Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik

Masterarbeit

Vergleich ausgewählter Datenaustauschstrategien im Ingenieurwesen

Autor:
Sara Kunze

17. Oktober 2012

Gutachter:
Prof. Gunter Saake
Fakultät für Informatik
Institut für Technische und Betriebliche Informationssysteme

Prof. Sandor Vajna
Fakultät für Maschinenbau
Institut für Maschinenkonstruktion

Betreuer:
Dipl.-Ing.-Inf. Maik Mory
Fakultät für Informatik
Institut für Technische und Betriebliche Informationssysteme

Dipl.-Ing.-Inf. Alexander Blankenburg
Fakultät für Maschinenbau
Institut für Maschinenkonstruktion

Kunze, Sara:
Vergleich ausgewählter Datenaustauschstrategien im Ingenieurwesen
Masterarbeit, Otto-von-Guericke-Universität Magdeburg, 2012.

Inhaltsangabe

Der Datenaustausch zwischen heterogenen Systemen ist auch heute noch ein Thema in der Forschung und Industrie. Diese Arbeit beschäftigt sich mit dem Vergleich von ausgewählten Datenaustauschstrategien im Ingenieurwesen. Dazu werden Klassifikationsschemata zur Unterscheidung solcher Strategien vorgestellt. Weiterhin betrachtet diese Arbeit den Datenaustausch bei der Montagesimulation als ein konkretes Anwendungsszenario des Ingenieurwesens. Bei diesen Betrachtungen werden verschiedene Variationen des Beispielszenarios identifiziert. Zudem werden die Klassen der möglichen Datenaustauschstrategien nach den Klassifikationsschemata herausgearbeitet. Danach werden die Datenaustauschlösungen STEP, JT, die Online-Kopplung mit einem CAx-Objektbus und die Online-Kopplung mit OpenGL verglichen. Dabei werden die ausgewählten Strategien vorgestellt, ihre Stärken und Schwächen identifiziert, die Strategien in die zuvor behandelten Klassifikationsschemata eingeordnet und für den Einsatz im Beispielszenario beurteilt. Die Beurteilung geht auf die identifizierten Variationen ein. Insgesamt ist die Arbeit als Beispiel für den Vergleich von Datenaustauschstrategien für ein konkretes Anwendungsszenario im Ingenieurwesen zu sehen. Beantwortet werden Fragen nach einer Einteilung von Datenaustauschstrategien, nach einem konkreten Anwendungsszenario und nach ausgewählten Datenaustauschstrategien.

Danksagungen

An dieser Stelle möchte ich mich bei meiner Familie, meinen Freunden und allen Bekannten, die mir bei der Erstellung dieser Arbeit geholfen haben, bedanken. Eure Unterstützung, Motivation und Vorschläge haben mir bei dieser Arbeit sehr geholfen. Für das mühevolle Korrekturlesen meiner Arbeit bedanke ich mich bei meiner Mutter, Kornelia Kunze, meiner Schwester, Liane Kunze, und meiner Freundin, Anja Bachmann. Besonderer Dank gilt meinen Eltern, Kornelia Kunze und Ronald Kunze. Ihr habt mir das Studium und damit das Schreiben dieser Arbeit erst ermöglicht und dafür möchte ich mich bei euch bedanken.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

ANICA	Analysis of Access Interfaces of Various CAx-Systems
B-Rep	Boundary Representation
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAx	Computer Aided x
CORBA	Common Object Request Broker Architecture
CSG	Constructive Solid Geometry
DMU	Digital Mock-up
ISO	International Organization for Standardization
JT	Jupiter Tessellation
ORB	Object Request Broker
PDM	Produktdatenmanagement
PMI	Product Manufacturing Information
STEP	Standard for the Exchange of Product Model Data
VDA	Verband der Automobilindustrie
VR	virtuelle Realitäten

1. Einleitung

1.1 Motivation

Die vorliegende Arbeit beschäftigt sich mit dem Datenaustausch im Ingenieurwesen. Der Begriff „*Ingenieurwesen*“ umfasst nach der Brockhaus Enzyklopädie [Bro06] alle „Disziplinen, die aus der systematisch theoretischen Bearbeitung technischer Probleme entstanden und heute Gegenstand des ingenieurwissenschaftlichen Studiums sind.“ Klassische Fächer des Ingenieurwesens sind das Bauwesen, der Bergbau, die Elektrotechnik, der Maschinenbau und die Verfahrenstechnik [Bro06]. Diese Fächer unterteilen sich weiter in unterschiedliche Fachrichtungen und Schwerpunktbereiche. Heute existieren für viele Aufgaben- und Fragestellungen der Bereiche verschiedene Softwaresysteme zur Unterstützung. Dabei ist es aufgrund des Umfangs der möglichen Aufgaben und der Vielzahl an Realisierungsmöglichkeiten nicht möglich, alle verfügbaren Funktionalitäten in einer Softwarelösung zu vereinen [VWB+09]. Ein Beispiel für einen solchen Ingenieurbereich ist die Produktentstehung [VWB+09]. Innerhalb eines Produktentstehungsprozesses sind viele stark voneinander unterschiedliche Aufgaben zu erfüllen. Dabei beschränken sich diese häufig nicht auf ein Fach des klassischen Ingenieurwesens. Häufig sind neben dem Maschinenbau auch die Elektrotechnik und die Verfahrenstechnik beteiligt [VWB+09]. Ein Softwaresystem ist daher nicht in der Lage, alle Aspekte der Produktentstehung zu unterstützen [VWB+09]. Es kommen somit mehrere Systeme zum Einsatz.

Das Problem bei mehreren Systemen ist die Weiterverwendung einmal erzeugter Daten. Eine neue Eingabe ist aufwendig und fehleranfällig. Daher wird ein Austausch von Daten zwischen den Systemen angestrebt. Dies stellt bereits seit Jahren eine Herausforderung dar, die auch heute noch nicht vollkommen gelöst ist [MPKS11]. 1999 schätze eine Studie [BM99], dass etwa eine Milliarde Dollar in den USA jährlich allein bei den Automobilzulieferern wegen fehlerhafter Interoperabilität ausgegeben werden. Eine Verbesserung dieser Situation ist nicht zu beobachten.

Der Grund für die Probleme beim Datenaustausch ist, dass die Systeme heterogen sind. Das bedeutet, dass die Systeme die Daten auf unterschiedliche Art und Weise

abbilden und speichern, sodass das jeweils andere System nicht direkt mit den Daten arbeiten kann.

Zur Lösung dieses Problems wurde eine Vielzahl von Strategien entwickelt [VWB+09]. Diese Vielzahl macht jedoch die Auswahl einer geeigneten Lösungsstrategie für die eigene Anwendung schwierig. Auch der Beurteilung neuer Ansätze wird durch die Menge an bestehenden Ansätzen erschwert.

Diese Arbeit beschäftigt sich mit dem Vergleich von Datenaustauschstrategien speziell im Ingenieurwesen. Im nächsten Abschnitt werden die gesetzten Ziele und Fragestellungen der Arbeit behandelt.

1.2 Ziel der Arbeit

Die Zielsetzung dieser Arbeit ist es, einen Vergleich von Datenaustauschstrategien für das Ingenieurwesen so durchzuführen, dass der Vergleich mit anderen Strategien und anderen Randbedingungen immer wieder durchgeführt werden kann. Diese Arbeit stellt somit eine Art Leitbeispiel für den Vergleich von Lösungsstrategien für den Datenaustausch im Ingenieurwesen dar.

Dabei werden folgende Fragestellungen beantwortet:

- Welche Klassifikationsschemata gibt es für Datenaustauschstrategien im Ingenieurwesen?
- Was ist ein konkretes Anwendungsszenario und welche Anforderungen hat es?
- Wie werden die ausgewählten Datenaustauschstrategien in die Klassifikationsschemata eingeordnet und wie sind sie für das Beispielszenario zu bewerten?

Zur Abgrenzung der umfangreichen Thematik beschränkt sich diese Arbeit auf den systemübergreifenden Austausch zwischen heterogenen Systemen. Ein funktionierender Datenversand wird als gegeben angenommen. Zudem werden ausschließlich Kopplungslösungen betrachtet. Bei dieser Art von Lösungen können die einzelnen Systeme als eigenständig identifiziert und die Verbindung jederzeit getrennt werden.

Der nächste Abschnitt geht kurz auf den Aufbau dieser Arbeit ein.

1.3 Aufbau der Arbeit

Die vorliegende Arbeit ist wie folgt aufgebaut. In Kapitel 2 werden Grundlagen erläutert. Dies beinhaltet die Vorstellung verschiedener Klassifikationsschemata. Das darauf folgende Kapitel 3 stellt das verwendete Beispielanwendungsszenario vor. Dabei wird auf die Anforderungen und Besonderheiten des Szenarios eingegangen. Als Beispielanwendungsszenario dient in dieser Arbeit der Datenaustausch bei der Montagesimulation. Das vierte Kapitel beschäftigt sich dann mit dem eigentlichen Vergleich von vier ausgewählten Datenaustauschstrategien. Diese werden in die Klassifikationsschemata eingeordnet und bezüglich ihrer Verwendbarkeit für das Beispielszenario beurteilt. Den Abschluss der vorliegenden Arbeit bildet das Kapitel 5. Dieses fasst die Ergebnisse zusammen und gibt einen Ausblick auf weitere mögliche Untersuchungen.

2. Grundlagen

Dieses Kapitel befasst sich mit den Grundlagen dieser Arbeit. Der erste Abschnitt erläutert grundlegende Begriffe zum besseren Verständnis der Arbeit. Danach folgt eine Betrachtung der Klassifikation von Lösungsstrategien für den Datenaustausch. Dazu werden verschiedene Klassifikationsschemata behandelt. Der vorletzte Abschnitt dieses Kapitels erläutert Stufen von Produktmodellen in der virtuellen Produktentstehung. Den Abschluss bildet eine kurze Zusammenfassung zu den Grundlagen.

2.1 Grundbegriffe

Die Arbeit beschäftigt sich mit der Thematik des Datenaustausches. *Daten* sind Werte, die elektronisch gespeichert und verarbeitet werden [VWB+09, S.495]. Die Werte liegen in numerischer, alphabetischer oder alphanumerischer Form vor [VWB+09, S.57]. Erst durch die Verknüpfung der Daten mit ihrer Bedeutung entstehen für den Menschen verständliche Informationen [LLS06, S.32][VWB+09, S.57f]. Ein *Datenformat* beschreibt eine spezifische Strukturierung von Daten, die von einer Software durchgeführt wird [BBD+03, S.218]. Eine *Datei* ist eine geordnete, elektronisch gespeicherte Datenmenge [BBD+03, S.207]. Das *Dateiformat* beschreibt die Art und Weise, in der die Daten in der Datei abgespeichert und ausgelesen werden [BBD+03, S.218]. Ein *Modell* ist eine Abbildung mit in der Regel verminderten Eigenschaften des Originals [BBD+03, S.588]. Der Aufbau eines Modelles im Rechner erfolgt durch verschiedene Daten zu den Eigenschaften des Originals.

Der *Datenaustausch* ist ein Prozessschritt, der für sich gesehen einen gerichteten Datenfluss vom Sender zum Empfänger darstellt [VDA02]. Für den Rückfluss von Daten ist ein weiterer Datenaustausch notwendig [VDA02]. Somit entsteht ein bidirektionaler Datenaustausch [DFB11].

Schumann [Sch01, S.3f] unterscheidet zwischen zwei grundlegenden Aspekten des Datenaustausches, dem *Datenversand* und dem *systemübergreifenden Datenaustausch*. Der *Datenversand* behandelt den reinen Transport der Daten und ist damit

unabhängig vom genauen Inhalt und der internen Struktur der Daten. Der *systemübergreifende Datenaustausch* beschäftigt sich hingegen mit der Frage, in welcher Art und Weise Daten zwischen verschiedenen Softwaresystemen ausgetauscht werden müssen, sodass das Empfängersystem diese Daten interpretieren und weiterverarbeiten kann. Somit ist der systemübergreifende Datenaustausch von Inhalt und Struktur der Daten abhängig. Beide Aspekte des Datenaustausches bedingen sich gegenseitig. Diese Arbeit beschäftigt sich ausschließlich mit dem Aspekt des systemübergreifenden Datenaustausches. Ein funktionierender Datenversand wird als gegeben angenommen.

Sender und Empfänger des Datenaustausches sind Systeme. Für den Begriff „*System*" gibt es je nach Anwendungsgebiet verschiedene Definitionen [VWB+09, S.101]. Diese Arbeit verwendet die Definition der DIN IEC 600050-351 [DIN06, S.11]. Nach dieser ist ein System eine „Menge miteinander in Beziehung stehender Elemente, die in einem bestimmten Zusammenhang als Ganzes gesehen und als von ihrer Umgebung abgegrenzt betrachtet werden." Diese Arbeit betrachtet eine konkrete Softwareinstallation mit seiner aktuellen Version und seiner Konfiguration als ein System. Softwareinstallationen mit verschiedenen Versionen oder verschiedenen Konfigurationen sind somit als unterschiedliche Systeme zu betrachten.

Beim Datenaustausch unterscheidet diese Arbeit zwischen *homogenen* und *heterogenen* Systemen. Der Begriff „*homogene Systeme*" beschreibt Softwaresysteme, die sich so ähnlich sind, dass beim Datenaustausch zwischen den Systemen nur der Datenversand zu beachten ist und die Interpretation und Weiterverarbeitung der Daten ohne Anpassungen möglich ist. In Gegensatz dazu sind zwei Systeme *heterogen*, wenn sie sich so weit unterscheiden, dass eine direkte Interpretation und Weiterverarbeitung der Daten vom Sender im Empfänger nicht möglich ist. Es kann beispielsweise passieren, dass zwei Systeme eines Herstellers jedoch unterschiedlicher Version heterogen sind, da der Austausch von Daten ohne Anpassung nicht möglich ist [BDP08]. Die Gründe für die Heterogenität sind vielfältig und reichen von der Notwendigkeit zur Realisierung spezieller Funktionalitäten bis zur Systempolitik der Hersteller [GA90, Sch01, BDP08]. Zum Austausch der Daten sind spezielle Lösungsstrategien zur Überwindung der Heterogenität notwendig. Diese Lösungsstrategien sind das Thema dieser Arbeit.

Grabowski et al. [GAG86] definierten 1986 eine Schnittstelle als „[...] ein System von Bedingungen, Regeln und Vereinbarungen, das den Informationsaustausch zweier miteinander kommunizierender Systeme oder Systemkomponenten festlegt." Eine Schnittstelle dient der Verbindung von Systemen zum Datenaustausch. Dabei verändert sie die Systeme selbst nicht [VWB+09, S.416]. In der Informatik wird zwischen *Hardware-* und *Softwareschnittstellen* unterschieden [VWB+09, S.416][GA90, S.26][And93, S.40]. Softwareschnittstellen verbinden Softwarekomponenten oder verschiedene Softwaresysteme miteinander [GA90, S.26][And93, S.40]. Analog dazu agieren Hardwarekomponenten, indem sie die Verbindung von Hardwarekomponenten oder -systemen festlegen [GA90, S.26][And93, S.40]. Eine Sonderrolle bei den Schnittstellen nimmt die *Benutzungsschnittstelle* ein, da sie nicht zwischen Softwaresystemen oder Softwarekomponenten wirkt, sondern zwischen Benutzer und Softwaresystem. Vajna et al. [VWB+09, S.502] und Anderl [And93, S.55] ordnen die Benutzungsschnittstelle den Softwareschnittstellen zu. Für diese Arbeit ist sie,

durch den Fokus auf den Datenaustausch zwischen heterogenen Softwaresystemen, nicht näher von Bedeutung. Aus diesem Grund werden auch die Hardwareschnittstellen nicht näher betrachtet. Daher ist im weiteren Verlauf dieser Arbeit unter dem Begriff „Schnittstelle“ stets eine Softwareschnittstelle zu verstehen.

In diesem Abschnitt wurden Begriffe zum grundlegenden Verständnis der gesamten Arbeit erläutert. Der folgende Abschnitt beschäftigt sich mit Klassifikationen von Lösungsstrategien für den systemübergreifenden Datenaustausch zwischen heterogenen Systemen.

2.2 Klassifikationen der Lösungsstrategien für den Datenaustausch

Die Einteilung der Strategien erfolgt nach unterschiedlichen Gesichtspunkten, wobei es zwischen den verschiedenen Einteilungen Abhängigkeiten und Zusammenhänge gibt. Diese werden gemeinsam mit der Einteilung näher erläutert.

Köppen et al. [KVGS11] teilen die Lösungsstrategien grundsätzlich nach ihrem Grobkonzept in *Integrationslösungen* und *Kopplungslösungen* ein. Die Integration ist definiert als „eine auf Dauer ausgelegte enge Verbindung mehrerer Systeme, die von einem Benutzer als Einheit empfunden werden [VWB+09, S. 502].“ Im Gegensatz dazu ist eine Kopplung „eine jederzeit trennbare Verbindung mehrerer Systeme, die jederzeit als eigenständig identifiziert werden können“ [VWB+09, S. 502].

Köppen et al. verwenden statt des Begriffes „Kopplungslösungen“ den Begriff „Schnittstellenlösungen“. Dieser ist jedoch nicht ganz eindeutig, da nach der Definition auf der vorherigen Seite eine Schnittstelle auch den Datenaustausch zwischen Systemkomponenten festlegt und bei einer Integration die einzelnen Systeme zu Systemkomponenten des integrierten Gesamtsystems werden. Zur Unterscheidung von Softwareschnittstellen nach den Verbindungspartnern verwenden Vajna et al. [VWB+09, S.416], Anderl [And93, S.52f] und Grabowski und Anderl [GA90, S.27] die Begriffe „*intern*“ und „*extern*“. Interne Schnittstellen realisieren die Verbindung zwischen Softwarekomponenten innerhalb eines Systems und externe Schnittstellen die Verbindung verschiedener eigenständiger Systeme. Diese Unterteilung ist jedoch nicht immer bei jeder Funktionseinheit für alle Systeme eindeutig möglich [SK97, S.335]. Je nach individuellem Systemaufbau und nach der Definition der individuellen Systemgrenzen ist die Zuordnung verschieden, da für diese Einteilung die Systemzugehörigkeit entscheidend ist. Eine Funktionseinheit, zum Beispiel die grafische Ausgabe, ist somit entweder eine Systemkomponente, die über eine interne Schnittstelle verbunden ist, oder ein eigenständiges System, das über eine externe Schnittstelle verbunden ist. Der Datenaustausch bei einer Kopplung von Systemen wird somit über externe Schnittstellen der Systeme realisiert. Entscheidend dafür ist, ob eine Funktionseinheit ein nach außen sichtbar eigenständiges Softwaresystem ist.

Eine Realisierungsmöglichkeit für den Datenaustausch zwischen den Systemen bei einer Integration ist die Nutzung einer *gemeinsamen Datenbasis*, in der alle Systeme die Daten in einem gemeinsamen Format ablegen [VWB+09, Mer09, GA90, And93].

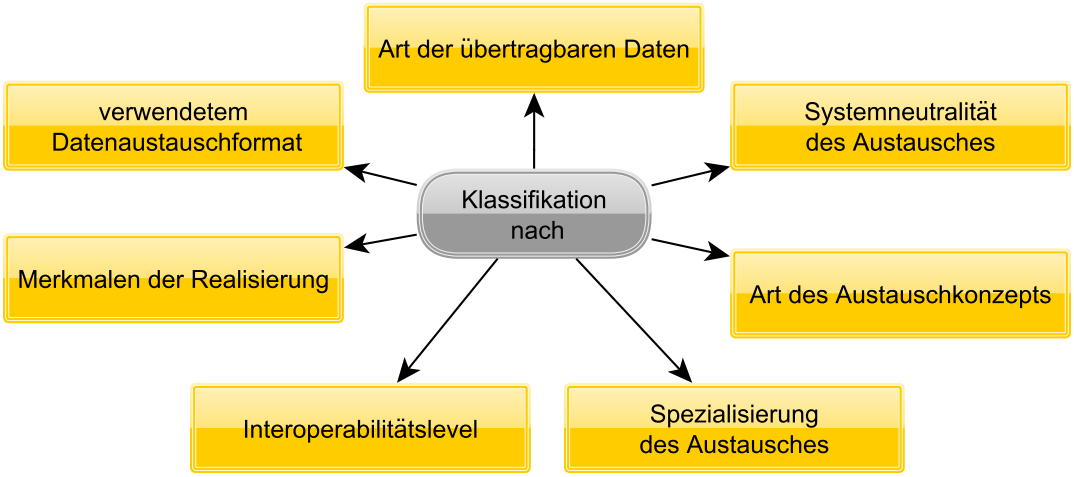

Abbildung 2.1: Übersicht der vorgestellten Klassifikationsschemata für Kopplungslösungen

Der Vorteil bei der Nutzung einer gemeinsamen Datenbasis ist die Verhinderung von Datenredundanz und den daraus resultierenden Problemen, da alle Systeme ihre Daten in der gemeinsamen Datenbasis und nicht in eigenen Datenbasen speichern. Somit entsteht keine Kopie von Daten. Drath et al. [DFB11] und Stekolschik [Ste07, S1ff] beschreiben, dass in der Praxis allerdings nur homogenen Systemlandschaften, bei der alle Softwaresysteme von einem Hersteller stammen, diesen Ansatz realisieren. Dies bedeutet für die Anwender jedoch eine Abhängigkeit vom Hersteller der Systemlandschaft. Die Integration heterogener Systeme über eine gemeinsame Datenhaltung konnte sich nicht durchsetzten [DFB11][Ste07, S1ff][Sch01, S.9]. Gründe dafür sind die fehlende Akzeptanz der Software-Hersteller [DFB11][Ste07, S1ff][Sch01, S.9] und der Anwender [DFB11] sowie die fehlende Einigung auf einheitliche Techniken und Methoden der Realisierung [DFB11][Sch01, S.9].

Vornholt et al. [VGL10] unterscheiden zwischen drei verschieden Architekturansätzen zur Integration. Bei der *Inclusion* wird ein System vollständig in ein anderes integriert. Dieser Ansatz beinhaltet sowohl die Daten wie auch die Funktionen und die Benutzungsoberfläche. Damit werden die Systeme vollständig zu Systemkomponenten des Gesamtsystems. Damit einher geht die gemeinsame Verwaltung der Daten in einer Datenbasis. Die beiden anderen Ansätze sind Konzepte zur automatischen Verbindung über Kopplungslösungen. Dies sind der Ansatz mit *„common objects“* und der Ansatz mit *„managed data“*. Die Integration mit „common objects“ arbeitet mit einer Datenbasis, in der alle Daten zu einem integrierten Modell verbunden werden. Die einzelnen Systeme verwenden jedoch weiterhin ihre eigenen Datenbasen. Dies bedeutet eine redundante Datenhaltung. Für den Austausch erfolgt die Konvertierung der Daten aus dem Format der gemeinsamen Datenbank in das lokale Format des Systems. Die „managed data“-Ansatz ist ähnlich aufgebaut. Eine Datenbasis verwaltet die Daten in den verschiedenen Formaten gemeinsam mit den Abhängigkeiten der Daten untereinander. Der Austausch erfolgt ebenfalls durch Konvertierung.

Auch Mertens [Mer09, S.1] beschreibt die automatische Verbindung über Kopplungslösungen als einfachste Form der Integration. Bei einer automatischen Kopplung bleiben die Systeme jedoch unabhängig und die Verbindung kann prinzipiell jederzeit getrennt werden. Von diesem systembezogenen Blickwinkel aus gehört die automatische Kopplung zu den Kopplungslösungen. Jedoch von Datenseite aus gesehen, ist auch eine Zuordnung zu den Integrationslösungen möglich, da die Daten der Systeme aufgrund des automatischen Austausches eng verbunden sind und eine Einheit bilden. Daher wird die automatische Kopplung in dieser Arbeit beiden grundlegenden Klassen von Lösungsstrategien zugeordnet.

Diese Arbeit beschäftigt sich im weiteren Verlauf mit verschiedenen Kopplungslösungen für den Datenaustausch im Ingenieurwesen. Reine Integrationslösungen werden nicht näher betrachtet, da sie sich in der Praxis nicht durchsetzten konnten.

Es gibt eine Vielzahl verschiedener Kopplungslösungen für den Datenaustausch im Ingenieurwesen [Sch01, S.15]. Eine Klassifikation dieser Lösungen ist anhand verschiedener Kriterien möglich. Abbildung 2.1 zeigt die Klassifikationsschemata, die im Folgenden näher vorgestellt werden.

Klassifikation nach Art der übertragbaren Daten

Eine Klassifikationsmöglichkeit von Datenaustauschstrategien ist die Unterteilung nach der Art der übertragbaren Daten. Austauschstrategien sind dabei in der Regel nicht ausschließlich auf eine Art von Daten beschränkt. Dennoch sind Unterschiede bezüglich des Inhalts bei den übertragbaren Daten zu beobachten.

Grabowski und Anderl [GA90] beschreiben eine Klassifizierung von Daten *nach produktionstechnischen Aspekten* in produktbezogene Daten, prozessbezogene Daten und auftragsbezogene Daten.

Zur Gruppe der *produktbezogenen Daten* gehören alle Daten, die das Produkt betreffen. Dies sind insbesondere Daten zur Gestalt sowie zu technischen, funktionalen und organisatorischen Aspekten des Produktes. In diesem Zusammenhang findet auch der Begriff „*Produktmodell*" Verwendung. Ein Produktmodell ist ein Modell, das ein Produkt im Rechner darstellt. Ziel ist es, ein Produkt vollständig mit allen Eigenschaften über den gesamten Produktlebenszyklus darzustellen [VWB+09][SK97]. Vajna et al. [VWB+09] und Spur und Krause [SK97] verwenden für solche vollständigen Darstellungen auch die Bezeichnung „integriertes Produktmodell". Der Abschnitt 2.3 beschäftigt sich näher mit verschiedenen Stufen von Produktmodellen in der virtuellen Produktentstehung.

Im Gegensatz zu den produktbezogenen Daten steht bei den *prozessbezogenen Daten* nicht das Produkt im Mittelpunkt der Betrachtungen, sondern der Produktionsprozess. Ein wesentliches Merkmal ist der direkte oder indirekte Zeitbezug. Beispiele für prozessbezogene Daten sind Montagepläne, NC-Programme und Anweisungen für Industrieroboter. Mit Hilfe der prozessbezogenen Daten werden *Prozessmodelle* erstellt. Prozessmodelle beschreiben in diesem Kontext Vorgehensweisen beziehungsweise Arbeitsprozesse bei der Entwicklung und Produktion [VWB+09]. Modelle von Werkzeugen, die für Simulationen von Fertigungsvorgängen genutzt werden, beschreiben einerseits Werkzeuge als Teil des Fertigungsprozesses und andererseits

Werkzeuge als eigenständige Produkte. Sie zählen daher sowohl zu den prozessbezogenen Daten als auch zu den produktbezogenen Daten.

Auftragsbezogene Daten beinhalten Daten zu Terminen, Mengen, Kapazitäten, Kosten und zur Qualität eines konkreten Auftrages [GA90, And93]. Diese Daten werden zum Aufbau verschiedener, auftragsbezogener Modelle verwendet. Ein Beispiel für auftragsbezogene Modelle sind Kostenmodelle.

Eine andere Möglichkeit der Unterscheidung von Daten ist *nach dem Geometriegehalt* der Daten [Kal06]. *Geometriedaten* beschreiben die Geometrie zum Beispiel von Produkten oder Werkzeugen und dienen zum Aufbau des Geometriemodelles. Der Inhalt und die Form der Daten hängen dabei von der Art des Geometriemodells ab. Abbildung 2.2 zeigt die Einteilung der Geometriemodelle. Geometriemodelle unterscheiden sich nach der Dimension der beschriebenen Geometrie in 2D-, 2,5D- und 3D- Modelle [VWB+09, SK97, VSG+11, BBD+03]. 3D-Modelle werden nach der mathematischen Repräsentationsform unterteilt in Punktmodelle [GB08, Ten02], Kanten- beziehungsweise Drahtmodelle [VWB+09, Ten02, SK97], Flächenmodelle [GB08, VWB+09, Ten02, SK97], Volumenmodelle [GB08, Ten02, VWB+09, SK97], Facettenmodelle [GB08, Ten02, SK97] und Voxel- beziehungsweise Zellenmodelle [VWB+09, Ten02, SK97]. Bei Punktmodellen erfolgt die Beschreibung über Körperpunkte. Dabei ist die Darstellung rein über Eckpunkte oder durch eine Punktewolke möglich. Der Zusammenhang der Punkte ist in dieser Beschreibung nicht enthalten. Ein Kanten- oder Drahtmodell beschreibt 3D-Geometrien über die Außenkanten des Modells. Der Informationsgehalt ist hier größer als beim Punktmodell, da der Zusammenhang der Eckpunkte beschrieben wird. Körperflächen sind jedoch nicht enthalten. Dadurch ist diese Darstellung nicht eindeutig. Körperflächen werden beim Flächenmodell beschrieben. Die Flächen bilden jedoch in dieser Art von Geometriemodell noch kein Volumen und haben dadurch einen geringeren Informationsgehalt als Volumenmodelle. Es fehlen Volumeninformationen, die beschreiben welcher Teil des Raumes Teil des Körpers ist und welcher nicht. Dies beschreiben Volumenmodelle. Sie bilden die Geometrie eines Körpers als Volumenobjekt mit spezifischen Eigenschaften ab. Dabei gibt es verschiedene mathematische Ansätze zur Beschreibung. Unterschieden wird zwischen generativen Modellen, akkumulativen Modellen und hybriden Modellen [VWB+09, SK97]. Generative Modelle beschreiben das Volumen analytisch über den Erzeugungsweg. Wichtigste Vertreter dieser Klasse sind die Constructive Solid Geometry (CSG)-Modelle. Sie beschreiben das Volumen über boolesche Verknüpfungen von Grundelementen. Im Gegensatz dazu beschreiben akkumulative Modelle die Geometrie deskriptiv. Die wichtigsten Vertreter hier sind die Boundary Representation (B-Rep)-Modelle. Die Volumenbeschreibung erfolgt hier über die geschlossenen Außenflächen und deren topologischen Zusammenhang. Hybride Modelle stellen eine Kombination beider Vorgehensweisen dar. Facettenmodelle sind spezielle B-Rep-Modelle, welche die Geometrie eines Körpers über Polygone approximieren. Voxel- oder Zellenmodelle beschreiben ein Volumen durch kleine Basiskörper, die jedoch nicht in Beziehung zueinander stehen. Das Prinzip ähnelt dabei dem eines Pixels. Die Menge der Basiskörper bildet das Geometriemodell. Für detaillierte Informationen zu den mathematischen Konzepten ist die einschlägige Literatur (zum Beispiel [VWB+09, SK97]) zu dieser Thematik zu verwenden.

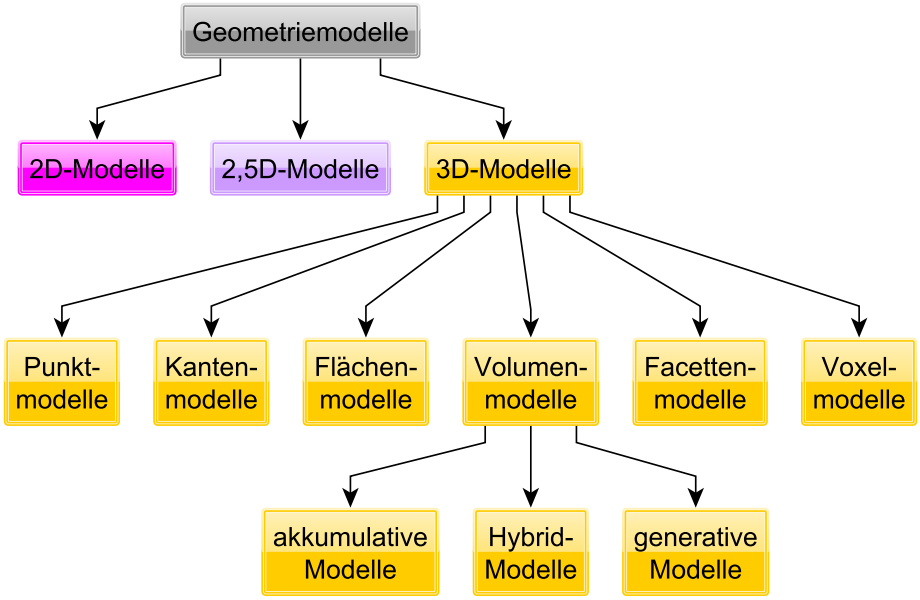

Abbildung 2.2: Arten von Geometriemodellen nach mathematischer Repräsentation

Alle Daten, die keine Geometrie beschreiben, gehören zu den *nicht-geometrischen Daten* [Kal06]. Diese Gruppe von Daten ist sehr groß [Kal06]. Sie enthält beispielsweise Preise, Materialien, Arbeitspläne, kinematische Beziehungen, Versionsverweise, Abhängigkeiten, Beziehungen und Parameter [Sch01, Kal06]. Auch Parameter, die geometriesteuernd wirksam sind, zählen zu dieser Gruppe von Daten [Sch01], da sie selbst keine Geometrie beschreiben, sondern wann Änderungen der Geometrie erfolgen. Besondere nicht-geometrische Daten sind Geometrieattribute oder darstellungsbezogene Daten [Sch01, Kal06]. Diese enthalten beispielsweise Angaben zu Toleranzen, Farben, Materialien und Symbolen [Sch01, Kal06]. Sie beschreiben jedoch selbst ebenfalls keine Geometrieelemente. Kalusche und Marcus [Kal06] unterscheiden zusätzlich noch organisations- und planungsbezogene Daten. Diese gehören nach der vorherigen Definition zu den nicht-geometrischen Daten und beinhalten beispielsweise Zugriffsrechte, Modellidentifikationen und Steuerungsinformationen zur Planung und Ausführung.

Formfeature enthalten geometrische Daten und optional zusätzliche nicht-geometrische Daten. Sie werden daher eher den geometrischen Daten zugeordnet [Sch01].

Mit Hilfe der vorgestellten Klassifikationen nach der Art der übertragbaren Daten ist es möglich, für ein Anwendungsszenario prinzipiell verwendbare Lösungsstrategien zu identifizieren oder die zu übertragenden Daten für neue Lösungsansätze festzulegen. Aussagen über weitere Aspekte des Austausches sind jedoch nicht möglich. Dazu sind weitere Klassifikationsschemata notwendig.

Klassifikation nach Systemneutralität des Austausches

Grundlegend unterteilen sich Datenaustauschstrategien nach der Systemneutralität in *systemspezifische* Austauschstrategien und *systemneutrale* Austauschstrategien [MPKS11, Sto11, GA90, Mac91, SK97]. Grabowski und Anderl verwen-

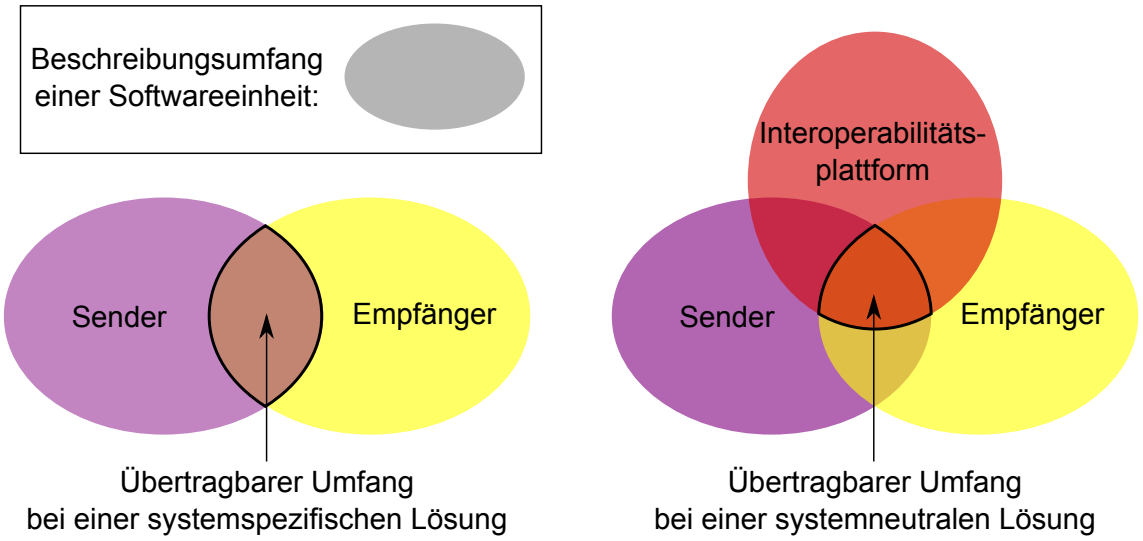

Abbildung 2.3: Übertragbarer Umfang im Vergleich nach Vornholt et al.[VSG+11]

den in [GA90] statt der Begriffe „systemspezifisch“ und „systemneutral“ die Begriffe „ungenormt“ und „genormt“. Sie gehen dabei allgemein von Schnittstellen zum Datenaustausch aus und beschränken sich nicht auf bestimmte Lösungskonzepte. Diese Begriffe führen jedoch zu der Annahme, dass alle neutralen Schnittstellen einen Normungsprozess durchlaufen haben. Dies ist jedoch nicht zwingend notwendig [FLL+11, VSG+11, Fac05, SVG11]. Aus diesem Grund werden in dieser Arbeit die Bezeichnungen „systemneutral“ und „systemspezifisch“ verwendet.

Systemspezifische Austauschstrategien koppeln immer genau zwei Softwaresysteme direkt miteinander [VGL10, MPKS11]. Das bedeutet, bei einer unidirektionalen Kopplung müssen für einen bidirektionalen Datenaustausch zwei Schnittstellen realisiert werden. Bei n Systemen sind somit $n * (n - 1)$ Schnittstellen notwendig [GA90, VWB+09, Sto11], was eine Komplexität von $O(n^2)$ bedeutet. Der Nachteil der systemspezifischen Lösungen ist der mit der Anzahl an Systemen steigende Aufwand für die Entwicklung und Pflege der Kopplungen [Sto11]. Der Vorteil von systemspezifischen Lösungsansätzen ist, dass ein möglichst großer gemeinsamer Beschreibungsraum übertragbar ist [VSG+11, VWB+09, Sto11]. Abbildung 2.3 verdeutlicht dies grafisch.

Systemneutrale Lösungsstrategien nutzen eine neutrale Zwischenstufe, die Mory et al. [MPKS11] als Interoperabilitätsplattform bezeichnen, zum Datenaustausch zwischen den Systemen [Sto11, VGL10]. Von jedem System erfolgt erst der Austausch in die neutrale Zwischenstufe und von dieser dann der Austausch in das Empfängersystem. Pro System sind dadurch zwei unidirektionale Kopplungen von und zur neutralen Zwischenstufe notwendig [Sto11]. Das bedeutet, bei n Systemen sind $2 * n$ Schnittstellen notwendig [GA90, VWB+09]. Damit ist die Komplexität $O(n)$. Der Vorteil systemneutraler Lösungsstrategien ist daher, dass der Aufwand für die Erstellung und die Pflege der Schnittstellen ab einer bestimmten Zahl an zu koppelnden Systemen geringer ist als bei systemspezifischen Lösungen. Ein Nachteil systemneutraler Lösungen ist der in der Regel geringere Umfang der übertragbaren

Daten, da nur Daten übertragbar sind, die sowohl in den Systemen wie auch in der neutralen Zwischenstufe abbildbar sind [VSG+11, VWB+09, Sto11]. Dies ist in Abbildung 2.3 am Beispiel des Austausches zwischen zwei Systemen dargestellt. Ein anderes Problem ist das Finden eines geeigneten, neutralen Ansatzes, auf dem die Interoperabilitätsplattform basiert [MPKS11].

Avgoustinov hat in [Avg97] weitere Untersuchungen zum Aufwand durchgeführt. Er untersucht den tatsächlichen Aufwand bei der Nutzung von Direktkonvertern und bei der Nutzung neutraler Formate. Direktkonverter gehören zu den systemspezifischen Austauschstrategien und neutrale Formate zu den systemneutralen Austauschstrategien. Dabei bezieht der Autor unter anderem den Aufwand für die Pflege des neutralen Formates mit ein. Das Ergebnis dieser Untersuchung ist, dass die Grenze für die Anzahl an Systemen, bei denen der Aufwand mit einem neutralen Format geringer ist als mit Direktkonvertern, sich in Abhängigkeit von bestimmten Parametern nach oben verschiebt. Ohne diese Betrachtung liegt die Grenze bei 3 Systemen. Bei Avgoustinov Betrachtung ist diese Zahl von unbekannten Parametern abhängig. Der Autor bestimmt daher keine genaue Grenze, sondern beschreibt verschiedene Formeln zur Bestimmung des Aufwandes. Für konkrete Details zu diesen Formeln wird hier auf die Arbeit von Avgoustinov [Avg97] verwiesen.

Zu beachten ist jedoch, dass bei standardisierten neutralen Austauschformaten der Aufwand für die Pflege des Formates nicht unbedingt beim nutzenden Unternehmen liegt, sondern von Standardisierungsstellen getragen wird [Sch01]. Ein besonderer Entwicklungsaufwand entsteht aber bei Veränderung des neutralen Formates, da in diesem Fall alle Prozessoren angepasst werden müssen [Sto11]. Dies gilt jedoch ebenso bei Änderung eines Systems für die systemspezifischen Lösungen. Diese Ausführungen zum Aufwand können allgemein auf alle systemspezifischen und systemneutralen Austauschstrategien übertragen werden. Zusammenfassend bedeutet dies, dass systemneutrale Lösungen im Gegensatz zu systemspezifischen Lösungen weiterhin ab einer bestimmten Zahl von Systemen Vorteile bezüglich des Aufwandes bringen. Diese Grenze ist jedoch abhängig von verschiedenen Parametern und kann daher hier nicht als pauschaler Wert angegeben werden.

Systemspezifische und systemneutrale Lösungsstrategien haben verschiedene Vor- und Nachteile. Daher muss im konkreten Anwendungsszenario entschieden werden, welche Klasse von Lösungsstrategien an dieser Stelle die bessere ist. Bei der Entwicklung neuer Lösungsansätze muss die Entscheidung für die Systemneutralität des Ansatzes grundlegend getroffen werden.

Klassifikation nach Art der Austauschkonzepte

Eine andere Möglichkeit der Klassifizierung von Lösungsstrategien basiert auf dem Konzept des Austausches. Schumann [Sch01] unterscheidet zwischen dem *übersetzenden Austausch*, dem *generierenden Austausch* und dem *einbindenden Austausch*. Beim übersetzenden und beim einbindenden Austausch differenziert er zusätzlich zwischen verschiedenen Realisierungsmöglichkeiten der Übersetzung. Abbildung 2.4 zeigt das Klassifikationsschema in der Übersicht.

Das Konzept des **übersetzenden Datenaustausches** basiert auf dem Prinzip der Konvertierung der Daten vom Format des Senders in das Format des Empfängers.

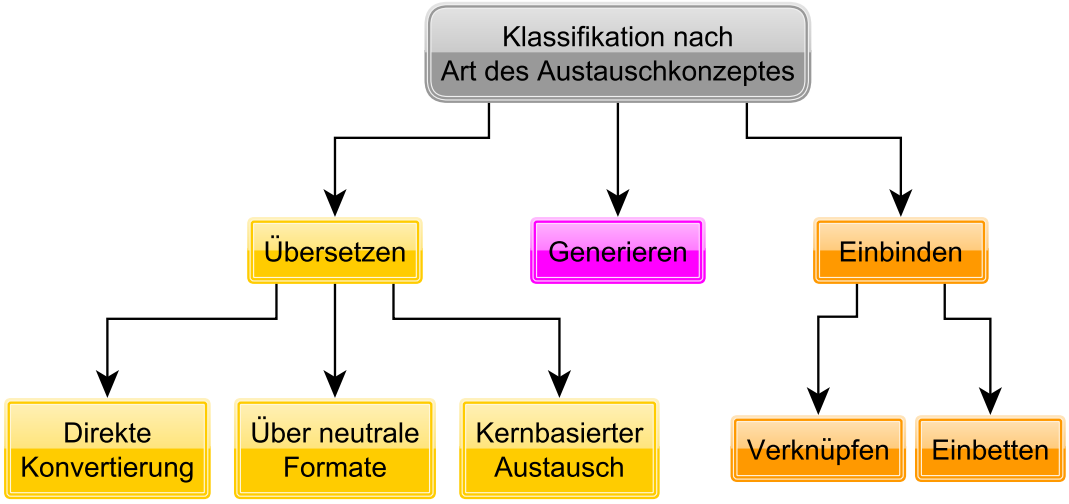

Abbildung 2.4: Klassifikation nach der Art des Austauschkonzeptes

Eine Konvertierung ist die Übersetzung von einer Sprache in eine gleichwertige Sprache. Ein Format ist eine spezielle Art von Sprache [Avg97]. Eine Konvertierung erfolgt mittels eines Konverters. In der Informatik beschäftigen sich neben dem Begriff „Konvertieren“ auch die Begriffe „Interpretieren“ und „Kompilieren“ mit dem Übersetzen von einer Sprache in eine andere. Bei diesen Arten der Übersetzung sind beide Sprachen jedoch nicht gleichwertig. Das bedeutet, beide Sprachen sind nicht auf dem gleichen Abstraktionsniveau. Im Zusammenhang mit dem Datenaustausch zwischen Systemen wird daher nur der Begriff „Konvertieren“ für die Übersetzung verwendet.

Vajna et al. [VWB+09] bezeichnen das Konzept der Übersetzung als Datenaustausch auf der Basis sequenzieller Daten. Grabowski und Anderl [GA90] nennen es auf Basis von Datenstrukturen. Stoye [Sto11] verwendet den Begriff „dateibasiert“, da er davon ausgeht, dass die zu übertragenden Daten in Form von Dateien vorliegen beziehungsweise in Dateien abspeicherbar sind. Diese Bezeichnungen unterstreichen den deskriptiven Charakter des Austauschobjektes, da beim übersetzenden Datenaustausch Datenstrukturausprägungen übertragen werden und keine Erzeugerlogik [GA90, Sch01].

Nachteile des übersetzenden Datenaustausches sind die redundante Datenhaltung, da immer eine Kopie der Daten im Empfängersystem erzeugt wird, und die Größe der Datenmengen, die durch den deskriptiven Ansatz entsteht [Sch01].

Schumann [Sch01] unterteilt die übersetzenden Datenaustauschstrategien weiter in die direkte Konvertierung, den Austausch über Standardschnittstellen und den kernbasierten Austausch.

Die *direkte Konvertierung* realisiert die direkte Kopplung von Sender und Empfänger durch die Übersetzung vom systemspezifischen Format des Senders in das systemspezifische Format des Empfängers [Sch01, Sto11, VWB+09, Haa95]. Der sogenannte Direktkonverter führt die Übersetzung durch [Sto11, VWB+09]. Dieser ist entweder

ein internes Modul eines der Systeme [And93, Ger03, VGL10] oder ein externes Programm [Mac91, VGL10, Haa95]. Andere Bezeichnungen dafür sind „programmierte Schnittstelle“ [Jäg91, Haa95] oder „Kopplungsschnittstelle“ [And93].

Die direkte Konvertierung gehört zu den systemspezifischen Austauschstrategien. Damit gelten alle Vor- und Nachteile der systemspezifischen Austauschstrategien auch für die direkte Konvertierung. Das bedeutet, der Nachteil der direkten Konvertierung ist der Aufwand, der für die Erstellung und Pflege der Direktkonverter notwendig ist. Die Vorteile bei der Verwendung von Direktkonvertern sind der Umfang der übertragbaren Daten und die geringe Transferzeit [Sch01]. Die geringe Transferzeit ist besonders bei großen zu übertragenden Datenmengen von Bedeutung und wird maßgeblich durch die Anzahl der Konvertierungen und den zeitlichen Aufwand der einzelnen Konvertierungen bestimmt [Sch01].

Insgesamt ist bei der direkten Konvertierung nur eine Konvertierung notwendig. Anders ist dies bei der Verwendung neutraler Formate, bei welcher zwei Konvertierungen notwendig sind. Der Austausch über *neutrale Formate* stellt eine systemneutrale Lösungsstrategie dar, bei der ein neutrales Format als neutrale Zwischenstufe genutzt wird [VWB+09, VGL10, Sto11, Haa95]. Die Realisierung des Datenaustausches erfolgt über zwei Konverter pro System [VWB+09, Sch01, Sto11]. Eine andere Bezeichnung für diese Konverter ist „Prozessoren“.

Anderl [And93] verwendet für die Kopplung über neutrale Formate auch die Bezeichnungen „Austausch über eine Datenschnittstelle“. Schumann [Sch01] nutzt die Bezeichnung „Austausch über Standardschnittstellen“. Der Begriff „Standardschnittstelle“ führt analog zum Begriff „genormt“ zu der Annahme, dass neutrale Formate immer formal standardisiert sind, also durch ein Standardisierungsgremium angenommen wurden [FLL+11]. Dies ist jedoch keine Voraussetzung für die Verwendung neutraler Formate, obwohl dies auf einige Formate zutrifft [WBTM00].

Der große Vorteil neutraler Formate als systemneutrale Austauschstrategie ist die Zahl der benötigten Prozessoren. Nach Friedewald et al. [FLL+11] sind neben dem geringeren allgemeinen Aufwand bei der Verwendung von neutralen Formaten auch geringere Lizenzkosten und ein besser beherrschbares Qualitätsmanagement für den Austausch zu erwarten. Die Nachteile bei der Verwendung neutraler Formate sind der Umfang der übertragbaren Daten, die Auswahl des geeignetsten Formates für die jeweilige Anwendung [Haa95] und die benötigte Konvertierungszeit bei sehr großen Datenmengen [Sch01]. Wenn die Realisierung der Prozessoren bei den Systemherstellern liegt, findet diese unter Umständen nicht einheitlich statt. Damit ist in diesem Fall kein reibungsloser Austausch möglich [Sch01].

Ein spezielles Konzept, welches die neutralen Formate mit den Direktkonvertern verbindet, ist das *Bypass-Konzept* [VWB+09, S.464][Sch01]. Haasis [Haa95] verwendet dafür den Begriff „Datei-Adaption“. Dabei erfolgt der Austausch der Daten erst über das neutrale Format. Danach werden fehlende Daten über eine direkte Konvertierung übertragen [VWB+09, S.464][Haa95]. Dies soll den Umfang der übertragbaren Daten auf das Niveau des Direktkonverters erhöhen, ohne den gleichen Aufwand, wie bei der Verwendung von Direktkonvertern [VWB+09, S.464]. Vajna et al. [VWB+09, S.464] beschreiben den Einsatz des Bypass-Konzepts hauptsächlich zur Übertragung von Altdatenbeständen. Der Einsatz ist jedoch auch beim Datenaustausch zwischen

heterogenen Systemen allgemein möglich. Zum Beispiel schlagen die Autoren das Konzept zur Übertragung von Geometrie und Parametrik vor, wobei ein neutrales Format die Geometrie überträgt und ein Direktkonverter die Parametrik.

Schumann [Sch01] beschreibt als dritte Art von übersetzenden Datenaustauschstrategien den *kernbasierten Datenaustausch*. Die Grundlage des kernbasierten Austausches ist der Modellierkern eines CAx-Systems. CAx steht dabei für verschiedene Softwaresysteme zur Unterstützung von Ingenieuraufgaben [Sch01, VWB+09]. Der Modellierkern ist der grundlegende Baustein eines CAx-Systems und stellt den Großteil der für die Modellierung benötigten Funktionen bereit. Dadurch ist der Datenaustausch zwischen Systemen mit gleichem Modellierkern, bis auf systemspezifische Aspekte, verlustfrei möglich. Spezielle kernspezifische Formate erleichtern den Austausch zusätzlich [Sch01, Ger03].

Vajna et al. [VWB+09, S.16] geben an, dass fast 80% aller CAx-Systeme einen von drei Modellierkernen oder einen auf diesen basierenden Modellierkern verwenden. Diese drei Kerne sind der Parasolid-Kern, der ACIS-Kern und der Pro/Engineer-Kern. Dieser Sachverhalt ermöglicht den Austausch über die Kerne mit einem geringeren Aufwand als über Direktschnittstellen zwischen kompletten Systemen oder über neutrale Formate [Ger03][VWB+09, S.16].

Systemspezifische Daten, die nicht zum Kern gehören, sind auf kernbasierte Weise nicht übertragbar. Dies betrifft beispielsweise Parametrik- und Featuredaten. Auch der Datenaustausch zwischen Systemen, die keinen Modellierkern in dieser Form verwenden, ist damit nicht möglich. Schumann [Sch01] beschränkt sich in seinen Ausführungen auf Geometriemodelle. Dennoch ist das Prinzip des kernbasierten Austausches auch für andere Szenarien denkbar. Voraussetzung ist jedoch das Vorhandensein von Funktionseinheiten ähnlich den Modellierkernen bei CAx-Systemen.

Ein alternatives Konzept zum übersetzenden Datenaustausch ist der **generierende Datenaustausch**. Die Grundidee dieses Ansatzes ist der Austausch von Erzeugungslogik statt fertiger Datenstrukturen [GA90, Sch01]. Andere Bezeichnungen für die Erzeugungslogik sind „prozedurale Daten“ [VWB+09] oder „implizite Daten“ [KB97]. Die Beschreibung der Erzeugungslogik erfolgt in Form von Programmstrukturen mit Prozeduren [GA90, Sch01]. Schumann [Sch01] unterscheidet bei den Prozeduren zwischen neutralen und systemspezifischen beziehungsweise systemnahen Prozeduren. Neutrale Prozeduren sind durch Standards festgelegt oder allgemeine Bestandteile der für das System verwendeten Programmiersprache. Systemspezifische Prozeduren sind Teil des Systems. Daher kann die Übertragung zum Teil schwierig sein. Es treten ähnliche Schwierigkeiten wie beim übersetzenden Austausch auf. Die Konvertierung systemspezifischer Prozeduren erfolgt analog zum übersetzenden Austausch systemspezifisch oder systemneutral. Ein systemspezifischer Ansatz setzt die Prozeduren des Sendesystems direkt in Prozeduren des Empfängers um. Ein systemneutraler Ansatz nutzt eine neutrale Sprache als Zwischenstufe, in welche die systemspezifischen Prozeduren des Senders übertragen werden und die dann selbst in die systemspezifischen Prozeduren des Empfängers übertragen wird. Ein Beispiel für eine solche neutrale Lösung ist VDA-PS [Haa95, GA90].

Der Import und Export von Programmstrukturen, welche die Erzeugungslogik beschreiben, erfolgt durch Programmierschnittstellen [Sch01, And93]. Eine andere Bezeichnung für Programmierschnittstellen ist „prozedurale Schnittstellen“ [Haa95, Jäg91]. Programmierschnittstellen ermöglichen, je nach Realisierung, den Zugriff auf existierende Daten, deren Änderung und die Erstellung neuer Daten [Haa95]. Sie beinhalten Funktionen systeminterner Programmschnittstellen sowie Kontrollstrukturen [And93]. Diese systeminternen Funktionen sind dabei entweder vollständig oder in eingeschränktem Umfang in der prozeduralen Schnittstelle enthalten [And93]. Schumann [Sch01] nennt die Erstellung von Erzeugungslogik „Ableiten“. Das Ableiten von Erzeugungslogik erfolgt entweder direkt im Sendesystem oder in externen zusätzlichen Programmen. Gleiches gilt für die Generierung.

Die Ableitung von Erzeugungslogik stellt jedoch ein Problem dar, wenn die vorhandenen Programmierschnittstellen eher auf das Manipulieren von Daten ausgelegt sind und nicht auf die Ableitung von Erzeugerlogik. Nach [Nat99] war dies 1999 noch ein Problem. Neuere Studien zu dieser Thematik sind nicht bekannt.

Ein weiteres Problem tritt nach Schumann [Sch01] auf, wenn der Sprachumfang der Schnittstellen nicht ausreicht oder Abweichungen durch die Konvertierung entstehen. Dies führt zu Datenverlusten, die es zu vermeiden gilt. Notwendig für den generierenden Austausch ist, dass die beteiligten Systeme die verwendeten Funktionalitäten besitzen und diese auch auf ähnliche Art und Weise verwenden. Zum Beispiel kann eine Geometrie nur zwischen Systemen generierend übertragen werden, wenn beide Systeme geometrieerzeugende Prozeduren besitzen. Ein Problem ist auch die Handhabung von Objekten und objektorientierten Beschreibungen, da deren Kommunikation zwischen den Elementen schwierig in Prozeduren abbildbar ist.

Der klare Vorteil des generierenden Datenaustausches ist, dass systeminterne Funktionen des Empfängers die Datenstrukturen aufbauen. Bei Produktdaten bedeutet dies, dass sowohl Geometrie als auch Attribute, Manipulationen, Parameter, Referenzen und Bedingungen ohne das Auftreten geometrisch-topologischer Verluste durch unterschiedliche systeminterne Genauigkeiten oder Toleranzen übertragbar sind [Sch01]. Zudem bleiben eine Modellhistorie und verschiedene Gestaltfreiheitsgrade aus Parametern, Regeln und Abhängigkeiten erhalten [Sch01]. Eine Übertragung von Features ist ebenfalls möglich [KB97]. Auch ist bei prozeduralen Daten von einem geringeren Datenumfang als bei deskriptiven Daten auszugehen, was einen schnelleren Datenversand ermöglicht [Sch01]. Relativ transparente Befehlsfolgen ermöglichen Eingriffe und Korrekturen [Sch01]. Damit ist auch eine einfache Realisierung von zusätzlichen Parametrikroutinen möglich [Sch01]. Beispiele für die Verwendung von generierenden Austauschlösungen wurden bei der Bereitstellung von Norm- und Zukaufteildaten beobachtet[Sch01, GA90].

Beim übersetzenden und beim generierenden Datenaustausch ist das Ergebnis, dass die Daten nach Abschluss des Austausches im Format des Empfängers vorliegen. Einen anderen Weg verfolgt der **einbindende Datenaustausch**. Dieses Konzept betrachtet Schumann [Sch01] ebenfalls in der Klassifikation. Eine andere Bezeichnung für den einbindenden Austausch ist „Objektaustausch“, da dieses Konzept Objekte zum Datenaustausch zwischen den Systemen verwendet. Voraussetzung für den einbindenden Datenaustausch ist die Verwendung objektorientierter Techniken und

Normen in den Systemen. *Objekte* sind einfache, zueinander kompatible Einheiten, die bestimmte Eigenschaften und Methoden besitzen und in Relation zueinanderstehen. Die Eigenschaften und Relationen stellen dabei die auszutauschenden Daten dar. Die Definition eines Objektes mit seinen Eigenschaften, Methoden und Relationen ist Teil des Systems, welches das Objekt erzeugt. Diese Definition kann bestimmten Normen und Standards unterliegen. Für das Empfängersystem sind Kenntnisse zu dem verfügbaren Objekt und den Nachrichten notwendig, um mit dem Objekt zu arbeiten [Sen95, S.43]. Bestehen keine einheitlichen Objektdefinitionen, so sind Interpreter zum Angleich notwendig. Übersetzungsprozessoren für Datenformate oder Prozeduren sind jedoch ebenso wenig notwendig wie detaillierte Kenntnisse zum Sendesystem selbst [Sen95, S.43].

Schumann [Sch01] sowie Sendler [Sen95] unterscheiden zwischen zwei Arten der Einbindung, dem Verknüpfen und dem Einbetten. Beim *Verknüpfen* (englisch: Linken) werden keine vollständigen Objekte übertragen, sondern Verweise zur Quelle. Das Empfängersystem greift über den Verweis auf das Objekt zu. Somit entsteht ein bidirektionaler Datenaustausch ohne redundante Datenhaltung. Der Sender überträgt Objekteigenschaften und Aktualisierungen zum Empfänger. Über die Rückkopplung gibt der Empfänger Abfragen von Eigenschaften und Methoden sowie Änderungsanweisungen direkt an den Sender weiter. Somit erfolgen Änderungen direkt an den Originaldaten und der Empfänger erhält alle Aktualisierungen sofort automatisch. Der Nachteil dieser Art von Einbindung ist, dass die Daten nur in Verbindung mit dem Sender verwendbar sind. Beide Systeme müssen gleichzeitig laufen. Anders ist dies bei einer *Einbettung*. Bei dieser Art von Einbindung erhält der Empfänger das Objekt vollständig als Kopie. Somit sind die Daten auch ohne das Sendesystem verwendbar. Änderungen erfolgen jedoch nicht direkt an den Originaldaten. Der Empfänger erhält zudem Aktualisierungen nicht automatisch. Diese Probleme zählen zum Problemkreis der redundanten Datenhaltung, welcher durch die Kopie des Objektes entsteht.

Schumann beschreibt verschiedene Vor- und Nachteile des einbindenden Datenaustausches. Unabhängig von der Art der Einbindung sind Nachteile des einbindenden Datenaustausches das Fehlen weitreichender Standards und unterschiedlicher Realisierungen der Objektorientierung. Unterschiede betreffen dabei folgende Aspekte [Sch01]:

- Objektdefinitionen
- Regelungen für den Objektzugriff
- Beschreibungssprachen und Entwicklungsumgebungen
- Regelungen für Vererbungen und Klassen
- Datenhaltung
- verwendeten Hard- und Softwareplattformen

Ein Vorteil des einbindenden Austausches im Vergleich zum übersetzenden und generierenden Austausch ist nach Schumann die Erhaltung von Originalstrukturen.

Komplexe Informationen zum Aufbau der Daten sind nicht zwingend zu übertragen. Des Weiteren ist die Kommunikation mit verschiedenen Sendern mit gleicher Objektdefinition ohne zusätzlichen Entwicklungsaufwand möglich. Ein zusätzlicher Vorteil bei verknüpfender Einbindung ist, dass eine redundante Datenhaltung vermieden wird und Aktualisierung und Änderungen direkt übertragen werden. Beispiele für einbindende Ansätze zum Datenaustausch sind der OLE-Ansatz und seine Erweiterung CLE4D&M sowie der CLEO-Ansatz und Komponentensystemansätze wie im ANICA-Projekt.

Die Einteilung nach dem verwendeten Konzept zeigt, wie unterschiedlich die verschiedenen Lösungsansätze sein können. Neue Ansätze können nach ihrem angedachten Konzept in diese Klassifikation eingefügt werden und die Auswahl bestehender Ansätze für konkrete Anwendungsszenarien wird erleichtert.

Klassifikation nach Spezialisierung des Datenaustausches

Anderl [And93] gliedert die Datenaustauschlösungen nach der Spezialisierung des Austausches in Verfahren für einen generalisierten Austausch und Verfahren für einen spezialisierten Austausch. Dabei bezieht der Autor sich auf Produktdaten. Die Klassifikation ist jedoch auch auf andere Daten ausweitbar.

Beim *generalisierten Datenaustausch* ist das Ziel, Daten möglichst universell zwischen Systemen auszutauschen. Der Austausch soll dabei unabhängig vom konkreten Inhalt der Daten, von den beteiligten Softwaresystemen, den beteiligten Unternehmen, den Unternehmensabteilungen und -bereichen sowie speziellen anderen Randbedingungen, wie konkreten Produktionstechniken, sein.

Im Gegensatz zum generalisierten Austausch ist das Ziel des *spezialisierten Austausches* die Unterstützung eines konkreten Austausches zwischen Systemen. Dabei ist der spezialisierte Austausch abhängig von bestimmten zweckgebundenen Randbedingungen und der konkreten Anwendung. Auch Abhängigkeiten von bestimmten Systemen, Unternehmen oder Unternehmensbereichen sind möglich. Anderl versteht den spezialisierten Austausch als Abbildung des generalisierten Austausches auf eine konkrete Anwendung.

Die Schwierigkeit bei der Zuordnung von Datenaustauschstrategien in diese Klassifikation ist, dass bezüglich des generalisierten Austauschs unterschiedliche Blickwinkel möglich sind. Konkrete Datenaustauschlösungen sind in der Regel auf bestimmte Daten und bestimmte Anwendungsgebiete beschränkt. Dies spricht für einen spezialisierten Austausch. Jedoch können einige Datenaustauschstrategien Systeme innerhalb ihrer Anwendungsgebiete weitgehend universell koppeln. Dies spricht für eine Zuordnung zu den Verfahren des generalisierten Austausches. Somit ist nach der bisherigen Definition der Klassen die Zuordnung nicht eindeutig möglich, sondern hängt vom Blickwinkel ab. Zu beachten ist jedoch, dass bisher es bisher keine vollkommen universellen Datenaustauschstrategien im Ingenieurwesen gibt und dass der Autor selbst sich auf das Anwendungsgebiet Produktdaten bezieht. Daher werden in dieser Arbeit Datenaustauschformate, die weitgehend universell innerhalb eines Anwendungsgebietes, das über ein konkretes Anwendungsszenario hinausgeht, einsetzbar sind, zu den Verfahren des generalisierten Austausches gezählt. Ein Beispiel für solch ein Verfahren ist das Datenaustauschformat JT. Die Zuordnung zu Verfahren des

spezialisierten Austausches erfolgt in dieser Arbeit somit, wenn die Datenaustauschlösung von den konkreten zu koppelnden Systemen, dem konkreten Anwendungsszenarios oder weiteren spezialisierenden Bedingungen abhängt. Ein Beispiel für solch ein Verfahren ist die Kopplung über Direktkonverter.

Vorteilhaft ist ein Verfahren für einen generalisierten Datenaustausch beispielsweise beim Austausch von Daten zu Norm- und Zukaufteilen. Währenddessen Verfahren für einen spezialisierte Austausch zur Unterstützung von konkreten Prozessketten in einem Unternehmen geeignet sind.

Diese Klassifikation zeigt jedoch auch die Abhängigkeit vom konkreten Anwendungsszenario bei der Auswahl geeigneter Austauschlösungen. Der Vorteil einer generalisierten Lösung ist, dass nur eine Lösung zu realisieren ist. Dabei ist es jedoch nicht immer möglich, alle abteilungsspezifischen Anforderungen vollständig zu berücksichtigen. Anders ist dies, wenn ein Austausch zwischen einer Gruppe von klar bestimmten Systemen zu einem konkreten Zweck gefordert ist. Eine spezialisierte Austauschlösung kann hier spezielle Anforderungen berücksichtigen. Die Einteilung nach der Spezialisierung des Austausches unterstützt die Auswahl an Lösungsmöglichkeiten, indem für konkrete Anwendungsszenarien die Zahl der möglichen Lösungsansätze eingeschränkt werden kann. Zusätzlich ist die Spezialisierung als ein Ausgangspunkt bei der Realisierung neuer Lösungsansätze einsetzbar.

Klassifikation nach den Interoperabilitätsleveln

Eine weitere Möglichkeit der Unterscheidung von Lösungsstrategien ist nach den unterstützten Interoperabilitätsleveln. Manso et al. geben in [MWB09] einen Überblick über die Thematik der Interoperabilität und identifizieren sieben verschiedene, allgemeine Interoperatiblitätslevel, die im Folgenden näher vorgestellt werden.

Manso et al. definieren die *Interoperabilität* als „the ability of different types of computers, networks, operating systems, and applications to exchange and reuse data and information.“ Diese Arbeit beschränkt sich dabei auf die Betrachtung von verschiedenen, heterogenen Softwaresystemen. Die Kopplung von Netzwerken, Computern und Betriebssystemen wird nicht betrachtet. Eine ähnliche Definition verwendet Handschuh [Han11], wobei er von einer möglichst nahtlosen Zusammenarbeit heterogener Systeme spricht. Er geht dabei jedoch nicht näher auf den Begriff „nahtlos“ ein. Der Kontext legt nahe, dass damit ein möglichst verlust- und fehlerfreier Austausch mit möglichst wenig benötigten Benutzerinteraktionen gemeint ist. Diese Aspekte werden in dieser Arbeit als grundlegende Anforderungen an den Datenaustausch behandelt und sind anzustrebende Merkmale der Interoperabilität. Grundlage für die Klassifikation nach den Interoperabilitätsleveln ist die erste Definition. Die von Manso et al. [MWB09] identifizierten, allgemeinen Interoperabilitätslevel sind in Abbildung 2.5 dargestellt.

Thema der *technischen Interoperabilität* ist der Versand von Daten. Die Kommunikation der Systeme betrachten die Autoren unabhängig vom Format der Daten. Wichtig sind dabei die verwendete Kommunikationsinfrastruktur und die verwendeten Kommunikationsprotokolle der Hard- und Software. Damit beschreibt die technische Interoperabilität den Datenversand. Dieser wird hier als gegeben angenommen. Daher wird die technische Interoperabilität im Folgenden nicht näher behandelt.

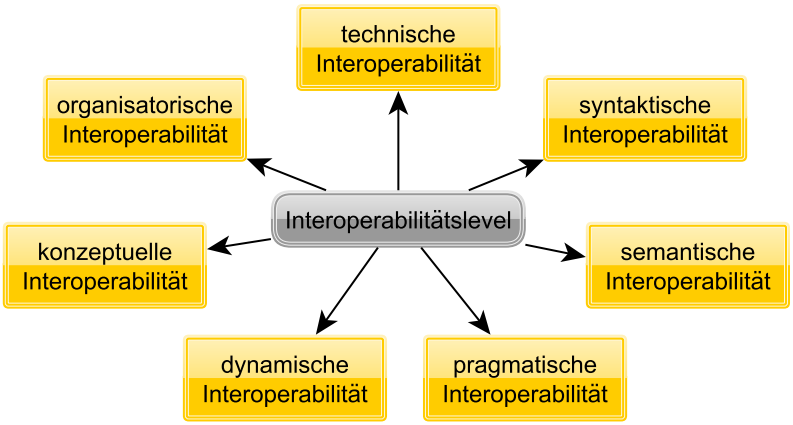

Abbildung 2.5: Interoperabilitätslevel nach Manso et al. [MWB09]

Die Grundlage für einen funktionierenden Datenaustausch zwischen heterogenen Systemen ist eine *syntaktische Interoperabilität*. Diese bezieht sich auf die Syntax der auszutauschenden Daten. Die Syntax ist die Form, in der die Daten übertragen werden. Besteht keine syntaktische Interoperabilität, so ist das Empfangssystem nicht in der Lage die Daten zu lesen und ein Datenaustausch schlägt fehl. Die Herstellung syntaktischer Interoperabilität ist Aufgabe der Datenaustauschstrategie.

Bei alleiniger syntaktischer Interoperabilität ist die gleiche Interpretation der übertragenden Daten in den Systemen jedoch nicht sichergestellt. Fehlinterpretationen sind möglich, da syntaktische Interoperabilität keine gemeinsame Semantik voraussetzt. Diese gemeinsame Semantik muss ebenfalls durch die Datenaustauschstrategie festgelegt werden, da ansonsten zwar Daten ausgetauscht jedoch nicht weiterverwendet werden können. Die *semantische Interoperabilität* wird durch ein gemeinsames Vokabular realisiert. So können Fehlinterpretationen und Ungenauigkeiten vermieden werden. Die Festlegung eines gemeinsamen Vokabulars erfolgt über Standards und Spezifikationen. Ziel ist die Erreichung einer eindeutigen Interpretation der Daten.

Ein weiteres Interoperabilitätslevel ist die *pragmatische Interoperabilität.* Voraussetzung hier ist, dass die Systeme gegenseitig auf Funktionalitäten und Schnittstellen zugreifen können. Diese Kopplung der Systeme ist bereits sehr eng. Dennoch können die Systeme je nach Realisierung unabhängig bleiben. Eine Integration ist somit nicht zwingend gegeben. Diese Art von Interoperabilität ist für einen funktionierenden Datenaustausch nicht zwingend notwendig, bringt jedoch Vorteile mit sich, da direkt Funktionalitäten der einzelnen Systeme in den anderen Systemen aufgerufen werden können.

Bei der *dynamischen Interoperabilität* überwacht ein System die anderen Systeme. Das überwachende System kann dann dynamisch Einfluss auf den Datenaustausch nehmen. Beim Datenaustausch zwischen Diensten sind verschiedene Formen von Einflüssen möglich. Beim Datenaustausch zwischen heterogenen Systemen im Sinne dieser Arbeit beschränkt sich der Einfluss an dieser Stelle weitgehend auf die Realisierung eines automatischen Austausches. Überwacht werden die Änderungen

der Daten. Diese können dann einen Datenaustausch auslösen. Dieser automatische Austausch kann durch eine Integration oder eine automatische Kopplung realisiert werden.

Grundsätzlich ist ein Datenaustausch nur zwischen Systemen möglich, die ähnliche Grundkonzepte verwenden. Das bedeutet Geometriedaten können nur zwischen Systemen ausgetauscht werden, die diese Art von Daten abbilden und verarbeiten können. Ein Austausch von einem 3D-Modell zwischen einem System, dass dieses Modell erstellt hat, und einem System, das rein zur Tabellenkalkulation gedacht ist und keine Geometrie verarbeiten kann, ist nicht möglich. Die Grundlage für diese *konzeptuelle Interoperabilität* ist das Vorhandensein von Wissen über die Systemfunktionalitäten. Dieses Wissen kann in Form von Systemdokumentationen abgelegt sein. Diese sollten standardisiert und austauschbar sein. Für die Realisierung eines Datenaustausches bedeutet dies, dass von vornherein zu klären ist, ob konzeptuelle Interoperabilität vorliegt. Datenaustauschstrategien können dies bisher nicht selbstständig.

Bei der *organisatorischen Interoperabilität* steht das Wissen um die Rahmenbedingungen des Datenaustausches im Vordergrund. Rahmenbedingungen sind dabei beispielsweise Ziele, Zugriffsrechte, Erwartungen und Verträge. Im Kontext dieser Arbeit gehört dies zum Anwendungsszenario.

Die Klassifikation nach den Interoperabilitätsleveln ermöglicht den Vergleich verschiedener Lösungsstrategien hinsichtlich des Levels der Kopplung. Sowohl die Zuordnung bestehender Lösungen zu realisierten Level, als auch die Berücksichtigung geforderter Level und ihrer Eigenschaften direkt bei der Entwicklung neuer Lösungen sind möglich.

Klassifikation nach Merkmalen der Realisierung

Kopplungslösungen unterscheiden sich anhand verschiedener Merkmale der Realisierung. Li [Li10] beschreibt auf Grundlage der Realisierungsmerkmale eine Klassifikation von Kopplungslösungen. Diese wird von Vornholt et al. in [VGL10] aufgegriffen und ebenfalls beschrieben.

Ein Merkmal ist die *Sicht auf die Daten* in der Benutzungsoberfläche des Datenmanagements. Die Sicht auf die Daten ist dabei unabhängig vom konkreten Format der Daten. Der Autor unterscheidet zwischen einer spezifischen und einer allgemeinen Sicht. Die allgemeine Sicht erlaubt einen domänübergreifenden Überblick auf die Daten. Sie wird entweder mit einem Produktdatenmanagement (PDM)-System oder durch neue Elemente der Benutzungsoberfläche von anzeigenden Systemen realisiert. Diese Elemente erlauben den Zugriff auf PDM-Funktionalitäten. Der Autor verwendet dafür den Begriff „Integrated Interface“. Diese Art von Sicht erlaubt die Kombination von Modellen unterschiedlicher Domäne zu einem Gesamtmodell. Insgesamt wird sie jedoch eher den Integrationslösungen zugeordnet. Die spezifische Sicht beschränkt sich auf eine einzelne Domäne und repräsentiert die Daten für diese Domäne. Jedes System hat hier ihre eigene Repräsentationsform. Daher ist diese Art der Sicht bei Kopplungslösungen zu finden.

Beide Arbeiten unterscheiden nicht zwischen der *Zahl unterschiedlicher Domänen* von einem Datenaustausch zwischen Systemen verschiedener Domänen aus. Dies beschreibt hingegen Avgoustinov in [Avg97]. Der Autor bezeichnet den Austausch

zwischen Systemen einer Domäne als *in-phasen* oder *horizontaler* Austausch. Den Austausch zwischen Systemen unterschiedlicher Domänen nennt er hingegen *inter-phasen* oder *vertikaler* Austausch. Diese Unterscheidung ist für die Auswahl und Entwicklung von Austauschlösungen bedeutsam, da damit Aussagen zu den verwendeten und benötigten Informationen der Systeme möglich sind. Systeme einer Domäne arbeiten mit ähnlichen Informationen, da ähnliche Problemstellungen mit den Systemen gelöst werden. Systeme unterschiedlicher Domänen verarbeiten hingegen unterschiedliche Informationen. Die Zahl der unterschiedlichen Domänen ist somit ein weiteres Realisierungsmerkmal für die Unterscheidung von Austauschlösungen.

Li [Li10] verwendet als weiteres Realisierungsmerkmal die *Art der Weitergabe von Änderungen* der Daten. Der Autor unterscheidet dabei zwischen dem *vollständigen* und dem *inkrementellen* Austausch von Daten. Beim inkrementellen Austausch werden, im Gegensatz zum vollständigen Austausch, nur Änderungen übertragen. Initial ist jedoch auch beim inkrementellen Austausch eine vollständige Übertragung notwendig. Für die Aktualisierung der Daten in einem System werden dann nur die Änderungen aus dem anderen System ausgetauscht. Dazu ist es notwendig, die Änderungen zu identifizieren. Dies geschieht über die Behandlung von Abhängigkeiten. Die Realisierung eines inkrementellen Datenaustausches erfolgt daher in Form von systemspezifischen Austauschlösungen oder über ein zentrales Kontrollzentrum. Vorteile des inkrementellen Austausches gegenüber dem vollständigen Austausch sind der geringere Umfang der zu übertragenden Daten, was Ressourcen wie Zeit und Speicher spart, und die Konzentration der Anwender auf Änderungen statt auf Versionen. Die Realisierung der Austauschlösung selbst ist jedoch anspruchsvoller.

Drath et al. [DFB11] unterscheiden nach der *Vollständigkeit der Daten*. Dabei verwenden sie als Beispiel Anlagenplanungsdaten. Die Autoren unterscheiden zwischen dem *Austausch aller Daten* und dem *Austausch von tatsächlich benötigten Daten*. Alle Daten bedeutet dabei, dass das alle Daten des Sendesystems, die die Strategie in der Lage ist zu übertragen, auch übertragen werden. Dies beinhaltet häufig auch für den Empfänger und das Anwendungsszenario nicht relevante Daten. Im Gegensatz dazu beschränkt sich der Austausch von tatsächlich benötigten Daten auf die relevanten Daten für den Empfänger und das Anwendungsszenario. Zusätzliche Daten, die das Sendesystem enthält und durch die Strategie prinzipiell übertragbar sind, werden nicht weitergegeben. Vorteile beim Austausch von tatsächlich benötigten Daten sind die geringeren Datenmengen und der erhoffte geringere Aufwand bei der Realisierung des Austausches, da weniger semantische Übereinstimmungen gefunden und realisiert werden müssen. Nachteil dieser Methode ist jedoch die Identifizierung der benötigten Datenmengen.

Ein weiteres Realisierungsmerkmal in der Klassifikation von Li [Li10] ist die *Bestimmung des Zeitpunktes* für den Datenaustausch. Die Bestimmung des Zeitpunktes geschieht demnach *manuell* auf Anfrage oder *automatisch*. Ein Austausch startet bei automatischer Bestimmung, sobald Änderungen der Daten im Sendesystem erkannt werden. Anders ist dies bei der manuellen Bestimmung. Hier startet eine Anfrage des Anwenders, eines Softwaresystems oder einer Middleware den Austauschprozess. Die automatische Bestimmung des Austauschzeitpunktes führt zu einer automatischen Kopplung der Systeme und realisiert so dynamische Interoperabilität bei Systemen im Sinne dieser Arbeit. Hier besteht ein Zusammenhang zwischen der Klassifikation

nach den Interoperabilitätsleveln und dieser hier vorgestellten Klassifikation nach Realisierungsmerkmalen.

Bei der Austauschkontrolle hinsichtlich der Weitergabe von Änderungen unterscheidet der Autor zwischen einem *zentralen und einem fortlaufenden Austauschmanagement*. Ein fortlaufendes Austauschmanagement reicht Änderungen an Daten über die Beziehung der Daten untereinander fortlaufend weiter. Das heißt, bei der Änderung eines Parameters erfolgt die Anpassung der davon abhängigen Daten Stück für Stück entlang der Abhängigkeitskette. Dafür müssen mehrfach Daten in der Kette ausgetauscht werden. Anders geht ein zentrales Austauschmanagement vor. Dort verwaltet ein zentrales Kontrollzentrum die Daten und Beziehungen zentral. Bei Änderung beispielsweise eines Parameters passt das zentrale Kontrollzentrum alle dazugehörigen Daten direkt an. Es entsteht keine Kette von Änderungen von einem System zum nächsten.

Die Klassifizierung von Li [Li10] verwendet zwei weitere Realisierungsmerkmale. Diese beziehen sich zum einen auf verschiedene Prozesstypen der Produktentwicklung und auf das Format bei dateibasierten Lösungen. Beide Merkmale sind somit nicht allgemeingültig, sondern stehen im Bezug zu einem Anwendungsbereich oder zu einer Klasse von Lösungsansätzen.

Bei dem Format unterscheidet der Autor die Austauschlösungen in Lösungen mit *systemspezifischen* und *systemneutralen Formaten*. Eine nähere Betrachtung der Klassifikation nach der Systemneutralität allgemein und beim dateibasierten Austausch im speziellen erfolgte bereits in vorherigen Klassifikationsschemata (siehe Seite 9 und 12).

Li [Li10] beschreibt das *linear*, das *simultanous* und das *concurrent* Engineering als Prozessvarianten der Produktentwicklung. Beim sogenannten linear Engineering werden die Aufgaben und Entwicklungsphasen nacheinander bearbeitet. Anders ist es beim concurrent Engineering. Dort werden die Aufgaben einer Prozessphase aufgeteilt und parallel bearbeitet. Beim sogenannten simultanous Engineering werden komplette Prozessphasen überlappend bearbeitet. Der Autor klassifiziert Austauschlösungen nach der Unterstützung dieser Prozessvarianten, da diese spezielle Anforderungen an den Datenaustausch stellen. Beim concurrent und dem simultanous Engineering werden Daten vor der Freigabe zwischen Aufgaben und Prozessphasen ausgetauscht. Das bedeutet, dass Änderungen dieser Daten jederzeit möglich sind und eine schnelle Weitergabe dieser Änderungen an die anderen Aufgabenbereiche gefordert ist. Anders ist dies beim linear Engineering. Dort erfolgt die Weitergabe von einer Phase zur anderen nach der Freigabe.

Die Klassifikation nach verschiedenen Realisierungsmerkmalen zeigt die Unterscheidung von Datenaustauschlösungen nach den verschiedensten Kriterien und die Abhängigkeit der Ausprägung der Kriterien vom konkreten Anwendungsszenario. Diese Kriterien unterstützen die Auswahl von Lösungsstrategien. Ein Heranziehen für die Entwicklung neuer Lösungen ist ebenfalls möglich. Zu bemerken ist dabei, dass diese Betrachtung nicht vollständig ist und konkrete Anwendungsszenarien weitere Merkmale besitzen.

Klassifikation nach verwendetem Datenaustauschformat

Der Datenaustausch über Dateien ist eine weitverbreitete Form des Datenaustausches [VSG+11]. Eine Möglichkeit des dateibasierten Austausches ist die Verwendung neutraler Formate, deren Klassifikation nach ihren Eigenschaften hier beschrieben wird. Datenaustauschformate sind Formate, die dem Zweck des Datenaustausches dienen [BBD+03, S.212]. Diese Formate unterscheiden sich in verschiedenen Eigenschaften. Beispielsweise sind dies folgende Eigenschaften:

- Art und Umfang der Daten [VSG+11][FLL+11][Fac05][KVGS11]
- Eignung für bestimmte Anwendungsszenarien [Thi04][VSG+11][FLL+11]
- Grad der Verbindlichkeit der Daten [Han11]
- Standardisierung [VSG+11][FLL+11][Fac05][WBTM00]
- Offenheit [FLL+11][Fac05][HL08][BDP07]
- Dateigröße [VSG+11][BDP08][FLL+11][BDP07]
- Komplexität und Umfang der Dokumentation [VSG+11][DFB11]
- Erweiterbarkeit [HL08]
- Sicherheit vor unerlaubtem Zugriff und Veränderung der Daten [HL08][BDP07]

Neutrale Formate für den Datenaustausch unterscheiden sich analog zu Datenaustauschstrategien allgemein nach der *Art der übertragbaren Daten* (siehe Seite 7) [KVGS11]. Beispielsweise übertragen Visualisierungsdaten hauptsächlich geometrische und geometriebezogene Daten zur Visualisierung von Geometrie [FLL+11]. Andere Formate dienen beispielsweise zum Austausch von Produkt- oder Prozessdaten [KVGS11]. Weiterhin unterscheiden sich die einzelnen Formate für eine Art von Daten im *Beschreibungsumfang* der übertragbaren Daten [VSG+11, FLL+11, Fac05]. Friedewald et al. untersuchen beispielsweise in [FLL+11] verschiedene Austauschformate hinsichtlich ihrer Eignung als universelles Austauschformat im Schiffsbau. Dafür stellen die Autoren verschiedene gewichtete Kriterien auf, zum Beispiel welche Geometrieelemente und welche Metadaten übertragbar sind. Darunter sind auch Kriterien den Beschreibungsumfang des neutralen Formates betreffend. Beispiele hierfür sind die Übertragbarkeit von Farbinformationen, Hilfsgeometrien und die mathematische Repräsentationsform der Geometriemodelle. Die Datenaustauschformate unterscheiden sich insbesondere in diesem Punkt, wobei eine Vielzahl verschiedener Daten möglich ist. Aus diesem Grund werden hier keine näheren Einzelheiten für Datenaustauschformate im Allgemeinen genannt.

Thierfelder unterscheidet in [Thi04] Formate zum Austausch von 3D-Geometriedaten *nach ihrem Anwendungsfeld* in CAD/CAM/CAE-Formate, Modellierungs- und Animationsformate, Echtzeitgrafik/VR-Formate und internet-basierte Formate. Der Autor beschreibt, dass bei Formaten zum Austausch zwischen Computer Aided Design (CAD)-, Computer Aided Manufacturing (CAM)- und Computer Aided Engineering (CAE)-Systemen die exakte Beschreibung der Modelle im Vordergrund

steht. Eine exakte und detaillierte Übertragung der Daten ist notwendig. Beispiele für solche Formate sind DXF, STEP, IGES und SET. Modellierungs- und Animationsformate beschreiben laut dem Autor statt einzelner Objekte komplette Szenen. Diese Szenen enthalten mehrere Objekte sowie Informationen zu Lichtern, Kameras und Effekten. Die exakte Darstellung rückt dabei in den Hintergrund. Approximierte Geometriebeschreibungen sind daher möglich. Beispielformate sind OBJ, 3DS und POV. Echtzeitgrafik/VR-Formate dienen dem Austausch zwischen Echtzeitsystemen und Systemen für virtuelle Realitäten (VR). Sie erweitern die Modellierungs- und Animationsformate um komplexe Animationen, Simulationen und Interaktionsmöglichkeiten. Dafür sinken in der Regel der Detaillierungsgrad und die Auflösung. Als Beispiele nennt der Autor OpenGL, Direct3D und QuickDraw3D. Bei internet-basierten Formaten steht die plattformunabhängige Darstellung in Browsern im Vordergrund. Dazu sind Modularisierungs-, Kompressions-, Streaming- und Skalierungsansätze erforderlich. Beispielformate sind VRML, X3D und MPEG-4. Diese Einteilung ist für 3D-Grafikformate interessant, jedoch gibt Thierfelder keine Quellen für seine Aussagen an. Daher ist diese Einteilung kritisch zu betrachten.

Handschuh [Han11] unterteilt Datenaustauschformaten *nach dem Grad der Verbindlichkeit* der beschriebenen Daten in primäre und sekundäre Formate. Sekundäre Datenformate haben rein informellen Charakter. Der Austausch verbindlicher Daten erfolgt über primäre Datenformate. Der Autor sagt aus, dass heute primäre Formate häufig noch systemspezifische Formate sind. Diese sind nicht zum Austausch zwischen heterogenen Systemen gedacht. Den Austausch zwischen heterogenen Systemen realisieren häufig sekundäre Formate. Als Hauptanwendungsgebiet dieser Einteilung nennt der Autor die Automobil-Industrie. Handschuh beschäftigt sich mit der Nutzung des neutralen Formates JT als primäres Datenformat und kommt zu dem Ergebnis, dass JT als primäres Datenformat verwendbar ist.

Datenaustauschformate unterscheiden sich *nach ihrer Standardisierung* in formal standardisierte Formate, nicht-standardisierte Formate und Industriestandards. Eine formale Standardisierung beziehungsweise Normung bedeutet, dass das Format von einem Standardisierungsgremium angenommen wurde [FLL+11]. Beispiele für standardisierte neutrale Formate sind STEP, IGES und VDAFS [WBTM00]. Nicht standardisiert ist beispielsweise das Format PLM XML [Fac05]. Der Vorteil von standardisierten Formaten ist die konkrete Definition der Verwendung und der Terminologie des Formates [WBTM00]. Nachteil der Standardisierung ist, dass die Standards möglicherweise zu groß oder unhandlich zur Implementierung oder zu restriktiv sind [WBTM00]. Die Industrie verwendet Industrie- beziehungsweise de-facto Standards ähnlich wie Standardformate [VSG+11, WBTM00]. Jedoch sind diese Formate nicht von einem Gremium angenommen. Dafür ist ihre Verbreitung in der Regel groß [VSG+11, WBTM00]. Ein Beispiel dafür ist DXF [WBTM00].

Friedewald et al. [FLL+11] beschreiben die *Offenheit* eines Formates als die Zugänglichkeit der Formatspezifikation. Ein Austauschformat gilt genau dann als offen, wenn die Formatspezifikation frei zugänglich publiziert wurde. Damit ist die Realisierung eigener Programme zur Interpretation einer Datei in dem bestimmten Format möglich. Proprietäre Formate sind nicht-offene Formate. Handschuh setzt in [Han11] proprietäre Formate und systemspezifische Formate gleich. Obwohl

systemspezifische Formate oft proprietär sind, können diese nicht im Allgemeinen gleichgesetzt werden, da sowohl systemspezifische Formate offen sein können als auch neutrale Formate proprietär sein können. Beispielsweise ist das neutrale Format DXF ein proprietäres Format [SVG11, VSG+11]. Hier ist auf korrekte Verwendung der Begriffe zu achten.

Die *Leichtgewichtigkeit* von Austauschformaten bezieht sich auf das Verhältnis der Dateigröße vom systemspezifischen Format zum neutralen Format [Han11]. Handschuh [Han11] bezeichnet ein neutrales Format demnach als leichtgewichtig, wenn die Datei kleiner als im systemspezifischen Format ist. Gerhardt verwendet in [Ger10] folgende Definition für leichtgewichtige Formate: „A neutral, lightweight and CAD-derived data format (NF) is a data format that is visualization-oriented with the possibility to store CAD-content at very small sizes (down to 10% compared to original CAD). An NF can be read and written by arbitrary applications and includes containers to store tessellated (approximated) geometry at different Levels of Detail [...], exact geometry, product structures, meta data for product structure nodes and positions (transformations) for product structure nodes [Ger10, S.16].“ Handschuh nutzt dafür den Begriff „Open Lightweight Strategy-Format“. Zu beachten ist, dass diese Definition von einem Austausch geometrischer Daten ausgeht. Nach der grundlegenden Definition von Leichtgewichtigkeit als Verhältnis der Dateigröße von systemspezifischen zu neutralem Format sind auch alle anderen neutralen Formate nach der Leichtgewichtigkeit klassifizierbar.

Die Klassifikation der Formate nach ihren Eigenschaften erlaubt eine Zuordnung der Formate zu Gruppen. Dies ermöglicht den Vergleich und die Beurteilung bestehender und neuer Formate.

Die verschiedenen vorgestellten Klassifikationsschemata zeigen, dass es große Unterschiede bei den Lösungsstrategien gibt und dass diese verschiedene Vor- und Nachteile aufweisen. Die erfolgreiche Verwendung der Strategien hängt daher häufig vom konkreten Anwendungsszenario und dessen Eigenschaften ab. Die Lösungsstrategien unterscheiden sich bezüglich verschiedener Aspekte. Eine allumfassende Einteilung ist nicht möglich. Die Klassifikationsschemata können zur Unterstützung der Auswahl bestehender Lösungen und zur Entwicklung neuer Ansätze verwendet werden. Im Anhang ab Seite 89 befindet sich eine Übersicht über die vorgestellten Klassifikationsschemata. Dieser Abschnitt hat somit die Frage nach bestehenden Klassifikationsschemata für Datenaustauschstrategien im Ingenieurwesen beantwortet.

2.3 Stufen von Produktmodellen in der virtuellen Produktentstehung

Dieser Abschnitt konzentriert sich auf die Entwicklungsstufen von Produktmodellen, die in der virtuellen Produktentstehung verwendet werden. Die VDI Richtlinie 2221 [VDI93, S.41] definiert ein *Produkt* als ein „Erzeugnis, das als Ergebnis des Entwickelns und Konstruierens hergestellt oder angewendet wird.“ Die Richtlinie unterscheidet zwischen materiell und immateriell. Materielle Produkte sind beispielsweise Maschinen und Bauteile. Währenddessen ist ein Softwareprogramm ein immaterielles Produkt. Diese Arbeit beschränkt sich auf materielle Erzeugnisse des Ingenieurwesens und verwendet für diese den Begriff „Produkt“.

Die *virtuelle Produktentstehung* beschäftigt sich mit der durchgängigen Rechnerunterstützung des Produktentstehungsprozesses [ES09, Eig09, War09]. Ziel ist die Verkürzung der Entwicklungszeit [KVGS11] durch die frühe virtuelle Untersuchung von Produkteigenschaften und Produktverhalten auf Grundlage digitaler Produktmodelle [ES09, Eig09]. Die virtuellen Untersuchungen sollen die Zahl der benötigten, teuren physischen Untersuchungen reduzieren [ES09, Eig09]. Dazu werden Simulationen und Berechnungen durchgeführt.

Neben der Bezeichnung „virtuelle Produktentstehung“ ist in der Literatur auch die Bezeichnung „virtuelle Produktentwicklung“ zu finden. Die Verwendung dieser Begriffe richtet sich nach dem Umfang der unterstützten Produktlebensphasen [ES09, Sto11]. Dabei umfassen die Begriffe „Produktentwicklung“ und „Produktentstehung“ je nach Einteilung des Produktlebenszyklus unterschiedliche Lebensphasen [ES09, Sto11, SK97]. Die *Produktentwicklung* umfasst nach Stoye [Sto11] und Spur und Krause [SK97] allgemein die Produktplanung, die Produktkonstruktion und die Produkterprobung. Die Produktion beziehungsweise Herstellung ist neben allen Phasen der Produktentwicklung Teil der *Produktentstehung*. Nach Auffassung dieser Autoren gehört die Produktforschung weder zur Produktentwicklung noch zur Produktentstehung.

Eigner und Stelzer [ES09] unterscheiden nicht zwischen Produktkonstruktion und Erprobung. Diese Phasen fassen die Autoren unter dem Begriff „Entwicklung“ zusammen. Dafür unterscheiden sie zwischen Produktentwicklung und Produktionsentwicklung. Die *Produktentwicklung* umfasst demnach die Aufnahme und Analyse der Anforderungen, die Produktplanung, große Teile der Entwicklung und kleine Teile der Prozessplanung. Der größte Teil der Prozessplanungsphase zählt zusammen mit dem Rest der Entwicklung zur Produktionsplanung. Die *Produktentstehung* definieren sie als Gesamtheit der Produkt- und Produktionsentwicklung. Die Produktion ist nicht Teil der Produktentstehung nach dieser Definition.

Bei allen drei Quellen ist die Produktentwicklung immer Bestandteil der Produktentstehung. Die *Produktentwicklung* ist in dieser Arbeit definiert als Oberbegriff für die Aufnahme und Analyse der Anforderungen, die Produktplanung, die Produktkonstruktion und die Produkterprobung. Die *Produktentstehung* ist definiert als Gesamtheit aller Phasen von der Aufnahme der Anforderungen bis zum Ende der Fertigung.

Für die virtuelle Produktentstehung ist eine digitale Repräsentation der Produkte von großer Bedeutung [Sto11, War09], da diese für die Durchführung von Berechnungen und Simulationen notwendig ist. Die Literatur verwendet verschiedene Begriffe für Produktmodelle in der virtuellen Produktentstehung. Diese werden im Folgenden vorgestellt und in Entwicklungsstufen eingeordnet. Abbildung 2.6 zeigt eine Übersicht über die Entwicklungsstufen.

Die einfachste Stufe der Produktmodelle im Rechner ist die *textuelle Beschreibung* der Produkte. Die Beschreibung der Produktgeometrie ist weniger anschaulich im Vergleich zu grafischen Darstellungen.

Eine Möglichkeit der grafischen Darstellung von Produkten im Rechner ist durch zweidimensionale Geometriemodelle. Diese *2D-Modelle* entsprechen den klassischen

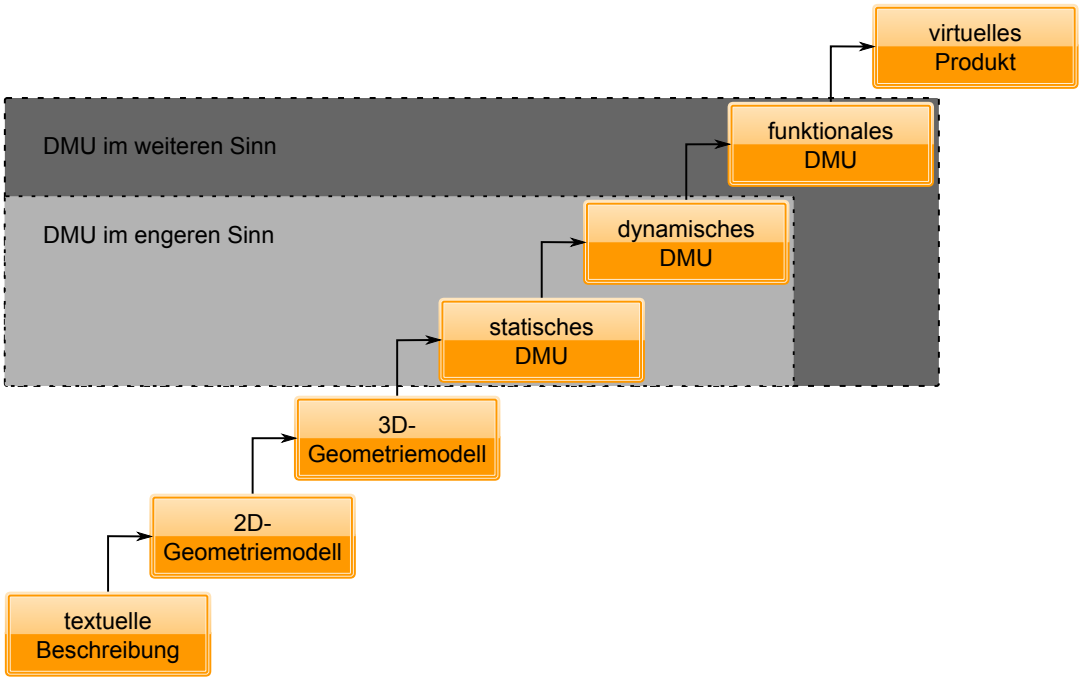

Abbildung 2.6: Entwicklungsstufen von Produktmodellen in der virtuellen Produktentstehung

Konstruktionszeichnungen [War09], wobei verschiedene Ansichten zur Beschreibung der gesamten Produktgeometrie notwendig sind. Ein 2D-Modell beschreibt genau eine Ansicht des Produktes. Dies hat die Nachteile, dass zum einen eine konsistente und widerspruchsfreie Beschreibung des Produktes schwierig ist und zum anderen bei Änderungen an der Geometrie aller Ansichten kontrolliert und gegebenenfalls angepasst werden müssen [Ten02, S.10]. Ein weiterer Nachteil von 2D-Geometriemodellen ist die wenig anschauliche Darstellung des Produktes für den Anwender und insbesondere für Laien [War09]. Insgesamt bestehen mit 2D-Modellen wenige Möglichkeiten für Simulationen und Berechnungen. Dennoch sind 2D-Modelle für bestimmte Einsatzbereiche ausreichend, beispielsweise für die Entwicklung elektronischer Schaltungen oder bei Fertigung von Blechteilen [Ten02, S.10]. In der FEM-Berechnung werden 2D-Modelle in bestimmten Situationen als Vereinfachung zur Verkürzung der Berechnungszeit und zur Verringerung des Aufwandes eingesetzt. Ein weiteres Einsatzgebiet von 2D-Modellen ist in Sketchern [VWB+09, S.159ff] [CKMH12]. Sketcher sind Skizzierwerkzeuge, die beispielsweise in der frühen Konstruktionsphase zum Einsatz kommen und einer ersten Darstellung dienen. Cheon et al. beschreiben in [CKMH12] einen Ansatz zur Generierung von 3D-Modellen aus 2D-Freihand-Sketchern-Modellen.

Dreidimensionale Geometriemodelle bilden die nächste Stufe von Produktmodellen [War09]. Mit ihnen ist eine eindeutige Beschreibung komplexer Produktgeometrie möglich [VWB+09, S.170][Ten02, S.10]. Ein Vorteil der 3D-Modelle gegenüber den 2D-Modellen ist die anschaulichere und realitätsnähere Darstellung der Produkte [Ste07, War09, VWB+09], wobei die Generierung von 2D-Ansichten und Schnitten automatisiert möglich ist [Ten02, S.10]. Das heißt, 3D-Modelle beinhalten sämtli-

che Informationen, die auch in 2D-Modellen abbildbar sind. Änderungen erfolgen direkt am Modell und wirken sich in allen Dimensionen aus. Eine manuelle Anpassung von einzelnen Ansichten ist nicht notwendig [Ten02, S.10]. 3D-Produktmodelle unterstützen die Durchführung verschiedener Simulationen und Berechnungen zum Produktverhalten, die mit 2D-Modellen nicht oder nur unvollständig möglich sind [Ten02, S.10][VWB+09, S.170][VWB+09, S.159ff]. Dies betrifft beispielsweise bestimmte FEM-Berechnungen und Mehrkomponentensimulationen. Zudem ist die Ableitung von bestimmten geometrischen Daten für NC-Programme oder Rapid Prototyp-Verfahren möglich [VWB+09, S.170][Ten02, S.10].

Nachteile bei der Verwendung von 3D-Geometriemodellen sind die meist hohen Lizenzkosten für die Erzeugersysteme und der relativ hohe Einarbeitungsaufwand in die Systeme [Ten02, S.10]. Nichtsdestotrotz bilden die 3D-Geometriemodelle die Grundlage für die virtuelle Produktentstehung [Ten02, S.10][AB10, S.18].

Stekolschik teilt in [Ste07] den Inhalt von 3D-Geometriemodellen in drei Schichten ein. Dies sind die Visualisierungsschicht, die Modellierungsschicht und die Semantikschicht. Die Visualisierungsschicht enthält die Geometriedaten und die Visualisierungsdaten. Die Modellhistorie, die mögliche Parametrik und Feature-Informationen sind in der Modellierungsschicht angesiedelt. Die Semantikschicht enthält Wissen zur Modellierung und zur Konstruktion. Die Modellierungs- und Semantikschicht enthalten oft Informationen zum Unternehmenswissen und werden daher häufig nicht beim unternehmensübergreifenden Datenaustausch weitergegeben.

Eine Einteilung von 3D-Geometriemodellen nach der mathematischen Repräsentation wurde bereits bei der Klassifikation nach der Art der Daten auf Seite 8 beschrieben und in Abbildung 2.2 abgebildet.

Die nächste Stufe der Produktmodelle in der virtuellen Produktentstehung bilden die Digital Mock-ups (DMUs) [AB10, War09]. Es gibt verschiedene Definitionen für DMU. Diese Definitionen unterscheiden sich im Umfang der möglichen Untersuchung und in den Bestandteilen des Produktmodells. Diese Arbeit unterscheidet daher zwischen DMU im engeren Sinn und DMU im weiteren Sinn.

DMUs im engeren Sinn dienen der virtuellen Untersuchung verschiedener Aspekte des Zusammenbaus von Produkten oder von Unterbaugruppen von Produkten [AG11, VDA02, SBD06, AB10]. Die Aufgaben sind dabei beispielsweise die Analyse der Struktur und des Aufbaus und die Betrachtung der räumlichen und funktionalen Gestaltung von vollständigen Produkten, Baugruppen und Bauteilen [KAW+07]. Konkrete Anwendungsbereiche von DMUs im engeren Sinn sind beispielsweise die Überprüfung und Simulation der Montage und Demontage [KAW+07, AG11, ES09, AJ00], die Berechnung von Zusammenbauten [AG11], verschiedene ergonomische Untersuchungen [KAW+07, ES09], die Kollisionsprüfung [KAW+07, AG11, AB10, AJ00], die Visualisierung der Produkte in kollaborativen Umgebungen [KAW+07] und die Ableitung von Modellen zur Präsentation in virtuelle Realitäten [AJ00].

Die für diese Anwendungen zwingend benötigen Bestandteile eines DMUs im engeren Sinn sind die Produktgeometrie und die Produktstruktur [AB10, AG11, SBD06, ES09, KAW+07, AJ00]. Die Verwendung exakter Geometriemodelle ist aufgrund der damit verbundenen Datenmengen für viele Anwendungen nicht möglich [SBD06,

ES09]. Dies gilt besonders für sehr große und komplexe Produkte wie Autos, Schiffe und Flugzeuge [ES09]. Daher werden oft vereinfachte Geometriemodelle verwendet [AG11, SBD06]. Dabei sind auch Approximationen möglich [ES09, AG11]. Ein Nachteil dieses Ansatzes ist, dass Änderungen an den Teilen des DMU nicht direkt durchgeführt werden können [ES09]. Änderungen müssen immer am Originalmodell erfolgen. Ein Problem bei der Approximation ist, dass der Aufwand und das Datenvolumen mit der Zahl gekrümmter Flächen zunehmen [ES09]. Somit führt eine Approximation in bestimmten Fällen zu größeren Datenvolumina als die exakte Geometrie. Eigner und Stelzer haben dies in [ES09] verdeutlicht. Wichtig ist somit die Betrachtung des konkreten Anwendungsfalls zur Bestimmung eines geeigneten Beschreibungsansatzes für die Produktgeometrie. Dies gilt sowohl für DMU im engeren als auch im weiteren Sinn.

Die Produktstruktur beschreibt die Zuordnungen und Beziehungen der Bauteile und Unterbaugruppen untereinander [ES09, S.233f]. Dies beinhaltet die korrekte Positionierung der Teile im Zusammenbau [ES09, S.233f]. Mit zusätzlich zugeordneten Materialeigenschaften ist zudem die Bestimmung verschiedener physikalischer Eigenschaften des Produktes wie Gewicht und Schwerpunkt möglich [AG11, AB10, AJ00].

Verschiedene Quellen [VWB+09, ES09, Wan02] setzen DMUs mit *virtuellen Prototypen* gleich. Diese Definitionen gehören zu den Definitionen von *DMUs im weiteren Sinn*. DMUs sind in diesem Sinne Produktmodelle, die ein Produkt möglichst realitätsnah im Rechner beschreiben [Kra97, ES09, KKM97, DR96] und für verschiedene Untersuchungen in unterschiedlichen Phasen des Produktlebenszyklus verwendet werden [GDSKM97, ES09, Wan02, VK10]. Dabei bilden DMUs im weiteren Sinn im Unterschied zu DMUs im engeren Sinn Produktfunktionalitäten im Produktlebenszyklus umfangreich ab [GDSKM97, ES09, KKM97, DR96]. DMUs im weiteren Sinne beschränken sich somit nicht auf Untersuchungen bezüglich des Zusammenbaus, sondern ermöglichen die verschiedensten Simulationen, Berechnungen und Visualisierungen zu Produkteigenschaften und zum Produktverhalten [AG05b]. Das DMU enthält dabei alle Daten, die zu diesem Zweck notwendig sind in ausreichender Genauigkeit [Sto11]. Dies beinhaltet alle Daten von DMUs im engeren Sinn [AB10, KAW+07, AJ00], Materialeigenschaften [AB10], physikalische und logische Eigenschaften [AB10, AJ00], Funktionsstrukturen, Simulationsmodelle und -ergebnisse sowie reale Messdaten [KAW+07]. Anwendungsbeispiele für DMUs im weiteren Sinne sind Machbarkeitsanalysen [Sto11], Auslegung und Optimierung von Produkten [Sto11], Strömungssimulationen [KAW+07, ES09] und Schwingungsanalysen [ES09].

Eigner und Stelzer unterscheiden in [ES09] drei verschiedene Klassen von DMUs. *Statische DMUs* beinhalten Daten zur Produktgeometrie und zur Produktstruktur, jedoch keine Daten zur Kinematik der Bauteile und zur Funktionalität. *Dynamische DMUs* erweitern die statischen DMUs um Daten zu funktionellen Bewegungen der Bauteile. Die Probleme dynamischer DMUs sind lange Berechnungszeiten und große Datenmengen [AG05b]. Jedoch ermöglichen diese DMUs dynamische Geometrieprüfungen [AG05a]. Diese beiden Klassen entsprechen den Definitionen von DMUs im engeren Sinn. Die dritte Klasse erweitert die dynamischen DMUs um Daten zum Produktverhalten und der Produktfunktionalität. Diese *funktionalen DMUs* entsprechen den Definitionen von virtuellen Prototypen oder DMU im weiteren Sinne [KAW+07].

Ein Produktmodell, das sich nicht nur auf einzelne Aspekte beschränkt, sondern Produkteigenschaften und -verhalten umfassend im Rechner darstellt, nennt sich *virtuelles Produkt* [SK97, AB10, Kra97, AJ00] oder *integriertes Produktmodell* [SK97, VWB+09]. Ziel ist die vollständige, digitale Beschreibung des Produktes über den gesamten Lebenszyklus [Kra97, SK97, AJ00]. Ein virtuelles Produkt stellt somit ein Produkt umfassend virtuell dar. Das virtuelle Produkt bildet somit die höchste Stufe der Produktmodelle in der virtuellen Produktentstehung.

Dieser Abschnitt hat verschiedene Stufen von Produktmodellen in der virtuellen Produktentstehung beschrieben und voneinander abgegrenzt. Mit den Stufen steigt der Informationsgehalt der Modelle. Die Produktmodelle beschreiben das Produkt immer umfassender in seinen Eigenschaften und seinem Verhalten.

2.4 Zusammenfassung

In diesem Kapitel wurden die Grundlagen der Arbeit erläutert. Dies beinhaltete neben Grundbegriffen auch bestehende Klassifikationsschemata für Lösungsstrategien und eine Einordnung von Produktmodellen der virtuellen Produktentstehung in verschiedene Stufen.

Mit der Vorstellung verschiedener Klassifikationsschemata wurde in diesem Kapitel die erste Fragestellung der Arbeit nach bestehenden Schemata beantwortet. Dabei wurde gezeigt, dass die Lösungsstrategien sich in verschiedenen Aspekten unterscheiden. Diese Aspekte sind sowohl für die Auswahl bestehender Ansätze als auch für die Entwicklung neuer Ansätze von großer Bedeutung. Das konkrete Anwendungsszenario mit seinen spezifischen Anforderungen spielt dabei eine wichtige Rolle, da jede Lösungsstrategie Vor- und Nachteile aufweist. Aus diesem Grund wird der Vergleich ausgewählter Datenaustauschstrategien für das konkrete Beispielszenario „Datenaustausch bei der Montagesimulation“ durchgeführt. Das nächste Kapitel betrachtet dieses Anwendungsszenario näher und beantwortet so die zweite Fragestellung der Arbeit nach einem konkreten Anwendungsszenario.

3. Anwendungsszenario

Das vorherige Kapitel hat Grundlagen dieser Arbeit erläutert. Dabei wurde gezeigt, dass das konkrete Anwendungsszenario Einfluss auf die Auswahl und Entwicklung von Lösungsstrategien hat. Diese Arbeit verwendet daher ein Beispielszenario für den Vergleich ausgewählter Lösungsstrategien. Als Beispielszenario dient der Datenaustausch bei der Montagesimulation. Den Beginn dieses Kapitels bildet ein Überblick über die Montagesimulation allgemein. Danach folgt eine Betrachtung des Datenaustausches bei der Montagesimulation. Dies beinhaltet das Aufzeigen verschiedener Variationen des Szenarios. Im Anschluss daran folgen Erläuterungen zu den Anforderungen an den Datenaustausch. Danach werden die Klassen von infrage kommenden Lösungsstrategien nach den Klassifikationsschemata ermittelt. Das Ende bildet eine kurze Zusammenfassung zum Anwendungsszenario.

3.1 Überblick Montagesimulation

Die VDI Richtlinie 3633 [VDI96, S.14] versteht unter einer *Simulation* „ein Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind." Im weiteren Sinn beinhaltet eine solche Simulation „das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell" [VDI96, S.14]. Der Begriff „System" beschreibt in dieser Definition Systeme im allgemeinen. Bei der Montagesimulation beinhaltet das nachgebildete System je nach untersuchtem Aspekt und Umfang der Simulation an der Montage beteiligte Elemente. Elemente sind hierbei vor allen Dingen die Bauteile und Unterbaugruppen eines Produktes, die zu montieren sind. Weiterhin ist es möglich, bei der Montagesimulation Werkzeuge, Montageanlagen [VWB+09] und Menschen [KKM97] miteinzubeziehen.

Im Mittelpunkt der Betrachtungen steht bei der Montagesimulation die Abbildung der Bewegungen bei der Montage und Demontage von Produkten [SBD06]. Das Ziel ist späte kosten- und zeitintensive Änderungen bei der Konstruktion eines Produktes durch frühzeitige Erkennung von Fehlern und Problemen zu verhindern

[SK97, SBD06, VWB+09]. Dabei ist es möglich, verschiedene Varianten der Montage und Demontage zu entwickeln und zu bewerten [SBD06]. Dies beinhaltet nach Vajna et al. [VWB+09] auch den zeitlichen Bedarf für verschiedene Konfigurationen einer Produktionsanlage. Dieser zeitliche Bedarf umfasst Änderungen der Produktionsanlage beispielsweise durch den Wechsel von Werkzeugen oder Spannvorrichtungen.

Die Montagesimulation untersucht die Montier- und Demontierbarkeit eines Produktes [AMPSS04]. Damit verbunden ist die Erkennung von Kollisionen bei Ein- und Ausbauvorgängen [Haa95]. Voraussetzung für die Montage und Demontage in der Realität sind kollisionsfreie Bewegungspfade [AMPSS04, Haa95, Ger10]. Werkzeuge zur Montagesimulation bieten heute zum Teil bereits Funktionen zur automatischen Erkennung dieser kollisionsfreien Ein- und Ausbaupfade an [SBD06]. Ein Beispiel für solch ein Werkzeug ist das Fitting Modul von CATIA V5 [Beh03]. Auch die Erstellung von Platzhaltern entlang der Bewegungspfade ist nach Scholz et al. [SBD06] heute möglich.

Bei der Erstellung von Montagepfaden sind neben den Bauteilen auch Werkzeuge und der Mensch zu berücksichtigen. Scholz et al. [SBD06] schlagen dafür die Verwendung von einzuhaltenden Mindestabständen vor. Eine andere Möglichkeit ist die Verwendung eigener Modelle für die Werkzeuge und den Menschen. Die Literatur geht nicht näher auf diese Thematik ein, aber bei den Abbildungen sind Varianten mit Mindestabständen (zum Beispiel in [VWB+09, Beh03, SBD06]) und Varianten mit Modellen von Werkzeugen und Menschen (zum Beispiel in [GDSKM97, WRSG11, Vin04]) zu finden. Bei der Verwendung von Modellen für die Werkzeuge und Menschen ist von einer höheren Qualität der Ergebnisse auszugehen, da die Werkzeuge und Menschen genauer abgebildet werden können als durch pauschale Mindestabstände. Anzunehmende Nachteile sind dabei jedoch die zusätzlichen Datenmengen und der Aufwand für die Erstellung oder Beschaffung geeigneter Modelle.

Die Untersuchung der Montierbarkeit und der Demontierbarkeit dienen der Feststellung, ob ein konstruiertes Produkt herstellbar ist. Weiterhin sind mit der Montagesimulation Untersuchungen zur Wartungsfreundlichkeit eines Produktes möglich [SBD06]. Die Wartungsfreundlichkeit eines Produktes wirkt sich auf die entstehenden Instandhaltungskosten und Stillstandszeiten aus. Ziel ist daher bei bestimmten Produkten, wie Fahrzeugen und Produktionsmaschinen, eine hohe Wartungsfreundlichkeit zu erreichen [SBD06]. Die Simulation der Montage und Demontage kann dies unterstützen [SBD06, AMPSS04, DR96]. Dies geschieht durch Prüfung, ob und wie gut Bauteile für eine Reparatur zugänglich sind.

Aus einer bestehenden Montagesimulation ist die Erstellung von Montagevideos möglich. Diese sind beispielsweise für die Schulung von Monteuren einsetzbar [SBD06].

Neben der kontinuierlichen Erprobung von Teilen während der Konstruktion ist die Unterstützung der Montageplanung ein Einsatzgebiet der Montagesimulation [VWB+09, Hol02]. Dabei ist nach Holle [Hol02] zu beachten, dass die Montageplanung nicht vollständig abgebildet wird. Die Montageplanung berücksichtigt weitere Prozessdetails wie Montagezeit und Arbeitsteilung zur Erstellung von Montageplänen. Die Betrachtung dieser Aspekte erfolgt in der Montagesimulation, die hier be-

trachtet wird, in der Regel nicht oder nur am Rande. Jedoch ist eine Unterstützung bei der Wahl von Montagemitteln und der Montagereihenfolge möglich [VWB+09].

Die aktuelle Situation bei der Produktentstehung ist, dass verschiedene Abteilungen und Personen an der Entwicklung und Produktion eines Produktes beteiligt sind [BKK+04]. Scholz et al. [SBD06] beschreiben die Verwendung der Montagesimulation als Kommunikationsmittel zwischen diesen verschiedenen Beteiligten. Mögliche Beteiligte, zwischen denen eine Kommunikation zu unterstützen ist, sind die Entwickler von Einzelteilen und Unterbaugruppen [KKM97], Verantwortliche für die Gesamtkonstruktion [KKM97], insgesamt Konstruktionsabteilungen [SBD06], die Produktionsplanung [SBD06], Werker sowie interne und externe Entwicklungspartner [KKM97]. Die Montagesimulation verdeutlicht diesen verschieden Beteiligten, die jeweils eine andere Sichtweise auf das Produkt haben, wichtige Aspekte der Montage und Demontage.

Die Montagesimulation gehört zu den Techniken der virtuellen Produktentstehung. Ziel ist die Verringerung kosten- und zeitintensiver physischer Montageexperimente durch virtuelle Techniken [GDSKM97, KKM97]. Dabei ist der Einsatz von Techniken der virtuelle Realitäten möglich [SK97, GDSKM97, ES09, KKM97, VWB+09]. Diese erlauben das Erleben der Montage in einer virtuellen Umgebung [GDSKM97].

In diesem Abschnitt wurde zum besseren Verständnis die Montagesimulation in einem Überblick vorgestellt. Die Montagesimulation benötigt verschiedene Daten zu den verwendeten Bauteilen und gegebenenfalls zu Werkzeugen und Montageanlagen. Diese Daten können aus verschiedenen Systemen stammen [SAKJ98]. Als Ergebnisse der Montagesimulation sind weiterhin beispielsweise erkannte Kollisionen von Bauteilen für verschiedene Systeme zur Weiterverarbeitung von Bedeutung [SAKJ98]. Der nächste Abschnitt beschäftigt sich näher mit dem Datenaustausch bei der Montagesimulation. Dabei wird auf den Inhalt der Daten und verschiedene Variationen des Beispielszenarios eingegangen.

3.2 Datenaustausch bei der Montagesimulation

Für den Vergleich ausgewählter Lösungsstrategien verwendet diese Arbeit den Datenaustausch bei der Montagesimulation als Beispielszenario. Im vorherigen Abschnitt wurde die Montagesimulation im Überblick vorgestellt.

Im Mittelpunkt der Montagesimulation steht die Montage eines Produktes. Eingangsdaten für die Montagesimulation sind Produktdaten [GDSKM97, KAW+07]. Diese beinhalten die Geometrie der einzelnen Bestandteile eines Produktes, die Produktstruktur und die Kinematik von bewegten Bauteilen [GDSKM97, KAW+07, Ger10]. Das verwendete Produktmodell in der Montagesimulation ist somit ein dynamisches DMU. Produktstruktur und Kinematik können dabei jedoch auch im Simulationssystem eingegeben werden. Damit sind auch Geometriemodelle und statische DMUs als Eingangsmodelle möglich.

Ein Geometriemodell beschreibt ein Bauteil eines Produktes. Eine grundlegende Voraussetzung für die Durchführung einer Montagesimulation sind dreidimensionale Geometriemodelle [Ste07, Ger10, ES09, Haa95]. 2D-Modelle sind als Eingangsmodelle nicht geeignet [ES09].

Statische und dynamische DMUs beschreiben Baugruppen. Beschriebene Baugruppen sind hier entweder komplette Produkte oder einzelne Unterbaugruppen des Produktes. Unterbaugruppen können zur Reduzierung der Datenmenge auch durch Ersatzgeometrien beschrieben sein [SBD06]. Die innere Struktur der Unterbaugruppe wird dabei vernachlässigt und die Unterbaugruppe durch ein Hüllenvolumen ersetzt. Ein Geometriemodell analog zu einem Modell eines Bauteils beschreibt das Hüllenvolumen. Für die Unterbaugruppen selbst kann eine separate Montagesimulation erfolgen.

Werden einzelne Bauteile und Unterbaugruppen an die Montagesimulation übergeben, so ist die Positionierung dieser nachträglich anzufügen. Dies gilt sowohl für die Beschreibung von Unterbaugruppen durch DMUs als auch für die Beschreibung durch eine Ersatzgeometrie. Bei statischen oder dynamischen DMUs des kompletten Produktes ist die Positionierung der Bauteile und Unterbaugruppen nicht notwendig, da das DMU die Produktstruktur bereits enthält. Bei statischen DMUs muss im Gegensatz zu dynamischen DMUs die Kinematik bewegter Teile angefügt werden.

Der Datenaustausch variiert somit bezüglich des *Umfanges der übertragen Modelle*. Möglich sind 3D-Modelle von Bauteilen und Ersatzgeometrien von Unterbaugruppen, statische und dynamische DMUs von Unterbaugruppen und kompletten Produkten. Produktmodelle höherer Stufe als die dynamischen DMUs enthalten weitere Informationen, welche die Montagesimulation in der Regel jedoch nicht benötigt.

Weiterhin von Bedeutung sind je nach Umfang der Simulation *Daten zu Werkzeugen und Montageanlagen*. Werkzeuge und Montageanlagen sind selbst Produkte. Modelle für Werkzeuge und Maschinen stammen dabei entweder von der Entwicklung dieser, werden für die Montagesimulation neu modelliert oder durch Techniken des Reverse Engineering erstellt. Die Montagesimulation benötigt für Werkzeuge und Maschinen analog zu Bauteilen Daten zur Geometrie und zu möglichen Bewegungen. Daten zur Struktur sind nicht notwendig.

Neben dem Inhalt und der Qualität der Daten hängt die Komplexität des Datenaustausches von den eingesetzten Systemen ab [Dyl02]. Sender der Daten für die Montagesimulation sind Erzeugersysteme. Dies sind beispielsweise verschiedene CAD-Systeme oder Systeme zur Digitalisierung von Objekten. Auch weitere CAD-ferne geometrieerzeugende Systeme sind als Erzeugersysteme für die Montagesimulation denkbar. Voraussetzung ist nur die Erzeugung von dreidimensionalen Geometriemodellen.

Der Datenaustausch variiert in der *Anzahl der Erzeugersysteme* zwischen einem oder wenigen und vielen Erzeugersystemen. Von Bedeutung ist hier auch die Version eines Systems. Sollen Daten aus verschiedenen Versionen eines Systems übertragen werden, bildet jede Version ein eigenes System. Der Grund für diese Festlegung ist, dass Daten über Versionsgrenzen innerhalb eines Systems nicht immer ohne Anpassungen im aktuellsten System verwendbar sind [BDP08]. Es ist ein Datenaustausch zwischen den verschiedenen Softwareversionen notwendig.

Empfänger der Daten ist das Simulationssystem. Bei der *Art des Simulationssystems* unterscheidet diese Arbeit drei Fälle. Im Fall A ist das Simulationssystem ein eigenständiges System. Das bedeutet, es benötigt in der Regel keine Daten, die

nicht für die Simulation notwendig sind. Eigenständige Systeme zur Montagesimulation sind beispielsweise spezielle VR-Systeme [WRSG11]. Zu diesem Fall zählen auch Simulationssysteme, die zwar Teil eines größeren Systems sind, jedoch nicht auf vollständige Modelle im Format dieses größeren Systems angewiesen sind. Der Fall B beinhaltet Simulationssysteme, die ein Modul eines beliebigen größeren Systems sind. Diese Simulationsmodule sind jedoch auf vollständige Modelle im Format des größeren Systems angewiesen. Dieses beinhaltet in der Regel auch Daten, die für die Simulation nicht notwendig sind. Die Zuordnung zu diesem Fall ist abhängig von der internen Realisierung. Der Fall C stellt einen Spezialfall des Falls B dar, da dieser sich auf CAD-Systeme als übergeordnete Systeme beschränkt. Beispielsweise existieren Simulationsmodule für eine Montagesimulation in den CAD-Systeme CATIA [Beh03, SAKJ98, SBD06] und HiCAD [ISD11]. Ob diese Systeme zum Fall C gehören, hängt jedoch von der internen Realisierung ab. Zu Fall C gehören nur Simulationsmodule von CAD-Systemen, die auf vollständige Modelle ihres übergeordneten CAD-Systems angewiesen sind. Damit sind zusätzliche Anforderungen bezüglich der notwendigen Daten zu erwarten. Im Folgenden wird der Fall C exemplarisch für den Fall B behandelt, da pauschale Aussagen für beliebige Systeme im Allgemeinen nicht möglich sind.

Ziel bei der Montagesimulation ist es entworfene Produkte auf Kollisionen während der Montage zu überprüfen. Dabei werden in der Regel kollidierte Bereiche durch Einfärbung gekennzeichnet. Im nächsten Bearbeitungsschnitt der Produktentstehung werden die Bauteile und in bestimmten Fällen die Werkzeuge iterativ so weiterentwickelt, dass am Ende eine kollisionsfreie Montage möglich ist. Dazu werden die Produktbestandteile und Werkzeuge verändert und eine erneute Montagesimulation durchgeführt. Dieser Vorgang wird, bis eine kollisionsfreie Montage möglich ist, wiederholt. Die Bestandteile des Produktes oder die Werkzeuge liegen dabei entweder als am Rechner entworfenes Modell oder als von einem realen Objekt digitalisiertes Modell vor. Am Rechner entworfene Modelle werden dann in der Regel auch im Rechner weiterentwickelt. Bei digitalisierten Modellen ist dies auch in bestimmten Systemen möglich. Ansonsten werden diese die realen Objekte verändert und für die erneute Montagesimulation neu digitalisiert. Sollen die Modelle am Rechner verändert werden, so kann die Übertragung der Kennzeichnungen am Modell zum Erzeugersystem für den Konstrukteur hilfreich sein [SAKJ98]. Diese Übertragung stellt einen Austausch von Ergebnisdaten dar [SAKJ98]. Somit variiert der Datenaustausch zwischen dem Austausch von *Eingabedaten* zum Simulationssystem und dem Austausch von *Ergebnisdaten* zu den jeweiligen Erzeugersystemen oder zu einem Bearbeitungssystem.

Die Schwierigkeit beim Austausch der Ergebnisdaten ist die Art der Erzeuger- beziehungsweise Bearbeitungssystemen. Hier müssen die Anforderungen an die jeweiligen Importdaten beachtet werden. Erzeuger- und Bearbeitungssysteme sind beispielsweise CAD-Systeme oder andere CAD-ferne geometrieerzeugende Systeme. Bei CAD-Systemen treten beim Austausch der Ergebnisdaten ähnliche Anforderungen wie bei Fall C für Eingangsdaten auf. Dabei ist der empfangende CAD-System statt dem Simulationssystem ein Erzeuger- oder Bearbeitungssystem ist. Hierbei ist zu beachten, dass das Simulationssystem selbst kein CAD-System sein muss und die Daten des Simulationssystems in einer anderen Form und einem anderen Umfang als für

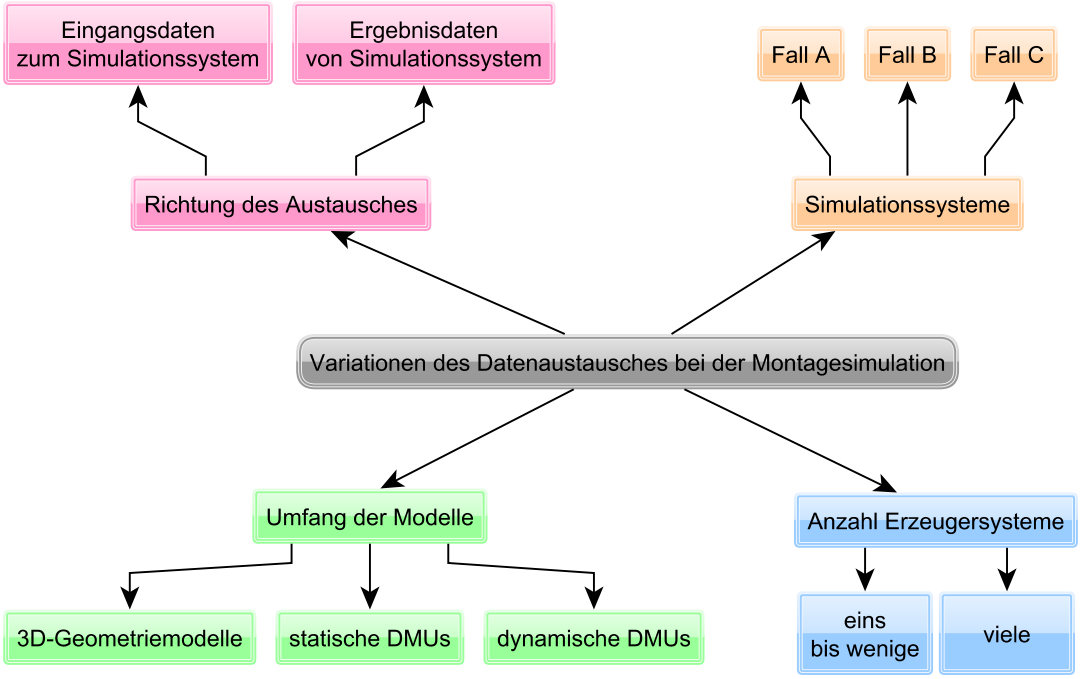

Abbildung 3.1: Variationen des Datenaustauschs bei der Montagesimulation

das CAD-System notwendig vorliegen können. Dies macht den Austausch schwierig. Auch der Austausch von Ergebnisdaten zu CAD-fernen geometrieerzeugenden Systemen und beliebigen Bearbeitungssystemen ist ein komplexes Problem, da Spezifika bezüglich der notwendigen Importdaten dieser Systeme zu berücksichtigen sind. Es kann dabei pauschal nicht davon ausgegangen werden, dass jedes Simulationssystem alle diese Daten besitzt. Insgesamt kann somit der Austausch der Ergebnisdaten vom Simulationssystem zu den Erzeugersystemen als komplexes Austauschproblem betrachtet werden.

Dieser Abschnitt beschreibt verschiedene Variationen des Anwendungsszenarios Datenaustausch bei der Montagesimulation. Abbildung 3.1 zeigt die Variationen im Überblick. Die Variation erfolgt an verschiedenen Aspekten des Datenaustausches. Dabei sind unterschiedliche Kombinationen der einzelnen Variationen möglich. Nur bei der Kombination von einem Erzeugersystem mit einem Modul als Simulationssystem ist zu beachten, dass, wenn das Modul zum Erzeugersystem gehört, kein Datenaustausch im Sinne dieser Arbeit notwendig ist. Die Daten bleiben innerhalb eines Systems. Daher wird bei der Kombination von einem Sendesystem und einem Modul für die Montagesimulation davon ausgegangen, dass das Modul nicht Teil des Sendesystems ist, sondern zu einem anderen System gehört.

Der nächste Abschnitt beschäftigt sich mit den Anforderungen an den Datenaustausch und damit an Datenaustauschstrategien.

3.3 Anforderungen an den Datenaustausch

Dieser Abschnitt geht näher auf die Anforderungen an den Datenaustausch bei der Montagesimulation ein. Grundsätzlich unterscheidet diese Arbeit zwischen den folgenden vier Gruppen von Anforderungen:

- qualitätsbezogene Anforderungen
- organisatorische Anforderungen
- anwendungsszenariospezifische Anforderungen
- lösungsspezifische Anforderungen

Zu den *qualitätsbezogenen Anforderungen* zählen Anforderungen, welche die Qualität des Datenaustausches festlegen. Allgemein fordern Vajna et al. [VWB+09] beispielsweise einen reibungsfreien Datenaustausch. Dies bedeutet, sie fordern einen verlustfreien und fehlerfreien Datenaustausch. Ein verlustfreier Datenaustausch wird auch von Schumann in [Sch01], Friedewald et al. in [FLL+11], Anderl in [And93] und Köppen et al. in [KVGS11] gefordert. Die Verlustfreiheit bezieht sich hier darauf, dass alle zu übertragenden Daten und ihre Zusammenhänge bei der Übertragung erhalten bleiben. Ein Verlust bei der Übertragung eines Flächenmodelles ist beispielsweise die Nichtübertragung einer Fläche. Dadurch ist das Flächenmodell nach der Übertragung nicht mehr vollständig und möglicherweise inkonsistent. Solche und andere Verluste von Daten gilt es zu vermeiden, da diese zusätzlichen Aufwand beim Nutzer erzeugen. Dieser muss im Falle eines Verlustes die fehlenden Daten ersetzten und die Modelle reparieren. Im schlechtesten Fall schlägt die komplette Übertagung aufgrund eines Datenverlustes fehl. Analog gilt dies für Fehler, die durch den Datenaustausch auftreten. Fehlerfreiheit beziehungsweise Fehlersicherheit wird neben Vajna et al. [VWB+09] auch von Friedewald et al. [FLL+11], Stoye [Sto11] und Anderl [And93] gefordert. Anderl schließt dabei das Auftreten von Fehlern nicht vollkommen aus, sondern nennt die Erkennung, Behandlung und Dokumentation von Fehlern als systembezogene Aufgaben von externen Schnittstellen. Weiterhin fordert Anderl [And93] eine konsistente und eindeutige Abbildung der Informationen beim Datenaustausch. Dies beugt Fehlern bei der Interpretation der Daten vor.

Organisatorische Anforderungen beziehen sich auf die Realisierung der Datenaustauschlösung sowie auf die Durchführung und die Rahmenbedingungen des Austausches. Eine organisatorische Anforderung, welche die Realisierung der Austauschlösung betrifft, ist die Forderung nach einer schnellen Entwicklung [Ren86]. Die Erstellung und Pflege der Austauschlösung soll mit einem möglichst geringen Aufwand möglich sein [Ten02, Avg97]. Ziel dieser Anforderungen ist, die Kosten für die Realisierung und Pflege der Austauschlösungen möglichst gering zu halten. Daher fordert Anderl [And93] auch erweiterbare und an das konkrete Szenario anpassbare Austauschstrategien. Zudem sollen nach Dyla [Dyl02] langfristig verwendbare Lösungen realisiert werden.

Anderl [And93] fordert für die Durchführung des Austausches benutzerfreundliche Lösungen. Die Software sollte einfach zu bedienen sein. Zudem fordert Renz in [Ren86], dass Dateien und Daten einfach zu lesen und zu ändern sein sollen.

Anforderungen zum organisatorischen Rahmen des Datenaustausches betreffen die Sicherheit der Daten vor unerlaubtem Zugriffen und Änderungen [BDP07], die anfallenden Lizenzkosten [BDP07, Fac05] und die Nachvollziehbarkeit des Austausches bezüglich der Daten und Modelle [VSG+11, KVGS11]. Köppen et al. [KVGS11] bezieht sich bei der Nachvollziehbarkeit auf Änderungen von Daten und Modellen und den Einfluss dieser beispielsweise auf Ergebnisse aus Berechnungen. Dies beinhaltet Abhängigkeit zwischen Versionen und Varianten von Daten und Modellen sowie verworfenen Modellen. Berechnungen und Simulationen beziehen sich in der Regel auf einen Datenstand und müssen für neue Datenstände erneut durchgeführt werden. Die Nachvollziehbarkeit fällt jedoch in den Bereich der Verwaltung von Daten und Austauschprozessen und ist nicht Thema konkreter Datenaustauschstrategien.

Anwendungsszenariospezifische Anforderungen beinhalten alle Anforderung, die durch das konkrete Anwendungsszenario vorgegeben werden. Dies betrifft zu großen Teilen die Art und den Umfang der zu übertragenden Daten. Auf die Art und den Umfang der Daten, die bei der Montagesimulation ausgetauscht werden soll, wurde bereits im vorherigen Abschnitt (auf Seite 33) eingegangen. Zusammenfassend benötigt die Montagesimulation je nach Umfang der übertragenen Modelle Daten zur Geometrie von Bauteilen und Unterbaugruppen, die Produktstruktur, Daten zur Kinematik bewegter Teile und gegebenenfalls Daten zu Werkzeugen und Montageanlagen. Die Geometrie kann dabei in der Regel approximiert und vereinfacht sein [SBD06]. Beim Fall B und C sind weitere Daten zur Erstellung des vollständigen Modells des Gesamtsystems notwendig. Welche Daten dies konkret im Fall B sind, hängt von System, das dem Simulationsmodul übergeordnet ist, ab. Pauschale Aussagen zu belieben Systemen sind hier nicht möglich. Anders ist dies beim Fall C. Hier müssen vollständige CAD-Modelle ausgetauscht werden. Diese beruhen in der Regel auf exakter Geometrie und enthalten heute oft Parametrik- und Featuredaten [Ger03, NJG09]. Für editierbare CAD-Modelle kann zudem die Modellierungshistorie notwendig sein [CKMH12]. Damit stellt der Fall C andere Anforderungen bezüglich dem Umfang und der Qualität der zu übertragenden Daten. Die Datenaustauschstrategien müssen daher für diesen Fall gesondert beurteilt werden. Für Fall B ist diese gesonderte Beurteilung nur in Ansätzen möglich, da keine pauschalen Aussagen zum Umfang und zur Qualität der Daten getroffen werden können.

Beim Austausch der Ergebnisdaten vom Simulationssystem zu Erzeuger- oder Bearbeitungssystemen müssen vornehmlich geometrische Daten mit Kennzeichnungen, zum Beispiel Einfärbungen, von Bestandteilen des Produktes übertragen werden. In bestimmten Fällen sind dabei auch Werkzeugmodelle möglich. Die Daten müssen dabei an CAD-Systeme oder andere geometrieerzeugende Systeme übertragen werden. Die einzelnen Systeme besitzen hierbei bestimmte Spezifika die Eingangsdaten betreffend. Diese sind zu berücksichtigen. Beim Austausch der Ergebnisdaten zu CAD-Systemen gelten die gleichen Spezifika, wie beim Fall C. Ziel beim Austausch der Ergebnisdaten ist die Anpassung der Modelle. Das bedeutet, für die direkte Veränderung der übertragenen Modelle müssen diese editierbar sein. Viele CAD-Systeme fordern hierzu die Übertragung der Modellhistorie und exakter Geometrien [CKMH12]. Insgesamt ist beim Datenaustausch von Ergebnisdaten bei der Montagesimulation der Umfang der notwendigen Importdaten für die Erzeuger- und

Bearbeitungssysteme problematisch, da nicht jedes Simulationssystem alle diese Daten besitzt. Diese fehlenden Daten machen den Austausch von Ergebnisdaten zur Weiterverarbeitung zu einem komplexen Problem.

Lösungsspezifische Anforderungen gelten für bestimmte Arten von Lösungsstrategien. Beispielsweise werden beim Austausch über neutrale Formate kleine Dateigrößen gefordert, um einen schnellen Datenversand und einen geringen Speicherplatzbedarf zu gewährleisten [BDP07, Ren86]. Ball et al. [BDP07] fordern zudem die Verwendung offener Formate. Dies erleichtert die Realisierung eigener Software zur Verarbeitung der Daten. Analog existieren verschiedene Anforderungen für andere Lösungsstrategien.

Dieser Abschnitt behandelte Anforderungen an den Datenaustausch. Dabei wurde eine Einteilung der Anforderungen in verschiedene Gruppen durchgeführt. Weiterhin wurden szenariospezifische Anforderungen an den Datenaustausch bei der Montagesimulation behandelt. Zudem erfolgte eine Betrachtung verschiedener Variationen des Szenarios. Der nächste Abschnitt beschäftigt sich mit den Klassen möglicher Lösungsstrategien für den Datenaustausch bei der Montagesimulation. Dabei wird auch auf die Variationen des Szenarios eingegangen.

3.4 Klassen von möglichen Lösungsstrategien

Dieser Abschnitt untersucht das Anwendungsszenario Datenaustausch bei der Montagesimulation bezüglich der Klassen von möglichen Lösungsstrategien. Dazu werden die Klassifikationsschemata aus Abschnitt 2.2 verwendet. Eingegangen wird dabei auch auf die verschiedenen Variationen des Anwendungsszenarios.

Ein Klassifikationsschema, das vorgestellt wurde, ist die Einteilung *nach der Art der Daten*. Dabei wurde zwischen einer Einteilung nach produktionstechnischen Aspekten und einer Einteilung nach dem Geometriegehalt der Daten unterschieden. Nach produktionstechnischen Aspekten wird zwischen Produktdaten, Prozessdaten oder Auftragsdaten unterschieden. Herausgestellt wurde, dass Lösungsstrategien nicht ausschließlich auf eine Art der Daten beschränkt sind, aber dennoch Schwerpunkte bei der Art der übertragbaren Daten besitzen. Nach dem Geometriegehalt werden geometrische und nicht-geometrische Daten unterschieden.

Beim Austausch zum Simulationssystem benötigt die Montagesimulation Produktdaten und gegebenenfalls Prozessdaten in Form von Werkzeugmodellen. Die benötigten Werkzeugmodelle ähneln dabei vom Inhalt her den Produktmodellen und können daher auch durch Strategien zum Austausch von Produktdaten übertragen werden.

Je nach Umfang der Produktmodelle sind unterschiedliche Daten zu übertragen. Grundsätzlich benötigt die Montagesimulation Geometriemodelle der Bauteile, Unterbaugruppen und gegebenenfalls der Werkzeuge. Damit sind Datenaustauschstrategien notwendig, die geometrische Daten übertragen. Sollen statische oder dynamische DMUs als Eingangsdaten dienen, müssen neben der Geometrie die Produktstruktur und bei dynamischen DMUs die Kinematik übertragen werden. Produktstruktur und Kinematik zählen zu den nicht-geometrischen Daten. Bei diesen beiden

Variationen sind somit Datenaustauschstrategien notwendig, die geometrische und bestimmte nicht-geometrische Daten übertragen.

Beim Datenaustausch zur Montagesimulation allgemein sind approximierte und vereinfachte Geometriedaten möglich, wenn das Simulationssystem dies unterstützt. In diesem Fall muss die Datenaustauschstrategie auch diese Art der Daten übertragen können.

Im Fall B und C sind weitere Daten zum Aufbau des Modells im Format des Gesamtsystems notwendig. Eine Einordnung der Daten in die Art ist im Fall B pauschal nicht möglich, da die konkreten Daten nicht bekannt sind und je nach System variieren. Beim Fall C sind nicht-geometrische Daten zu Modellhistorie, Parametrik und Features möglich. Welche Daten davon konkret benötigt werden, hängt vom konkreten CAD-System ab.

Beim Austausch der Ergebnisdaten vom Simulationssystem zu den Erzeuger- und Bearbeitungssystemen muss dieselbe Art von Daten wie beim Austausch der Eingangsdaten übertragen werden. Zusätzlich müssen Kennzeichnungen an der Geometrie übertragen werden, diese gehören zu den Geometrieattributen und sind nicht-geometrische Daten. Mögliche Datenaustauschstrategien müssen dies unterstützen. Zu beachten ist hier, dass bei beliebige anderen Erzeuger- und Bearbeitungssystemen die Spezifika bezüglich der notwendigen Import-Daten zu berücksichtigen sind. Pauschale Aussagen zur Art dieser spezifischen Daten sind für beliebige Systeme jedoch nicht möglich. Bei CAD-Systemen sind hier meist zusätzliche nicht-geometrische Daten zur Modellhistorie für editierbare Modelle notwendig.

Die Klassifikation *nach der Systemneutralität* wurde in Abschnitt 2.2 auf Seite 9 vorgestellt. Lösungsstrategien für den Datenaustausch unterscheiden sich demnach in systemspezifische und systemneutrale Strategien. Systemspezifische Strategien sind bei wenigen zu koppelnden Systemen vorteilhaft. Währenddessen sind systemneutrale Strategien bei einer großen Zahl an zu koppelnden System sinnvoll. Der Datenaustausch bei der Montagesimulation variiert in der Anzahl der Systeme. Somit ist die Entscheidung zwischen diesen Strategien von der konkreten Situation im Unternehmen abhängig. Grundsätzlich sind beide Formen der Kopplung für das Anwendungsszenario möglich.

Ein weiteres vorgestelltes Klassifikationsschema unterscheidet Datenaustauschstrategien *nach dem Konzept des Austausches* in übersetzende, generierende und einbindende Strategien (auf Seite 11). Übersetzende Strategien unterscheiden sich weiter in die Verfahren mit Direktkonvertern, Strategien mit einem neutralen Format und kernbasierte Strategien. Verfahren mit Direktkonvertern sind systemspezifische Verfahren. Es gilt das Gleiche, wie für die gesamte Klasse dieser Verfahren. Sie sind bei wenigen Systemen vorteilhaft, aber insgesamt unabhängig von den unterschiedlichen Variationen möglich. Analog verhält es sich bei Verfahren, die neutrale Formate verwenden. Diese gehören zu den systemneutralen Datenaustauschstrategien und sind bei vielen zu koppelnden Systemen sinnvoll. Ihr Einsatz ist jedoch unabhängig von den verschiedenen Variationen möglich. Anders ist es beim kernbasierten Austausch. Die Voraussetzung hier ist das Vorhandensein von Modellierkernen in den Erzeuger- und im Simulationssystem. Im Fall C ist davon auszugehen, dass diese Voraussetzung erfüllt ist, wenn die Sendesysteme ebenfalls CAD-Systeme sind. Daher ist hier

ein kernbasierter Austausch möglich. Allgemein kann jedoch nicht davon ausgegangen werden, dass alle Erzeuger- und das Simulationssystem einen Modellierkern in gleicher Art und Weise verwenden. Ob kernbasierte Austauschstrategien möglich zur Realisierung des Datenaustausches bei der Montagesimulation sind, hängt daher von den konkreten zu koppelnden Systemen ab. Außerdem können über eine kernbasierte Austauschstrategie nur Daten, die im Kern abbildbar sind, übertragen werden. Andere Daten sind nicht übertragbar. Fraglich ist hier, ob Produktstrukturdaten und Kinematikdaten übertragen werden können. Dies hängt vom Funktionsumfang der Kerne ab. Zusammenfassend sind kernbasierte Strategien beim Datenaustausch bei der Montagesimulation allgemein nur unter der Voraussetzung verwendbar, dass alle Systeme Modellierkerne von der gleichen Art und Weise besitzen. Beim Fall C ist ein kernbasierter Austausch in der Regel möglich, wenn die Sendesysteme ebenfalls CAD-Systeme sind. DMUs können nur übertragen werden, wenn die Kerne die notwendigen Funktionen besitzen.

Ähnlich dem kernbasierten Austausch verhält es sich beim generierenden Austausch. Voraussetzung ist hier, dass die Systeme ähnliche erzeugende Funktionen besitzen. Das bedeutet, ein generierender Austausch zwischen einem Erzeugersystem und einem reinen Konsumenten ist nicht möglich. Beide Systeme müssen in der Lage sein beispielsweise Geometrie zu erzeugen. Hier liegt das Problem dieser Art von Austauschstrategie. Nicht jedes Simulationssystem ist in der Lage, Geometrie zu erzeugen. Selbst wenn alle Systeme Geometrie erzeugen können, ist ein generierender Datenaustausch möglicherweise problematisch. Dies ist der Fall, wenn die Systeme Modelle auf eine vollkommen andere Art und Weise erzeugen. Hier ist es egal, ob Eingangsdaten oder Ergebnisdaten der Montagesimulation ausgetauscht werden sollen. Insgesamt bedeutet dies, dass eine generierende Austauschstrategie nur unter bestimmten Voraussetzungen beim Datenaustausch in der Montagesimulation möglich sind. Anders ist dies beim Fall C, wenn die Erzeugersysteme ebenfalls CAD-Systeme sind. Hier werden Daten zwischen CAD-Systemen ausgetauscht. Diese Systeme erzeugen Geometrie auf ähnliche Art und Weise. Ein generierender Datenaustausch ist somit beim Fall C unter der Bedingung, dass die Sendesysteme ebenfalls CAD-Systeme sind, möglich.

Verfahren zum einbindenden Austausch unterscheiden sich in der Form der Einbindung. Formen sind dabei das Verknüpfen und das Einbetten. Beide Formen sind beim Datenaustausch bei der Montagesimulation möglich. Dies ist unabhängig von den verschiedenen Variationen des Austausches. Zu beachten ist, dass beim Verknüpfen der Rückaustausch von Ergebnisdaten entfällt, da die Kennzeichnungen direkt am Originalmodell vorgenommen werden. Insgesamt ist der einbindende Datenaustausch bei der bidirektionalen Kopplung von Vorteil. Die Ergebnisdaten liegen auch beim einbettenden Austausch direkt im Format des Erzeugersystems vor. Das Objekt muss für den Austausch der Ergebnisdaten vom Simulationssystem nur an das Erzeugersystem zurückgegeben werden. Eine weitere Anpassung ist nicht notwendig.

Anderl [And93] unterscheidet *nach der Spezialisierung des Austausches* zwischen Verfahren zum generalisierten Datenaustausch und Verfahren zum spezialisierten Austausch. Dieses Klassifikationsschema wurde im Abschnitt 2.2 auf Seite 17 erläutert. Der Datenaustausch bei der Montagesimulation kann als spezialisierter Austausch realisiert werden, da die konkrete Anwendung und die Randbedienungen

bekannt sind. Bei der Realisierung sind zudem die zu koppelnden Systeme, Unternehmen und Unternehmensabteilungen hinreichend bekannt. Somit kann auf die konkreten Anforderungen des Anwendungsszenarios eingegangen werden. Nichtsdestotrotz sind Verfahren zum generalisierten Austausch möglich. Anzunehmen ist jedoch, dass im Gegensatz zu Verfahren des spezialisierten Datenaustausches Einschränkungen bei der Realisierung der konkreten Anforderungen zu erwarten ist.

Eine andere Möglichkeit ist die Klassifikation *nach den Interoperabilitätsleveln*. Dieses Schema wurde in Abschnitt 2.2 auf Seite 18 erläutert. Manso et al. unterscheiden in [MWB09] zwischen sieben verschiedenen Leveln. Die technische Interoperabilität wird in dieser Arbeit als gegeben angenommen. Für den Datenaustausch bei der Montagesimulation sind unabhängig von den verschiedenen Variationen die syntaktische und die semantische Interoperabilität zwingend notwendig. Die Systeme müssen sowohl in der Lage sein die Daten zu lesen als auch sie mit der richtigen Bedeutung zu interpretieren. Damit sind die syntaktische und die semantische Interoperabilität die Grundvoraussetzungen für einen Datenaustausch zwischen heterogenen Systemen. Vorteilhaft ist für das konkrete Beispielszenario auch die pragmatische Interoperabilität. Der Zugriff auf Funktionalitäten des Erzeugers könnte hier die Übertragung der Ergebnisdaten erleichtern. Dynamische Interoperabilität ist für dieses Anwendungsszenario weniger bedeutsam, da eine automatische Weitergabe von Änderungen der Modelle nicht notwendig ist. Montagesimulationen werden zu bestimmten Entwicklungsständen des Produktes durchgeführt. Eine stetige Weitergabe aller Änderungen der Bestandteile des Produktes ist nicht notwendig. Stattdessen kann die automatische Kopplung zu Problemen führen, wenn Änderungen gleichzeitig mit der Durchführung der Simulation erfolgen. Eine automatische Weitergabe der Daten kann dabei zu Inkonsistenz führen, sodass die Ergebnisse der Simulation möglicherweise fehlerhaft sind oder die Simulation fehlschlägt. Bei der Verwendung von Datenaustauschstrategien, die dynamische Interoperabilität realisieren, ist es hier notwendig Maßnahmen zu realisieren.

In Abschnitt 2.2 auf Seite 20 wurde die Klassifikation *nach Merkmalen der Realisierung* vorgestellt. Ein Merkmal ist die Anzahl unterschiedlicher Domänen. Allgemein ist beim der Montagesimulation von verschiedenen Domänen der Systeme auszugehen. Eine Domäne ist die Erzeugung der notwendigen Daten in Erzeugersystemen. Das Simulationssystem gehört zur Domäne der Simulation und Erprobung. Daher findet bei der Montagesimulation im Allgemeinen ein inter-phasen Austausch der Daten statt. Anders ist dies beim Fall C. Hier gehören alle Systeme den Erzeugersystemen an. Das System, das die Simulation durchführt, ist ein Subsystem eines der Erzeugersysteme und verwendet die Daten im Format dieses Erzeugersystems. Hier findet ein in-phasen Austausch zwischen den Systemen statt.

Ein weiteres vorgestelltes Merkmal der Realisierung ist die Art der Weitergabe von Änderungen. Unterschieden wird zwischen der Übertragung aller Daten bei einer Änderung und der Übertagung ausschließlich von Änderungen. Auch beim Beispielszenario tritt der Fall auf, dass aufgrund von Veränderungen die Montagesimulation erneut durchgeführt wird. Hier sind beide Varianten der Realisierung möglich. Vorteil der inkrementellen Übertagung ist die geringere Datenmenge, die übertragen werden muss. Der Nachteil ist, dass Änderungen von der Datenaustauschlösung erkannt werden müssen. Bei der Rückgabe von Ergebnisdaten vom Simulationssystem

zu den Erzeugersystemen ist es möglich, die Kennzeichnungen als Änderungen zu betrachten und ausschließlich diese zu übertragen. Dabei können verschiedene Probleme des Austausches von Ergebnisdaten umgangen werden.

Die Vollständigkeit der übertragenden Daten ist ein weiteres Realisierungsmerkmal. Im Fall A ist die Übertagung tatsächlich notwendiger Daten bei der Montagesimulation im Allgemeinen ausreichend. Nur der Fall B und C benötigen mehr Daten für den Aufbau der Modelle, die für die Montagesimulation nicht relevant sind.

Ein weiteres Realisierungsmerkmal, dass in Abschnitt 2.2 vorgestellt wurde, ist die Art des Austauschmanagements. Dieses Merkmal bezieht sich auf die Weitergabe von Änderungen bei Abhängigkeiten der Daten untereinander. Abhängigkeiten können im Beispielszenario nur innerhalb der Modelle eines Erzeugersystems und zwischen den Modellen verschiedener Erzeugersysteme bestehen. Dies bedarf einer Betrachtung bei der Erzeugung der Daten. Für die Montagesimulation ist dies nur relevant, wenn ein inkrementeller Datenaustausch realisiert wird. Dabei sind beide Varianten des Austauschmanagements möglich.

Die zu unterstützenden Prozesstypen der Produktentwicklung hängen von konkreten Anwendungsunternehmen ab. Prinzipiell sind alle Prozesstypen möglich. Diese sind dann bei der Realisierung des Datenaustausches zu beachten.

Voraussetzung für die Klassifikation *nach dem Datenaustauschformat* ist die Realisierung des Datenaustausches durch eine übersetzende Datenaustauschstrategie mit einem neutralen Format. Die Klassifikation bezieht sich hier auch auf das verwendete Format. Beschrieben wurde diese Klassifikation in Abschnitt 2.2 auf Seite 23. Dabei wurde auf verschiedene Eigenschaften von Datenaustauschformaten eingegangen.

Eine dieser Eigenschaften ist das Anwendungsfeld. Für 3D-Geometriedaten wurde dabei eine Klassifikation vorgestellt. Diese unterscheidet zwischen Formaten für CAD/CAM/CAE, für die Modellierung und Animation, für Echtzeit- und VR- Anwendungen und für das Internet. Diese Klassifikation kann für den Datenaustausch bei der Montagesimulation eingesetzt werden, da hier unter anderem dreidimensionale Geometriedaten auszutauschen sind. CAD/CAM/CAE-Formate sind für Anwendungsszenarien wie den Fall C ausgelegt. Sie sind allgemein für den Datenaustausch bei der Montagesimulation verwendbar. Jedoch sind keine Geometrieapproximation in diesen Formaten vorgesehen. Bei Modellierungs- und Animationsformaten und Echtzeit/VR-Formaten ist dies möglich. Nach Thierfelder [Thi04] beschreiben diese Formate komplette Szenen unter anderem mit mehreren Objekten, Licht, Effekten und bei Echtzeit/VR-Formaten mit Animationen und Interaktionsmöglichkeiten. Diese Daten können nicht alle Erzeugersysteme im Beispielszenario bereitstellen. Daher sind diese Formate nur bedingt verwendbar. Internetbasierte Formate sind ebenfalls verwendbar, jedoch auf einen anderen Zweck ausgerichtet. Daher sind diese Formate für das Beispielszenario nicht die optimale Lösung. Diese Ausführungen bestätigen, dass diese Klassifikation von Thierfelder [Thi04] kritisch zu betrachten ist. Das konkrete Beispielszenario ist in diesem Klassifikationsschema nur unzureichend abgebildet. Anwendungen ohne Szenen aber mit approximierten Geometrien werden nicht berücksichtigt. Diese Anwendungen haben ähnliche Anforderungen wie CAD/CAM/CAE-Anwendungen, erweitern diese jedoch um die Möglichkeit approximierter Geometrien.

Datenformate unterscheiden sich weiter nach dem Grad der Verbindlichkeit in primäre und sekundäre Formate. Beim Datenaustausch bei der Montagesimulation sind beide Arten von Formaten möglich. Die Zuordnung ist abhängig vom Stellenwert der Montagesimulation im Unternehmen. Ist die Montagesimulation nur eine zusätzliche Kontrolle, dann ist es ausreichend, die Daten zu Informationszwecken auszutauschen. Dies bedeutet, sekundäre Formate sind ausreichend. Hat die Montagesimulation einen höheren Stellenwert bei der Produktentwicklung, so müssen verbindliche Daten ausgetauscht werden. Dies geschieht über primäre Formate. Das Problem ist, dass primäre Formate häufig systemspezifisch sind und somit nicht für den Datenaustausch zwischen heterogenen Systemen geeignet sind. In diesem Fall erfolgt der Datenaustausch über sekundäre Formate.

Weitere vorgestellte Eigenschaften für Datenaustausch sind die Standardisierung, die Offenheit und die Leichtgewichtigkeit. Bei der Standardisierung und der Leichtgewichtigkeit gibt es im Beispielszenario keine Einschränkungen. Bezüglich der Offenheit sind offene Formate zu bevorzugen, da diese es erlauben eigene Software zur Verarbeitung zu realisieren. Dies erleichtert die Realisierung der Konverter und ermöglicht es, eigene Zusatzsysteme zu entwickeln.

Dieser Abschnitt hat die vorgestellten Klassen von Lösungsstrategien auf die Verwendbarkeit beim Datenaustausch bei der Montagesimulation untersucht. Bezug wurde dabei auf die Klassifikationsschemata aus Abschnitt 2.2 genommen. Die Ausführungen zeigen, dass viele Strategien als Lösungsstrategien für das Beispielszenario möglich sind. Dabei erfolgte ebenfalls die Untersuchung des Einflusses der Variationen auf die möglichen Klassen von Lösungsstrategien. Hier konnte gezeigt werden, dass diese für bestimmte Klassen von Lösungsstrategien von großer Bedeutung sind. Es konnten jedoch auch Einschränkungen und Voraussetzungen für den Einsatz einiger Strategien aufgezeigt werden.

Die Ergebnisse dieses Abschnittes sind in der Übersicht im Anhang (Tabelle A.1) durch farbige Hinterlegung der Klassen und Fußnoten zusammenfassend dargestellt. Der nächste Abschnitt zieht ein Fazit zu diesem Kapitel.

3.5 Zusammenfassung

Dieses Kapitel hat sich mit dem Anwendungsszenario Datenaustausch bei der Montagesimulation beschäftigt. Den Anfang bildete ein Überblick über die Montagesimulation. Dies diente dem besseren Verständnis des Kontextes, in dem das Anwendungsszenario angesiedelt ist. Danach wurde auf das Anwendungsszenario selbst eingegangen und verschiedene Variationen aufgezeigt. Dies beinhaltete Ausführungen zu den benötigten Daten in der Montagesimulation, der Anzahl an Erzeugersystemen, dem Simulationssystem und der Richtung des Austausches. Der darauffolgende Abschnitt hat sich mit den Anforderungen an den Datenaustausch beschäftigt. Dort konnte festgestellt werden, dass für die einzelnen Variationen verschiedene anwendungsszenariospezifische Anforderungen gelten. Dies bedeutet, dass diese Variationen beim Vergleich ausgewählter Strategien beachtet werden müssen. Neben den szenariospezifischen Anforderungen wurden zudem grundlegende allgemeine Anforderungen vorgestellt. Im vorletzten Abschnitt dieses Kapitels erfolgte die Untersuchung der in Abschnitt 2.2 vorgestellten Klassen von Lösungsstrategien bezüglich

der Verwendung für den Datenaustausch bei der Montagesimulation. Besondere Beachtung fanden hier die verschiedenen Variationen des Anwendungsszenarios. Insgesamt konnte in diesem Abschnitt gezeigt werden, dass viele verschiedene Klassen von Lösungsstrategien für das Beispielszenario in Frage kommen. Dabei sind jedoch bestimmte Voraussetzungen und Einschränkungen für einige Klassen zu beachten.

Zusammenfassend behandelte dieses Kapitel Variationen, Anforderungen und Klassen von möglichen Lösungsansätzen des Beispielszenarios. Damit beantwortet dieses Kapitel die zweite Fragestellung dieser Arbeit. Analog kann diese Betrachtung auch für andere Anwendungsszenarien im Ingenieurwesen erfolgen. Im nächsten Kapitel folgt ein Vergleich ausgewählter Lösungsstrategien für den Datenaustausch. Dabei wird auf die Klassifikationsschemata und die Eignung für das in diesem Kapitel vorgestellte Anwendungsszenario eingegangen.

4. Vergleich von Lösungsansätzen

Thema dieses Kapitels ist der Vergleich von Lösungsansätzen. Dabei ist es aufgrund der Vielzahl von Ansätzen nicht möglich, alle bestehenden Ansätze hier zu betrachten. Daher wurden vier Ansätze zum Vergleich ausgewählt. Dies sind die Datenaustauschformate STEP und JT sowie zwei Ansätze zur Online-Kopplung. Die beiden Datenaustauschformate sind in der Praxis sehr weit verbreitet [VSG+11, SVG11] und wurden deshalb ausgewählt. Die Online-Kopplung von CAx-Systeme mit einem CAx-Objektbus wurde vom Projekt ANICA entwickelt [SAKJ98, SAKJ00, AJSK98] und ist eine interessante Alternative zum klassischen übersetzenden Austausch. *Online-Kopplung* bedeutet dabei, dass alle Systeme zur Laufzeit miteinander gekoppelt werden und Daten austauschen. Ähnlich dem Ansatz des ANICA-Projekts ist eine neuere Datenaustauschstrategie, welche die Kopplung mit Hilfe von bestehenden OpenGL-Streams der Systeme realisiert. Dies ist der zweite Online-Kopplungsansatz, der in den Vergleich eingeht.

Im Folgenden werden die vier Lösungsansätze einzeln vorgestellt und verglichen. Dies beinhaltet die Einordnung der Ansätze in die Klassifikationsschemata aus Abschnitt 2.2. Zudem wird für jeden Lösungsansatz die Eignung für das Anwendungsszenario „Datenaustausch bei der Montagesimulation" diskutiert. Am Ende dieses Kapitels fasst ein Fazit die gewonnenen Erkenntnisse zusammen.

4.1 STEP

Dieser Abschnitt beschäftigt sich mit STEP als Datenaustauschlösung. Diese Abkürzung „STEP" ist die inoffizielle jedoch weit verbreitete Bezeichnung für die Normenreihe *ISO 10303* [Kal06, Pra01, AEK+00]. Daher verwendet auch diese Arbeit die Bezeichnung „STEP".

STEP selbst ist kein klassisches Datenaustauschformat, sondern die Spezifikation für eine Menge von anwendungsspezifischen neutralen Formaten [VSG+11, SVG11]. Weitere neutrale Formate sind dabei nach dem Baukastenprinzip erstellbar [VSG+11, SVG11].

Die Entwicklung von STEP begann 1984 durch die International Organization for Standardization (ISO) [Pra01] und baute auf ältere bestehende Datenaustauschformate wie IGES, SET und VDAFS auf [Kal06]. Erste Teile von STEP wurden 1994 veröffentlicht [Pra01]. Seitdem wurde die Norm stetig weiterentwickelt. Dabei spielt der ProSTEP iViP Verein eine wichtige Rolle für die Verbreitung und Weiterentwicklung des Standards [Han11].

Das Ziel ist die vollständige, digitale Abbildung eines Produktes über den gesamten Lebenszyklus zu realisieren [Kal06, Pra01, And93, GA90]. Dazu verwendet STEP einen Partialmodellansatz [VWB+09, Kal06]. Verschiedene Sichten auf ein Produkt innerhalb von Abschnitten des Produktlebenszyklus werden durch verschiedene Partialmodelle beschrieben. Die Gesamtheit aller Partialmodelle bildet das vollständige Produktmodell, das im Abschnitt 2.3 als virtuelles Produkt oder integriertes Produktmodell bezeichnet wird [VWB+09]. Dieser Ansatz erlaubt es zudem, dass STEP für verschiedene Branchen und Anwendungen entwickelt werden konnte [Kal06]. Der Nachteil dabei ist, dass die Integration zu einem Gesamtmodell die Definition von Randbedingungen erfordert [Kal06]. Die Eingabe und Überprüfung dieser Bedingungen kann bei der Implementierung und in der konkreten Anwendung problematisch sein.

STEPals Normenreihe besteht aus verschiedenen Teilen [Kal06, Pra01, AEK+00, AD00]. Diese Teile werden in sogenannten Serien gruppiert. Dabei ist die Nummer der Serie in der Dokumentnummer codiert. Beispielsweise gehört das Dokument ISO 10303-22 zur 20er Serie von STEP. Eine Serie behandelt dabei immer einen Aspekt der Norm. Insgesamt umfasst STEP neun Serien von Dokumenten, die der Spezifikation dienen. Im Folgenden wird ein kurzer Überblick über diese Serien gegeben. Für detailliertere Informationen zu den Bestandteilen von STEP wird an dieser Stelle auf die von Anderl und Trippner herausgegebene Arbeit [AT00] verwiesen. Von besonderem Interesse sind hier der zweite Teil [AEK+00] und der Anhang [AD00]. Abbildung 4.1 zeigt den Aufbau von STEP im Überblick.

Den Kern von STEP bilden die 0er und die 10er Serie. Die 0er Serie dient dabei der Einführung in die Norm und beschreibt den Aufbau, die Zielsetzung und die Konzepte der STEP-Normenreihe. Die Dokumente der 10er Serie definieren die Methoden für die Spezifikation. Dies beinhaltet die Datenbeschreibungssprache EXPRESS und deren graphische Repräsentation EXPRESS-G. Das Besondere an EXPRESS ist, dass die Schemata in dieser Sprache vom Rechner verarbeitet und verifiziert werden können [Pra01]. Dies bedeutet, dass neben der Syntax die Existenz von vorher bestimmten Verknüpfungen zu anderen Schemata überprüft werden kann. Hinsichtlich der Semantik ist es mit EXPRESS möglich, bei der Definition von Entitäten festgelegte Regeln zum Übersetzungszeitpunkt zu prüfen [Pra01]. EXPRESS selbst ist Teil der Norm und gehört zu den formalen Spezifikationssprachen. Für das Verständnis und die Implementierung der einzelnen Spezifikationen in STEP selbst sind Kenntnisse zu EXPRESS notwendig. Daher beschreiben Arlt et al. diese Sprache in [AEK+00] ausführlich. Für das weitere Verständnis dieser Arbeit ist dies hier nicht notwendig.

Methoden zur Implementierung und zur Konformitätsprüfung sind in den 20er, 30er und 300er Serien definiert. Die 20er Serie enthält die Normen für die Implementierung der Softwarelösungen. Dies beinhaltet die Normen für die Abbildung der

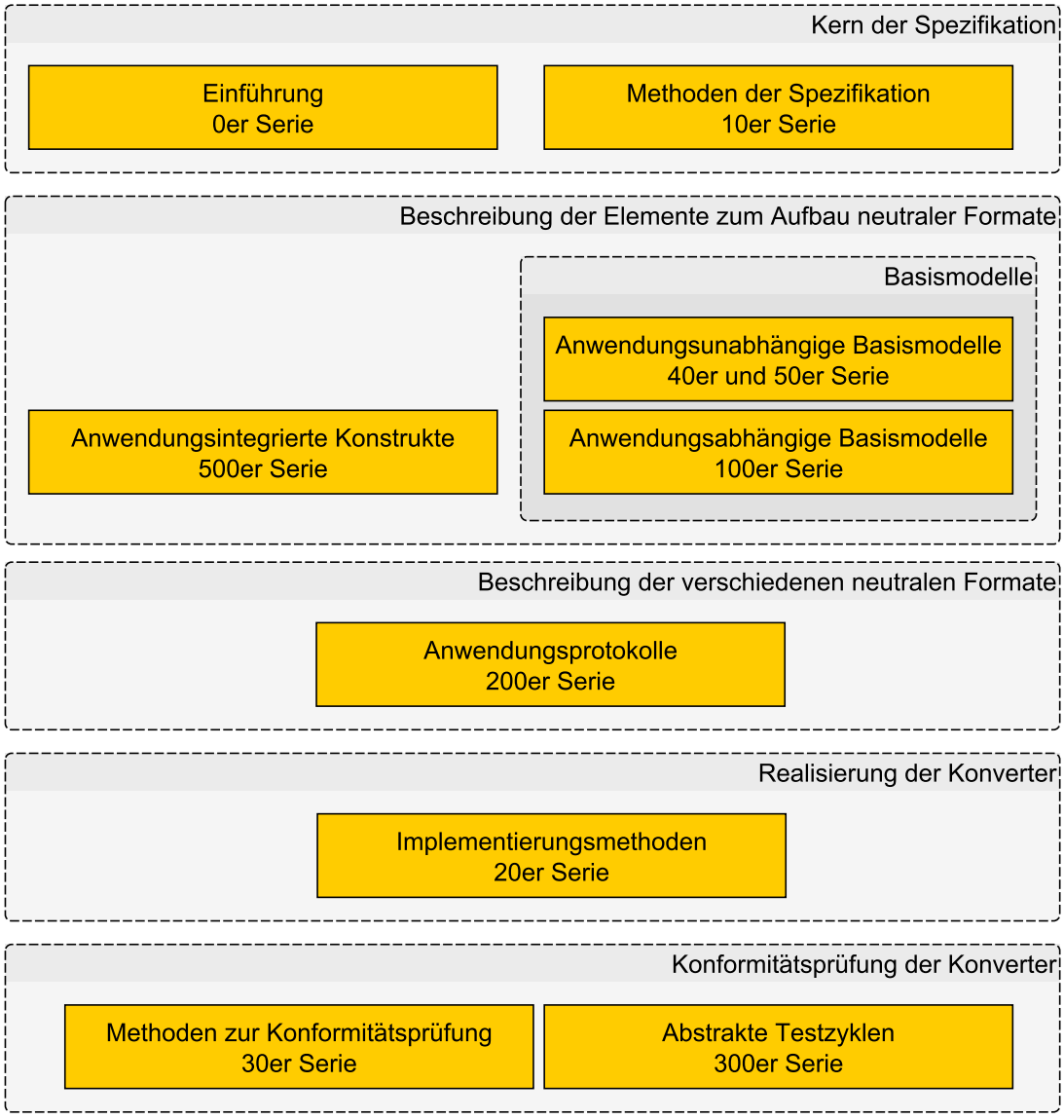

Abbildung 4.1: Aufbau von STEP (ISO 10303)

Daten in einer Datei. Dort werden neben dem Aufbau einer STEP-Datei auch die Schlüsselworte und die Abbildung von EXPRESS auf die Datenstrukturen definiert. Die 30er und die 300er Serien spezifizieren, wie eine konkrete Implementierung auf Konformität mit dem Standard zu prüfen ist. Die 300er Serie legt dabei für jedes Anwendungsprotokoll der 200er Serie die abstrakten Testzyklen fest. Abstrakt bezieht sich hier auf die Unabhängigkeit von der konkreten Implementierung.

Die bisher beschriebenen Serien beschäftigen sich noch nicht mit dem Produktmodell an sich. Dies geschieht in den 40er, 50er und 100er Serien. Hier wird das integrierte Produktmodell der STEP-Normenreihe behandelt. Die 40er und 50er Serie spezifiziert anwendungsunabhängige Merkmale von Produkten. Dies sind beispielsweise Geometriemodelle, Toleranzmodelle, Modelle zur Produktstruktur und zu den Materialeigenschaften. Gemeinsam ist diesen Modellen, dass sie nicht nur in einzelnen Anwendungsgebieten verwendet werden, sondern in einem breiten Bereich relevant sind. Anders ist dies bei den Modellen, die in den Dokumenten der 100er Serie definiert werden. Diese Modelle sind anwendungsspezifisch. Dazu gehören beispielsweise FEM-Modelle, Zeichnungsmodelle und Kinematikmodelle. Somit bilden die 40er, 50er und die 100er Serie die Basismodelle für das integrierte Produktmodell.

Die Anwendungsprotokolle werden in den Dokumenten der 200er Serie beschrieben. Anwendungsprotokolle dienen der Spezifikation von Produktmodellen für konkrete Anwendungsbereiche. Aufgebaut ist das spezifizierte Produktmodell aus den anwendungsunabhängigen und den anwendungsspezifische Basismodellen, die in den 40er und 100er Serien beschrieben sind. Somit legt ein Anwendungsprotokoll die Verwendung der Basismodelle für die konkrete Anwendung fest. Conformance Classes bilden dabei Implementierungseinheiten eines Anwendungsprotokolls, deren komplette Implementierung innerhalb eines Softwaresystems gefordert ist. Das bedeutet, alle beschriebenen Aspekte einer Conformance Class sollen realisiert werden. Eine Conformance Class beinhaltet dabei Anwendungsobjekte, die aus einer oder mehreren Funktionseinheiten des Anwendungsprotokolls stammen. Anwendungsprotokolle und Conformance Classes ermöglichen so eine benutzerorientierte Anpassung von STEP an das konkrete Anwendungsgebiet.

Die 500er Serie der STEP-Normenreihe beschreibt anwendungsintegrierte Konstrukte, die Datenstrukturen für die Verwendung in allen Anwendungsprotokollen festlegen. Ziel ist, eine Kompatibilität von Systemen mit gleichen Sachverhalten zu erreichen.

STEP wurde für den Einsatz in verschiedenen Anwendungsgebieten und Branchen entwickelt. Dies spiegelt sich auch im Aufbau der Normenreihe wieder. In der Praxis wird STEP weltweit in vielen Branchen auch tatsächlich erfolgreich eingesetzt [VWB+09]. Dazu gehören beispielsweise die Automobilindustrie, das Bauwesen, der Maschinen- und Anlagenbau, die Luftfahrtindustrie und der Schiffsbau [VWB+09, Kal06, Han11]. Die große Verbreitung von STEP ist der Grund für die Auswahl dieser Datenaustauschstrategie für diesen Vergleich.

Friedewald et al. [FLL+11] geben an, dass die Anwendungsprotokolle 203 und 214 dabei am häufigsten realisiert sind. Andere Anwendungsprotokolle sollen nach den Autoren weniger häufig realisiert sein. Weiter nennen die Autoren das Anwendungsprotokoll 242 als Zusammenfassung der Protokolle 203 und 214. Diese ist laut

ISO heute noch in der Entwicklung [ISOd] und kann daher nicht näher betrachtet werden.

Pratt [Pra01] beschreibt, dass das Anwendungsprotokoll 203 häufig zum Austausch von geometrischen Daten verwendet wird. Der Titel dieses Anwendungsprotokolls lautet „Configuration Controlled Design“ [Pra01, AT00]. Es dient dem Austausch von Daten über mechanische Teile und Baugruppen. Mit Hilfe des Anwendungsprotokolls 203 ist es möglich, 3D-Geometriedaten, Zusammenbaustrukturen und weitere Konfigurationsdaten zu übertragen. Zu den Konfigurationsdaten gehören hier beispielsweise Versionsnummern und Statusangaben. Der genaue Umfang an übertragbaren Daten inklusive des Typs des übertragbaren Geometriemodells ist abhängig von der realisierten Conformance Class. So erlaubt die Conformance Class 1 nur die Übertragung von Konfigurationsdaten ohne Geometriemodelle. Diese Klasse ist dabei Teil aller weiteren Klassen. Diese unterscheiden sich im Typ der übertragbaren Geometriemodelle zwischen Drahtmodellen mit Topologie, Flächenmodellen mit Topologie und facettierten oder erweiterten B-Rep-Modellen. Parametrik und akkumulative Modelle mit Modellhistorie sind in der Version von 1994 nicht enthalten. Nach Aussage von Pratt 2001 sind sie jedoch in Arbeit. Volumenmodelle mit Konstruktionshistorie sind in der aktuellsten Version von 2011 laut ISO [ISOa] bereits enthalten. Angaben zur Parametrik werden nicht gemacht.

Das Anwendungsprotokoll 214 „Core data for automotive mechanical design processes“ wurde, wie der Titel bereits sagt, für den automotiven Anwendungssektor entwickelt. Die Entwicklung begann 1992 [AEK+00]. Die erste vollständige Version dieses Anwendungsprotokolls wurde erst 2001 [ISOb] veröffentlicht. Darauf folgten eine Version 2003 und die letzte Version 2010 [ISOc]. Mit Hilfe des Anwendungsprotokolls können nach Friedewald et al. [FLL+11] verschiedene Typen von Geometriemodellen, Hilfsgeometrien, Farben, Materialeigenschaften, Metadaten, kinematische Beziehungen und Baugruppenstrukturen übertragen werden. 2003 wurde vom ProSTEP iViP Verein [Pro03] eine Untersuchung verschiedener Prozessoren für das STEP Anwendungsprotokoll 214 durchgeführt. Diese Untersuchung konnte eine steigende Qualität der Prozessoren feststellen. Die Übertragung von Volumenmodellen einzelner Bauteile gilt als stabil. Jedoch wurden auch Unterschiede bezüglich der übertragenen Daten bei bestimmten Sender-Empfänger Kombinationen festgestellt. Untersucht wurde die Übertagung von Bauteile als Volumenmodelle und von Baugruppen. Friedewald et al. [FLL+11] haben ebenfalls verschiedene Prozessoren von CAD-Systemen auf die Konvertierungsergebnisse hin untersucht. Dabei sind in den Ergebnissen ebenfalls Unterschiede bei den einzelnen Prozessoren sichtbar. Dies betrifft insbesondere die Hilfsgeometrie, Farbinformationen und Metadaten. Gründe dafür werden jedoch nicht genannt. Weiter führen die Autoren an, dass bei der Implementierung von STEP teilweise die Conformance Classes missachtet werden. Die Folge davon sind Probleme beim Datenaustausch, da die Prozessoren unterschiedliche Umfänge an Daten übersetzten können.

Insgesamt gesehen bringt STEP einige Probleme mit sich. STEP ist sehr umfangreich und komplex [Avg97, Kal06, Han11, Sto11]. Dies hat zur Folge, dass die Implementierung von Prozessoren aufwendig und teuer ist [Avg97, Sto11]. Der Umfang der übertragbaren Daten ist abhängig vom realisierten Anwendungsprotokoll [Sto11]. Diese sind in der Regel inkompatibel zu einander [Kal06]. Damit ist es notwendig,

verschiedene Konverter für die Anwendungsprotokolle zu realisieren. Diese müssen zudem verwaltet und gepflegt werden. Damit kann jedes Anwendungsprotokoll als ein neutrales Format betrachtet werden. Nachteilig für die Implementierung ist außerdem, dass die Spezifikation zwar grundsätzlich offen ist, jedoch kostenpflichtig bei der ISO erworben werden muss [Sto11]. Zudem besitzt STEP keine Mechanismen zur Minimierung der Dateigrößen [FLL+11, Ger10]. Dies hat nach Friedewald et al. [FLL+11] zur Folge, dass STEP nicht in allen Prozessen verwendet werden kann. Eine weitere Folge sind große Dateien, die sich nachteilig auf den Datenversand auswirken.

Verbreitet ist STEP hauptsächlich im CAD-nahen Bereich. CAD-ferne Bereiche unterstützen STEP kaum [Han11]. Dies ist auch in den Ergebnissen von Friedewald et al. [FLL+11] sichtbar. Hier sind die Importmöglichkeiten für STEP-Dateien bei Echtzeit/VR-Systemen begrenzt.

Neben diesen Nachteilen und Problemen bietet STEP auch verschiedene Vorteile. Mit STEP kann eine Vielzahl verschiedener Daten ausgetauscht werden. STEP ist durch seine Anwendungsprotokolle an die Bedürfnisse einzelner Anwendungsbereiche angepasst. Für CAD-Systeme stehen bereits verschiedene Konverter zur Verfügung, die direkt vom Anwender verwendet werden können. Jedoch sind diese Realisierungen häufig unterschiedlich [Pro03, FLL+11], was zu Fehlern und Verlusten bei der Übertagung führt. Ein weiterer Vorteil von STEP ist, dass eine stetige Weiterentwicklung des Standards stattfindet. Damit ist STEP auch zukünftig relevant.

Im folgenden Unterabschnitt wird STEP allgemein in die Klassifikationsschemata aus Abschnitt 2.2 eingeordnet.

Einordnung in Klassifikationsschemata

Im Abschnitt 2.2 wurden verschiedene Klassifikationsschemata für Datenaustauschstrategien vorgestellt. In diese kann auch STEP eingeordnet werden. In der Übersicht im Anhang sind die Klassen, zu denen STEP gehört, durch ein x gekennzeichnet.

Bei der Einteilung *nach der Art der austauschbaren Daten* wird zwischen der Einteilung nach produktionstechnischen Aspekten und der Einteilung nach dem Geometriegehalt der Daten unterschieden. STEP dient primär dem Austausch von Produktdaten. Dies wird bereits durch die Bezeichnung „Standard for the Exchange of Product Model Data“ deutlich. Aufgrund der Ähnlichkeit der Daten und der Tatsache, dass Werkzeuge und Maschinen ebenfalls Produkte sind, ist es zudem möglich, bestimmte Daten zu diesen Betriebsmitteln mittels STEP auszutauschen. Diese Daten beziehen sich dabei auf Aspekte, die auch bei Produktdaten bestehen. Dies sind beispielsweise die Geometrie und Kinematik von Werkzeugen und Maschinen. Für den Austausch anderer prozessbezogener Daten ist STEP nicht ausgelegt.

Grundsätzlich können mit STEP sowohl geometrische als auch nicht-geometrische Daten ausgetauscht werden. Deutlich wird dies bereits bei den Ausführungen zu den Anwendungsprotokollen 203 und 214 (zusammen auf der vorherigen Seite). Der genaue Umfang der übertragbaren Daten hängt bei STEP von den realisierten Anwendungsprotokollen und Conformance Classes ab.

Bei der Klassifikation nach der *Systemneutralität* der Datenaustauschstrategie gehört STEP eindeutig in die Klasse der systemneutralen Strategien. Die neutrale

Zwischenstufe ist beim Datenaustausch mit STEP die STEP-Datei mit dem durch das Anwendungsprotokoll spezifizierten Format. STEP selbst definiert eine ganze Reihe verschiedener neutraler Datenaustauschlösungen. Diese sind durch die Anwendungsprotokolle und deren Conformance Classes definiert und auf die Anforderungen verschiedener Anwendungsgebiete ausgelegt.

Nach der Einteilung bezüglich des *Konzeptes des Datenaustausches* gehört STEP zu den übersetzenden Datenaustauschstrategien mit einem neutralen Format. Diese Einordnung trifft bereits Schumann [Sch01] bei der Vorstellung des Klassifikationsschemas. Damit werden bei der Verwendung eines STEP-Formates für den bidirektionalen Austausch pro System und STEP-Format zwei Konverter benötigt. Soll ein unidirektionaler Austausch von einem oder mehreren Sendern zu einem Empfänger realisiert werden, so benötigt jeder Sender einen Konverter vom eigenen Format in das STEP-Format. Der Empfänger benötigt dann für jedes eingehende STEP-Format einen Konverter von diesem Format in das eigene systemspezifische Format. Vorteilhaft ist es hier, wenn der Austausch über ein STEP-Format also ein Anwendungsprotokoll erfolgen kann und nicht mehrere notwendig sind. Dazu muss dieses Protokoll jedoch alle notwendigen Daten abbilden können.

Die Anwendungsprotokolle sind auch für die Klassifikation nach der *Spezialisierung des Datenaustausches* relevant. STEP und seine spezifizierten Formate realisieren einen generalisierten Datenaustausch. STEP als Ganzes hat das Ziel, Produktdaten möglichst universell austauschen zu können. Die Anwendungsprotokolle stellen zwar eine Spezialisierung auf bestimmte Anwendungsgebiete dar, doch innerhalb dieser können die neutralen STEP-Formate weitgehend universell eingesetzt werden. Es bestehen keine Abhängigkeiten mit den konkreten Systemen, dem konkreten Unternehmen und anderen Randbedingungen. STEP, als Ganzes betrachtet, kann zudem durch die verschiedenen Anwendungsprotokolle universell in verschiedenen Branchen und Anwendungsgebiete verwendet werden. Damit ist der Standard insgesamt weitgehend universell einsetzbar und die Anwendungsprotokolle sind in ihren bestimmten Anwendungsgebieten universell einsetzbar.

Die Klassifikation nach den *Interoperabilitätsleveln* unterscheidet verschiedene Interoperabilitätslevel. Für den Datenaustausch mit STEP sind die syntaktische und semantische Interoperabilität von Bedeutung. In STEP wird die Syntax durch die im Standard enthaltene Beschreibungssprache EXPRESS festgelegt. Semantische Aspekte werden durch die Spezifikationen für die Basisressourcen und die Anwendungsprotokolle behandelt. Pragmatische und dynamische Interoperabilität sind durch den Standard nicht abgedeckt. Der Standard schließt diese Level jedoch nicht aus. Daher können sie bei der konkreten Realisierung zusätzlich berücksichtigt werden. Für pragmatische Interoperabilität muss dazu der gegenseitige Zugriff von Systemen auf Systemfunktionalitäten realisiert werden. Dynamische Interoperabilität kann durch ein Überwachungssystem realisiert werden. Dieses überwacht die Sendesysteme bezüglich Änderungen der Daten und stößt bei Änderungen einen Datenaustausch über STEP automatisch an.

Bei der Klassifikation nach *Realisierungsmerkmalen* werden verschiedene Merkmale betrachtet. Eines dieser Merkmale ist die Anzahl unterschiedlicher Domänen, zwischen denen ein Datenaustausch unterstützt wird. STEP ist sowohl für den Austausch innerhalb von Domänen als auch für den Austausch zwischen verschiedenen

Domänen geeignet. Voraussetzung ist hier, dass das verwendete Anwendungsprotokoll alle benötigten Daten unterstützt. Ist dies der Fall, so können prinzipiell Konverter in den verschiedensten Domänen realisiert werden. In der Praxis ist jedoch zu beobachten, dass STEP-Konverter im CAD-fernen Bereichen weniger verbreitet sind [FLL+11, Han11]. Konkrete Untersuchungen zum Austausch über den CAx-Bereich hinaus sind jedoch nicht bekannt. Es wird jedoch angenommen, dass hier aufgrund unterschiedlicher Strukturen und Sichtweisen der Programme und Daten die Verwendung von STEP schwierig ist. Dies würde die geringe Verbreitung in diesen Bereichen erklären.

Li [Li10] unterscheidet bei der Art der Weitergabe von Änderungen zwischen einem vollständigen und einem inkrementellen Austausch. Die ISO 10303 selbst enthält keine Maßnahmen für einen inkrementellen Austausch. Diese müssten somit gegebenenfalls zusätzlich bei der Realisierung implementiert werden. Grundsätzlich sieht STEP nur einen vollständigen Austausch vor. Dies wirkt sich auch auf das Realisierungsmerkmal der Austauschkontrolle von Änderungen aus. Der Autor ordnet STEP zwar der zentralen Austauschkontrolle zu, doch liefert er dazu keine genauen Erläuterungen. Denkbar ist, dass sich dies hier auf das integrierte Produktmodell und die Abhängigkeiten zwischen den Partialmodellen bezieht. Änderungen werden demnach bei einem bei einem Datenmanagement der Partialmodelle zentral an die verschiedenen Partialmodelle weitergeleitet. Das dazu notwendige Datenmanagement ist nicht Teil der Spezifikation. Bei einem einfachen Datenaustausch mittels STEP-Formaten müssen die Abhängigkeiten vom Sendesystem verwaltet werden, um Datenkonsitenz zu gewährleisten.

Die Vollständigkeit der übertragenen Daten ist ein weiteres Realisierungsmerkmal. Beim Datenaustausch mit einem STEP-Format erfolgt in der Regel ein Austausch vollständiger Daten. Das bedeutet, alle Daten, die das Format abbilden kann, werden übertragen. Diese können über die für das konkrete Anwendungsszenario benötigten Daten hinausgehen. Der Grund dafür ist, dass die Konverter in der Regel nicht für nur ein bestimmtes Anwendungsszenario entwickelt werden. Je nach Realisierung kann es jedoch möglich sein, den Umfang der zu übersetzenden Daten beim Konverter einzustellen. Dies ist von der konkreten Realisierung des Konverters abhängig und kann nicht als allgemein gültig betrachtet werden. Das Anwendungsprotokoll und die realisierte Conformance Class legen die abbildbaren Daten des STEP-Formates fest. Das heißt, für das Anwendungsprotokoll beziehungsweise die Conformance Class benötigte Daten werden übertragen. Das konkrete Anwendungsszenario kann dabei weniger Daten benötigen. Für das Realisierungsmerkmal bedeutet das, dass die Daten des Anwendungsprotokolls beziehungsweise der realisierten Conformance Class vollständig übertragen werden und nicht nur für das konkrete Anwendungsszenario tatsächlich benötigte Daten.

Nach Li [Li10] ist STEP für linear Engineering-Prozesse ausgelegt. Spezielle Anforderungen des simultanous oder concurrent Engineering werden durch den Standard nicht abgedeckt.

STEPspezifiziert verschiedene *Datenaustauschformate*, daher ist eine Klassifikation nach Eigenschaften von Austauschformaten möglich.

Thierfelder [Thi04] ordnet STEP in der von ihm vorgestellte Klassifikation von Datenaustauschformaten für 3D-Geometrien in die Klasse der CAD/CAM/CAE-Formate ein. Diese Einordnung wird für die Anwendungsprotokolle 203 und 214 durch die Ausführungen im vorherigen Abschnitt unterstützt. Diese STEP-Formate erlauben den Austausch exakter Geometrien und werden für den Datenaustausch zwischen CAD-Systemen eingesetzt. Inwieweit die Einordnung auf alle Anwendungsprotokolle zum Austausch von 3D-Geometrien zutrifft, kann an dieser Stelle aufgrund fehlender, detaillierter Informationen zu diesen Protokollen nicht beurteilt werden. Dazu sind weitere Untersuchungen notwendig.

Auch was die Einordnung nach dem Grad der Verbindlichkeit betrifft, sind für STEP keine endgültigen Aussagen an dieser Stelle möglich, da hier keine Untersuchungen diesen Aspekt betreffend bekannt sind. Nach den Ausführungen von Handschuh [Han11] scheint STEP in der Praxis nicht als primäres Format genutzt zu werden. Der Autor spricht davon, dass in der Praxis häufig systemspezifische Formate diese Rolle besitzen. Dennoch kann die Verwendung von STEP als primäres Format nicht ausgeschlossen werden. Für eine solche Verwendung von STEP sprechen der große Umfang übertragbarer Daten und die große Verbreitung von STEP im CAD-Bereich. Voraussetzung für diesen Einsatz von STEP als primäres Format ist das unternehmensweite Vorhandensein qualitativ hochwertiger Konverter. Für konkretere Aussagen zu STEP als primäres Format sind jedoch nähere Untersuchungen notwendig.

STEPist ein veröffentlichter Standard der ISO. Der Zugriff auf die Spezifikation ist jedoch kostenpflichtig.

Der Standard besitzt selbst keine Mechanismen zur Minimierung der Dateigröße [FLL+11, Ger10]. Zudem werden STEP-Dateien im ASCII-Format und nicht in binärer Form gespeichert [Han11, Ger10]. Dies hat den Vorteil, dass die Dateien für Anwender lesbar sind. Der Nachteil ist jedoch die Dateigröße. Allgemein wird STEP daher nicht zu den leichtgewichtigen Formaten gezählt [Han11].

Dieser Abschnitt hat STEP in die Klassifikationsschemata aus Abschnitt 2.2 eingeordnet. Dabei wurde festgestellt, dass nicht immer eindeutige Aussagen für alle STEP-Formate getroffen werden können. Dazu sind nähere Untersuchungen dieser Aspekte im konkreten Anwendungsszenario und an den konkreten STEP-Formaten notwendig. Zudem wird STEP heute noch immer weiterentwickelt. Neue Anwendungsprotokolle werden erstellt und bestehende Teile des Standards aktualisiert und verändert. Daher können alle Betrachtungen dieser Arbeit nur als Momentaufnahme aufgefasst werden.

Der nächste Abschnitt beurteilt STEP für die Verwendung für das Beispielszenario „Datenaustausch bei der Montagesimulation“. Dabei wird auf die verschiedenen Variationen des Szenarios eingegangen.

Beurteilung für das Anwendungsszenario

Dieser Abschnitt untersucht die Eignung von STEP für den Datenaustausch bei der Montagesimulation. Der Datenaustausch bei der Montagesimulation wird in dieser Arbeit als Beispielszenario für den Vergleich der Datenaustauschlösungen verwendet.

Dieses Anwendungsszenario wurde in Kapitel 3 vorgestellt. Dabei wurden neben Anforderungen auch Variationen des Datenaustausches identifiziert. Diese gehen in die Beurteilung ein.

Ein Aspekt, in dem der Datenaustausch bei der Montagesimulation variiert, ist der *Umfang der zu übertragenden Modelle*. Variiert wird hier zwischen 3D-Geometriemodellen und statischen oder dynamischen DMUs. Bei 3D-Geometriemodellen sind geometrische Daten auszutauschen. Baugruppenstrukturen kommen bei statischen DMUs noch hinzu. Bei dynamischen DMUs wird dies weiter durch Daten zur Kinematik ergänzt. Für STEP bedeutet dies, dass nach der Beschreibung der Anwendungsprotokolle 203 und 214 beide grundsätzlich bezüglich den übertragbaren Daten geeignet sind. Das Anwendungsprotokoll 203 beschreibt den Austausch von 3D-Geometriemodellen und statischen DMUs. Im Anwendungsprotokoll 214 sind zudem kinematische Beziehungen enthalten, was eine Übertragung von dynamischen DMUs ermöglicht. Dennoch ist zu beachten, dass die verschiedenen Conformance Classes der Anwendungsprotokolle unterschiedliche Mengen von Daten und unterschiedliche Modelle unterstützen. Zudem hängt der Umfang der übertragbaren Daten von der Realisierung der Konverter ab. Hier sind Unterschiede bei verschiedenen Benchmarks zwischen den einzelnen Realisierungen festzustellen. Solche Benchmarks wurden beispielsweise vom ProSTEP iViP Verein [Pro03] und Friedewald et al. [FLL+11] durchgeführt. Die Konverter der konkreten Systeme müssen daher für die Realisierung des Datenaustausches bei der Montagesimulation im Unternehmen betrachtet werden.

Allgemein sind nach den Anforderungen bei den Geometriemodellen in der Montagesimulation Approximationen möglich und zum Teil vorteilhaft. Bei den anwendungsintegrierten Konstrukten der 500er Serie in der ISO 10303-512 werden Facettenmodelle beschrieben [SK97]. Laut Spezifikation unterstützt STEP somit approximierte Geometrien. Inwieweit existierende Konverter diese Form der Geometriemodelle unterstützen ist nicht bekannt. Hier sind nähere Untersuchungen notwendig.

Weiterhin variiert der Datenaustausch bei der Montagesimulation in der *Zahl der Erzeugersysteme* zwischen einem oder wenigen und vielen Systemen. STEP als systemneutrale Datenaustauschstrategie ist bei vielen Sendesystemen bezüglich dem notwendigen Aufwand für Entwicklung und Pflege vorteilhaft. Dies wird durch die große Verbreitung von STEP in vielen CAD-Systemen weiter unterstützt. Bei einem oder wenigen Systemen, die bisher keinen STEP-Konverter besitzen, ist STEP hingegen weniger geeignet, da hier mit gleichem oder geringerem Aufwand systemspezifische Lösungen realisierbar sind. Diese haben den Vorteil eines größeren und angepassteren Datenumfangs beim Austausch.

Die *Art des Simulationssystems* ist ein weiteres Merkmal des Datenaustausches, das variiert wird. Hier unterscheidet diese Arbeit drei Fälle. Im Fall A bei einem eigenständigen Simulationssystem ist ein Datenaustausch mit STEP prinzipiell möglich, solange das Simulationssystem eine zu STEP kompatible Datenstruktur verwendet. Besteht kein STEP-Konverter so ist hier ein Konverter für das STEP-Format zu realisieren. Diese Ausführungen gelten auch für Simulationssysteme, die ein Modul eines größeren Systems sind, jedoch kein vollständiges Modell im Format des größeren Systems benötigen. Pauschale Aussagen, ob STEP für den Datenaustausch zu einem speziellen System geeignet ist, sind jedoch nicht möglich. Dies hängt vom

konkreten Simulationssystem und dessen internen Strukturen ab. Auch für den Fall B sind keine pauschalen Aussagen möglich, da hier die notwendigen Importdaten und die Kompatibilität der Strukturen pauschal nicht bekannt sind. Somit hängt die Eignung von STEP auch hier vom konkreten System und dessen interner Struktur ab. Für den Fall C ist STEP hingegen in der Regel geeignet, da CAD-Systeme in der Regel eine kompatible Struktur besitzen und die STEP Anwendungsprotokolle 203 und 214 alle grundsätzlich notwendigen Daten unterstützen. Vorteilhaft bei der Verwendung von STEP für diesen Fall ist zudem, dass viele CAD-Systeme bereits vom Hersteller aus Konverter zum Export und Import von STEP-Dateien besitzen [FLL+11, Pro03]. Zu beachten ist dabei jedoch stets, welche Anwendungsprotokolle und Conformance Class vom konkreten System unterstützt werden und wie gut diese Unterstützung ist.

Schließlich variiert der Datenaustausch bei der Montagesimulation noch in der *Richtung des Austausches*. Beim Austausch mit STEP als dateibasierte Austauschstrategie sind der Austausch von Eingangsdaten in das Simulationssystem und der Austausch vom Ergebnisdaten aus dem Simulationssystem als getrennte eigenständige Austauschprozesse zu betrachten. Grundsätzlich treten bei beiden Austauschvorgängen im Beispielszenario die gleichen Probleme auf. Das STEP-Format muss alle zu übertragenden Daten abbilden können. Diese sind bei beiden Austauschvorgängen fast gleich. Beim Austausch der Ergebnisdaten kommen nur die Kennzeichnungen hinzu. Damit ist das Anwendungsprotokoll 203 nicht zum Austausch der Ergebnisdaten geeignet, da diese Daten nicht unterstützt werden. Dafür unterstützt das Anwendungsprotokoll 214 Farben. Das Problem dabei ist, dass bestehende Konverter laut der Untersuchung von Friedewald et al. in [FLL+11] dies nicht im gleichen Maße realisieren. Dies muss daher für die konkreten Systeme bei der Realisierung des Anwendungsszenarios untersucht werden. Unklar ist weiterhin, ob das Anwendungsprotokoll 214 Modellierhistorien übertragen kann. Daher ist keine Aussage möglich, ob editierbare Modelle an CAD-Systeme, die diese Art von Daten benötigen, übertragen werden können. Weitere Aussagen zu beliebigen Erzeuger- und Bearbeitungssystemen sind ebenfalls nicht möglich, da hierzu Wissen zu den Spezifika der Importdaten der einzelnen Systeme notwendig ist.

Allgemein gilt für die Realisierung des Datenaustausches mit Hilfe von STEP die Voraussetzung, dass Sender und Empfänger eine zum STEP-Format kompatible Struktur der Daten unterstützen müssen. Ist dies nicht der Fall, so kann grundsätzlich kein Datenaustausch mit STEP erfolgen. Diese Voraussetzung ist an den konkreten Systemen, die am Datenaustausch beteiligt sind, zu untersuchen. Für die am Beispielszenario möglicherweise beteiligten Systeme kann an dieser Stelle keine allgemeingültige Aussage getroffen werden, da verschiedenste Erzeuger- und Simulationssysteme an der Montagesimulation beteiligt sein können.

Dieser Abschnitt hat sich mit der Eignung von STEP für den Datenaustausch bei der Montagesimulation beschäftigt. Insgesamt ist STEP unter bestimmten Bedingungen für das Beispielszenario geeignet. Dabei bringt diese Strategie bestimmte Vorteile und Probleme mit sich. Die Bedingungen, Vorteile und Probleme wurden in diesem Abschnitt erläutert. Die grundlegende Bedingung für den Einsatz von STEP sind kompatible Strukturen bei den beteiligten Systemen. Es können sowohl 3D-Geometriemodelle wie auch statische und dynamische DMUs übertragen werden.

Auch der Austausch von Ergebnisdaten ist prinzipiell möglich, hängt jedoch auch von den konkreten Systemen ab. Vorteilhaft ist die große Verbreitung von STEP im CAD-Bereich. Die Nachteile von STEP sind neben dem Umfang und dem damit verbundenen Implementierungsaufwand, der Preis für die Spezifikationsdokumente und die unterschiedlichen Realisierungen der Konverter durch die Systemhersteller. Beachtet werden muss zudem, welches Anwendungsprotokoll und welche Conformance Class jeweils realisiert ist.

Der nächste Abschnitt betrachtet ein weiteres Datenaustauschformat, das ähnlich wie STEP in der Industrie weit verbreitet ist.

4.2 JT

Neben STEP ist Jupiter Tessellation (JT) ein anderes weit verbreitetes Datenaustauschformat im Ingenieurwesen [SVG11, VSG+11, Fac05]. Im Gegensatz zu STEP beschreibt die Spezifikation von JT keine Menge von Formaten, sondern genau ein Format, das JT-Format [SIE]. Damit ist JT ein Datenformat im klassischen Sinn. Es müssen somit keine verschiedenen Ausprägungen von JT in der Form, wie bei STEP betrachtet werden. Daraus ist zu folgern, dass für jedes System genau ein JT-Konverter für den Import und ein Konverter für den Export der JT-Dateien notwendig sind.

Deisinger beschreibt in [Dei10] die Entstehung und die bisherige Entwicklung des JT-Formates. Das Format wurde ursprünglich von Hewlett Packard und Engineering Animation Inc. 1997 entwickelt. Die Engineering Animation Inc. wurde 1999 durch UGS gekauft. Dadurch wurde JT Teil dessen Produktpalette. Heute gehört JT zur Siemens PLM Software Inc. Seit der ersten Spezifikation von JT wurde das Format stetig weiterentwickelt. Die Version 9.5 Revision D [SIE] ist heute die aktuellste Version. Die Formatspezifikation wurde erstmals 2007 veröffentlicht. Seitdem ist diese Spezifikation kostenfrei online verfügbar. Heute ist die Version 8.1 der JT-Spezifikation ein öffentlich zugänglicher ISO-Standard mit der Kennung ISO/PAS 14306:2011 [ISOf]. Der Standard wird dabei stetig überarbeitet und aktualisiert. Zum jetzigen Zeitpunkt ist die zweite Version noch nicht fertig. Sie befindet sich jedoch bereits am Ende der Prüfung (Phase 40). Bis zur Veröffentlichung (Phase 60) muss nur die Zustimmungsphase (Phase 50) noch durchlaufen werden [ISOe].

Zur Förderung der Verbreitung von JT wurde 2003 die JT-Open Initiative gegründet. In Deutschland unterstützen zudem der ProSTEP iViP Verein und der Verband der Automobilindustrie (VDA) die Verbreitung von JT. Dies erfolgt beispielsweise durch Untersuchung von realen Anwendungsszenarien in der Industrie.

Ziel bei der Entwicklung von JT war und ist ein Format, dass einfach in die bestehenden Unternehmensprozesse eingefügt und zur Unterstützung aller die Produktgeometrie betreffenden Downstream-Prozesse genutzt werden kann [SIE]. Downstream-Prozesse sind Prozesse, die Ergebnisse der Konstruktion näherer betrachten. Dazu gehören neben der reinen Darstellung der Produkte und seiner Bestandteile beispielsweise DMU-Untersuchungen wie die Montagesimulation. Im Fokus steht die Wiederverwendung erstellter digitaler Geometriemodelle [SIE].

JTist im Gegensatz zu STEP ein visualisierungs-orientiertes Format, bei dem kleine Dateigrößen von großer Bedeutung sind [Ger10]. Dies spiegelt sich auch im Aufbau

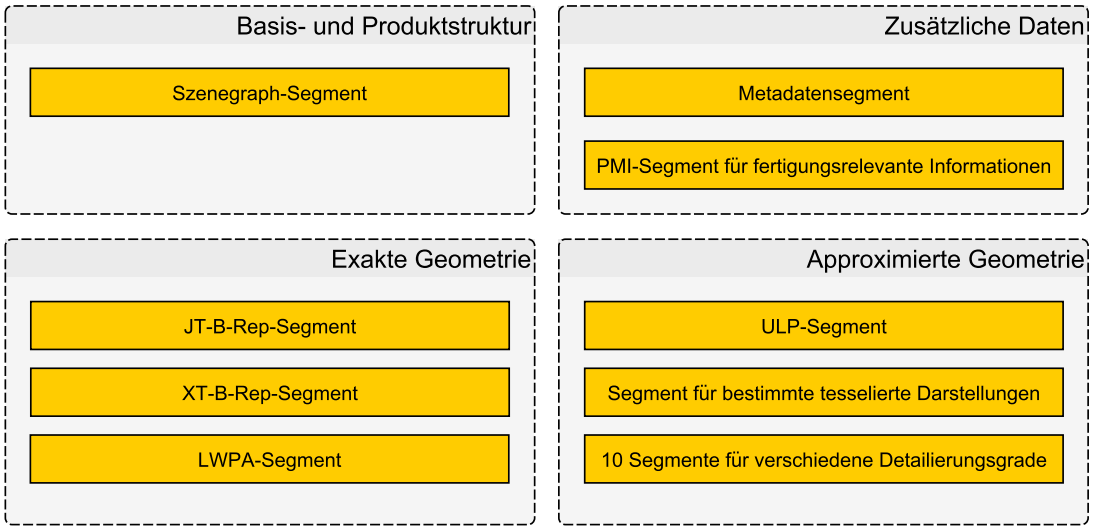

Abbildung 4.2: Mögliche Segmente einer JT-Datei nach [SIE]

von JT wieder. An dieser Stelle erfolgt eine kurze Betrachtung der Bestandteile von JT. Für detailliertere Informationen zum konkreten Aufbau einer JT-Datei wird auf die aktuelle Spezifikation [SIE] verwiesen.

Eine JT-Datei besteht aus bis zu 19 verschiedenen Arten von Datensegmenten. Jedes dieser Segmente beinhaltet verschiedene Daten, die vom JT-Format übertragen werden können. Je nach konkretem Anwendungsfall und der konkreten Konfiguration sind diese Segmente unterschiedlich gefüllt [Dei10]. Jedes dieser Segmente kann dabei einzeln adressiert und ausgelesen werden [Ger10]. Dadurch ist es möglich, einzelne Aspekte des abgebildeten Produktes näher zu betrachten [Ger10]. Es ist nicht notwendig, alle Daten dazu auszulesen. Eine JT-Datei kann so auch auf mehrere Dateien aufgeteilt werden, die auf verschiedenen physikalischen Speichern verteilt sein können [Ger10]. Die möglichen Datensegmente einer JT-Datei sind in Abbildung 4.2 dargestellt. Die Position eines Segmentes innerhalb der Datei ist nicht global festgelegt, sondern in einer Tabelle am Anfang der Datei beschrieben.

Die Basis jeder JT-Datei ist der logische Szenengraph. Ein *Szenengraph* ist eine objektorientierte Datenstruktur, die zur Beschreibung von komplexen Szenen in der Computergraphik verwendet wird [Wik]. Der Szenengraph einer JT-Datei ist azyklisch und gerichtet. Dadurch entsteht eine Teile-Ganzes Struktur, welche die Produktstruktur darstellt. Dazu besteht der Graph aus verschiedenen Arten von Knoten, diese beschreiben entweder Gruppenzugehörigkeiten oder sind Referenz auf weitere JT-Datei oder auf die anderen Segmente der Datei. Neben Knoten enthält der Szenengraph Attributelemente, die Eigenschaften der Geometrieelemente wie Farben beschreiben, und atomare Eigenschaftselemente, die Metadaten zu Attributelementen oder Knoten enthalten können. Insgesamt betrachtet, beschreibt dieses Datensegment die Produktstruktur mit bestimmten Attributen und Metadaten.

Die Geometrie kann mit JT in vier verschiedenen Repräsentationsformen gleichzeitig beschrieben werden. Dabei ist eine Darstellungsform in einem Datensegment gekapselt. Möglich ist die exakte Darstellung der Geometrie als Kantenmodell, als

JT-B-Rep-Modell, als XT-B-Rep-Modell oder als LWPA-Modell. JT-B-Rep und XT-B-Rep sind unterschiedliche Varianten von B-Rep. XT-B-Rep entspricht dabei dem B-Rep-Format vom Parasolid-Modellierkern und JT-B-Rep ist ein eigenes B-Rep-Format von JT. Ein LWPA-Modell ist ein besonderes B-Rep-Modell, das im Gegensatz zu anderen B-Rep-Modellen keine Informationen und keine Punkte oder Kurven enthält. Weiterhin erlaubt JT die Darstellung der Geometrie als tessellierte Modelle in 10 Detaillierungsgraden plus einer zusätzlichen tessellierten Darstellung für bestimmte Einzelfälle. Solch ein Einzelfall ist beispielsweise ein unbekannter Detaillierungsgrad eines approximierten Modells. Eine semi-exakte Darstellung ist als ULP-Modell möglich. Dies ist ein spezielles B-Rep-Modell, das durch die semi-exakte Darstellung besonders wenig Speicherplatz benötigt. Nachteil ist, dass das Modell nicht vollständig exakt ist. Für detaillierte Informationen zu den Darstellungsformen wird auf die Spezifikation des Formates [SIE] verwiesen. Für diese Arbeit ist hier von Bedeutung, dass JT sowohl zum Austausch von exakter Geometrie als auch zum Austausch approximierter Geometrie in unterschiedlichen Detaillierungsgraden verwendbar ist.

Neben der Abbildung von Produktstruktur und Geometrie erlaubt JT auch die Abbildung von Metadaten und fertigungsrelevanten Informationen, die Product Manufacturing Information (PMI) genannt werden. Dafür sind in der JT-Spezifikation zwei Segmente vorgesehen, ein Metadaten-Segment und ein PMI-Segment. Ein separates PMI-Segment ist dabei für Konverter, die mit älteren Versionen der Spezifikation als Version 8 arbeiten, erhalten geblieben. Die aktuelle Version 9.5 der Spezifikation sieht vor, dass alle Metadaten und fertigungsrelevanten Informationen in einem Segment gespeichert werden. Zusammenfassend kann JT somit Geometriedaten in unterschiedlichen Repräsentationsformen, die Produktstruktur, darstellungsbezogene Attribute, Metadaten und fertigungsrelevante Informationen übertragen.

JTist ein weit verbreitetes Datenaustauschformat [VSG+11, SVG11], dessen Bedeutung in der Industrie hoch ist [Fac05]. Eine Studie des ProSTEP iViP Vereins empfiehlt den Einsatz von JT besonders für Kollaborations- und Visualisierungszwecke. Besonders verbreitet ist JT in der Automobilindustrie. Dort konnte JT bereits vor der Standardisierung als Industriestandard betrachtet werden [Ger10]. Die große Verbreitung von JT ist auch der Grund für die Betrachtung in dieser Arbeit.

Das JT-Format wurde bereits in anderen Untersuchungen mit weiteren Datenaustauschformaten verglichen. Ein Beispiel hierfür ist die Arbeiten von Friedewald et al. [FLL+11]. Friedewald et al. untersuchten verschiedene Formate hinsichtlich ihrer Eignung als universelles Format im Schiffsbau. Dabei verwendeten die Autoren unterschiedliche Kriterien für ihren Vergleich. Die Autoren schlussfolgerten aus ihren Ergebnissen, dass JT zuverlässig zum Austausch von Modellen für ihre Zwecke geeignet ist. Sie erkannten jedoch auch, dass die JT-Konverter Daten nicht immer einheitlich übertragen. Dieser Aspekt wurde auch in einem Benchmark verschiedener JT-Konverter vom ProSTEP iViP Verein [Pro10] untersucht. Die Ergebnisse sind hier ähnlich. Produktstruktur und exakte Produktgeometrie werden weitgehend richtig übersetzt. Probleme gibt es jedoch besonders bei den fertigungsrelevanten Informationen. Hier sind große Unterschiede bei den verschiedenen Konvertern zu beobachten.

Weitere Untersuchungen des JT-Formates wurden durch die ProSTEP AG durchgeführt. Beckers et al. [BFS10] beschreiben Ergebnisse dieser Untersuchungen. Diese Untersuchungen stellen ebenfalls die Unterschiede der Konverter insbesondere bei der Konvertierung fertigungsrelevanter Informationen und Farben fest. Diese werden teilweise richtig, gar nicht oder als Geometrie übersetzt. Auch werden typische Konvertierungsprobleme, wie der Einfluss der Modellierungsmethodik auf das Austauschergebnis festgestellt. Nach diesen Untersuchungen werden Farben häufig nicht korrekt übertragen und Texturen komplett nicht übertragen, obwohl sie Teil der Spezifikation sind. Positiv ist jedoch, dass Konvertierungen teilweise vollkommen fehlerfrei möglich waren und fast alle Testmodelle übertragbar waren. Beckers et al. sehen die Gründe für die Unterschiede bei den Konvertern in einer zu geringen Abstimmung der Hersteller untereinander. Sie schließen in ihren Ausführungen jedoch auch nicht aus, dass die Spezifikation möglicherweise nicht eindeutig genug ist. Das Problem der Unterschiede bei den Konvertern ist auch bei STEP festgestellt wurden. Jedoch sind hier andere Gründe ausschlaggebend.

Neben Problemen hat das JT-Format auch verschiedene Vorteile. Im Gegensatz zu STEP besitzt JT verschiedene Mechanismen zur Minimierung der Dateigröße [SIE]. Zudem sind JT-Dateien in einem binären Format gespeichert [VSG+11, SVG11, Han11, Ger10]. Dies wirkt sich positiv auf die Dateigröße aus. Es hat jedoch zur Folge, dass die Daten nicht ohne spezielle Programme zur Darstellung der JT-Daten lesbar sind [Ger10]. Zu beachten ist an dieser Stelle, dass mit der Zahl der Repräsentationsformen und der Menge an weiteren Daten die Dateigröße steigt, da mehr Daten zu speichern sind. Dennoch wird JT zu den leichtgewichtigen Formaten gezählt [Han11], da es im Vergleich beispielsweise mit STEP und systemspezifischen Formaten in der Regel geringere Dateigrößen aufweist. Untersucht wurde diese Thematik beispielsweise von Friedewald et al. [FLL+11].

Ein anderer Vorteil von JT ist, dass Modell-Assoziativitäten zum Originalmodell über IDs gespeichert werden können [Ste07]. Dies erleichtert die Aktualisierung sowohl der JT-Daten als auch der Originaldaten, sollten Änderungen am JT-Modell vorgenommen werden [Ste07].

Zur reinen Darstellung der Daten einer JT-Datei steht JT2Go als kostenloser Viewer zu Verfügung [Ger10, HL08]. Dies senkt anfallende Lizenzkosten. Teil der Viewer-Installation ist auch ein Microsoft Office Plug-In, das es ermöglicht, JT-Dateien in Dokumente dieser Software einzubetten [Ger10, BDP07]. Zudem steht ein kostenpflichtiges, proprietäres Toolkit „JTOpenToolkit“ für die Interaktion zur Verfügung [Ger10]. Konverter können jedoch auch ohne dieses Toolkit mit Hilfe der frei verfügbaren Spezifikation realisiert werden. Anders als bei STEP ist die Spezifikation von JT aktuell kostenfrei verfügbar und weniger umfangreich. Dies erleichtert die Implementierung. Zu beachten ist jedoch, dass JT im Vergleich zu STEP weniger Möglichkeiten bietet. Daher ist das Ziel, eine gemeinsame Verwendung beider Formate für den Datenaustausch in einem Unternehmen und zwischen Unternehmen [Sto11].

Dieser Abschnitt hat einen Überblick über das Datenaustauschformat JT gegeben. Der folgende Unterabschnitt ordnet JT in die vorgestellten Klassifikationsschemata ein.

Einordnung in Klassifikationsschemata

Dieser Abschnitt beschäftigt sich mit der Einordnung von JT als Datenaustauschlösung in die Klassifikationsschemata aus Abschnitt 2.2. Dabei geht dieser Abschnitt auch auf Parallelen und Unterschiede zum Datenaustauschformat STEP ein.

Bei der Klassifikation *nach der Art der übertragbaren Daten* wird zwischen der Einteilung nach produktionstechnischen Aspekten und der Einteilung nach dem Geometriegehalt der Daten. Der vorherige Abschnitt beschreibt, welche verschiedenen Daten in einer JT-Datei gespeichert werden können. Dazu zählen die Geometriedaten, Strukturdaten, darstellungsbezogene Attribute, Metadaten und Daten zu fertigungsrelevanten Informationen. Diese Daten können Produkte und Werkzeuge oder Maschinen betreffen. Nach produktionstechnischen Aspekten ist JT somit in der Lage, Produktdaten und Prozessdaten in Form von bestimmten Werkzeugdaten zu übertragen. Der Schwerpunkt liegt hier jedoch genau wie bei STEP auf Produktdaten. Bezogen auf den Geometriegehalt der Daten ist JT in der Lage, sowohl geometrische als auch nicht-geometrische Daten zu übertragen. Jedoch ist der Umfang nicht-geometrischer Daten auf darstellungsbezogene Attribute, Metadaten und fertigungsrelevante Daten begrenzt. STEP mit seinen verschiedenen Anwendungsprotokollen und Conformance Classes bietet an dieser Stelle mehr Möglichkeiten.

JTgehört analog zu STEP zu den systemneutralen Austauschstrategien in der Klassifikation *nach der Systemneutralität*. Die neutrale Zwischenstufe ist das JT-Format. Damit hat JT auch alle Vor- und Nachteile dieser Klasse von Austauschlösungen.

Auch bei der Klassifikation *nach dem Konzept des Datenaustausches* gehört JT in die gleiche Klasse wie STEP. Mit Hilfe von JT als Datenaustauschformat wird ein übersetzender Austausch mit einem neutralen Format realisiert. Das neutrale Format ist das JT-Format. Für den Austausch wird somit in der Regel pro Sendesystem ein Konverter von dessen Format in das JT-Format und pro Empfänger ein Konverter vom JT-Format in das Empfängerformat benötigt. Eine Studie des ProSTEP iViP Vereins [Fac05] stellt jedoch fest, dass für JT, im Gegensatz zu STEP, Systeme existieren, die JT-Dateien direkt ohne Übersetzung verarbeiten können. In diesem Fall benötigen solche Systeme keine Konverter. Somit können an dieser Stelle auch keine Verluste oder Fehler auftreten. Auch der Aufwand für Entwicklung und Pflege der Konverter entfällt. Das executive Summary nennt jedoch keine Beispiele für solche Systeme.

Anderl [And93] unterscheidet bei der Klassifikation *nach der Spezialisierung des Austausches* zwischen einem generalisierten und einem spezialisierten Austausch. Zur Realisierung dieser Formen des Austausches dienen unterschiedliche Verfahren. JT zählt nach den Ausführungen auf Seite 17 zu den Verfahren zur Realisierung eines generalisierten Austauschen in einem bestimmten Anwendungsgebiet. Das Anwendungsgebiet von JT ist dabei der Austausch von Visualisierungsdaten. In diesem Gebiet kann JT weitgehend universell eingesetzt werden.

In die Klassifikation *nach den Interoperabilitätsleveln* ist JT analog zu STEP einzuordnen. Die JT-Spezifikation legt syntaktische und semantische Interoperabilität fest. Ein automatischer Austausch oder der Zugriff auf Systemfunktionalitäten sind in der Spezifikation nicht beschrieben. Dynamische und pragmatische Interoperabilität werden jedoch nicht ausgeschlossen. Diese sind analog zu STEP durch

zusätzliche Methoden zu realisieren. Bei der semantischen Interoperabilität ist zu beachten, dass sie in der Spezifikation nur für die Geometrie, die Produktstruktur und Geometrieattribute festgelegt ist [SIE]. Die Semantik von nicht für Metadaten und fertigungsrelevante Informationen ist nicht definiert. Festgelegt sind diesbezüglich nur Datumsangaben. Texte und Zahlenwerte werden in der Spezifikation nicht bezüglich ihrer Semantik festgelegt.

In Abschnitt 2.2 wurde auch die Klassifikation *nach Realisierungsmerkmalen* behandelt. Ein Realisierungsmerkmal ist die Zahl der unterstützten Domänen. JT dient hauptsächlich dem Austausch zwischen unterschiedlichen Domänen. Domänen sind hier beispielsweise Entwicklung, Produktion und Fertigung, bis hin zur Dokumentation und Wartung [Dei10]. Auch der Einsatz im Marketing ist durch die Verwendbarkeit mit VR-Systemen denkbar. Neben diesem inter-phasen Austausch ist auch ein in-phasen Austausch im Entwicklungsbereich möglich. Dies ist an den bestehenden Import- und Exportkonvertern von verschiedenen CAD-Systemen ersichtlich [FLL+11]. Zu beachten ist hier jedoch, dass JT bisher keine Übertragung von Features, Parametrik und Modellhistorie vorsieht [SIE]. JT ist analog zu STEP für den in-phasen und für den inter-phasen Austausch einsetzbar. Beachtet werden muss, dass STEP auf einen größeren Umfang an Daten als JT ausgelegt ist.

Ein weiteres Realisierungsmerkmal beschäftigt sich mit der Art der Weitergabe von Änderungen. Daten werden mit JT analog zu STEP vollständig übergeben. Die reine Übertragung von Änderungen und somit ein inkrementeller Austausch ist in der Spezifikation nicht vorgesehen. Zur Realisierung eines solchen Austausches mit JT müssen zusätzliche Programmroutinen zur Erkennung der Änderungen an der JT-Datei implementiert werden. Vorteilhaft bei JT ist, dass Modell-Assoziativitäten zwischen Originalmodellen und JT-Datei erhalten werden können [Ste07]. Dies geschieht durch IDs [Ste07]. Damit wird der Austausch von Änderungen vereinfacht, da die Verbindung zwischen den Objekten erhalten bleibt und nicht neu erkannt werden muss [Ste07].

Je nach Vollständigkeit der Daten unterscheiden Drath et al. [DFB11] zwischen dem Austausch aller Daten und dem Austausch tatsächlich benötigter Daten. JT unterstützt das Konzept des Austausches tatsächlich benötigter Daten teilweise durch seinen segmentellen Aufbau. Dieser erlaubt, je nach konkretem Anwendungsszenario und Konfiguration, die einzelnen Segmente unterschiedlich stark zu befüllen[Dei10]. So können beispielsweise exakte Darstellungen bei Nichtgebrauch weggelassen werden. Problematisch daran ist, dass die bestehenden Konverter dieses Konzept unterstützt müssen, um die Anpassung auf das Anwendungsszenario zu realisieren. Das bedeutet bei den Konvertern muss einstellbar sein, welche Daten in die JT-Datei geschrieben werden und welche nicht. Untersuchungen, inwieweit bestehende Konverter die Einstellbarkeit unterstützen sind nicht bekannt. Doch prinzipiell ermöglicht JT den Austausch auf tatsächlich benötigte Daten zu begrenzen. Hier unterscheidet sich JT von STEP, da bei STEP solche Mechanismen in der Form nicht bekannt sind.

Zum Austauschmanagement von Änderungen trifft die Spezifikation keine Aussagen [SIE]. Abhängigkeiten sind innerhalb der JT-Datei zwischen den verschiedenen gespeicherten Repräsentationsformen einzelner Bauteile und zwischen den verschiedenen Bauteilen vorhanden. Aufgrund des konkreten Aufbaus von JT-Dateien wird

an dieser Stelle angenommen, dass bei Änderungen die komplette Datei mit allen Segmenten neu aufgebaut wird. Abhängigkeiten zwischen den Bauteilen werden dann durch die Erzeugersysteme behandelt. Die unterschiedlichen Repräsentationsformen werden durch den Neuaufbau der Datei neu generiert, wodurch diese konsistent gehalten werden. Somit besitzt JT keine Austauschkontrolle gemäß der Klassifikation.

JTunterstützt anders als STEP neben dem linear Engineering auch simultanous und concurrent Engineering-Prozesse. Eine Studie des ProSTEP iViP Vereins [Fac05] empfiehlt den kollaborativen Einsatz von JT. Konkrete Gründe dafür werden jedoch nicht genannt. Anzunehmen ist, dass hier die im Vergleich zu STEP geringeren Dateigrößen und die einfache Visualisierung der Produkte ausschlaggebend ist.

JTals *Datenaustauschformat* kann auch nach Eigenschaften dieser klassifiziert werden. Thierfelder [Thi04] unterscheidet Datenaustauschformat für 3D-Daten nach ihrem Anwendungsfeld. In diese Klassifikation gehen insbesondere die Daten ein, die ein Format abbildet. JT kann demnach den Modellierungs- und Animationsformaten zugeordnet werden. Grund für die Zuordnung ist, dass JT in der Lage ist, komplette Szenen inklusive darstellungsbezogenen Attributen abzubilden. Eine Zuordnung zu den CAD/CAM/CAE-Formaten ist ebenfalls möglich, da JT neben approximierten auch exakte Geometrien übertragen kann. Demnach ist die Klassifikation nach Thierfelder nicht exklusiv. Die Klassen können sich überschneiden.

Das JT-Format kann nach der Untersuchung von Handschuh [Han11] sowohl als sekundäres als auch als primäres Format eingesetzt werden. Anders als bei STEP sind somit hier klare Aussagen möglich.

Der Standardisierungsprozess von JT ist aktuell noch nicht abgeschlossen. Bisher existiert für die Version 8.1 eine öffentlich zugängliche Spezifikation, die ISO/PAS 14306:2011 [ISOf]. In der Automobilbranche ist JT so weit verbreitet, dass das Format dort als Industriestandard angesehen werden kann [SIE]. Die Spezifikation des Formates wurde 2007 erstmals veröffentlicht [Dei10]. Aktuell ist die Version 9.5 [SIE] kostenlos online verfügbar[1]. Damit kann JT als offenes Datenaustauschformat eingeordnet werden.

JT besitzt verschiedene Mechanismen zur Minimierung der Dateigröße [Han11]. Zudem erfolgt die Speicherung der Datei im Binär-Format [Han11, VSG+11, SVG11]. Daher ist JT im Gegensatz zu STEP ein leichtgewichtigen Datenaustauschformaten [Han11].

Dieser Abschnitt hat JT in vorgestellten Klassifikationsschemata eingeordnet. Dabei wurde JT mit STEP verglichen und Unterschiede sowie Gemeinsamkeiten erkannt. Die wichtigsten Unterschiede beider Formate betreffen den Umfang der übertragbaren Daten und die Leichtgewichtigkeit. STEP besitzt zwar einen größeren Umfang an übertragbaren Daten, ist jedoch nicht leichtgewichtig, währenddessen JT ein leichtgewichtiges Format mit begrenztem Übertragungsumfang ist. Die Ergebnisse der Einordnung von JT in die Klassifikationsschemata sind in der Übersicht im Anhang eingetragen.

[1] http://www.plm.automation.siemens.com/en_us/Images/JT_v95_File_Format_Reference_Rev-D_tcm1023-134827.pdf Stand 02.10.2012 15:00 Uhr

Der folgende Abschnitt beurteilt JT für den Einsatz im Beispielszenario. Dabei wird ebenfalls auf Unterschiede und Parallelen zu STEP eingegangen.

Beurteilung für Anwendungsszenario

Dieser Abschnitt befasst sich mit der Beurteilung von JT als Datenaustauschlösung für den Einsatz im Beispielszenario „Datenaustausch bei der Montagesimulation“. Dazu wird auch auf die in Abschnitt 3.2 identifizierten Variationen des Szenarios eingegangen. Im Zuge dieser Betrachtungen werden zudem Unterschiede und Parallelen zu STEP aufgezeigt.

Ein Aspekt, der beim Beispielszenario variiert, ist der *Umfang der übertragenen Modelle*. Unterschieden wird hier zwischen der Übertragung von 3D-Geometriemodellen, statischen und dynamischen DMUs. JT erlaubt den Austausch von 3D-Geometriemodellen und statischen DMUs. Dynamische DMUs können im Gegensatz zu STEP durch fehlende Kinematik nicht abgebildet werden [SIE, Fac05]. Dafür ermöglicht JT die Übertragung approximierter Geometrien in unterschiedlichen Detaillierungsgraden. Dies ist bei Montagesimulation von Produkten mit vielen Bauteilen vorteilhaft und mit STEP in dieser Form nicht möglich.

Weiterhin variiert der Datenaustausch bei der Montagesimulation in der *Anzahl der Erzeugersysteme*. Hier verhält sich JT analog zu STEP. Bei einem oder wenigen Systemen ohne bestehende Konverter ist eine systemspezifische Lösung vorteilhaft, da so mit geringerem oder gleichem Aufwand bessere Übertragungsergebnisse erreichbar sind. Sind bereits Konverter vorhanden, so sind der Aufwand und der Nutzen in der konkreten Situation abzuwägen. Bei vielen Erzeugersystemen sind hingegen neutrale Formate wie JT hinsichtlich des Aufwandes vorteilhaft.

Die *Art des Simulationssystems* ist ein weiterer Aspekt, bei dem der Datenaustausch variiert. Diese Arbeit unterscheidet hier drei Fälle. Analog zu STEP ist auch bei JT keine pauschale Beurteilung für die Fälle A und B möglich. Diese hängt von den konkreten Systemen, den konkret benötigten zusätzlichen Daten und der Kompatibilität zum JT-Format ab. Anders ist dies bei Fall C. Hier ist JT häufig als Datenaustauschlösung geeignet. Dies zeigen die Ergebnisse von Friedewald et al. [FLL+11] bezüglich den vorhandenen Konvertern für den Export und Import von JT-Dateien in verschiedene CAD-Systeme. Hier sind somit ebenfalls Parallelen zu STEP festzustellen, obwohl JT keine Daten zur Parametrik, zu Features und zur Modellhistorie austauschen kann. Folge ist, dass die Modelle mit hoher Wahrscheinlichkeit nicht editierbar sind [Dei10]. Dies wirkt sich auf die Möglichkeiten der Rückgabe von Ergebnissen aus.

Die *Richtung des Datenaustausches* ist das letzte Kriterium, das variiert werden kann. Bei der Montagesimulation können sowohl Eingabedaten als auch Ergebnisdaten auszutauschen sein. Der Austausch von Eingabedaten und der Austausch von Ergebnisdaten stellen bei einem Datenaustausch mit JT analog zu STEP zwei Datenaustauschvorgänge dar. Bei Ergebnisdaten sind zusätzlich zur Geometrie Kennzeichnungen zu übertragen. JT erlaubt prinzipiell die Übertragung von Farben [SIE]. Die Schwierigkeiten hier sind, dass dazu eine JT-Datei vom Simulationssystem erstellt werden muss und das Erzeuger- oder Bearbeitungssystem das entsprechende Modell mit der Kennzeichnung importieren können muss. Dieses Problem muss durch den

Exportkonverter des Simulationssystems gelöst werden. Dieser muss dazu in der Lage sein, Kennzeichnungen an approximierten Geometrien auf exakte Geometrien zu übertragen, da ansonsten Systeme, die exakte Daten fordern, die Ergebnisse nicht verwenden können. Grundsätzlich sind beim Austausch der Ergebnisdaten exakte Modelle zu bevorzugen, wenn Änderungen direkt an diesen Modellen erfolgen sollen. Hier besteht bei JT ein weiteres Problem, da einige CAD-Systeme für editierbare Modelle eine Modellierhistorie fordern. Diese kann mit JT jedoch nicht übertragen werden. Bei diesen Systemen kann JT somit nicht zum Austausch editierbarer Modelle genutzt werden. Jedoch kann JT zur Darstellung der gekennzeichneten Bauteile verwendet werden. Dazu kann der kostenfreie Viewer für JT-Dateien genutzt werden. Damit ist es für den Konstrukteur einfacher die Kollisionsbereiche zu identifizieren als wenn er nur einen mündlichen oder schriftlichen Bericht der Ergebnisse erhält. Durch die kostenfreien Viewer fallen zudem keine zusätzlichen Lizenzkosten an. Ein weiteres Problem bei der Übertragung von Ergebnisdaten ist die unterschiedliche Realisierung der Konverter [FLL+11, Pro10, BFS10]. Hier ist wieder eine Parallele zu STEP vorhanden, da auch STEP-Konverter an diesem Punkt unterschiedlich realisiert sind.

Eine weitere Parallele zu STEP betrifft die Grundvoraussetzung zur Verwendung des JT-Formates. Auch JT kann nur verwendet werden, wenn alle beteiligten Systeme Daten in einer zu JT kompatiblen Struktur verarbeiten können. Ist dies nicht der Fall, so ist ein Austausch mit JT nicht möglich. Diese Grundvoraussetzung ist stets an den konkreten Systemen zu untersuchen.

Dieser Abschnitt hat JT hinsichtlich der Eignung für den Datenaustausch bei der Montagesimulation untersucht. Dabei wurde festgestellt, dass JT nicht für alle Variationen gleichermaßen geeignet ist und bestimmte Aspekte an den konkreten Systemen untersucht werden können. Auch wurden Unterschiede und Parallelen zu STEP aufgezeigt. Gemeinsamkeiten gibt es, bei der Beurteilung der unterschiedlichen Art der Simulationssysteme, bei den Unterschieden bei bestehenden Konvertern und bei der Voraussetzung von kompatiblen Strukturen. Vorteile birgt JT bezüglich den Dateigrößen, der kostenfreie Spezifikation und dem kostenfreier Viewer. Jedoch können insgesamt weniger Daten als bei STEP abgebildet werden. So ermöglicht JT im Gegensatz zu STEP keine Übertragung von dynamischen DMUs. Dafür können zwar approximierte Geometrien in verschiedenen Detaillierungsgraden übertragen werden, jedoch sind weniger nicht-geometrische Daten als bei STEP abbildbar. Stoye [Sto11] sieht es daher als Ziel an, beider Formate zur Realisierung des Datenaustausches gemeinsam einzusetzen.

Bisher wurden somit zwei weitverbreitete Datenaustauschformate untersucht. Im nächsten Abschnitt wird eine Alternative zu diesen dateibasierten Datenaustauschstrategien näher betrachtet.

4.3 Online-Kopplung mit einem CAx-Objektbus

Eine Alternative zum dateibasierten Datenaustausch mit Datenaustauschformaten wie STEP und JT ist die Online-Kopplung mittels einem CAx-Objektbus. Diese Austauschstrategie ist das Thema der folgenden Erläuterungen.

Das Konzept des CAx-Objektbusses wurde im Projekt Analysis of Access Interfaces of Various CAx-Systems (ANICA) entwickelt [AJSK98, SAKJ98, SAKJ00]. Das Projekt lief von 1995 bis 1998 an der Universität Karlsruhe und wurde von zahlreichen Partnern aus der Industrie unterstützt [AJSK98]. Dazu gehörten neben Anwendern von CAx-Systemen auch Hersteller solcher Systeme. Arnold et al. [AJSK98], Swienczek et al. [SAKJ00] und Swienczek et al. [SAKJ98] beschreiben Ergebnisse dieses Projektes.

Der Ansatz des ANICA-Projektes beschreibt die Schaffung eines Komponentensystems, in dem jede Komponente seine Funktionalitäten als Dienste bereitstellt. Komponenten sind ehemals eigenständige bestehende Systeme mit ihren Modulen oder direkt ANICA-konforme Komponenten. Die Verbindung der Systeme und der Zugriff auf CAx-Objekte erfolgt über den sogenannten CAx-Objektbus. Dieser besteht intern aus einer Menge von definierten Schnittstellen, die Common Interface genannt wird, und Implementierungen zur Infrastruktur. Der CAx-Objektbus realisiert eine einheitliche gemeinsame Zugriffsschnittstelle, auf dessen Grundlage die Systeme kommunizieren. Der Austausch erfolgt nur zur Laufzeit der Systeme. Zudem gilt die Voraussetzung, dass die Systeme technisch Kontakt zueinander haben. Das bedeutet, die Systeme müssen entweder gleichzeitig auf einem Rechner oder gleichzeitig innerhalb eines Netzwerkes laufen. Die Realisierung des Datenaustausches erfolgt dann über ein Client-Server Konzept. Der Server stellt die Daten bereit, die vom Client angefordert werden.

Die Realisierung des Ansatzes im ANICA-Projekt basiert auf zwei Standards. Das ist zum einen das STEP Anwendungsprotokoll 214 in der Conformance Class 1 und zum anderen die Common Object Request Broker Architecture (CORBA), ein Industriestandard für die Interoperabilität verteilter Systeme.

STEPdient dabei als Grundlage für den CAx-Objektbus die zu übertragenden Daten betreffend. Zu beachten ist hier jedoch, dass das Anwendungsprotokoll 214 während der Projektlaufzeit noch nicht vollständig veröffentlicht war. Für die Entwicklung innerhalb des Projektes konnten als nur sich in Bearbeitung befindende Spezifikationsdokumente verwendet werden. Ob dies Einfluss auf die Realisierung hatte, ist jedoch nicht bekannt. Übertragbare Daten der Conformance Class wurden jedoch von Arnold et al. in [AJSK98] beschrieben und umfassen die Bauteilbeschreibung, das Produktdatenmanagement, Draht-, Flächen- und Volumenmodelle. Produktstruktur und Kinematik werden beispielsweise nicht beschrieben.

Neben Daten haben die verbundenen Systeme auch Zugriff auf bestimmte Funktionalitäten der einzelnen Systeme. Diese werden nicht durch STEP beschrieben und müssen daher manuell bei der Realisierung des Ansatzes identifiziert und generalisiert werden. So können auch weitere zusätzliche Daten der Austauschlösung hinzugefügt werden.

CORBAist ein Industriestandard für die transparente Interaktion von Objekten zwischen heterogenen verteilten Systemen. CORBA gehört zur Object Management Architecture der Object Management Group. CORBA ist eine plattform- und programmmiersprachenunabhängige Spezifikation. Zentraler Kern dieser Spezifikation ist der Object Request Broker (ORB). Dieser realisiert die Kommunikation der Systeme untereinander und ermöglicht so den Zugriff auf Objekte eines Servers ohne

Kenntnisse zu Details der Implementierung dieses Servers. Der ORB wird im Ansatz des ANICA-Projektes Objektbus genannt. Die Schnittstellen zu den Objekten mit Methoden zum Zugriff und zur Änderung werden durch eine spezielle Spezifikationssprache für Schnittstellen definiert. Diese Sprache heißt Interface Definition Language und dient als Implementierungsvorlage für die Schnittstellen.

Für die Entwicklung des CAx-Objektbusses wurde ein Common Interface Entwurfsmuster verwendet. Dieses Entwurfsmuster gleicht der Verwendung eines neutralen Formates und realisiert eine neutrale Zwischenstufe zwischen den Systemen. Diese definiert jedoch anders als neutrale Formate Objekte und Methoden zum Zugriff. Vorteil dieses Ansatzes ist, dass für Systeme, die nicht direkt zum Objektbus konform sind, nur ein Adapter zu realisieren ist. Die Grundlage für die neutrale Zwischenstufe sind die Programmierschnittstellen der einzelnen Systeme. Deren Funktionsumfang bestimmt den Funktionsumfang des CAx-Objektbusses. Dabei sind drei Ansätze möglich. Zum einen kann der kleinste gemeinsame Nenner der Systeme verwendet werden. Vorteil dieser Möglichkeit ist, dass für alle Systeme die gleichen Funktionen verwendbar sind. Der Nachteil ist jedoch, dass somit die Möglichkeiten des Objektbusses und damit des Austausches von dem System bestimmt wird, dessen Programmierschnittstelle den kleinsten Funktionsumfang besitzt. Ein zweiter Ansatz zur Festlegung des Common Interface ist es für Teilbereiche die beste Lösung zu verwenden. Damit ist der Funktionsumfang des CAx-Objektbusses größer, dafür unterstützen nicht alle Systeme die gleichen Funktionen. Die dritte Möglichkeit ist eine Kombination aus den beiden vorherigen Ansätzen. Zuerst wird über den kleinsten gemeinsamen Nenner die Grundfunktionalität bestimmt und danach um zusätzliche Funktionen der besten Lösungen der Teilbereiche ergänzt.

Zusammenfassend arbeitet das Konzept des ANICA-Projektes somit in dieser Art und Weise, dass die Systeme über einen CORBA-basierten Objektbus verbunden sind. Die nicht ANICA-konformen Systeme sind über Adapter an den Bus gekoppelt. Diese Adapter nutzen die Programmierschnittstelle der Systeme. Der Bus stellt so in generalisierter Form Schnittstellen zu Objekten und Funktionalitäten bereit. Darüber greifen die Systeme auf einander zu. Dabei wird eine Server-Client-Verbindung zwischen den Systemen realisiert.

Das Ziel des Projektes war die Verifikation eines CAx-Architekturansatzes zur Integration von heterogenen CAx-Systemen. Nach dieser Aussage zählt der Ansatz zu den Integrationsansätzen und nicht zu den Kopplungsansätzen. Integrationsansätze sind eigentlich nicht Thema dieser Arbeit. Ausschlaggebende Kriterien sind hier die Abgrenzung der Systeme und die Dauer der Verbindung. Laut der Definition aus Abschnitt 2.2 ist eine Integration erst vorhanden, wenn eine Verbindung auf Dauer ausgelegt ist und die Systeme nicht mehr als eigenständig identifiziert werden können. Eine Kopplung beschreibt genau die gegenteilige Situation, eine lose, jederzeit trennbare Verbindung eigenständiger Systeme. Im Konzept des ANICA-Projektes behält jedes System seine Benutzungsschnittstelle und seine Funktionalitäten. Die Systeme werden durch den CAx-Objektbus miteinander verbunden und können darüber auf Objekte und Funktionalitäten anderer Systeme zugreifen. Jedoch sind je nach Realisierung die Systeme weiterhin als eigenständig identifizierbar. Dies wird erst dann schwierig, wenn die Benutzungsoberflächen der Einzelsysteme auf die Aufnahme von Bedienelementen für zusätzliche Funktionalitäten anderer Systeme erwei-

tert werden. Dies hängt jedoch von der konkreten Realisierung des Lösungskonzeptes ab. Weiterhin sind nicht-ANICA-konforme Systeme nur durch einen Adapter, der Programmierschnittstelle der Systeme verwendet, mit dem Bus verbunden. Daher ist diese Verbindung einfach trennbar. Aus diesen Gründen wird der Ansatz des CAx-Objektbusses hier als Kopplungsansatz mit der Möglichkeit zur Realisierung einer Integration eingeordnet und geht daher auch in diesen Vergleich ein.

Ein aktueller Einsatz des Ansatzes in der Praxis ist nicht bekannt. Der Lösungsansatz wurde jedoch nach der Entwicklung im ANICA-Projekt in der Automobilindustrie getestet [Sch01]. Ergebnisse dazu sind jedoch nicht bekannt. Ein Teilprojekt untersuchte die Verwendbarkeit des Ansatzes jedoch exemplarisch am Datenaustausch bei der virtuellen Einbauuntersuchung [AJSK98, SAKJ00, SAKJ98]. Dabei wurde ein Austausch zwischen zwei CAD-Systemen realisiert. Eines dieser Systeme enthielt ein Modul zur Durchführung der Untersuchung. Das andere System wurde zur Modellierung von Bauteilen verwendet. Unklar ist dabei, ob die virtuelle Einbauuntersuchung eine Montagesimulation im Sinn dieser Arbeit darstellt oder eine statische Untersuchung der Bauteile in Einbaulage. In der Literatur wird der Begriff „virtuelle Einbauuntersuchung“ unterschiedlich verwendet. Bei einer statischen Einbauuntersuchung wird im Gegenteil zur Montagesimulation geprüft, ob ein Bauteil in Einbaulage mit einem anderen kollidiert oder ob andere Probleme beispielsweise thermische Unverträglichkeiten bestehen. Eine Bewegung der Bauteile wird nicht betrachtet. Der Testeinsatz vom Ansatz des ANICA-Projektes war erfolgreich. Untersuchungen mit mehr Systemen und anderen Anwendungen sind jedoch nicht bekannt.

Ein Problem des Ansatzes ist, dass alle am Austausch beteiligten Systeme zum Austauschzeitpunkt laufen und miteinander verbunden sein müssen. Das bedeutet, wenn ein Systemen auf mehrere Daten von mehreren Systemen angewiesen ist, müssen alle diese Systeme entweder auf den Rechner oder im Netzwerk laufen und alle notwendigen Daten geladen sein. Bei vielen Sendesystemen und vielen Daten kann dies ein logistisches Problem darstellen, da die Rechner- und Netzwerkleistung technisch bedingt begrenzt ist. Zudem kann der Absturz eines Systems zum Scheitern des gesamten Austausches führen. Hier sind daher bei der Realisierung des CAx-Objektbusses geeignete Maßnahmen zur Lösung und Vorbeugung dieser Aspekte zu entwickeln.

Der Ansatz des ANICA-Projektes beschränkt sich auf die Kopplung von CAx-Systemen. CAx-ferne Systeme werden nicht berücksichtigt. Inwieweit diese daher unterstützt werden können, ist nicht bekannt. Auch der Austausch über Unternehmensgrenzen ist auf diese Weise als schwierig zu betrachten, da ein Zugriff auf unternehmensinterne Netzwerke und Rechner häufig nicht erwünscht ist.

Ein weiteres Problem des Ansatzes ist, dass die Adapter von den Systemen zum CAx-Objektbus vorhandene Programmierschnittstellen der Systeme verwenden. Damit ist eine Voraussetzung für die Anwendbarkeit des Ansatzes das Vorhandensein geeigneter Programmierschnittstellen. Dies kann bei CAx-Systemen und besonders bei CAx-fernen Systemen nicht pauschal als gegeben angesehen werden. Hier sind nähere Untersuchungen der konkreten beteiligten Systeme notwendig. Weiterhin hängt der Umfang, in dem Objekte und Funktionen ausgetauscht werden können, direkt vom Umfang der Programmierschnittstellen ab. Über Schnittstellen, die auf das Auslesen

von Daten ausgelegt sind, kann die Eingabe von Daten nicht realisiert werden. Das bedeutet, ein System mit solch einer Programmierschnittstelle kann keinen Client in der Verbindung darstellen. Umgekehrt gilt dies bei Servern ebenso. Damit hängt der Austausch über einen CAx-Objektbus stark von den konkreten Systemen ab.

Neben diesen Problemen bietet der Ansatz auch Vorteile gegenüber dateibasierten Austauschstrategien. So vermeidet dieser Ansatz unerwünschte Datenredundanzen in Dateien. Realisiert wird dies durch die Art des Objektaustausches zwischen Sender und Empfänger. Die notwendigen Daten zum Objekte werden vom Sendesystem über den CAx-Objektbus an das Empfängersystem übergeben [Sch01, SAKJ98]. Der Empfänger baut aus diesen Daten ein lokales Modell auf, das jedoch über den Objektbus mit dem Originalmodell verbunden bleibt. Über diese Verbindung können Daten nachgeladen und Änderungen übertragen werden. Die Übertragung der Änderungen kann dabei sofort oder zu einem späteren Zeitpunkt erfolgen. Wichtig ist es dabei jedoch, die Konsistenz der Daten zu sichern. Während Änderungen am lokalen Modell durchgeführt werden, sind Änderungen am Originalmodell zu verhindert bis die Änderungen an das Originalmodell übergeben sind. Änderungen am Originalmodell müssen direkt an das lokale Modell weitergegeben werden, um die Aktualität dieses Modells zu gewährleisten. Dabei ist zu beachten, dass bei Untersuchungen am lokalen Modell des Empfängers ebenfalls Änderungen am Originalmodell zu verhindern sind, da Änderungen am lokalen Modell während einer Untersuchung zu fehlerhaften Ergebnissen oder dem Misslingen der Untersuchung führen kann. Diese Art der Zugriffs- und Änderungsverwaltung ist durch die Infrastruktur des CAx-Objektbusses zu realisieren. Damit wird jedoch eine Datenredundanz in gespeicherten Dateien verhindert. Zum werden durch den inkrementellen Austausch der Daten Aktualisierungen meist schneller als beim vollständigen dateibasierten Austausch übertragen.

Außerdem soll der Einarbeitungsaufwand in die einzelnen Systeme minimiert werden. Funktionalitäten anderer Systeme sollen im eigenen System des Anwenders integriert werden. Dies zählt jedoch zu den Aspekten, die weg von der Kopplung hin zur Integration führen. Pauschal für alle Realisierungen kann der Aspekt somit nicht im vollen Maße betrachtet werden.

Der Ansatz des ANICA-Projektes bringt somit verschiedene Vorteile und Probleme mit sich. Konkrete Beispiele für den praktischen Einsatz des Ansatzes konnten nicht gefunden werden, sind jedoch nicht ausgeschlossen.

Der folgende Unterabschnitt ordnet den Ansatz in die vorgestellten Klassifikationsschemata ein. Dabei wird der Vergleich zu den behandelten Datenaustauschformaten gezogen.

Einordnung in Klassifikationsschemata

In Abschnitt 2.2 wurden verschiedene Klassifikationsschemata zur Einteilung von Datenaustauschstrategien vorgestellt. Dieser Abschnitt ordnet den Ansatz zur Online-Kopplung mit einem CAx-Objektbus, wie er im Projekt ANICA entwickelt wurde, in die Schemata ein.

Bei der Einteilung *nach der Art der Daten* gibt es zwei verschiedene Arten, nach denen sich Datenaustauschstrategien unterscheiden. Einerseits unterscheiden sich

Datenaustauschstrategien nach produktionstechnischen Aspekten in Strategien zum Austausch von Produkt-, Prozess- oder Auftragsdaten. Andererseits ist eine Klassifikation nach dem Geometriegehalt der Daten möglich. Der CAx-Objektbus des ANICA-Projektes basiert datenseitig auf der Conformance Class 1 des STEP Anwendungsprotokoll 214. Daher überträgt der CAx-Objektbus hauptsächlich Produktdaten und Daten zu Werkzeugen, die nicht über die Produktdaten hinausgehen. Diese Daten sind sowohl geometrisch als auch nicht-geometrisch. Damit ähnelt dieser Ansatz den beiden vorgestellten Datenaustauschformaten im Hinblick auf die Art der übertragbaren Daten. Der Unterschied ist jedoch die Erweiterbarkeit des CAx-Objektbus Ansatzes. Weitere Daten können den Objekten individuell durch einen zusätzlichen Aufwand bei der Realisierung hinzugefügt werden. Bei Datenaustauschformaten ist dies schwierig und nur durch Änderung des Formates möglich. Dies hätte jedoch einen enormen Aufwand zu Folge, da im schlimmsten Fall alle Konverter an die neue Version angepasst werden müssen.

Weiterhin werden Datenaustauschstrategien *nach der Systemneutralität* unterschieden. Durch das Common Interface Entwurfsmuster des Objektbusses ist der Ansatz des ANICA-Projektes eine systemneutrale Austauschstrategie. Der CAx-Objektbus stellt dabei die neutrale Zwischenstufe dar. Anders als bei den Datenaustauschformaten beschreibt diese Objekte und Funktionalitäten.

In die Klassifikation *nach dem Austauschkonzept* ist die Online-Kopplung mit einem CAx-Objektbus in die Klasse der einbindenden Austauschstrategien einzuordnen. Diese Zuordnung trifft bereits Schumann [Sch01] bei seinen Ausführungen zum Klassifikationsschema. Der Objektbus realisiert den Austausch der Objekte. Dabei wird eine Zwischenstufe zwischen Einbetten und Verknüpfen durchgeführt. Die notwendigen Daten zum Objekte werden vom Sendesystem über den CAx-Objektbus als Kopie an das Empfängersystem übergeben [Sch01, SAKJ98]. Das entspricht einer Einbettung dieser Daten. Dabei bleibt jedoch eine verknüpfende Verbindung zum Objekt im Sendesystem erhalten [SAKJ98]. Über diese können weitere Daten nach geladen werden oder Änderungen, wie Einfärbungen, direkt an das Originalmodell übertragen werden. Die Übertragung der Änderungen kann dabei sofort oder zu einem späteren Zeitpunkt erfolgen. Während des gesamten Prozesses laufen die Systeme und sind über den Objektbus verbunden. Im Ergebnis entsteht eine Verknüpfung zum Objekt und eine Einbettung der benötigten Daten des Objektes.

Nach der Spezialisierung des Datenaustausches ist der CAx-Objektbus den Verfahren zur Realisierung eines spezialisierten Austausches zuzuordnen. Grund für diese Zuordnung ist, dass die konkreten beteiligten Systeme direkten Einfluss auf den Objektbus haben. Damit ist ein generalisierter Austausch nicht gegeben.

Bei der Einordnung der Datenaustauschstrategie *nach den unterstützten Interoperabilitätsleveln* realisiert der Ansatz des ANICA-Projektes neben syntaktischer und semantischer Interoperabilität auch pragmatische Interoperabilität. Voraussetzung für pragmatische Interoperabilität ist die Kommunikation der Systeme miteinander. Dies wird durch den CAx-Objektbus realisiert. In einer Client-Server-Verbindung greifen die Systeme auf einander zu. Somit realisiert dieser Ansatz ein Level der Interoperabilität, das durch Datenaustauschformate, wie STEP und JT, nicht unterstützt wird. Ein Vorteil ist ein individuellerer Datenaustausch. Zudem realisiert

die Austauschstrategie dynamische Interoperabilität durch Überwachung und Regelung von Änderungen an den Daten. Die notwendigen Mechanismen zur Sicherung der Konsistenz der Modelle wurden bereits im vorherigen Abschnitt auf Seite 70 beschrieben.

Weiterhin wurde in Abschnitt 2.2 die Klassifikation *nach Realisierungsmerkmalen* vorgestellt. Ein Merkmal dabei ist die Zahl der unterstützten Domänen. Der Ansatz des ANICA-Projektes legt sich hierbei nicht fest. Sowohl der Austausch zwischen Systemen einer Domäne, wie beim Testbeispiel, als auch zwischen CAx-Systemen allgemein wird beschreiben. Analog zu den beiden Datenaustauschformaten STEP und JT ist daher ein in-phasen und ein inter-phasen Austausch möglich.

Ein anderes Realisierungsmerkmal ist die Art der Weitergabe von Änderungen. Im Gegensatz zu den bisher vorgestellten Datenaustauschstrategien realisiert die Online-Kopplung mit einem CAx-Objektbus einen inkrementellen Datenaustausch. Arnold et al. [AJSK98] beschreiben, dass die Erstübertragung der Objekte noch schlechter in der Performanz ist als die vollständige Übertragung durch Datenaustauschformate. Jedoch stellen die Autoren fest, dass die Übertragung von Aktualisierungen meist besser an dieser Stelle ist. Dies ist auch schlüssig, da hierbei insgesamt weniger Daten auszutauschen sind.

Ausgetauscht werden bei der Verwendung des CAx-Objektbusses nur benötigte Daten. Dies geht beim Testbeispiel, der virtuellen Einbauuntersuchung, soweit, dass statt kompletter Bauteile nur relevante Teilbereiche übertragen werden. Nach der Vollständigkeit der Daten ist der Ansatz somit den Ansätzen, die nur tatsächlich benötigte Daten austauschen, zuzuordnen.

Zwischen den Originalobjekten in den einzelnen Systemen sind auch Abhängigkeiten möglich. Die Verwaltung dieser insbesondere bei Änderungen kann prinzipiell auch durch den CAx-Objektbus geschehen. Dabei sind beide Austauschmanagementvarianten, die Li [Li10] beschreibt, denkbar.

Die Architektur der Austauschlösung des ANICA-Projektes erlaubt neben dem Einsatz im linear Engineering auch den Einsatz im simultanous und concurrent Engineering, da Änderungen direkt übertragen werden und eine dynamische Zusammenarbeit der Systeme und damit der Anwender unterstützt wird. Somit ist der Ansatz für die drei Prozesstypen der Produktentwicklung geeignet.

Das letzte vorgestellte Klassifikationsschema unterscheidet Datenaustauschstrategien *nach dem verwendeten Datenaustauschformat*. Der Ansatz des ANICA-Projektes verwendet in diesem Sinne kein Datenaustauschformat. STEP wird zwar als datenseitige Basis für den Objektbus verwendet, tritt hier aber nicht als Datenaustauschformat im eigentlichen Sinne auf. Daher ist eine Einordnung in dieses Klassifikationsschema schwierig und nur teilweise möglich. Bei den Anwendungsfeldern kann der Ansatz durch die Verwendung von STEP und die Konzentration auf CAx-Systeme zu den Strategien für CAD/CAM/CAE-Anwendungen gezählt werden. Der Austausch der Daten ist insoweit verbindlich, wie diese vom jeweiligen Sendesystem bereitgestellt werden. Doch da der Ansatz kein Format beschreibt, kann er nicht den primären Formaten zugeordnet werden. Eine Standardisierung des Ansatzes erfolgte bisher nicht. Jedoch verwendet der Ansatz bestehende Standards und Industriestandards. Die Offenheit der Spezifikation und die Leichtgewichtigkeit können hier nicht

beurteilt werden, da in dieser Form keine Spezifikation besteht und die Leichtgewichtigkeit nicht ermittelt werden kann.

Dieser Abschnitt hat die Strategie der Online-Kopplung mit einem CAx-Objektbus in die vorgestellten Klassifikationsschemata eingeordnet. Dabei wurde auf Unterschiede und Gemeinsamkeiten zu den vorher behandelten Datenaustauschformaten STEP und JT eingegangen. Gemeinsamkeiten sind bezüglich der Art von übertragbaren Daten und der Systemneutralität festzustellen. Unterschiede betreffen das grundlegende Austauschkonzept, die Spezialisierung des Austausches, die realisierten Interoperabilitätslevel und verschiedene Realisierungsmerkmale. So realisiert die Online-Kopplung mit einem CAx-Objektbus die Übertragung wirklich benötigter Daten und einen inkrementellen Austausch von Änderungen. Insgesamt ist der Ansatz eine interessante Alternative zu Strategien mit Datenaustauschformaten. Der nächste Abschnitt beurteilt den Ansatz für die Realisierung des Datenaustausches bei der Montagesimulation. Dort wird weiterhin auf Unterschiede und Parallelen zu den betrachteten Datenaustauschformaten eingegangen.

Beurteilung für das Anwendungsszenario

Die Beurteilung der Online-Kopplung mit einem CAx-Objektbus für das Beispielszenario ist Thema dieses Abschnittes. Dazu erfolgt die Betrachtung der Variationen des Anwendungsszenarios. Weiterhin wird auf Unterschiede und Parallelen zu bereits behandelten Austauschstrategien eingegangen.

Das Beispielszenario dieser Arbeit ist der Datenaustausch bei der Montagesimulation. Ein Merkmal, das variiert, ist der *Modellumfang*. Möglich ist hier der Austausch von 3D-Geometriemodellen, statischen und dynamischen DMUs. 3D-Geometriemodelle beschreiben dabei entweder einzelne Bauteile oder Ersatzgeometrien für Baugruppen. DMUs enthalten die Geometrie mehrerer Bauteile und beschreiben Unterbaugruppen oder komplette Produkte. Dynamische DMUs enthalten im Gegensatz zu statischen zudem kinematische Beschreibungen.

Mit dem CAx-Objektbus ist die Übertagung von 3D-Geometriemodellen möglich, da die als Basis für die Datenseite verwendete Conformance Class des STEP Anwendungsprotokolls 214 diese Art von Daten unterstützt. Produktstruktur und Kinematik der Bauteile werden von dieser Conformance Class nicht unterstützt. Sollen statische oder dynamische DMUs übertragen werden, ist es notwendig, diese Daten dem Common Interface hinzuzufügen. Dies bedeutet jedoch zusätzlichen Aufwand bei der Realisierung des Objektbusses. Eine andere denkbare, jedoch nicht beschriebene Möglichkeit Kinematik einzelner Bauteile zwischen Sender und Empfänger auszutauschen, ist Lageänderungen und Bewegungen insgesamt von Originalmodel als Kinematik auszulegen und diese zu übertragen. Ein Problem dabei ist das Erkennen, welche Bewegungen Kinematik für die Montagesimulation beschreiben und welche nicht. Automatisch ist dies in der Regel auf Grundlage der übertragen Daten nicht möglich. Für das Erkennen ist zusätzliches Konstruktionswissen beispielsweise in Form von Features notwendig. Zudem bedeutet auch die Übertragung der Bewegungen zusätzlichen Aufwand für die Erweiterung des Common Interface. Im Vergleich zu den vorgestellten Datenaustauschformaten JT und STEP ist die Übertragung von Produktstruktur und Kinematik somit schwierig. Zu bemerken ist hierbei, dass

JT ebenfalls keine Kinematik unterstützt, dafür aber die Übertragung von Produktstrukturen. Andere STEP Conformance Classes des Anwendungsprotokolls, als die für den Objektbus verwendete, unterstützen DMUs besser, da je nach STEP-Format Produktstrukturen und Kinematikdaten unterstützt werden können. Hier hängt dieser Austausch jedoch auch von der Realisierung der Konverter ab.

Weiterhin variiert der Datenaustausch bei der Montagesimulation in der *Anzahl der Erzeugersysteme*. Bei einem oder wenigen Systemen ist die Verwendung des Ansatzes einer Online-Kopplung über einen CAx-Objektbus oft nicht sinnvoll, da systemspezifische Austauschlösungen mit weniger oder dem gleichen Aufwand realisierbar sind, dabei aber einen höheren Übertragungsumfang besitzen.

Als systemneutrale Strategie ist die Online-Kopplung mit einem CAx-Objektbus eher zur Kopplung mehrere Sende- und Empfängersysteme geeignet. Problematisch ist jedoch an dieser Stelle, dass alle am Austausch beteiligten Systeme laufen müssen und alle zu übertragenden Daten geladen sein müssen. Die technisch verfügbare Rechner- und Netzwerkleistung ist jedoch begrenzt. Daher ist hier dieser Strategie eine Obergrenze gesetzt. Untersuchungen dazu sind jedoch bisher nicht bekannt. Besonders der Aspekt, dass alle Daten in den Sendesystemen geladen sein müssen, kann bei der Montagesimulation ein Problem darstellen, da komplexe Produkte aus einer Vielzahl von Teilen bestehen können. Das Testbeispiel des ANICA-Projektes schlägt vor, Teilbereiche von Bauteilen bei Untersuchungen wie der virtuellen Einbauuntersuchung zu betrachten [SAKJ98]. Bei der Montagesimulation ist dies jedoch nicht immer zielführend, da hier zum einen komplexe Produkte und deren Montage untersucht wird, wodurch die Zahl der benötigten Teile immer noch hoch sein kann, und zum anderen Montagevorgänge im Zusammenhang betrachtet werden sollen. Bei der Untersuchung von Teilbereichen können nur Aspekte betrachtet werden, nicht die Montage des Gesamtproduktes oder der Gesamtbaugruppe. Zur Beurteilung der Aspekte ist dieser Ansatz, Teilbereiche zu untersuchen, dennoch durchaus nützlich, da weniger Aufwand zu erwarten ist als bei einer Gesamtuntersuchung.

Ein weiteres Problem bei sehr vielen Systemen ist der Aufwand für die Verwaltung der Systeme innerhalb des Objektbusses. Auch diesen Aspekt gilt es zu berücksichtigen. Insgesamt sind hier weitere Untersuchungen des Ansatzes zur Online-Kopplung über einen CAx-Objektbus notwendig um konkretere Aussagen treffen zu können.

Das Problem der technischen Grenze bei besonders vielen Bestandteilen, die in einer Montagesimulation verwendet werden sollen, besteht auch bei dateibasierten Datenaustauschstrategien. Hier verschiebt sich diese Grenze jedoch nach oben, da nicht verschiedene Systeme sondern nur das Simulationssystem laufen muss. Zudem muss bei dateibasierten Datenaustauschstrategien das Problem der technischen Grenze nicht für den Datenaustausch gelöst werden, sondern für die Simulation im Simulationssystem. Entsprechende Simulationssysteme, die auf besonders komplexe und große Produkte ausgelegt sind, besitzen hier eigene Lösungsstrategien zur Beherrschung der Komplexität.

Ein weiteres Merkmal, bei dem der Datenaustausch bei der Montagesimulation variiert, ist die *Art des Simulationssystems*. Diese Arbeit unterscheidet hier drei Fälle. Der CAx-Objektbus ist darauf ausgelegt CAx-Systeme zu koppeln. Inwieweit CAx-ferne Systeme unterstützt werden, ist nicht bekannt. In den Fälle A und B sind

jedoch sowohl CAx-Systeme als auch CAx-ferne Systeme möglich. Zudem ist nicht bekannt, ob die Systeme die notwendigen Programmierschnittstellen besitzen und welche spezifischen Daten zum Aufbau eines lokalen Modells benötigt werden. Auch fehlen allgemeingültige Informationen zur Kompatibilität der Systeme zum Objektbus bezüglich der verwendeten Datenstrukturen. Daher sind für die Fälle A und B keine pauschalen Aussagen möglich. Anders ist dies beim Fall C. Hier ist ein Datenaustausch zu einem CAD-System zu realisieren. Diese Systeme gehören zu den CAx-Systemen und sind in der Regel kompatibel bezüglich der verarbeitbaren Datenstrukturen. Das Testbeispiel des Projektes zeigt den Austausch von einem CAD-System zu einem anderen CAD-System. Das bedeutet, der Fall C wurde mit einem CAD-System als Sendesystem in diesem Testbeispiel erfolgreich untersucht. Prinzipiell sollte auch ein Austausch von einem nicht CAD-System zu einem CAD-System mit entsprechendem Simulationsmodul möglich sein, da die Art der übertragbaren Daten gleich bleibt. Jedoch wurde beim Testeinsatz auch festgestellt, dass der Austausch nicht zu allen CAD-Systemen möglich ist. Geplant war auch ein umgedrehter Datenaustausch vom vorherigen Empfänger zum vorherigen Sender zu realisieren. Dabei trat jedoch die Schwierigkeit auf, dass die Programmierschnittstelle dieses zweiten CAD-Systems nicht auf das Erzeugen von Geometrie ausgelegt war. Ob weitere Untersuchungen zur Kopplung in dieser Austauschrichtung durchgeführt wurden ist nicht bekannt. Jedoch zeigt dies, dass der Funktionsumfang der Programmierschnittstellen großen Einfluss auf die Eignung des Ansatzes hat.

Das letzte Merkmal, in dem der Datenaustausch bei der Montagesimulation variiert, ist die *Austauschrichtung*. Unterschieden wird hier zwischen dem Austausch von Eingabedaten zum Simulationssystem und dem Austausch von Ergebnisdaten vom Simulationssystem. Durch das Konzept des Ansatzes realisiert die Online-Kopplung mit einem CAx-Objektbus beide Variationen gleichzeitig, da Änderungen wie farbige Kennzeichnungen direkt nach der Untersuchung an die Originaldaten weitergegeben werden können. Hier liegt ein großer Vorteil dieser Datenaustauschstrategien im Vergleich zu den vorgestellten dateibasierten Strategien. Bei diesen stellt die Rückgabe von Ergebnisdaten zum ursprünglichen Erzeuger einen weiteren Datenaustausch dar.

Voraussetzung für den Austausch über einen CAx-Objektbus, wie er vom ANICA-Projekt vorgeschlagen wird, ist analog zu den vorgestellten Datenaustauschformaten die Kompatibilität der Datenstrukturen der Systeme untereinander und zum CAx-Objektbus. Dieser basiert datenseitig auf der Conformance Class 1 des STEP Anwendungsprotokolls 214. Somit sind hier die gleichen Systeme, wie bei STEP, zu erwarten.

Weiterhin setzt das Konzept das Vorhandensein geeigneter Programmierschnittstellen voraus. Dies schränkt die Zahl der koppelbaren Systeme weiter ein, da dies insbesondere bei CAx-fernen Systemen nicht pauschal voraussetzbar ist. Ob die Online-Kopplung mit einem CAx-Objektbus als Datenaustauschstrategie für das verwendete Beispielszenario geeignet ist, hängt somit von den konkreten beteiligten Systemen ab und muss im Einzelfall untersucht werden.

Insgesamt zeigen die Ausführungen, dass der Ansatz unter bestimmten Bedingungen für den Datenaustausch bei der Montagesimulation geeignet ist und im Vergleich zu dateibasierten Austauschstrategien Vorteile besitzen kann. Doch ist der Ansatz nicht immer geeignet und bringt neben Vorteilen auch Probleme mit sich.

Ob sich die Realisierung dieses Ansatzes lohnt, hängt von der Komplexität der Produkte, der Zahl der Systeme und den Eigenschaften der Systeme ab. Vorteilhaft sind im Vergleich zu den bisher vorgestellten Strategien, die Redundanzfreiheit und die Möglichkeit der Weitergabe von Ergebnissen. Nachteilig sind unter anderem der zusätzliche Aufwand für weitere Daten, wie Kinematik und Produktstruktur, die Problematik der technischen Grenze der Online-Kopplung und die Voraussetzung geeigneter Programmierschnittstellen. Insgesamt ist die Online-Kopplung mit einem CAx-Objektbus jedoch eine interessante Alternative zu dateibasierten Strategien.

Der nächste Abschnitt beschäftigt sich mit einer weiteren Austauschstrategie, die eine Alternative zum dateibasierten Austausch darstellt. Dies arbeitet im Gegensatz zum vorgestellten Ansatz mit OpenGL statt mit einem CAx-Objektbus.

4.4 Online-Kopplung mit OpenGL

Eine neuere Strategie zum Austausch geometrischer Daten zwischen heterogenen Systemen ist die Online-Kopplung mit Hilfe von OpenGL-Streams. Dieser Ansatz wird in diesem Abschnitt näher betrachtet.

OpenGL ist eine plattformunabhängige Spezifikation für eine Programmierschnittstelle zwischen Softwaresystem und Grafikeinheit zur Darstellung von 2D- und 3D-Geometrien [Cla97, Shr10]. Je nach Realisierung greifen die Methoden dabei direkt auf die Hardware zu und nutzen so spezielle Funktionalitäten dieser möglichst optimal aus [Cla97]. Werden Methoden nicht direkt von der Hardware unterstützt, so werden sie in unterstützte einfachere Befehle übersetzt [Cla97]. Dies ist jedoch Teil der OpenGL-Realisierung und für das Datenaustauschkonzept nicht von Bedeutung.

Die Darstellung von geometrischen Objekten des Softwaresystems erfolgt über Methodenaufrufe [Cla97]. Dies beinhaltet die Definition der Objekte selbst, sowie ihrer Bewegung und ihrer darstellungsbezogenen Eigenschaften [Cla97]. Initial bietet OpenGL nur die Definition über konvexe Flächen an [Cla97]. Das heißt, Körper werden durch ihre Oberflächen beschreiben. Dabei muss jede Fläche einzeln definiert werden. Unterstützt wird dies in OpenGL durch Darstellungslisten, welche die Wiederverwendung von definierten Objekten ermöglichen [Cla97]. Durch die Zusatzbibliothek GLUT sind jedoch auch einfache Primitiven möglich [Cla97, Shr10]. Diese können als Drahtmodell oder als Volumenmodell definiert werden. Darstellungsbezogene Eigenschaften wie Farben, Beleuchtung, Schatten, Transparenz und Füllung von Flächen werden über zusätzliche Methoden gesteuert, wobei gesetzte Parameter solange gelten bis diese neu besetzt werden [Cla97, Shr10]. Die Objekte können mit OpenGL translatorisch oder rotatorisch bewegt werden [Cla97, Shr10]. Zudem sind verschiedene Projektionsansichten und Skalierungen möglich [Cla97, Shr10]. Methodenaufrufe realisieren die komplette Darstellung der Objekte. Sie bilden dabei eine spezifische Abfolge, die als OpenGL-Stream bezeichnet wird.

OpenGL nimmt unter den Grafikbibliotheken die Stellung eines Industriestandards ein [WBTM00, SLLT06]. Eine andere ebenfalls weit verbreitete Grafikschnittstelle für 3D-Darstellungen ist Direct 3D [WBTM00]. Nachteil dieser Schnittstelle ist die Plattformabhängigkeit zu Windows [WBTM00]. Daher verwendet die Strategie OpenGL statt einer anderen 3D-Graphikbibliothek.

Der Ansatz zur Online-Kopplung von heterogenen Systemen mit Hilfe von OpenGL sieht vor, bestehende OpenGL-Streams der Sendesysteme zu ihren Grafikeinheiten auszulesen, diese auszuwerten und die Erkenntnisse daraus an den Empfänger weiterzugeben. Eine Voraussetzung dafür ist, dass alle notwendigen Sendesysteme mit OpenGL arbeiten. Ansonsten ist auf diese Weise kein Austausch möglich.

Ansätze, die OpenGL zur Realisierung von Interoperabilität nutzen, sind beispielsweise Lumino [SLLT06], BroadcastGL [IRK05] und Chromium [HHN+02]. Dabei beschränken diese sich auf die verteilte Nutzung der OpenGL-Streams. Aspekte zur Interpretation der Streams werden nicht behandelt. In [MMMK12] beschreiben Mory et al. bei der Realisierung eines Austausches zwischen einem proprietären System und einem neuartigen Display auch die Umsetzung vom OpenGL-Streams zur Semantik des Displays. Nähere Informationen zu diesen Ansätzen sind der Literatur zu entnehmen. Beispielsweise geben Mory et al. [MMMK12] hier einen Überblick. Im Bereich der kommerziellen Systeme nennen Mory et al. [MMMK12] TechVizXL und ICIDO's Capture als Systeme, die OpenGL für die Interoperabilität verwenden. Jedoch sind zu beiden Systemen keine weiteren Informationen verfügbar. Daher sind hierzu keine Aussagen möglich.

Der Ansatz mit OpenGL-Streams einen Datenaustausch zu realisieren, birgt verschiedene Herausforderungen und Nachteile gegenüber anderen Datenaustauschstrategien. Herausforderungen sind das Auslesen und Interpretieren des Streams sowie die Trennung von Streams, wenn ein System mehrere Streams produziert. Die Aspekte Auslesen und Interpretieren sind Teil der Strategie und müssen für einen funktionierenden Austausch realisiert sein. Die Trennung von Streams ist notwendig, um eine korrekte Interpretation gewährleisten zu können. Je nach Realisierung wird dies zur Herausforderung, wenn mehrere Streams aus einem System stammen. Dies ist der Fall, wenn ein System (z.B. ein CAD-System) mehrere getrennte Objekte (z.B. Bauteile) geladen hat und diese in getrennten Fenstern dargestellt werden.

Ein Nachteil dieser Datenaustauschstrategie ist der Umfang der übertragbaren Daten. OpenGL wird verwendet, um Geometrien darzustellen. Daher kann diese Strategie nur zum Austausch von darstellungsbezogenen Daten verwendet werden. Darstellungsbezogene Daten sind dabei Geometrien an sich, Farben, Texturen, Licht, Schatten und Bewegungen der Geometrien in der Darstellung. Andere Daten sind nicht möglich.

Für die Realisierung des Austausches ist es zudem notwendig, für alle Daten darstellende Systeme zu besitzen. Das bedeutet, dass für diese Systeme im Ingenieurwesen häufig Lizenzkosten gezahlt werden müssen. Im Gegensatz dazu ist es bei einem übersetzenden Austausch möglich, gegebene Dateien in einem systemspezifischen Format eines Fremdsystems über einen externen Konverter in ein neutrales Format oder das systemspezifisches Format eines eigenen Systems zu übersetzen. In diesem Fall sind keine Lizenzkosten für das Fremdsystem fällig. Dies ist jedoch beim Austausch über OpenGL in einer Online-Kopplung nicht möglich.

Der Vorteil bei der Verwendung von OpenGL als Interoperabilitätsplattform ist die weitgehende Unabhängigkeit von internen Strukturen der Sendesysteme. Das hat zur Folge, dass auch Systeme verbunden werden können, deren verarbeitbaren Datenstrukturen inkompatibel sind. Hier unterscheidet sich diese Strategie von den bisher

beschriebenen Strategien. Einzige Voraussetzung ist die Verwendung von OpenGL innerhalb der Systeme zur Kommunikation mit der Grafikeinheit. Ein weiterer Vorteil ist, dass der Austausch in Echtzeit erfolgt. So wird immer mit den aktuellsten Daten gearbeitet und Veränderungen sind direkt sichtbar.

Dieser Abschnitt hat eine neuere Strategie für den Austausch geometrischer Daten beschrieben. Die Entwicklung dieser Strategie ist heute noch nicht abgeschlossen. Daher sind alle Ausführungen als Momentaufnahmen zu werten. Auch sind bisher keine direkten Vergleiche zwischen der neuen Strategie und etablierten Strategien bekannt. Hier ist weiterer Untersuchungsbedarf zu sehen. Der nächste Abschnitt ordnet diese neue Strategie in die vorgestellten Klassifikationsschemata ein und zeigt weitere Unterschiede und Gemeinsamkeiten zu den bisher beschriebenen Datenaustauschstrategien.

Einordnung in Klassifikationsschemata

Dieser Abschnitt beschäftigt sich mit der Einordnung der Online-Kopplung mittels OpenGL in die in Abschnitt 2.2 vorgestellten Klassifikationsschemata. Die Ergebnisse dieser Einordnung sind im Anhang in Tabelle A.1 im Überblick dargestellt.

Die Klassifikation *nach der Art der Daten* unterteilt Datenaustauschstrategien nach produktionstechnischen Aspekten und nach dem Geometriegehalt in unterschiedliche Gruppen. Bei der Online-Kopplung mit OpenGL können nur Geometrien und darstellungsbezogene Eigenschaften übertragen werden. Damit gehört die Online-Kopplung mit OpenGL in die Klasse der Datenaustauschstrategien zur Übertragung geometrischer Daten. Übertragbare nicht-geometrische Daten sind ausschließlich Geometrieattribute und Bewegungen der Geometrien in der Darstellung. Beim Austausch der Daten ist es nicht relevant, ob diese produktbezogen, prozessbezogen oder auftragsbezogen sind. Somit können über die Online-Kopplung mit OpenGL alle produktionstechnischen Arten von Daten übertragen werden. Die Einschränkung liegt hier im Geometriegehalt der Daten. Hier unterscheidet sich diese Strategie von den anderen vorgestellten Strategien.

Die Datenaustauschstrategie gehört bei der Klassifikation *nach der Systemneutralität* in die Klasse der systemneutralen Lösungen. Die Interoperabilitätsplattform ist hierbei OpenGL. Der Vorteil im Gegensatz zu den bisher betrachteten Strategien ist, dass von den Sendesystemen keine Konvertierung in die neutrale Zwischenstufe erforderlich ist. Als Adapter ist lediglich eine Programmeinheit zum Auslesen und gegebenenfalls Manipulieren des OpenGL-Streams notwendig. Wird der ausgelesene OpenGL-Stream nur an anderer Stelle direkt zur Darstellung benutzt, so ist auch keine Interpretation des Streams notwendig. Soll hingegen mit den Geometriedaten weitergearbeitet werden, ist diese Interpretation des OpenGL-Streams notwendig und muss durch einen Adapter realisiert werden.

N*ach ihrem Konzept* wird der OpenGL-basierte Lösungsansatz den einbindenden Ansätzen zugeordnet. Das ausgetauschte Objekt ist dabei der OpenGL-Stream. Durch das Auslesen wird der Ansatz eher den einbettenden Ansätzen zugeordnet, da statt Referenzen das Objekt selbst übertragen wird. Die Zuordnung hier ist nicht endgültig, da der OpenGL-Stream kein Objekt im klassischen Sinn ist. Vom Lösungsprinzip her ähnelt der Ansatz unter dieser Bedingung den einbindenden und speziell den einbettenden Austauschstrategien am meisten.

Bei der Einteilung *nach der Spezialisierung des Austausches* gehört die Online-Kopplung mittels OpenGL analog zu JT und STEP zu den Verfahren für einen generalisierten Austausch. Der Ansatz ist durch die Verwendung des Industriestandards OpenGL und sein Konzept weitgehend unabhängig von konkreten Sendern. Durch die Beschränkung auf Geometriedaten, Geometrieattribute und Bewegungen sind die Anwendungsbereiche des Ansatzes jedoch eingeschränkt. Zudem ist der Ansatz durch die Online-Kopplung nur in bestimmte Online-verfügbare Systeme einsetzbar. Ein Austausch mit beliebigen Systemen auch außerhalb des Netzwerkes ist nicht möglich. Des Weiteren gilt zur Realisierung des Austausches für die Systeme die Voraussetzung, dass sie mit OpenGL arbeiten müssen. Dennoch zählt der Ansatz nicht zu den Verfahren zur Realisierung eines spezialisierten Austausches, da der Ansatz trotz allem nicht auf konkrete Anwendungsszenarien, konkrete Systeme oder Randbedingungen beschränkt ist.

Vorgestellt wurde auch die Klassifikation *nach Interoperabilitätsleveln*. Die Online-Kopplung mittels OpenGL realisiert syntaktische, semantische und zum Teil pragmatische und dynamische Interoperabilität. Syntax und Semantik werden durch die OpenGL-Spezifikation festgelegt. Pragmatische Interoperabilität bezieht sich nach der Definition aus Abschnitt 2.2 (auf Seite 18) auf die Möglichkeit der Systeme auf einander zuzugreifen. Bei der Online-Kopplung mittels OpenGL ist dies bezüglich des OpenGL-Streams für die Darstellung möglich. Realisiert wird dies durch Manipulation des Originalstreams und der Rückgabewerte der OpenGL-Methode. Ein direkter Eingriff in das Sendesystem oder das Empfängersystem ist jedoch nicht möglich ohne zusätzliche Programmeinheiten.

Da der OpenGL-Stream fortlaufend ist, müssen alle Methodenaufrufe auch fortlaufend ausgelesen und protokolliert werden. In dieser Form erfolgt der Austausch automatisch. Jedoch muss die Analyse und Weitergabe der Daten nicht zwingend automatisch erfolgen. Abhängig ist die Bestimmung des Zeitpunktes für die Übergabe der Änderungen an den Empfänger von der konkreten Realisierung. Hierbei ist auch von Bedeutung, ob ein reines Auslesen der Daten erfolgt oder ob über neue Befehle oder die Rückgabewerte, der ausgelesenen Befehle, Einfluss auf die Darstellung genommen werden soll. Bei einem reinen Auslesen der Daten ohne jeglichen Einfluss auf den Stream und somit ohne Realisierung pragmatischer Interoperabilität ist die manuelle Bestimmung des Analyse- und Weitergabezeitpunktes möglich. Dabei ist auch eine Online-Kopplung nicht zwingend notwendig. Der Stream kann auch in einer Protokolldatei zwischengespeichert werden. Anders ist die, wenn eine Manipulation des Streams oder der Rückgabewerte erfolgen soll. In diesem Fall ist eine Online-Kopplung zwingend notwendig. Die Daten müssen dabei in Echtzeit analysiert und an den Empfänger weitergeben werden. Dies gilt auch für die Manipulationen der Rückgabewerte. Hier ist somit ein automatischer Austausch zu realisieren. Insgesamt bedeutet dies für die dynamische Interoperabilität, dass der Sender durch das Auslesen des OpenGL-Streams stetig auf Änderungen überwacht wird. Somit ist ein Einfluss des überwachenden Systems auf den Datenaustausch möglich und dynamische Interoperabilität wird realisiert.

Bei der Klassifikation *nach Realisierungsmerkmalen* ist ein Merkmal die Zahl der Domänen. Für den Austausch über eine Online-Kopplung mittels OpenGL ist die Domäne eines Systems nicht von Bedeutung. Solange das System die Geometriedaten

über OpenGL an die Grafikeinheit weitergibt, ist ein Datenaustausch möglich. Was konkret dargestellt wird, hat keinen Einfluss auf den Austausch an sich. Ob die Daten, die ausgetauscht werden, sinnvoll sind, ist für den Austauschprozess nicht von Bedeutung. Insgesamt erlaubt die Online-Kopplung mit OpenGL sowohl einen in-phasen als auch einen inter-phasen Austausch.

Über den OpenGL-Stream werden Änderungen fortlaufend weitergegeben. Daher gehört diese Lösungsstrategie nach der Art der Weitergabe von Änderungen zur Klasse der inkrementellen Austauschstrategien. Der Vorteil ist, dass bei einer Änderung weniger Daten übertragen werden müssen. Durch den Stream ist eine Identifizierung der Änderungen nicht notwendig, da alle neuen Befehle Änderungen an der Darstellung sind. Jedoch ist damit bei Realisierung pragmatischer Interoperabilität auch eine stetige Analyse und Interpretation der Daten notwendig. Zudem müssen hier bei Untersuchungen, wo Geometrieänderungen innerhalb der Untersuchung nicht erlaubt sind, Sperren realisiert werden, um eine unerwünschte Änderung in diesem Zeitraum zu verhindern. Wird keine pragmatische Interoperabilität realisiert, dann ist auch ein Sammeln der Änderungen und Übertragung dieser zu einem sicheren Zeitpunkt möglich. Dabei sollte darauf jedoch geachtet werden, Untersuchungsergebnisse nicht versehentlich durch Einfügen der gesammelten Änderungen zu vernichten.

Ein weiteres Realisierungsmerkmal ist die Vollständigkeit der Daten. Es werden ausschließlich Geometriedaten übertragen. Der Austausch beschränkt sich damit zwar einerseits auf Geometriedaten. Andererseits werden alle Darstellungsbefehle des Streams vollständig ausgelesen. Welche Daten daraus an das Empfängersystem weitergegeben werden, hängt von der Interpretation der Daten ab und davon, in wie weit diese auf den Empfänger abgestimmt ist. Insgesamt ist damit die Zuordnung zu beiden Klassen möglich und hängt von der konkreten Realisierung ab.

Abhängigkeiten der Daten, die durch die Datenaustauschstrategie verwaltbar sind, existieren nicht. Daher ist eine Einordnung in die Klassifikation nach dem Austauschmanagement nicht möglich.

Das letzte behandelte Realisierungsmerkmal beschäftigt sich mit den unterstützten Prozesstypen der Produktentwicklung. Bisher sind wenig konkrete Untersuchungen für die Strategie bekannt. Daher ist es nicht möglich, hier gesicherte Aussagen zu treffen. Doch theoretisch unterstützt diese Austauschstrategie alle drei Prozesstypen, da durch die Online-Kopplung ein schneller Austausch von Änderungen zwischen verschiedenen Beteiligten möglich ist. Im einfachsten Fall werden dabei reine Darstellungen ermöglicht, indem der übertragene OpenGL-Stream ohne Interpretation nur erneut an eine Grafikeinheit gesendet wird. Jedoch sind mit der Interpretation der Daten auch weiterführende Untersuchungen der Geometrieobjekte möglich. Vorteil des Ansatzes ist insgesamt, dass alle Geometrien übertragbar sind. Somit sind in der Produktentwicklung neben Bauteil- und Werkzeuggeometrien auch Diagramme und andere Visualisierungen übertragbar und gegebenenfalls gemeinsam darstellbar. So können Beziehungen zwischen den verschiedenen Geometriedaten hergestellt werden. Dies kann auch zur Unterstützung der Produktentstehung und der Kommunikation zwischen Beteiligten insgesamt eingesetzt werden.

Das letzte vorgestellte Klassifikationsschema unterscheidet Datenaustauschstrategien *nach dem verwendeten Datenaustauschformat*. Die Online-Kopplung mit OpenGL ist keine dateibasierte Austauschstrategie und kann daher hier nur schwer eingeordnet werden.

OpenGL ist kein Datenformat sondern eine Grafikbibliothek. Dennoch ordnet Thierfelder in [Thi04] OpenGL den Echtzeit und VR-Formaten zu. Zu beachten ist hier, dass alle dieser Klasse zugeordneten Beispiele des Autors keine Datenformate sind. Jedoch stimmt diese Zuordnung zumindest für OpenGL insoweit, dass diese Grafikbibliothek in Echtzeit und VR-Anwendungen verwendet wird. Alles in allem ist die Klassifikation von Thierfelder an dieser Stelle kritisch zu betrachten und bedarf Anpassungen.

Weitere Eigenschaften von Datenaustauschformaten wie Verbindlichkeit der Daten, Standardisierung, Offenheit und Leichtgewichtigkeit sind nur schwer zu beurteilen. Standardisiert ist die Strategie bisher noch nicht. Jedoch existieren verschiedene Veröffentlichungen zur Austauschstrategie und die Spezifikation von Chromium ist frei verfügbar. Daher kann die Strategie als teilweise offen beurteilt werden. Die übertragenen Daten haben dabei informellen Charakter. Eine Zuordnung zu primären und sekundären Formaten ist dennoch nicht möglich, da die Strategie kein Datenaustauschformat verwendet. Zur Leichtgewichtigkeit sind keine Aussagen möglich, da keine Dateigrößen verglichen werden können.

Dieser Abschnitt hat die Online-Kopplung mit OpenGL in die vorgestellten Klassifikationsschemata eingeordnet. Im Zuge dieser Einordnung wurden Unterschiede und Parallelen zu den drei vorher betrachteten Datenaustauschstrategien aufgezeigt. Der größte Unterschied hierbei ist die Art der Kopplung. Im Gegensatz zu den drei anderen Strategien verwendet diese Strategie eine interne Schnittstelle des Sendesystems zum Auslesen der Daten. Durch OpenGL ist dabei ein Austausch von beliebigen Geometrien möglich. Dies beschränkt sich nicht auf wie bei den drei anderen Strategien auf Produkt- oder Werkzeuggeometrien. So sind beispielsweise auch Diagramme übertragbar. Änderungen können jedoch durch die Strategie im Gegensatz zur Online-Kopplung mittels Objektbus nicht an den Originaldaten sondern nur an der Darstellung vorgenommen werden. Die Online-Kopplung mit OpenGL realisiert genau wie die Datenaustauschformate STEP und JT einen generalisierten Austausch. Gemeinsam ist allen vier Strategien, dass sie sowohl für den in-phasen Austausch als auch für den inter-phasen Austausch geeignet sind. Ebenso können alle betrachten Strategien geometrische Daten übertragen. Insgesamt zeigt die Einordnung, dass die Strategie, den Datenaustausch über den OpenGL-Stream zu realisieren, eine Ergänzung zu bisherigen Strategien sein kann. Jedoch sind weitere Untersuchungen notwendig. Zudem erweckt die Einordnung den Anschein, dass die Strategie nicht für jedes Anwendungsszenario geeignet ist. Der nächste Abschnitt beurteilt die Strategie für das Beispielszenario dieser Arbeit. Dabei wird auch auf die Variationen des Szenarios eingegangen.

Beurteilung für das Anwendungsszenario

Dieser Abschnitt beschäftigt sich mit der Beurteilung der OpenGL-basierten Online-Kopplung für das Beispielszenario, dass in Kapitel 3 näher erläutert wurde. Bei der

Betrachtung des Datenaustausches bei der Montagesimulation wurden verschiedene Variationen des Szenarios identifiziert. Diese gehen in die Beurteilung der Austauschstrategie mit ein. Zudem werden Gemeinsamkeiten und Unterschiede zu den bisher vorgestellten und beurteilten Datenaustauschstrategien aufgezeigt.

Ein Merkmal, in dem der Datenaustausch bei der Montagesimulation variiert, ist der *Modellumfang*. Variationen sind hier die Übertragung von 3D-Geometriemodellen, von statischen oder dynamischen DMUs. Bei der Online-Kopplung mit OpenGL können aufgrund des Funktionsumfanges von OpenGL nur 3D-Geometriemodelle übertragen werden. Dabei ist es egal, ob dieses Modell die Geometrie von Bauteilen, Ersatzgeometrie von Unterbaugruppen oder Werkzeugen beschreiben. Durch Analyse der Bewegungen, die über den OpenGL-Stream mit übertragen werden können, ist es eventuell möglich, Kinematiken von Bauteilen für die Montagesimulation zu identifizieren. Die Schwierigkeit dabei ist es, die Bewegungen, die Kinematiken für die Montagesimulation beschreiben, von Bewegungen, die der Betrachtung oder anderen Animationen gelten, zu unterscheiden. Eine Bewegung, die für die Montagesimulation von Bedeutung ist, ist beispielsweise die eingeschränkte Drehung eines Bauteils um die Achse eines Gelenkes. Diese Drehung gilt es von einer Drehung zur Betrachtung einer anderen Körperseite zu unterscheiden. Dies ist automatisch jedoch aufgrund der verfügbaren Daten zur Geometrie nicht möglich. Hier muss der Anwender eingreifen und bestehendes Konstruktionswissen beispielsweise zu Gelenken nutzen. Somit ähneln sich die Online-Kopplung mit OpenGL und die Online-Kopplung mit einem CAx-Objektbus im übertragbaren Modellumfang.

In der Regel sind die Geometriemodelle, die mit OpenGL dargestellt werden, approximiert. Gründe dafür sind zum einen der initiale Funktionsumfang von OpenGL und zum anderen die Verringerung der Renderingzeit durch die Approximation. Über die Online-Kopplung mit OpenGL können nur die dargestellten Modelle übertragen werden. Einfluss auf die Genauigkeit der Daten hat hier nur das Sendesystem.

Bei der Variation der *Anzahl an Erzeugersystemen* treten ähnliche Schwierigkeiten, wie bei der Online-Kopplung mit einem CAx-Objektbus auf. Bei einem oder wenigen Systemen kann eine direkte systemspezifische Austauschlösung bei ähnlichem Aufwand eine größere Menge an Daten übertragen. Bei vielen Sendesystemen sind systemneutrale Lösungen wie die Online-Kopplung mit OpenGL bezüglich des Aufwandes vorteilhaft. Das Problem bei der Online-Kopplung allgemein ist, die notwendige Rechenleistung um alle Sendesysteme online zu halten. Nicht zu vernachlässigen ist dabei, dass alle Systeme die notwendigen Geometrieobjekte geladen haben müssen. Bei der Online-Kopplung mittels OpenGL kommt hier noch die Verwaltung der Streams und die Antwortfähigkeit bei Echtzeitbetrieb hinzu. Gerade bei komplexen Produkten mit vielen Bauteilen und Unterbaugruppen kann dies ein Problem darstellen. Bei dateibasierten Datenaustauschstrategien verlagert sich das Problem der begrenzten Rechenleistung zum Simulationssystem. Diese muss dort die vielen Geometrieobjekte handhaben. Vorteil hier ist, dass die notwendige Rechenleistung für die einzelnen Sendesysteme nicht belegt wird, da diese nicht geladen sein müssen. Zudem besitzen spezielle Simulationssysteme für komplexe Produkte eigene Mechanismen für dieses Problem. Im Vergleich sind daher die beiden Online-Kopplungs-Ansätze zu den Datenaustauschformaten im Nachteil und können nur bedingt bei

besonders vielen Systemen eingesetzt werden. Konkrete Untersuchgen zur Grenze bei der Online-Kopplung mit OpenGL sind jedoch nicht bekannt.

Zur *Art von Simulationssystemen* können für die Online-Kopplung mittels OpenGL keine konkreten Aussagen getroffen werden. Hierzu fehlen bestehende Untersuchungen. Jedoch die Weitergabe der interpretierten Daten an dieses System wird als eine Herausforderung betrachtet, da diese Daten in einer Form sein müssen, die von diesem System verarbeitet werden kann. Diese Schwierigkeit besteht sowohl bei Fall A wie auch bei Fall B und Fall C. Zu beachten ist hier auch, dass häufig keine exakte Geometrie übertragen werden kann. Dies führt beim Fall C möglicherweise zu Problemen. Eine denkbare Lösung der Schwierigkeiten bei der Weitergabe an das Simulationssystem ist die Entwicklung eines eigenen Systems zur Simulation der Montage. Vorteile dieser Lösung sind, dass eine optimale Anpassung an die Eingangsdaten möglich ist und der Interpreter in das System integriert werden kann. Nachteilig ist jedoch der zusätzliche Aufwand für die Entwicklung.

Ein weiteres Merkmal, das variiert, ist die *Richtung des Austausches*. Möglich ist neben dem Austausch von Eingangsdaten in das Simulationssystem auch der Austausch von Ergebnisdaten aus dem Simulationssystem. Je nach Realisierung der Online-Kopplung mit OpenGL erlaubt die Strategie auch die Rückgabe von Ergebnisdaten. Voraussetzung dafür ist jedoch, dass die Online-Kopplung so realisiert ist, dass eine Manipulation des Originalstreams möglich ist. Das bedeutet, pragmatische Interoperabilität muss realisiert sein. Dann können Einfärbungen an bestimmten Stellen der Darstellung eingefügt werden. Der Nachteil hierbei ist jedoch, dass eine direkte Änderung der Daten im Sendesystem nicht möglich ist. Damit können die Einfärbungen auch nicht gespeichert werden. Nur die Darstellung durch die Grafikeinheit wird verändert. Bei den anderen vorgestellten Datenaustauschstrategien ist dies anders. Bestimmte STEP-Formate und das JT-Format erlauben prinzipiell die Übertragung von Farben gemeinsam mit dem Modell in einer Datei. Ein Problem hierbei ist unterschiedliche Realisierung der Konverter. Zudem ist unklar, ob und in welchen Systemen die Modelle editierbar sind. Die Online-Kopplung mit einem CAx-Objektbus erlaubt im Gegensatz dazu die Einfärbung der Ausgangsmodelle. Voraussetzung dazu ist, dass das Ausgangssystem des Modells eine Programmierschnittstelle besitzt, die den Zugriff auf die erforderlichen Funktionalitäten realisiert.

Voraussetzung für die Datenaustauschstrategien STEP, JT und die Online-Kopplung über einen CAx-Objektbus ist die Kompatibilität der Systeme mit den Datenstrukturen der Austauschstrategien. Dies grenzt die Systeme, zwischen denen ein Austausch möglich ist, ein. Diese Voraussetzung gilt für die Online-Kopplung mit OpenGL nicht. Der Austausch ist unabhängig von der internen Datenstruktur der Systeme. Voraussetzung ist hingegen die Verwendung von OpenGL in den Sendesystemen. Hier unterscheidet sich die Online-Kopplung mittels OpenGL drastisch von den anderen Strategien.

Thema dieses Abschnittes war die Beurteilung von der Online-Kopplung mittels OpenGL für das Beispielanwendungsszenario. Dabei zeigte sich, dass diese nicht für alle Variationen des Anwendungsszenarios gleichermaßen geeignet ist. Erläutert wurden Bedingungen, Vorteile und Schwierigkeiten beim Einsatz der Strategie. Zudem wurde auf Unterschiede und Gemeinsamkeiten zu den anderen drei behandelten Datenaustauschstrategien eingegangen. Geeignet ist die Online-Kopplung mittels

OpenGL für den Austausch von 3D-Geometriemodellen. Die Übertragung von Kinematik ist mit Einsatz des Anwenders und dessen Konstruktionswissen ebenfalls denkbar. Jedoch kann immer nur ein Bauteil oder eine Ersatzgeometrie einer Unterbaugruppe übertragen werden, da die Identifikation einzelner Produktbestandteile in einem Stream, bei dem mehrere dargestellt werden, nicht möglich ist. Analog zur Online-Kopplung mit einem CAx-Objektbus bestehen bei der Strategie mit OpenGL Probleme bei besonders vielen Erzeugersystemen hinsichtlich der notwendigen Rechenleistung. Erkenntnisse zur Verwendung der Strategie mit bestehenden Simulationssystemen gibt es nicht. Kennzeichnungen an der Bauteilen können zwar im Erzeugersystem dargestellt werden, jedoch nicht gespeichert. Insgesamt bestehen bei der Online-Kopplung mit OpenGL somit viele Einschränkungen für die Realisierung des Datenaustausches bei der Montagesimulation. Der große Vorteil der Strategie gegenüber den anderen vorgestellten Strategien ist jedoch, dass der Austausch unabhängig von internen Strukturen der Erzeugersysteme ist. Damit müssen auch keine Kompatibilitätsbedingungen bezüglich Datenstrukturen zur Austauschstrategie beachtet werden. Die Voraussetzung ist lediglich, dass die Sendesysteme OpenGL verwenden. Vorteilhaft dabei ist die große Verbreitung dieser Grafikbibliothek als Schnittstelle zur Grafikeinheit.

Der nächste Abschnitt fasst die Ergebnisse dieses Kapitels zusammen und gibt ein Fazit.

4.5 Zusammenfassung

Dieses Kapitel hat sich mit dem Vergleich von vier ausgewählten Lösungsansätzen für den Datenaustausch beschäftigt. Dabei wurde jeder Lösungsansatz einzeln vorgestellt, in die behandelten Klassifikationsschemata eingeordnet und für die Verwendung im Beispielszenario beurteilt. Im Zuge dieser Erläuterungen wurden Gemeinsamkeiten und Unterschiede sowie Vor- und Nachteile der einzelnen Ansätze aufgezeigt. Tabelle 4.1 fasst diese Vor- und Nachteile vergleichen zusammen.

Die Vor- und Nachteile der einzelnen Strategien haben auch Einfluss auf die Verwendbarkeit beim betrachteten Beispielszenario. Hier war zudem zu beobachten, dass die unterschiedlichen Variationen des Szenarios von den einzelnen Strategien unterschiedlich gut unterstützt wurden. Tabelle 4.2 fasst die Ergebnisse hierzu zusammen

Die Untersuchung erfolgte jedoch nur theoretisch. Eine praktische Umsetzung wurde nicht durchgeführt. Zudem waren an mehreren Stellen keine konkreten, allgemeingültigen Aussagen möglich. Hier sind ebenfalls weitere Untersuchungen notwendig. Weiterhin ist zu bemerken, dass die Strategien STEP, JT und Online-Kopplung mit OpenGL heute immer noch weiterentwickelt werden. Die Betrachtungen stellen daher eine Momentaufnahme dar.

Die durchgeführte Untersuchung beschränkte sich in dieser Arbeit auf vier Datenaustauschstrategien. Weitere Strategien können analog dazu untersucht werden. Zudem sind weitere Untersuchungen mit anderen Anwendungsszenarien möglich.

Strategie	Vorteile	Nachteile
STEP	- viele bestehende Konverter - großes Verbreitung - viele Anwendungsgebiete - viele verschiedene Daten - stetige Weiterentwicklung	- Spezifikation kostenpflichtig - sehr umfangreich - Realisierung von Konvertern teuer - unterschiedlich umfangreiche Realisierung der Konverter - keine Minimierung der Dateigröße - inkompatible Anwendungsprotokolle - viele Formate zu realisieren und verwalten
JT	- Leichtgewichtig - verschiedene exakte und approximierte Modelle übertragbar - kostenfreie Spezifikation - kostenfreier Viewer - binäre Datei - Modell-Assoziativitäten möglich	- unterschiedliche Realisierung der Konverter - weniger Anwendungsgebiete und Daten
CAx-Objektbus	- keine Datenredundanz in Dateien - direkte Übertragung von Änderungen möglich - inkrementeller Austausch von Änderungen	- alle Systeme müssen laufen und alle Daten geladen sein - nur CAx-Systeme - Programmierschnittstelle notwendig - stark von Funktionsumfang der Systeme und Programmierschnittstellen abhängig - ein System muss Daten initial laden
OpenGL	- keine Änderungen/ Realisierungen am Sendesystem notwendig - Austausch in Echtzeit - Unabhängig von Inhalt der Daten (Bauteile, Diagramme, Textanzeigen) - Unabhängig von innerer Struktur der Sendesysteme	- nur darstellungsbezogene Daten - nur wenn Sender OpenGL verwendet möglich - noch in der Entwicklung - darstellende Systeme für Daten notwendig - Änderungen nur an OpenGL-Darstellung und Rückgabewerten von OpenGL-Methoden möglich

Tabelle 4.1: Vor- und Nachteile der betrachteten Datenaustauschstrategien

Merkmale		STEP	JT	CAx-Objektbus	OpenGL
Modellumfang	3D-Geometriemodelle	x	x	x	x
	statische DMUs	x	x	(x)	-
	dynamische DMUs	x	-	(x)	-
Approximationen		x	x	(x)	x
Zahl der Sendesysteme	ein oder wenige Systeme				
	viele Systeme	x	x	(x)	(x)
Art des Simulationssystems	Fall A	?	?	?	?
	Fall B	?	?	?	?
	Fall C	x	x	x	?
Richtung des Austausches	Eingangsdaten	x	x	x	x
	Ergebnisdaten	(x)	(x)	x	(x)
Voraussetzung	kompatible Datenstrukturen	x	x	x	-
	Programmierschnittstelle	-	-	x	-
	Verwendung von OpenGL	-	-	-	x

x = ja; (x) = bedingt; - = nein; ? = unbekannt

Tabelle 4.2: Unterstützung des Beispielszenarios durch die betrachten Austauschstrategien im Vergleich

5. Zusammenfassung und Ausblick

Diese Arbeit beschäftigte sich mit dem Vergleich von Lösungsstrategien zum Austausch von Daten zwischen heterogenen Systemen im Ingenieurwesen. Dabei beschränkte sich die Arbeit auf Kopplungslösungen. Integrationslösungen wurden nicht näher betrachtet. Auch der Datenversand war nicht Thema dieser Arbeit und wurde als gegeben angenommen. Betrachtet wurden Lösungen zur Überwindung der Heterogenität der Systeme.

In Kapitel 1 wurden die Motivation und die Zielsetzung dieser Arbeit erläutert. Das Ziel war, ausgewählte Datenaustauschstrategie im Ingenieurwesen beispielhaft an einem konkreten Beispielszenario zu vergleichen, sodass weitere Vergleiche mit anderen Lösungsstrategien und anderen Anwendungsszenarien auf ähnliche Weise möglich sind. Weiterhin wurden drei Fragestellungen zur Beantwortung in dieser Arbeit genannt:

- Welche Klassifikationsschemata gibt es für Datenaustauschstrategien im Ingenieurwesen?
- Was ist ein konkretes Anwendungsszenario und welche Anforderungen hat es?
- Wie werden die ausgewählten Datenaustauschstrategien in die Klassifikationsschemata eingeordnet und wie sind sie für das Beispielszenario zu bewerten?

Kapitel 2 hat sich mit den Grundlagen dieser Arbeit beschäftigt. Es wurden Grundbegriffe geklärt, bestehende Klassifikationsschemata vorgestellt und verschiedene Produktmodelle der virtuellen Produktentstehung in Stufen eingeordnet. Durch die Betrachtung der bestehenden Klassifikationsschemata wurde in diesem Kapitel die erste Fragestellung beantwortet und eine Einteilung der Datenaustauschstrategien nach verschiedenen Gesichtspunkten möglich. Weiterhin zeigte die Betrachtung der Klassifikationsschemata, dass der erfolgreiche Einsatz verschiedener Lösungsstrategien vom konkreten Anwendungsszenario abhängt. Nur im Zusammenhang mit diesem Szenario ist eine Bewertung der Strategien möglich. Das bedeutet auch, dass für

unterschiedliche Anwendungsszenarios verschiedene Vergleiche von Datenaustauschstrategien durchzuführen sind.

Das nächste Kapitel hat dann den Datenaustausch bei der Montagesimulation als ein Beispielszenario vorgestellt. Dabei wurden verschiedene Variationen des Szenarios identifiziert und Anforderungen an den Datenaustausch behandelt. Diese Anforderungen konnten im Zuge dessen in verschiedene Klassen eingeteilt werden. Danach beschäftigte sich ein Abschnitt mit den möglichen Klassen von Lösungsstrategien nach den in Kapitel 2 vorgestellten Klassifikationsschemata. Festgestellt wurde dabei, dass bei bestimmten Klassen von Strategien spezifische Einschränkungen und Voraussetzung zu beachten sind. Insgesamt beantwortet Kapitel 3 die zweite Fragestellung dieser Arbeit.

Mit der dritten Fragestellung nach der Einordnung und Beurteilung ausgewählter Datenaustauschstrategien beschäftigte sich Kapitel 4. Vorgestellt und verglichen wurden die Datenaustauschstrategien STEP, JT, die Online-Kopplung mit einem CAx-Objektbus und die Online-Kopplung mit OpenGL. Festgestellt wurde in diesem Kapitel, dass es gewisse Unterschiede und Gemeinsamkeiten zwischen den Strategien gibt. Jede Strategie hat ihre spezifischen Vor- und Nachteile, die sich auf die Eignung für den Datenaustausch bei der Montagesimulation auswirken. Dabei wurden Unterschiede bei der Eignung für die verschiedenen Variationen des Szenarios festgestellt. Zudem wurde gezeigt, dass die Voraussetzungen für den Einsatz der Strategien allgemein unterschiedlich sind. So sind STEP, JT und die Online-Kopplung mit einem CAx-Objektbus abhängig von der Kompatibilität der Datenstruktur der Systeme zur Datenstruktur der Datenaustauschstrategie. Dies schränkt die Systeme, die gekoppelt werden können, ein. Diese Einschränkung gilt bei der Online-Kopplung mit OpenGL nicht. Hier ist die Voraussetzung, dass die Systeme über OpenGL mit der Grafikeinheit kommunizieren. Damit gibt es hier keine Beschränkung bezüglich der Datenstruktur der Systeme. Dafür sind die Systeme auf OpenGL-verwendende Systeme beschränkt. Vorteilhaft ist dabei die große Verbreitung von OpenGL. Nachteil dieser Kopplungslösung im Vergleich zu den anderen Lösungen ist die Beschränkung auf Geometriemodelle, die durch OpenGL beschrieben werden können. So hat jede der vorgestellten Datenaustauschstrategien ihre Vor- und Nachteile.

Insgesamt beantwortet diese Arbeit alle gestellten Fragestellungen. Zu beachten ist jedoch, dass der vorliegende Vergleich theoretisch erfolgt ist. Eine praktische Untersuchung der vorgestellten Datenaustauschstrategien für das Beispielszenario kann ein noch detaillierteres Bild zur Eignung darstellen. Auch sind bestimmte Aspekte der vorgestellten Datenaustauschstrategien noch nicht ausreichend untersucht. Hier können weitere Arbeiten ansetzten. Weiterhin ist zu bedenken, dass dieser Vergleich eine Momentaufnahme darstellt, da einige Datenaustauschstrategien noch immer weiterentwickelt werden. Neue Versionen der Strategien müssen somit neu beurteilt werden. Zusammenfassend gesehen wird das Ziel eines Beispielvergleichs jedoch erfüllt. So können auf ähnliche Art und Weise weitere Datenaustauschstrategien und weitere konkrete Anwendungsszenarien betrachtet werden.

A. Anhang

Tabelle A.1: Übersicht über Klassifikationsschemata mit der Einordnung ausgewählter Lösungsstrategien. Für das Anwendungsszenario Datenaustausch bei der Montagesimulation sind die möglichen Klassen von Lösungsstrategien farbig hinterlegt. Grün hinterlegte Klassen sind allgemein möglich, gelb hinterlegte nur unter bestimmten Voraussetzungen.

			STEP	JT	CAx-Objektbus	OpenGL
Art der Daten	Produktions-technische Aspekte	Produktdaten	x	x	x	x[1]
		Prozessdaten[2]	(x)[3]	(x)[3]	(x)[3]	x[1]
		Auftragsdaten				x[1]
	Geometrie-gehalt	geometrische Daten	x	x	x	x
		Nicht-geometrische Daten[4]	x	x	x	(x)[5]
Systemneutralität	systemspezifisch[6]					
	systemneutral[7]		x	x	x	x

[1] nur darstellungsbezogene Daten

[2] gegebenenfalls Werkzeugmodelle, abhängig von Umfang der Simulation

[3] Daten zu Werkzeugen analog zu Produktdaten

[4] bei Übertragung von DMUs Produktstruktur und Kinematik; möglicherweise spezifische, notwendige Importdaten für das Simulationssystem, Fall C zum Beispiel möglicherweise Modellhistorie und Parametrik notwendig

[5] nur Geometrieattribute und Bewegungen in der Darstellung

[6] bei wenigen zu koppelnden Systemen vorteilhaft

[7] bei vielen zu koppelnden Systemen vorteilhaft

			STEP	JT	CAx-Objektbus	OpenGL
Konzept des Austausches	Übersetzen	Direktkonverter[6]				
		neutrales Format[7]	x	x		
		Kernbasiert[8]				
	Generieren[9]					
	Einbinden	Verknüpfen[10]			x	
		Einbetten[11]			x	x
Spezialisierung	Generalisiert		x	x		x
	Spezialisiert[12]				x	
Interoperabilitäts-level	Syntaktische[13]		x	x	x	x
	Semantisch[13]		x	x	x	x
	Pragmatisch				x	x[14]
	Dynamisch				x	x[14]
Realisierungs-merkmale	Zahl der Domänen	in-phasen Austausch[15]	x	x	x	x
		inter-phasen Austausch[16]	x	x	x	x
	Weitergabe von Änderungen	vollständig	x	x		
		inkrementell			x	x
	Vollständigkeit	vollständig	x	x		x[17]
		benötigte Daten		x	x	x[17]

[8] nur unter bestimmter Voraussetzung möglich; Übertagung von DMUs von konkreten Kernen abhängig

[9] nur unter bestimmten Voraussetzungen möglich

[10] Austausch der Ergebnisdaten vom Simulationssystem zurück zu den Erzeugersystemen automatisch realisiert

[11] einfache Realisierung des Austausch der Ergebnisdaten vom Simulationssystem zurück zu den Erzeugersystemen

[12] vorteilhaft, da auf konkrete anwendungsszenariospezifische Anforderungen eingegangen werden kann

[13] notwendig für Realisierung des Austausches zwischen heterogenen Systemen

[14] bezüglich Darstellung mit OpenGL möglich

[15] je nach beteiligten Systemen möglich, zum Beispiel bei bei Fall C alle Systeme sind Erzeugersysteme

[16] in Fall A und B Erzeuger- und Simulationssysteme

[17] abhängig von Realisierung

			STEP	JT	CAx-Objektbus	OpenGL
	Austausch-management	zentral[18]			x[17]	
		fortlaufend[18]			x[17]	
	Unterstützte Prozesstypen	linear	x	x	x	x
		concurrent		x	x	x
		simultanous		x	x	x
Datenaustausch-format	Anwendungsfeld von 3D-Geometrie-daten	CAD/CAM/CAE[19]	x	x	(x)	
		Modellierung und Animation[20]		x		
		Echtzeit/VR[20]				(x)
		Internet[21]				
	Grad der Verbindlichkeit	primär[22]		x	(x)	
		sekundär[23]		x	(x)	(x)
	Standardisierung	Standard	x	x		
		Industriestandard		x		
	Offenheit	offen[24]	x[25]	x		
		proprietär				
	Leichtgewichtigkeit			x		

[18] nur bei inkrementellen Austausch und Abhängigkeiten der Modelle relevant

[19] möglich, setzt jedoch exakte Geometrie voraus, für Fall C möglich

[20] möglich, setzt jedoch weitere Daten zur Szene voraus, für Fall C möglicherweise nicht geeignet

[21] möglich, ist jedoch für andere Anwendung entwickelt, daher Einschränkungen bei Montagesimulation zu erwarten

[22] wenn das Format als Datenaustauschformat verwendbar ist und die Montagesimulation auf verbindlichen Daten erfolgen soll

[23] wenn das primäre Format nicht für Austausch verwendbar ist oder die Montagesimulation informellen Charakter hat

[24] zu bevorzugen, da eigene Systeme und Konverter realisierbar

[25] kostenpflichtig

Literaturverzeichnis

[AB10] Anderl, Reiner ; Binde, Peter: *Simulation mit NX*. Hanser Verlag, 2010 (zitiert auf Seite 28, 29 und 30)

[AD00] Anderl, Reiner ; Daum, Bernd: Anhang. In: *STEP*. Teubner Verlag, 2000, S. 195–240 (zitiert auf Seite 48)

[AEK+00] Arlt, Martin ; Endres, Michael ; Katzenmaier, Jörg ; Philipp, Martin ; Pütter, Christian: Teil II STEP. In: *STEP*. Teubner Verlag, 2000, S. 39–126 (zitiert auf Seite 47, 48 und 51)

[AG05a] Alesi, Robert ; Gaigl, Hermann: First Time Right durch DMU. In: *CAD/CAM* (2005), Nr. 3, S. 47–49 (zitiert auf Seite 29)

[AG05b] Alesi, Robert ; Gaigl, Hermann: Neue Wege beim DMU. In: *Digital Engineering Magazin* (2005), Oktober, S. 22–24 (zitiert auf Seite 29)

[AG11] Anderl, Reiner ; Grabowski, Hans: Virtuelle Produktentstehung. In: *Dubbel*. Springer Verlag, 2011. – ISBN 9783540681915, Kapitel Y 3, S. Y15–Y29 (zitiert auf Seite 28 und 29)

[AJ00] Anderl, Reiner ; John, Harald: Teil I Produktdatentechnologie. In: *STEP*. Teubner Verlag, 2000, S. 9–36 (zitiert auf Seite 28, 29 und 30)

[AJSK98] Arnold, Florian ; Janocha, Andrzej T. ; Swienczek, Bernd ; Kilb, Thomas: Die CAx-Integrationsarchitektur ANICA und ihre erste Umsetzung in die Praxis. In: *Workshop Integration heterogener Softwaresysteme*, 1998, S. 43–54 (zitiert auf Seite 47, 67, 69 und 72)

[AMPSS04] Anderl, Reiner ; Melk, Katharina ; Pfeifer-Silberbach, Ullrich ; Schöfer, Frank: Digital-Mock-Up in der verteilten Produktentwicklung. In: *CAD-CAM Report* (2004), Nr. 6, S. 28–31 (zitiert auf Seite 32)

[And93] Anderl, Reiner: *CAD - Schnittstellen*. Hanser Verlag, 1993 (zitiert auf Seite 4, 5, 8, 13, 15, 17, 37, 41, 48 und 62)

[AT00] Anderl, Reiner (Hrsg.) ; Trippner, Dietmar (Hrsg.): *STEP*. Teubner, 2000 (zitiert auf Seite 48 und 51)

[Avg97] Avgoustinov, Nikolay: *Minimizing the Labour for Exchange of Product Definition Data Among N CAx-Systems*, Universität des Saarlandes, Diss., 1997 (zitiert auf Seite 11, 12, 20, 37 und 51)

[BBD+03] Barnert, Silvia ; Boeckh, Martin ; Delbrück, Matthias ; Greulich, Walter ; Heinische, Carsten ; Karcher, Ruht ; Lienhrat, Klaus ; Radons, Gunnar ; Voets, Stephan ; Wallenstein, Klaus: *Fachlexikon Computer*. Brockhaus Verlag, 2003 (zitiert auf Seite 3, 8 und 23)

[BDP07] Ball, Alexander ; Ding, Lian ; Patel, Manjula: Lightweight formats for product model data exchange and preservation. In: *Proceedings of PV 2007: Ensuring the Long-Term Preservation and Value Adding to Scientific and Technical Data*, 2007 (zitiert auf Seite 23, 38, 39 und 61)

[BDP08] Ball, Alexander ; Ding, Lian ; Patel, Manjula: An approach to accessing product data across system and software revisions. In: *Advanced Engineering Informatics* 22 (2008), Nr. 2, S. 222–235 (zitiert auf Seite 4, 23 und 34)

[Beh03] Behnisch, Susanne: *Digital Mockup mit CATIA V5*. Hanser Verlag, 2003 (zitiert auf Seite 32 und 35)

[BFS10] Beckers, Raphael ; Fröhlich, Arnulf ; Stjepandic, Josip: Anwendung und Potenziale universeller Visualisierungsformate. In: *9. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung*, 2010 (zitiert auf Seite 61 und 66)

[BKK+04] Brecheisen, Stefan ; Kriegel, Hans-Peter ; Kunath, Peter ; Pfeifle, Martin ; Renz, Matthias: Der virtuelle Prototyp: Datenbankunterstuetzung fuer CAD-Anwendungen. In: *Datenbank-Spektrum* 10 (2004), Nr. 9, S. 22–29 (zitiert auf Seite 33)

[BM99] Brunnermeier, Smita B. ; Martin, Sheila A.: Interoperability Cost Analysis of the U.S. Automotive Supply Chain / National Institute of Standards and Technology (NIST) study. 1999. – Forschungsbericht (zitiert auf Seite 1)

[Bro06] Brockhaus (Hrsg.): *Brockhaus Enzyklopädie Band 13*. Brockhaus Verlag, 2006 (zitiert auf Seite 1)

[CKMH12] Cheon, Sang-Uk ; Kim, Byungchul ; Mun, Duhwan ; Han, Soonhung: A procedural method to exchange editable 3D data from a free-hand 2D sketch modeling system into 3D mechanical CAD systems. In: *Computer-Aided Design* 44 (2012), Nr. 2, S. 123–131 (zitiert auf Seite 27 und 38)

[Cla97] Claussen, Ute: *Programmieren mit OpenGL*. Springer Verlag, 1997 (zitiert auf Seite 76)

[Dei10] Deisinger, Joachim: JT optimiert Zusammenarbeit und Transparenz. In: *interface* (2010), 3, Nr. 13, S. 12–13 (zitiert auf Seite 58, 59, 63, 64 und 65)

[DFB11] Drath, Rainer ; Fay, Alexander ; Barth, Mike: Interoperabilität von Engineering-Werkzeugen. In: *Automatisierungstechnik* 59 (2011), Nr. 7, S. 451–460 (zitiert auf Seite 3, 6, 21, 23 und 63)

[DIN06] ; DIN (Veranst.): *Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik (IEC 60050-351:2006)*. 2006 (zitiert auf Seite 4)

[DR96] Dai, Fan ; Reindl, Peter: Enabling Digital Mock-Up with Virtual Reality Techniques - Vision, Concept, Demonstrator. In: *Proceedings of The 1996 ASME Design Engineering Technical Conferences and Computer in Engineering Conferrences*, 1996 (zitiert auf Seite 29 und 32)

[Dyl02] Dyla, Andreas: *Modell einer durchgängig rechnerbasierten Produktentwicklung*, Technische Universität München, Diss., 2002 (zitiert auf Seite 34 und 37)

[Eig09] Eigner, Martin: IT-Lösungen für den Produktentwicklungsprozess. In: *Handbuch Unternehmensorganisation: Strategien, Planung, Umsetzung*. Springer, 2009, Kapitel 4.3, S. 247–260 (zitiert auf Seite 26)

[ES09] Eigner, Martin ; Stelzer, Ralph: *Product Lifecycle Management*. Springer Verlag, 2009 (zitiert auf Seite 26, 28, 29 und 33)

[Fac05] Fachgebiet Datenverarbeitung in der Konstruktion (DiK), Technische Universität Darmstadt: Visualisierungsformate und Strukturdatenaustausch / ProSTEP iViP e.V. 2005. – Forschungsbericht (zitiert auf Seite 10, 23, 24, 38, 58, 60, 62, 64 und 65)

[FLL+11] Friedewald, Axel ; Lödding, Hermann ; Lukas, Uwe Freiherr v. ; Mesing, Benjamin ; Roth, Matthias ; Schleusener, Sebastian ; Titov, Fedor: Benchmark neutraler Formate für den prozessübergreifenden Datenaustausch im Schiffbau / Frauenhofer IDG. 2011. – Forschungsbericht (zitiert auf Seite 10, 13, 23, 24, 37, 50, 51, 52, 54, 55, 56, 57, 60, 61, 63, 65 und 66)

[GA90] Grabowski, Hans ; Anderl, Reiner: *Produktdatenaustausch und CAD-Normteile*. expert Verlag, 1990 (zitiert auf Seite 4, 5, 7, 8, 9, 10, 12, 14, 15 und 48)

[GAG86] Grabowski, H. ; Anderl, R. ; Glatz, R.: CAD/CAM-Schnittstellenproblematik für den Anwender. In: *wt - Zeitschrift für industrielle Fertigung* (1986), S. 212–213 (zitiert auf Seite 4)

[GB08] Grote, Karl-Heinz ; Beyer, Christiane: Generative Fertigungsverfahren. In: *Einführung in die Fertigungslehre*. Shaker Verlag, 2008, Kapitel 8, S. 303–326 (zitiert auf Seite 8)

[GDSKM97] Gomes De Sà, A. ; Kress, H. ; Müller, St.: Digital Mock-up in der Einbau- und Montagesimulation. In: *Neue Generation von CAD/CAM-Systemen*, 1997 (zitiert auf Seite 29, 32 und 33)

[Ger03] Gerbino, Salvatore: Tools for the interoperability among CAD systems. In: *Proceedings of the XIII Adm-XV Ingegraf International Conference on Tools and Methods Evolution in Engineering Design, Cassino, Italia, June 2003*, 2003 (zitiert auf Seite 13, 14 und 38)

[Ger10] Gerhardt, Florian: *Supporting Virtual Produkct Engineering Processes by Integrating a Neutral, Lightweight and CAD- Derived Data Format*, Technische Universität Kaiserslautern, Diss., 2010 (zitiert auf Seite 25, 32, 33, 52, 55, 58, 59, 60 und 61)

[Haa95] Haasis, Siegmar: *Integrierte CAD-Anwendungen*. Springer Verlag, 1995 (zitiert auf Seite 12, 13, 14, 15, 32 und 33)

[Han11] Handschuh, Sebastian: *Wertextrahierende Nutzung von offenen leichtgewichtigen Datenformaten in automobilen Kollaborations- und Entwicklungsprozessketten*, Technische Universität Kaiserslautern, Diss., 2011 (zitiert auf Seite 18, 23, 24, 25, 48, 50, 51, 52, 54, 55, 61 und 64)

[HHN+02] Humphreys, Greg ; Houston, Mike ; Ng, Ren ; Frank, Randall ; Ahern, Sean ; Kirchner, Peter D. ; Klosowski, James T.: Chromium: a stream-processing framework for interactive rendering on clusters. In: *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2002, San Antonio, Texas, USA, July 23-26, 2002*, 2002, S. 693–702 (zitiert auf Seite 77)

[HL08] Hartman, Nathan W. ; Lim, Adrian: Examining neutral formats for visualization and data exchange. In: *Proceedings of the 2008 IAJC-IJME International Conference: Nashville, TN*, 2008 (zitiert auf Seite 23 und 61)

[Hol02] Holle, Wolfgang: *Rechnerunterstützte Montageplanung*. Hanser Verlag, 2002 (zitiert auf Seite 32)

[IRK05] Ilmonen, Tommi ; Reunanen, Markku ; Kontio, Petteri: Broadcast GL: An Alternative Method for Distributing OpenGL API Calls to Multiple Rendering Slaves. In: *WSCG (Journal Papers)*, 2005, S. 65–72 (zitiert auf Seite 77)

[ISD11] ISD Software und Systeme GmbH: *Innovationen nutzen - Ideen verwirklichen Lösungen für das Engineering*. HiCAD Broschüre, 2011 (zitiert auf Seite 35)

[ISOa] *ISO 10303-203:2011 - Industrial automation systems and integration – Product data representation and exchange – Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies*. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=44305. – Stand 02.10.2012 13:00 Uhr (zitiert auf Seite 51)

[ISOb] *ISO 10303-214:2003 - Industrial automation systems and integration – Product data representation and exchange – Part 214: Application protocol: Core data for automotive mechanical design processes.* http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=38727. – Stand 26.9.2012 15:00 Uhr (zitiert auf Seite 51)

[ISOc] *ISO 10303-214:2010 - Industrial automation systems and integration – Product data representation and exchange – Part 214: Application protocol: Core data for automotive mechanical design processes.* http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=43669. – Stand 26.9.2012 15:00 Uhr (zitiert auf Seite 51)

[ISOd] *ISO/CD 10303-242 - Industrial automation systems and integration – Product data representation and exchange – Part 242: Application protocol: Managed Model-based 3D Engineering.* http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=57620. – Stand 26.9.2012 15:00 Uhr (zitiert auf Seite 51)

[ISOe] *ISO/FDIS 14306 - Industrial automation systems and integration – JT file format specification for 3D visualization.* http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=60572. – Stand 02.10.2012 15:00 Uhr (zitiert auf Seite 58)

[ISOf] *ISO/PAS 14306:2011 - Industrial automation systems and integration – JT file format specification for 3D visualization.* http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54606. – Stand 02.10.2012 15:00 Uhr (zitiert auf Seite 58 und 64)

[Jäg91] Jäger, Karl W.: Rechnerintegrierte Produktion. In: *Schnittstellen bei CAD/CAE-Systemen.* VDI Verlag, 1991, Kapitel 1, S. 1–35 (zitiert auf Seite 13 und 15)

[Kal06] Kalusche, Marcus: *Methoden, Datenmodell und Schnittstellengestaltung in der Fabrikplanung für ein integriertes Planungssystem im Einsatz bei kleinen und mittleren Unternehmen (KMU)*, Technische Universität Dresden, Diss., 2006 (zitiert auf Seite 8, 9, 47, 48, 50 und 51)

[KAW+07] Krause, Frank-Lothar ; Anderl, Reiner ; Weber, Christian ; Bierwerth, Marc ; Rothenburg, Uwe ; Wanke, Sören ; Wöhler, Thomas: CAD-DMU-FMU. In: *Innovationspotenziale in der Produktentwicklung.* Hanser Verlag, 2007, Kapitel 7, S. 117–128 (zitiert auf Seite 28, 29 und 33)

[KB97] Krause, Frank-Lothar ; Baumann, Richard: Austausch parametrischer Informationen auf der Basis impliziter Produktbeschreibungen. In: *Features verbessern die Produktentwicklung - Integration von Prozessketten*, 1997 (zitiert auf Seite 14 und 15)

[KKM97] Katzenbach, Alfred ; Kreutz, Matthias ; Müller, Frank: *Digital Mock-up in der PKW-Entwicklung*. 3. CAD/CAM-Forum, 1997 (zitiert auf Seite 29, 31 und 33)

[Kra97] Krause, Frank-Lothar: Auf dem Weg zur virtuellen Produktentwicklung. In: *Neue Generation von CAD/CAM-Systemen*, 1997 (zitiert auf Seite 29 und 30)

[KVGS11] Köppen, Veit ; Vornholt, Stephan ; Geist, Ingolf ; Saake, Gunter: Ganzheitliche Unterstützung für die simultane und virtuelle Produktentwicklung. In: u.a., Roland K. (Hrsg.) ; Otto-von-Guericke-Universität Magdeburg (Veranst.): *10. Magdeburger Maschinenbau-Tage 27.-29.09.2011*. Magdeburg, September 2011 (zitiert auf Seite 5, 23, 26, 37 und 38)

[Li10] Li, Yuexiao: *Exchange and Integration Solutions for Heterogeneous Data in Concurrent Virtual Engineering*, Otto-von-Guericke Universität Magdeburg, Diplomarbeit, 2010 (zitiert auf Seite 20, 21, 22, 54 und 72)

[LLS06] Laudon, Kenneth C. ; Laudon, Jane P. ; Schoder, Detlef: *Wirtschaftsinformatik*. Pearson Verlag, 2006 (zitiert auf Seite 3)

[Mac91] Mache, Holm-Rainer: Stand der Schnittstellen-Normierung für den Produktdatenaustausch im Bereich Maschinenbau. In: *Schnittstellen bei CAD/CAE-Systemen*. VDI Verlag, 1991, Kapitel 2, S. 36–53 (zitiert auf Seite 9 und 13)

[Mer09] Mertens, Peter: *Integrierte Informationsverarbeitung 1*. Gabler Verlag, 2009 (zitiert auf Seite 5 und 7)

[MMMK12] Mory, Maik ; Masik, Steffen ; Müller, Richard ; Köppen, Veit: Exposing Proprietary Virtual Reality Software to Nontraditional Displays. In: *Proceedings of the 20th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Union Agency, 2012 (WSCG Communication Proceedings), S. 35–43 (zitiert auf Seite 77)

[MPKS11] Mory, Maik ; Pukall, Mario ; Köppen, Veit ; Saake, Gunter: Evaluation of Techniques for the Instrumentation and Extension of Proprietary OpenGL Applications. In: *2nd International ACM/GI Workshop on Digital Engineering (IWDE)*. Magdeburg, Germany, 2011, S. 50–57 (zitiert auf Seite 1, 9, 10 und 11)

[MWB09] Manso, Miguel-Ángel ; Wachowicz, Monica ; Bernabé, Miguel Á.: Towards an Integrated Model of Interoperability for Spatial Data Infrastructures. In: *Transaktion in GIS* 13 (2009), Nr. 1, S. 43–67 (zitiert auf Seite 18, 19 und 42)

[Nat99] National Center for Manufacturing Sciences Inc.: *Open CAD Architecture for Interoperability - Phase 1 Project Summary Report*. NCMS Project No. 96075, 03 1999 (zitiert auf Seite 15)

[NJG09] Nambiar, Arun N. ; Judd, Robert P. ; Greenblatt, Judah: Data exchange among different modules in a manufacturing or service framework. In: *International Journal of Product Development* 8 (2009), Nr. 2, S. 122–140 (zitiert auf Seite 38)

[Pra01] Pratt, Michael J.: Introduction to ISO 10303 - the STEP Standard for Product Data Exchange. In: *Journal of Computing and Information Science in Engineering* 1 (2001), Nr. 1, S. 102–103 (zitiert auf Seite 47, 48 und 51)

[Pro03] ProSTEP iViP e.V.: *8th STEP Processor Benchmark short report*. 2003 (zitiert auf Seite 51, 52, 56 und 57)

[Pro10] ProSTEP iViP e.V.: *ProSTEP iViP-VDA JT-Translator-Benchmark Ed2 Short-Report*. April 2010 (zitiert auf Seite 60 und 66)

[Ren86] Renz, W.: VDAFS a pragmatic interface for the exchange of sculptured surface data. In: *Proc. of a seminar of the ZGDV on Product data interfaces in CAD/CAM applications: design, implementation and experiences*. New York, NY, USA : Springer-Verlag, 1986. – ISBN 0–387–15118–4, S. 144–149 (zitiert auf Seite 37 und 39)

[SAKJ98] Swienczek, Bernd ; Arnold, Florian ; Kilb, Thomas ; Janocha, Andrzej: Online-Kopplung von CAx-Systemen für die virtuelle Produktentwicklung - Ein Vergleich mit dem dateibasierten Datenaustausch. In: *Informationsverarbeitung in der Konstruktion* (1998), S. 219–238 (zitiert auf Seite 33, 35, 47, 67, 69, 70, 71 und 74)

[SAKJ00] Swienczek, Bernd ; Arnold, Florian ; Kilb, Thomas ; Janocha, Andrzej: *Maßgeschneiderte CAx-Systeme auf Basis von Komponenten*. 2000 (zitiert auf Seite 47, 67 und 69)

[SBD06] Scholz, Eckhard ; Burkhardt, Christian ; Dietrich, Sascha: *Digital Mock-Up in der Produktentwicklung*. Podium HTWK - Leipzig, 2006 (zitiert auf Seite 28, 29, 31, 32, 33, 34, 35 und 38)

[Sch01] Schumann, Sören: *Methoden zur Optimierung des Austausches von Geometrie- und Parametrikdaten*, Otto-von-Guericke Universität Magdeburg, Diss., 2001 (zitiert auf Seite 3, 4, 6, 7, 9, 11, 12, 13, 14, 15, 16, 37, 53, 69, 70 und 71)

[Sen95] Sendler, Ulrich: *CAD & Office Integration*. Springer Verlag, 1995 (zitiert auf Seite 16)

[Shr10] Shreiner, Dave: *OpenGL Programming Guide*. Addison Wesley, 2010 (zitiert auf Seite 76)

[SIE] SIEMENS (Hrsg.): *JT File Format Reference Version 9.5 Rev-D*. SIEMENS (zitiert auf Seite 58, 59, 60, 61, 63, 64 und 65)

[SK97] Spur, Günter ; Krause, Frank-Lothar: *Das virtuelle Produkt*. Hanser Verlag, 1997 (zitiert auf Seite 5, 7, 8, 9, 26, 30, 32, 33 und 56)

[SLLT06] Stavrakakis, John ; Lau, Zhen-Jock ; Lowe, Nick ; Takatsuka, Masahiro: Exposing application graphics to a dynamic heterogeneous network. In: *Proceedings of the 14-th International Conferences in Central Europe on Computer Graphics* (2006), S. 71–78 (zitiert auf Seite 76 und 77)

[Ste07] Stekolschik, Alexander: *Ein Beitrag zum ganzheitlichen Qualitätsmanagement von CAD-Modellen in der Produktentstehung*, Ruhr-Universität Bochum, Diss., 2007 (zitiert auf Seite 6, 27, 28, 33, 61 und 63)

[Sto11] Stoye, Michael: *Entwicklung eines Datenmodells zur Unterstützung des dateibasierten Datenaustauschs in der Produktentwicklung*, Otto-von-Guericke Universität Magdeburg, Diplomarbeit, 2011 (zitiert auf Seite 9, 10, 11, 12, 13, 26, 29, 37, 51, 52, 61 und 66)

[SVG11] Stoye, Michael ; Vornholt, Stephan ; Geist, Ingolf: Management of Flexible Data Exchange Processes in Virtual Product Development. In: *2nd International ACM/GI Workshop on Digital Engineering (IWDE) 2011 Magdeburg, Germany*, 2011 (zitiert auf Seite 10, 25, 47, 58, 60, 61 und 64)

[Ten02] Tenbusch, Alexander: *Rechnergestützte geometriedaten-bestimmte Prozessketten zur Produktentwicklung*, Otto-von-Guericke Universität Magdeburg, Diss., 2002 (zitiert auf Seite 8, 27, 28 und 37)

[Thi04] Thierfelder, Kolja: *Dateiformate für dreidimensionale Daten*. Universität Münster Seminarausarbeitung, 2004 (zitiert auf Seite 23, 43, 55, 64 und 81)

[VDA02] VDA: *VDA 4956 Empfehlung Produktdatenaustausch*. 2002 (zitiert auf Seite 3 und 28)

[VDI93] ; VDI (Veranst.): *VDI-Richtlinie 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. 1993 (zitiert auf Seite 25)

[VDI96] ; VDI (Veranst.): *VDI-Richtlinie 3633 - Simulation von Logistik-, Materialfluß- und Produktionssystemen - Begriffsdefinitionen*. 1996 (zitiert auf Seite 31)

[VGL10] Vornholt, Stephan ; Geist, Ingolf ; Li, Yuexiao: Categorisation of data management solutions for heterogeneous data in collaborative virtual engineering. In: *First International Workshop on Digital Engineering (IWDE) 2010 Magdeburg*, 2010, S. 9–16 (zitiert auf Seite 6, 10, 13 und 20)

[Vin04] Vince, John: *Introduction to Virtual Reality*. Springer Verlag, 2004 (zitiert auf Seite 32)

[VK10] Vornholt, Stephan ; Köppen, Veit: Data-driven and Integrated Engineering for Virtual Prototypes. In: *The 3rd International Multi-Conference on Engineering and Technological Innovation: IMETI 2010*, 2010, S. 164–168 (zitiert auf Seite 29)

[VSG+11] Vornholt, Stephan ; Stoye, Michael ; Geist, Ingolf ; Köppen, Veit ; Saake, Gunter: Datenmodell zur flexiblen Verwaltung von Datenaustauschprozessen in der virtuellen Produktentwicklung. In: u.a., Roland K. (Hrsg.) ; Otto-von-Guericke-Universität Magdeburg (Veranst.): *10. Magdeburger Maschinenbau-Tage 27.-29.09.2011*. Magdeburg, September 2011 (zitiert auf Seite 8, 10, 11, 23, 24, 25, 38, 47, 58, 60, 61 und 64)

[VWB+09] Vajna, Sandor ; Weber, Christian ; Bley, Helmut ; Zeman, Klaus ; Hehenberger, Peter: *CAx für Ingenieure*. Bd. 2. Springer Verlag, 2009 (zitiert auf Seite 1, 2, 3, 4, 5, 7, 8, 10, 11, 12, 13, 14, 27, 28, 29, 30, 31, 32, 33, 37, 48 und 50)

[Wan02] Wang, G. G.: Definition and Review of Virtual Prototyping. In: *Journal of Computing and Information Science in Engineering* 2 (2002), Nr. 3, S. 232–236 (zitiert auf Seite 29)

[War09] Warschat, Joachim: Virtual Engineering. In: *Handbuch Unternehmensorganisation: Strategien, Planung, Umsetzung*. Springer, 2009, Kapitel 8.2, S. 530–544 (zitiert auf Seite 26, 27 und 28)

[WBTM00] Whyte, J. ; Bouchlaghem, N. ; Thorpe, A. ; McCaffer, R.: From CAD to virtual reality: modelling approaches, data exchange and interactive 3D building design tools. In: *Automation in Construction* 10 (2000), Nr. 1, S. 43–55 (zitiert auf Seite 13, 23, 24 und 76)

[Wik] *Szenengraph - Wikipedia*. http://de.wikipedia.org/wiki/Szenengraph. – Stand 09.10.2012 10:55 Uhr (zitiert auf Seite 59)

[WRSG11] Wack, Karl-Josef ; Riegmann, Tobias ; Strassburger, Steffen ; Guenther, Ulrich: Virtuelle Produktabsicherung am Beispiel Montage Powertrain. In: *Digitales Engineering und virtuelle Techniken zum Planen, Testen und Betreiben technischer Systeme*, 2011 (zitiert auf Seite 32 und 35)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Magdeburg, den 17. Oktober 2012