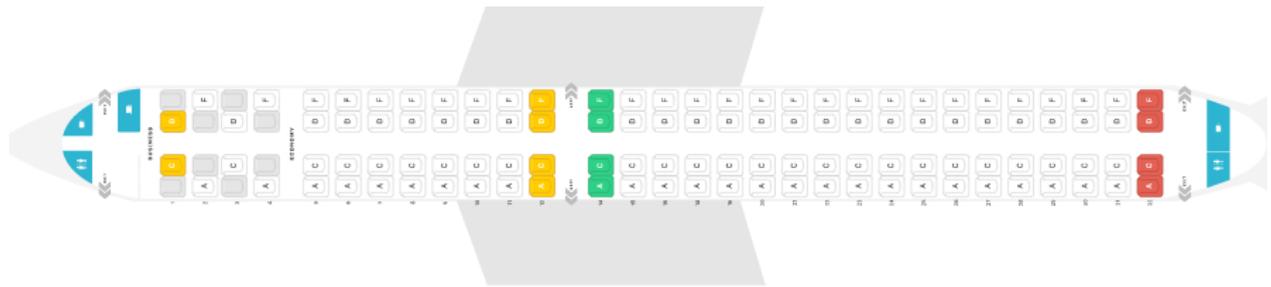


Abbildung 6.1: Kabinenlayout der Embraer E-195 der Lufthansa (TripAdvisor LLC, 2020)

Beide Layouts weisen viele Gemeinsamkeiten auf. Zum einen sind im Bereich der vorderen und hinteren Eingänge Toiletten und Bordküchen vorhanden. Die einzelnen Konfigurationen sind dabei allerdings unterschiedlich. Eine weitere Gemeinsamkeit ist in der Anordnung der Sitze zu erkennen. Beide Kabinenlayouts sind Business- und Economy-Klassen eingebaut, wobei die Business-Sitze die ersten Reihen und die Economy-Sitze die restlichen bilden. Lediglich die Anzahl an Sitzen in den jeweiligen Klassen unterscheidet beide Layouts. Eine der wichtigsten Gemeinsamkeiten besteht in der Anzahl der Notausgänge pro Seite. In beiden Kabinenlayouts existieren 3 Notausgänge, bestehend aus den beiden Eingängen und einem Notausgang mittig der Kabine im Bereich der Tragflächen.

Jedoch zeigen beide Kabinenlayouts auch Unterschiede auf. Während das Layout von Lufthansa den Kabinenplatz voll ausnutzt, weist das Layout des KBE-Entwurfs großflächige Lücken auf. Außerdem überschreitet die Bordküche am Heck des Flugzeugs den Rand der Kabine.

Der Vergleich der Doppelgangkonfiguration des KBE-Entwurfs soll mit einem Layout der Airbus A330-330 der Lufthansa erfolgen. (s. Abbildung 6.2)



Abbildung 6.2: Kabinenlayout der A330-300 der Lufthansa (TripAdvisor LLC, 2020)

Beide Layouts weisen ebenfalls Toiletten und Bordküchen in Bereichen der Eingänge auf. Jedoch sind im Gegensatz zum Layout von Lufthansa im KBE-Entwurf keine in mittleren Abschnitten der Kabine verbaut. Beide Layouts beinhalten unterschiedliche Sitzklassen. Allerdings sind im KBE-Entwurf nur zwei Klassen statt vier, wie bei der Lufthansa eingebaut. Ein weiterer Unterschied ist der mittleren Reihe der Economy-Sitze zu erkennen. Während im KBE-Entwurf die Anzahl an Sitzen der mittleren Sitzmodule der Economy-Klasse konstant bleibt, variiert sie im Kabinenlayout von Lufthansa. In der Anzahl an Notausgängen pro Seite haben beide Layouts eine Gemeinsamkeit, unterscheiden sich jedoch in deren Positionen.

Der Vergleich der Entwürfe aus der KBE-Anwendung mit bereits bestehenden Layouts in zwei entgegengesetzten Beispielen hat zu einigen Erkenntnissen geführt. Der Entwurf von einfachen Kabinen ist mithilfe der KBE-Anwendung problemlos möglich. Erhöht man die Größe und die Anzahl an Sitzen, nehmen auch die Anzahl an gestalterischen Möglichkeiten der Kabine zu. In seinem jetzigen Stadium ist die KBE-Anwendung noch nicht in der Lage, alle Möglichkeiten abzudecken, da in das KBE-System das entsprechende Wissen nicht implementiert ist.

## 7 Bewertung der Anwendung

Die Ergebnisse der Kabinenentwürfe und der Vergleich mit bestehenden Layouts haben Stärken und Schwächen der entwickelten KBE-Anwendung offengelegt.

Eine Stärke liegt in der Flexibilität der Anwendung. 3D-Modelle sind schnell austauschbar und können je nach Bedarf und Belieben ein- und ausgebaut werden. Ein weiterer Vorteil ist der geringe Zeitaufwand eines Entwurfs. Es können eine höhere Anzahl an Kabinenlayouts generiert und bewertet werden. In Kombination mit der Flexibilität des Programms können somit auch neue Konfigurationen und Konzepte erforscht und getestet werden.

Die Schwächen der KBE-Anwendung liegen zum einen in ihren hohen Anforderungen an Rechenleistung und Arbeitsspeicher für den Aufbau und die Darstellung der 3D-Kabinenmodelle. Eine weitere große Schwäche ist der Mangel an Wissen, um komplexere Kabinenmodelle zu entwerfen.

## 8 Fazit und Ausblick

Im Umfang dieser Bachelorarbeit wurde das Thema der Automatisierung durch Knowledge Based Engineering im Kontext des Kabinenentwurfs aufgegriffen und angewandt.

Der Kabinenentwurf ist ein wichtiger und unerlässlicher, aber auch kosten- und zeitintensiver Teil des Flugzeugentwurfs. Er besteht größtenteils aus repetitiven Prozessen und unterliegt gesetzlichen Regularien. Diese beiden Aspekte bilden eine optimale Grundlage für einen automatisierten Kabinenentwurf mittels Knowledge Based Engineering.

Knowledge Based Engineering ist ein wissensbasiertes System zur automatisierten Produktentwicklung durch Erfassung und Wiederverwendung von Produkt- und Prozesswissen. KBE verspricht eine Kosten- und Zeitreduktion in Entwurfsprozessen. Daher sollte in dieser Arbeit mithilfe von KBE eine Anwendung entwickelt werden, die den Entwurf von Kabinen automatisieren soll.

Die „Methodology and software tools Oriented to Knowledge based engineering Applications“, kurz MOKA, ist eine Methodik zur Entwicklung von KBE-Anwendungen. Sie beinhaltet einen Lebenszyklus, der die einzelnen Schritte der Entwicklung beschreibt. Die Kernelemente bilden dabei das informelle und das formelle Modell.

Mithilfe der MOKA-Methodik wurde eine KBE-Anwendung für den Kabinenentwurf basierend auf dem Datenschema CPACS entwickelt. Als Plattform dafür dient die 3D-Modellierungs-Software Blender und ihre Python-Schnittstelle.

Die Ergebnisse der KBE-Anwendung sind 3D-Modelle von Kabinenentwürfen. Diese wurden mit bestehenden Layouts verglichen und es wurden Gemeinsamkeiten und Unterschiede zwischen ihnen, sowie Stärken und Schwächen der KBE-Anwendung in seiner jetzigen Form aufgezeigt.

Schlussendlich lässt sich sagen, dass die Anwendung des wissensbasierten Ansatzes Knowledge Based Engineering auf die Problematik des Kabinenentwurfs ein geeignetes Mittel zur Lösung dieser ist. Jedoch ist die Problematik noch nicht in Gänze gelöst. Viele Aufgaben, wie z.B. die Wartung und Erweiterung der KBE-Anwendung, bleiben weiterhin offen. Ein weiteres Themengebiet, welches sich mit der Thematik KBE befassen bzw. überschneiden könnte, wäre die Digitalisierung weiterer Prozesse im Flugzeugentwurf, mit dem Ziel irgendwann eine komplett digitale Lösung zu haben.

# Literaturverzeichnis

**Bibliographisches Institut GmbH.** Duden. [Online] [Zitat vom: 07. Juli 2020.] <https://www.duden.de/rechtschreibung/Wissen>.

**Center, C. 2012.** *KMB-PR Praktikum aus Knowledge Management im Bildungsbereich.* 2012.

**Chapman, C. B und Pinfold, M. 1999.** Design engineering - a need to rethink the solution using knowledge based engineering. *Knowledge-Based Systems.* Volume 12, 1999.

**European Union Aviation Safety Agency. 2020.** *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25.* 2020.

**German Aerospace Center.** cpacs. [Online] [Zitat vom: 11. Juli 2020.] <https://www.cpacs.de>.

**Gobbin, Andreas. 2015.** *Numerische Modellierung des Auslegungsprozesses für Passagierkabinen von Verkehrsflugzeugen unter Berücksichtigung der wichtigsten Auslegungsforderungen und Implementierung in MatLab.* Berlin : Technische Universität Berlin, 2015.

**Gollnick, Volker. 2015.** *Methoden des Flugzeugentwurfs I.* Hamburg : Technische Universität Hamburg-Harburg, 2015.

**International Civil Aviation Organization. 2019.** *Aviation Benefits Report 2019.* 2019.

**La Rocca, Gianfranco. 2011.** *Knowledge Based Engineering Techniques to Support Aircraft Design and Optimization.* 2011. ISBN/EAN: 978-90-9026069-3.

**McGoey, P.J. 2011.** *A Hitch-hikers Guide to: Knowledge Based Engineering in Aerospace (& other Industries) Do not Panic!! Definition, History, Value, Technology, and Re-sources.* 2011.

**Reinmann-Rothmeier, Gabi und Mandl, Heinz. 2000.** Spektrum. *Spektrum.de.* [Online] 2000. [Zitat vom: 09. Juli 2020.] <https://www.spektrum.de/lexikon/psychologie/wissen/16892>.

**Singer, Dominic. 2014.** *Entwicklung eines Prototyps für den Einsatz von Knowledge-based Engineering in frühen Phasen des Brückenentwurfs.* s.l. : Technische Universität München, 2014.

**Steinkellner, Stefan. 2012.** *Knowledge-based Engineering in der Technischen Logistik.* Graz : TU-Graz, Institut für Technische Logistik, 2012.

**Stokes, Melody. 2001.** *Managing Engineering Knowledge: MOKA: Methodology for Knowledge Based Engineering Applications.* s.l. : Professional Engineering Publishing Limited, 2001. ISBN 1 86058 295 8.

**TripAdvisor LLC. 2020.** SeatGuru. [Online] 2020. [Zitat vom: 14. 09 2020.] [https://www.seatguru.com/airlines/Lufthansa/Lufthansa\\_Airbus\\_A330-300\\_B.php](https://www.seatguru.com/airlines/Lufthansa/Lufthansa_Airbus_A330-300_B.php).

—. 2020. SeatGuru. [Online] 2020. [Zitat vom: 14. 09 2020.]  
[https://www.seatguru.com/airlines/Lufthansa/Lufthansa\\_Embraer\\_195.php](https://www.seatguru.com/airlines/Lufthansa/Lufthansa_Embraer_195.php).

**Tripathi, K P. 2011.** A Review on Knowledge-based Expert System: Concept and Architecture. [Buchverf.] AIT. *IJCA Special Issue on "Artificial Intelligence Techniques - Novel Approaches & Practical Applications"*. Kolhapur, India: Bharati Vidyapeeth Deemed University, 2011.

**Verein Deutscher Ingenieure. 2017.** *Wissensmanagement im Ingenieurwesen: Wissensbasierte Konstruktion (KBE)*. 2017. VDI 5610 Blatt 2.

# Anhang

## A.1 Dateien auf der beigefügten CD

- \1\_Bachelorarbeit\Bachelorarbeit\_Kocacan\_Bahadir.pdf
- \2\_informelles\_Modell\ACTIVITY.xlsx
- \2\_informelles\_Modell\CONSTRAINT.xlsx
- \2\_informelles\_Modell\ENTITY.xlsx
- \2\_informelles\_Modell\ILLUSTRATION.xlsx
- \2\_informelles\_Modell\RULE.xlsx
- \3\_formelles\_Modell\DesignProcessModel.png
- \3\_formelles\_Modell\ProductModel.png
- \4\_KBE\_Programm\ACD.zip
- \5\_3D\_Modelle\BusinessSeatDouble.stl
- \5\_3D\_Modelle\BusinessSeatSingle.stl
- \5\_3D\_Modelle\Ceiling.stl
- \5\_3D\_Modelle\EconomySeatDouble.stl
- \5\_3D\_Modelle\EconomySeatQuadro.stl
- \5\_3D\_Modelle\EconomySeatSingle.stl
- \5\_3D\_Modelle\EconomySeatTriple.stl
- \5\_3D\_Modelle\f1L.stl
- \5\_3D\_Modelle\f1R.stl
- \5\_3D\_Modelle\f2L.stl
- \5\_3D\_Modelle\f2M.stl
- \5\_3D\_Modelle\f2R.stl
- \5\_3D\_Modelle\FirstclassSeat.stl
- \5\_3D\_Modelle\Galley.stl
- \5\_3D\_Modelle\OverheadbinMid.stl

- \5\_3D\_Modelle\OverheadbinSide.stl
- \5\_3D\_Modelle\r1L.stl
- \5\_3D\_Modelle\r1M.stl
- \5\_3D\_Modelle\r1R.stl
- \5\_3D\_Modelle\r2M.stl
- \5\_3D\_Modelle\Sidewall.stl
- \6\_CPACS\_Datensätze\CPACS\_large.xml
- \6\_CPACS\_Datensätze\CPACS\_small.xml