

# Datenmodell zur flexiblen Verwaltung von Datenaustauschprozessen in der virtuellen Produktentwicklung

Dipl.-Inf. Stephan Vornholt <sup>1)</sup>, Dipl.-Ing.-Inf. Michael Stoye <sup>2)</sup>, Dipl.-Inf. Ingolf Geist <sup>3)</sup>, Dr. rer. pol. Veit Köppen <sup>4)</sup>, Prof. Dr. rer. nat. habil. Gunter Saake <sup>5)</sup>

<sup>1)</sup> Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, Deutschland, vornholt@iti.cs.uni-magdeburg.de,  
+49 (0) 391 67-12869

<sup>2)</sup> Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, Deutschland, stoye@ovgu.de

<sup>3)</sup> Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, Deutschland, ingolf.geist@ovgu.de

<sup>4)</sup> Center for Digital Engineering, Otto-von-Guericke-Universität Magdeburg, Deutschland, vkoeppen@ovgu.de

<sup>5)</sup> Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, Deutschland, saake@iti.cs.uni-magdeburg.de

## Zusammenfassung

Bei der digitalen Produktentwicklung finden die Produktentwicklungsphasen (Entwurf und Simulation) auf der Basis computergenerierter Produktmodelle statt. Die virtuelle Produktentwicklung ist ein kollaborativer Prozess, der heterogene Daten aus verschiedenen Anwendungsbereichen nutzt, welche ausgetauscht, transformiert und verwaltet werden müssen. Dabei müssen die Arbeitsabläufe und Zuständigkeiten ebenso wie die entstehenden Daten gesteuert werden, um den Entwicklungsprozess eines Produktes nachvollziehen zu können.

Workflow-Management-Systeme unterstützen Prozessabläufe durch die Definition fester Abläufe. Diese Werkzeuge und Austauschformate können sich während der Produktentwicklung durch Erweiterungen oder neue Versionen von Werkzeugen verändern. Zur Verwaltung dieser Variabilität wird ein Metadatenmodell vorgeschlagen, um die benötigte Nachvollziehbarkeit von Ergebnissen bei der flexiblen Produktentwicklung zu unterstützen. In diesem Modell können komplexe, sich ändernde Abläufe in der Produktentwicklung abgebildet werden. Insbesondere für die Nachvollziehbarkeit ist ein solcher Ansatz sinnvoll, damit in flexiblen Umgebungen die „richtigen“ Austauschformate genutzt und später wieder verwendet werden.

Ein Metadatenmodell beschreibt semantische Informationen zu den Daten. Zur Erstellung eines geeigneten Datenmodells ist detailliertes Wissen über Möglichkeiten des Austauschvorgangs, eingesetzte Systeme und Formate sowie die Prozesse notwendig. In diesem Beitrag werden der Datenaustausch und häufig genutzte Formate für einzelne Prozessmodelle analysiert und darauf aufbauend in zwei Teilschritten das Datenmodell über Systemlandschaften und Prozessverwaltung in der Produktentwicklung vorgestellt. Die Evaluierung des Metadatenmodells für den Datenaustausch erfolgt anhand einer prototypischen Implementierung.

Schlüsselwörter: PDM, virtuelle Produktentwicklung, Datenverwaltung, Workflow-Management

## 1. Einleitung

Während der Entwicklung neuer Produkte stehen eine hohe Qualität bei gleichzeitig geringen Entwicklungskosten und oftmals kurzen Entwicklungszeiten im Vordergrund. Um diese Anforderungen zu befriedigen, setzen insbesondere die Vorreiter auf diesem Gebiet (Fahrzeug-, Luft- und Raumfahrttechnik) auf computergestützte Simulations- und Visualisierungstechniken. Bei der Entwicklung von automatisierten Systemen, im Bereich der mechatronischen Konstruktion und Simulation, werden computergestützte virtuelle Modellierungs- und Testverfahren genutzt. Diese virtuelle Produktentwicklung ist durch die Heterogenität der beteiligten Domänen, u.a. Konstruktion, Mechatronik, FEM-Simulation und Mehrkörpersimulation, mit eigenen Spezialisten und Werkzeugen gekennzeichnet. Zwischen den Domänenexperten muss während des gesamten Entwicklungsprozesses ein Informationsaustausch durch Transformation der Daten stattfinden. So entstehen innerhalb und zwischen Unternehmen komplexe CAX-Systemlandschaften, die aus Werkzeugen verschiedener Hersteller bestehen und unterschiedliche Schnittstellen anbieten. Ein Beispiel dafür sind 58 Schnittstellen der „Dr. Ing. h.c. F. Porsche AG“, um 28 CAX-Systeme für Simulationszwecke miteinander zu verbinden [1].

Für die Unterstützung der Anwender beim Produktentwicklungsprozess stehen Versionsverwaltung, standardisierte Prozessabläufe und Workflow-Management-Systeme zur Verfügung. Diese sind bspw. bereits in Produktdatenmanagement (PDM)-Systemen integriert. Dennoch treten bei einem intensiven Datenaustausch in der Produktentwicklung zwei grundsätzliche Probleme auf. Zum Einen muss eine komplexe Systemlandschaft abgebildet werden, welche sich während des Entwicklungsprozesses ändern kann. Zum Anderen ist die

Nachvollziehbarkeit von durchgeführten Datenaustauschprozessen über den gesamten Verlauf zu gewährleisten [2]. Metadatenmodelle ermöglichen die Speicherung und Verwaltung von Informationen zum Prozessverlauf. Zur Unterstützung von flexiblen Entwicklungsprozessen wird daher die Speicherung und Verwaltung der Daten mit Hilfe eines Metadaten-Repository in diesem Paper behandelt. Dieses Metadaten-Repository muss Eigenschaften sich verändernder Systeme und Austauschformate abbilden können. Um das Datenmodell zu entwickeln, müssen zuerst prozessspezifische Eigenschaften analysiert werden. Der Fokus dieser Arbeit liegt dabei auf den Werkzeugen und Transformationen.

Zuerst werden Datenaustauschprozesse, die Problematik mit dem Datentransfer und typische Dateiformate vorgestellt (Abschnitt 2). In einem Beispielprozess zur Mehrkörpersimulation (Abschnitt 2.4) werden der Ablauf und die Problematik erläutert. Abschnitt 3 umreißt den aktuellen Stand der Prozessunterstützung in der virtuellen Produktentwicklung. In Abschnitt 4 wird ein geeignetes Metadatenmodell vorgestellt, welches die in den Produktmodellierungsschritten entstehenden Daten verwalten kann. Das Metadatenmodell wird in zwei separaten Teilen: „Systemverwaltung“ (Abschnitt 4.1) und „Prozessverwaltung“ (Abschnitt 4.2) erläutert und in der Implementierung (Abschnitt 5) in einem Datenmodell zusammengeführt. Diese Arbeit schließt mit einer Zusammenfassung der erzielten Ergebnisse und einem Ausblick über zukünftige Nutzungsmöglichkeiten des Metadatenmodells.

## 2. Analyse der Datenaustauschprozesse

Die Simulation und Analyse eines virtuellen Produktmodells erfordert einen Datenaustausch zwischen einer Vielzahl an CAx-Systemen [3]. Anfänglich wurden CAx-Systeme isoliert voneinander eingesetzt und bildeten monolithische Systeme mit jeweils eigenen Datenstrukturen. Um dennoch Daten digital auszutauschen, wurden integrative CAx-Systeme auf der Grundlage eines integrierten Produktmodells entwickelt (siehe z.B.: [4]). Das Produktmodell bildet dabei eine einheitliche Datenbasis für heterogene Teilmodelle. Dieser Ansatz hat sich aber aufgrund der mangelnden Akzeptanz der CAx-Systemhersteller bisher nicht durchsetzen können [5]. Dennoch ist dieser Ansatz weiterhin Basis verschiedener Forschungsansätze [6, 7], welche sich auf spezielle Prozesse oder leichtgewichtige Ansätze konzentrieren. Die weitergehende Vernetzung der Entwicklungsdomänen und steigende Komplexität der Produkte macht eine solche Lösung notwendig.

Viele Systemhersteller nutzen die entwickelten Techniken integrierter Datenmodelle, um darauf aufbauend Produkte zu entwickeln, welche Funktionen für die Bearbeitung verschiedener Fachbereiche in einer integrierten Umgebung anbieten (bspw. Pro/ENGINEER). Hier ist jedoch häufig der Funktionsumfang der integrierten Module eingeschränkt und die Oberfläche nicht an die Tätigkeit der Domänenexperten angepasst.

### 2.1 Datenaustausch

Ein Informationsaustausch setzt oft eine Transformation der Daten voraus. Dieser Datenaustausch kann unterschiedlich organisiert werden, konzentriert sich aber dennoch auf eine Form des dateibasierten Datenaustausches, unabhängig davon ob die Dateien gemeinsam in einem PDM-System, einer Workflow-Management-Umgebung oder in verteilten Datenbeständen organisiert werden. Die Informationen, die in einem Format gespeichert sind, werden als Abbildungs- oder Merkmalsraum bezeichnet. Es können daher nur Daten ineinander überführt werden, die sowohl im Abbildungsraum des Ausgangssystems als auch in dem Austauschformat als auch dem Abbildungsraum des Zielsystems abgebildet werden können (vgl. Abbildung 1). So ist es beispielsweise in einem CAx-System möglich Materialdaten abzubilden, in einem Austauschformat für 3D-Daten zur Visualisierung jedoch nicht.

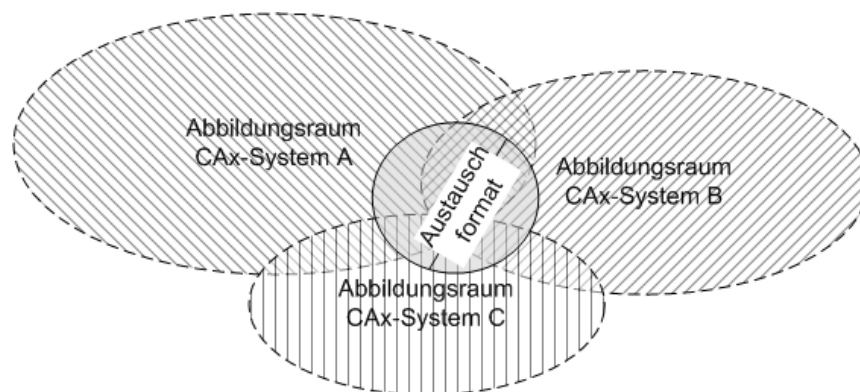


Abbildung 1: Abbildungsräume und Datenaustausch zwischen CAx-Systemen

Für die Realisierung dateibasierter Schnittstellen gibt es zwei unterschiedliche Ansätze [8, 9]:

- **Systemspezifische Schnittstellen** zeichnen sich durch eine Vielzahl an Schnittstellen aus, sind aber auf eine spezifische Übertragung zwischen zwei Systemen ausgelegt und bilden einen großen gemeinsamen Merkmalsraum.
- **Systemneutrale Schnittstellen** legen ein gemeinsames Übertragungsformat zwischen mehreren Systemen fest, wobei jedoch der Umfang des Übertragungsformats sehr groß wird und im Allgemeinen der gemeinsame Merkmalsraum zwischen zwei Systemen nicht komplett abgebildet werden kann. Als Beispiel kann das Gewicht eines Körpers dienen, das in zwei Werkzeugen definiert werden kann, in einem gemeinsamen Übertragungsformat für den Austausch der Gestalt von Modellen keine Berücksichtigung findet.

## 2.2 Dateiformate

Native Dateiformate sind system- beziehungsweise herstellerabhängig, d.h. dienen zur Speicherung aller erzeugbaren Daten eines oder mehrerer CAx-System eines Herstellers, und sind oft nicht frei verfügbar sowie erweiterbar. Für die Konvertierung kann ein spezielles Konvertierungsprogramm (z.B. PolyTrans<sup>1</sup>) genutzt werden. Systemneutrale Dateiformate werden nicht nur zum Datenaustausch, sondern auch direkt in Programmen genutzt, um Daten zu speichern. Im Gegensatz zu nativen Dateiformaten sind sie häufig genormt und legen systemneutrale Dateiformate für eine unabhängige, einheitliche Definition fest. Zum Vergleich der Dateiformate werden für die Entwicklungsdomänen der mechatronischen Simulation typische Standardformate untersucht. Diese sind gegenüber nativen Austauschformaten dokumentiert und frei verfügbar. Weitere Vergleiche von Dateiformaten finden sich z.B. in [8, 10, 11 und 12]:

- **Initial Graphics Exchange Specification (IGES)** ist eines der ersten entwickelten CAD Austauschformate, welches heute noch in vielen Systemen als Standard „ANSI Y 14.26M“ genutzt wird.
- **Standard for the Exchange of Product Model Data (STEP)** ist als Standard der Normreihe (ISO 10303) eingetragen, in der mehrere Standards zusammengefasst werden. Es werden nicht nur geometrische Daten unterstützt, sondern verschiedene Produktdaten, die im Lebenszyklus eines Produktes entstehen [13]. STEP beschreibt eine Menge an anwendungsspezifischen neutralen Dateiformaten und kann als Baukasten, mit dem verschiedene Produktdatenmodelle unter Verwendung von vorhandenen Grundbausteinen zusammengesetzt werden können, für die Definition weiterer neutraler Dateiformate verwendet werden.
- **Jupiter Tessellation (JT)** ist ebenfalls sehr verbreitet und stellt einen Industriestandard dar, der sich in der Entwicklung zu einem ISO-Standard befindet [14]. In einer JT-Datei können unterschiedliche Repräsentationen von 3D-Modellen zusammen abgelegt werden. Weiterhin gibt es Möglichkeiten Product-Manufacturing-Information (PMI) und visualisierungsspezifische Eigenschaften (z.B. Texturen) abzubilden.
- **Drawing Interchange Format (DXF)** ist ein proprietäres und kein genormtes neutrales Standardformat. Das DXF-Format ist übersichtlich und wird von vielen CAD-Systemen als Export- und Import-Format unterstützt.
- **Surface Tessellation Language (STL)** dient der Beschreibung von Oberflächen. Es ist nicht standardisiert, wird aber beim Austausch zwischen CAD- und CAM-Systemen (z.B.: Rapid Prototyping Anlagen) in der Praxis häufig eingesetzt.
- **Virtual Reality Modeling Language (VRML)** ist eine Beschreibungssprache von 3D-Szenen, deren Geometrien, Ausleuchtungen und Animationen. Es entwickelte sich vom ISO-Standard [15] zu einem XML basierten X3D-Format [16]. In der Produktentwicklung wird es hauptsächlich zum Austausch zwischen CAD- und VR-Systemen eingesetzt.

## 2.3 Vergleich und Eigenschaften

Zur Vergleichbarkeit der einzelnen Formate gehört neben Standardisierung auch der Beschreibungsumfang (Merkmalsraum), die Eignung (bspw. für Visualisierungszwecke), die Dateigröße, sowie Komplexität und Umfang der Dokumentation. Einen Überblick bietet Tabelle 1, in der grundlegende Eigenschaften der beschriebenen Dateiformate gezeigt werden.

---

<sup>1</sup> <http://www.okino.com/conv/conv.htm> (14.08.2011)

Eigenschaft	IGES	STEP	JT	DXF	STL	VRML
Norm (n) / Industriestandard (i)	n	n	i	i	i	n
ASCII (a) / Binär (b) / XML (x)	a, b	a, x	b	a	a, b	a
2D Modelle	✓	✓	✗	✓	✗	✗
3D Modelle (facettiert)	✓	✓	✓	✗	✓	✓
3D Modelle (parametrisch)	✓	✓	✓	✓	✗	✗
Nichtgeometrische CAD-Daten	✓	✓	✓	✓	✗	✓
✓ ist abbildbar, ✗ ist nicht abbildbar						

**Tabelle 1: Vergleich neutraler Datenaustauschformate**

Weiterhin unterscheiden sich die Möglichkeiten zur Datensicherung beim Import und Export in einzelnen Dateiformaten. Oft stehen ähnliche oder ältere Versionen von Dateiformaten für den Austausch zur Verfügung. Beim **Export** sind zum Beispiel verschiedene *Modelltypen* und *Teilelemente* wählbar (bspw. Drahtmodell, Flächenmodell oder Bezugskurven). Weiterhin sind Einstellungen zur *Qualität*, *Konfigurationssicherung*, und *Anzahl der benötigten Dateien* für einige Dateiformate möglich oder notwendig.

Der **Import** unterscheidet zum Beispiel anhand unterschiedlicher *Profile* und *Typen* (z.B.: Einzelteil, Baugruppe oder Zeichnung).

*Log- und Informationsdateien* können einen Teil der Auswahlmöglichkeiten beim Import und Export verwalten, stehen aber längst nicht für alle Dateiformate zur Verfügung. Eine Analyse mit detaillierten Beschreibungen zu Eigenschaften von Import- und Exportoptionen ist in [17] zu finden.

#### 2.4 Beispielprozess: Mehrkörpersimulation

Um die entstehenden Daten und deren Austauschprozesse zu untersuchen, wurde die Vorgehensweise der Entwicklung eines mechatronischen Modells zur Analyse von kinematischem und dynamischem Verhalten mit Hilfe einer Mehrkörpersimulation (MKS) untersucht. Auch wenn das verwendete Beispiel nur eine geringe Komplexität aufweist, enthält es die grundlegenden Fragestellungen an die Nachvollziehbarkeit des Austauschprozesses.

Abbildung 2a illustriert die am Prozess beteiligten CAx-Systeme und Prozessschritte zur Durchführung einer MKS-Simulation. Die Grundlage stellt ein CAD-Modell eines beliebigen Produktes dar. Dieses wird über ein Format für 3D-Grafikmodelle in ein von Pro/ENGINEER importierbares CAD-Format gespeichert (1), oder über einen entsprechenden externen Konverter (2) in ein solches transformiert. Alternativ dazu kann auch STEP als Standardformat für ein beliebiges CAD-System genutzt werden (3). Der Strukturkonverter stellt daraufhin einen Teil der Daten auf Basis von XML bereit (Daten zur Produktstruktur, Masse der Einzelteile, etc.), während ein beliebiges anderes CAD-Programm ein 3D-Modell im VRML-Format bereitstellt (4). Auf der Basis des RobotMax Systems [18] wird ein Modelica MKS-Modell (5) und darauf für Matlab mit C-Code (6) ein MKS-Modell die Simulation erstellt.

Die Erstellung kann somit auf verschiedenen Wegen entweder über ein CAD-Programm und ein Strukturkonverter oder über das Programm Pro/ENGINEER erfolgen.

Die in diesem Ablauf erzeugten Daten sind in Abbildung 2b mit ihren Abhängigkeiten dargestellt. **CAD-Daten für RobotMax** beschreibt die CAD-Daten, die an RobotMax übergeben werden müssen (ein bis mehrere **VRML-Dateien** und eine **XML-Datei**). Zwei unterschiedlichen Möglichkeiten, diese Dateien zu erhalten, stellen der **Strukturkonverter** und **Pro/ENGINEER** bereit. Wird der Prozessschritt über den Strukturkonverter gewählt, fallen zwei native Dateien an (**Strukturkonverter-Dateien**). Außerdem basiert der Export vom Strukturkonverter auf einem CAD-Modell im **STEP-Format**, welches gegebenenfalls aus einem weiteren **CAD-Modell** durch einen Konverter entstanden ist. Ein CAD-Modell wird je nach Format und Umfang durch eine oder mehrere **CAD-Dateien** repräsentiert. Für ein CAD-Modell im STEP-Format existiert zum Beispiel eine **STEP-Datei**. Der Export der CAD-Daten für RobotMax über Pro/ENGINEER basiert auf einem CAD-Modell im nativen Pro/ENGINEER-Format, welches durch eine oder mehrere **ASM-** und **PRT-Dateien** repräsentiert wird. Es kann gegebenenfalls auf einem vorhandenen CAD-Modell aufbauen, welches in Pro/ENGINEER importiert wurde. Aus den CAD-Daten für RobotMax entsteht zunächst ein **mechanisches MKS-Modell**, welches durch eine oder mehrere **Modelica-Dateien** repräsentiert wird. Aus dem mechanischen MKS-Modell können unterschiedliche **mechatronische MKS-Modelle** entstehen, indem mit Hilfe von RobotMax unterschiedliche Motoren zum mechanischen Modell hinzugefügt oder die Parameter der Motoren verändert werden. Das mechanische Modell und die CAD-Daten werden dabei nicht geändert.

Mit einem mechatronischen MKS-Modell können außerdem unterschiedliche **MKS-Simulationen** durchgeführt werden, da in Dymola zur Simulation bestimmte Einstellungen vorgenommen werden müssen. Es entstehen dann jeweils unterschiedliche **Simulationsergebnisse**, die in Form von verschiedenen Dateien abgespeichert werden. Aufbereitete Simulationsergebnisse, zum Beispiel Diagramme, zählen auch zu diesen Dateien. Den Datentransfer über unterschiedliche Versionen und Varianten in einer flexiblen Umgebung nachzuvollziehen, wird mit zunehmenden Austauschmöglichkeiten und verwendeten Werkzeugen komplexer.

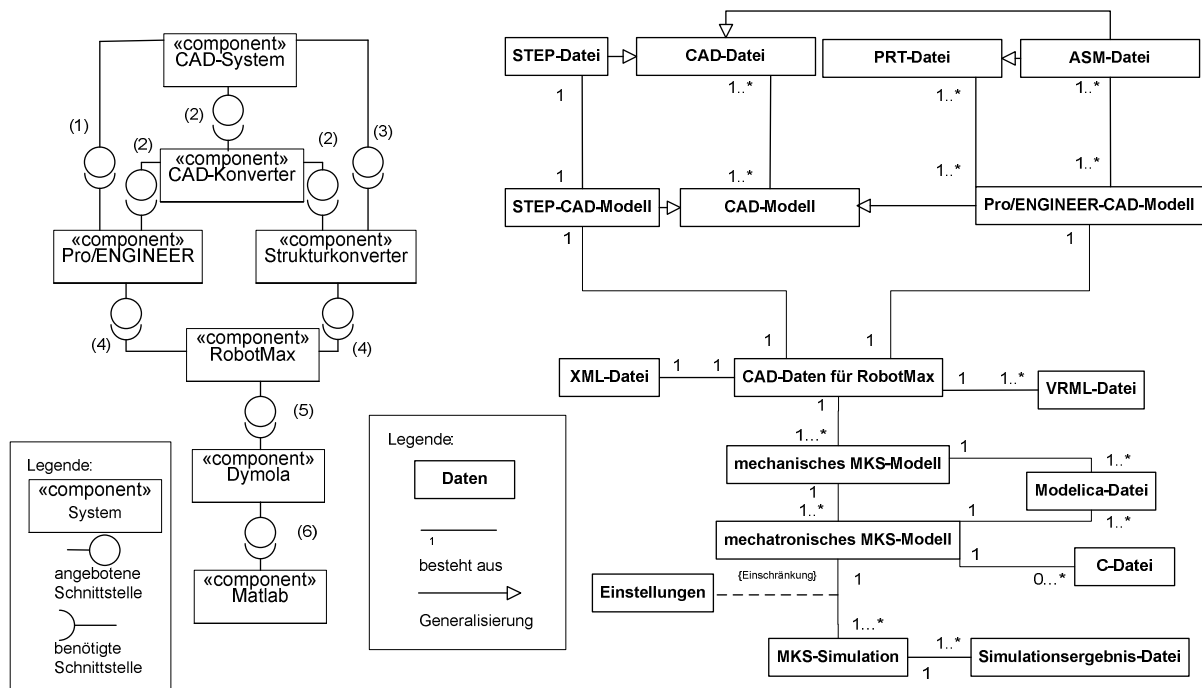


Abbildung 2: Beispielprozess Vorbereitung und Durchführung einer MKS-Simulation

### 3. Prozessunterstützung in der virtuellen Produktentwicklung

Zur Unterstützung der Steuerung von komplexen Datenaustauschprozessen können Workflow-Management (WFM)-Systeme eingesetzt werden. WFM-Systeme ermöglichen die Bereitstellung der richtigen Daten zu den richtigen Personen zur richtigen Zeit. Dafür müssen die Prozesse zunächst im WFM-System definiert werden. Anschließend können Instanzen der Prozesse vom WFM-System gesteuert und ausgeführt werden. WFM-Systeme bieten sich vor allem zur Unterstützung von strukturierten, vorgeplanten und sich häufig wiederholenden Prozessen an [19].

Im Gegensatz zu anderen Prozessen, wie zum Beispiel in der Produktion und Logistik, lassen sich die Prozesse in der Produktentwicklung nicht immer vorausplanen [20]. Der genaue Prozess ergibt sich oft erst im Laufe der Entwicklung. Außerdem ist die Entwicklung eines Produktes durch Kreativität und dynamische Abläufe (zum Beispiel Iterationen) geprägt. Durch eine Vorgabe des Entwicklungsprozesses würde zum einen der kreative Spielraum der Entwickler und zum anderen die dynamische Reaktionsfähigkeit eingeschränkt werden.

Es gibt aber Teilprozesse während der Produktentwicklung, die vorgeplant und damit automatisiert unterstützt werden können, wie zum Beispiel Freigabe- und Änderungsprozesse. Dafür werden in der Produktentwicklung prinzipiell keine eigenständigen WFM-Systeme eingesetzt, sondern PDM-Systeme mit integrierten WFM-Komponenten [21]. Sie verwalten Daten und Dateien (Produktdaten), die während der Produktentwicklung von CAx-Systemen und weiteren Software-Systemen erzeugt werden. Außerdem wird eine Verbindung zwischen den Produktdaten und den zur Erzeugung, Bearbeitung und Weiterleitung notwendigen Prozessen durch diese Softwaresysteme unterstützt.

## 4. Datenmodell zur Verwaltung flexibler Datenaustauschprozesse

Als Grundlage für das Metadaten-Repository wurde in [17] ein Datenmodell entwickelt, das zwei einzelne Datenmodelle zur System- und Prozessverwaltung integriert, wobei das Modell zur Systemverwaltung durch die zugehörige Prozessverwaltung erweitert wird. Zur weiteren Unterstützung von Prozessen müssen die beteiligten Werkzeuge der Entwicklung, Austauschformate und Transformationen beschrieben werden. Das Metadaten-Repository ermöglicht die Erfassung dieser Informationen und bietet darüber hinaus die Möglichkeit diese zu verwalten. Auf der einen Seite verwaltet das Metadaten-Repository Informationen über CAx-Systeme, Austauschformate und Einstellungen bei Export- sowie Importvorgängen. Auf der anderen Seite erfasst das Prozessmodell die Teilprozesse, Aufgaben und deren Abhängigkeiten. Anhand des vereinten Metadaten-Repository können Fragestellungen zur Herkunft von Produkt- und Simulationsdaten in flexiblen und adaptiven Umgebungen ebenso wie die Frage nach einer alternativen Prozessausführung nachvollzogen werden. Die beiden Teilmodelle werden einzeln vorgestellt, um eine bessere Übersichtlichkeit zu gewährleisten, sind jedoch im vorliegenden Prototypen als Einheit umgesetzt worden. Das Modell zur Systemverwaltung wird durch die zugehörige Prozessverwaltung erweitert.

### 4.1 Systemverwaltung

Eine Systemlandschaft besteht aus vorhandenen Systemen unterschiedlicher Hersteller sowie den Schnittstellen zwischen diesen. Das Datenmodell, in Abbildung 3 als UML-Klassendiagramm dargestellt, ermöglicht die Beschreibung von CAx-Systemen sowie ihrer Versionen (*CAx-System-Version*, *Dateiformat-Version*), Schnittstellen bezüglich Import und Export (*Schnittstelle*, *Austauschvorgang*) sowie *Dateiformat* und *Spezifikation*. Verschiedene *Einstellungen* und *Auswahlmöglichkeiten* ermöglichen die Nachvollziehbarkeit von import- oder exportspezifischen Vorgängen (z.B. bei der Auswahl von Teilmodellen). Die *Einstellungsmöglichkeiten* unterscheiden zwischen Export und Import, um in Kombination mit dem später vorgestellten Prozessmodell alternative Prozessabläufe vorzuschlagen.

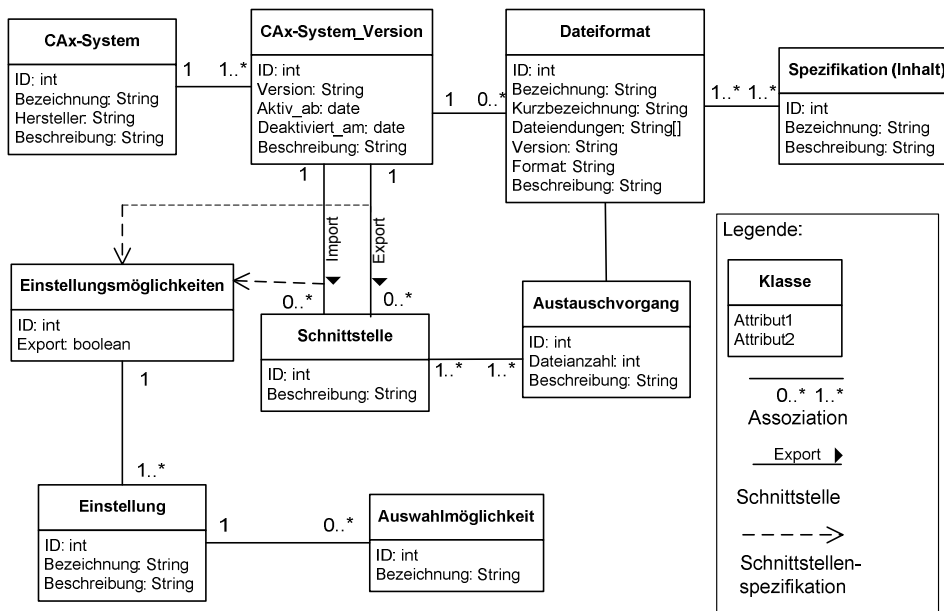


Abbildung 3: Klassendiagramm zur Systemverwaltung (Datenmodell) [17]

Zur Erläuterung des Schemas nutzen wir das Beispiel aus Abschnitt 2, wobei wir uns auf die beiden CAx-Systeme Pro/ENGINEER und RobotMax beschränken. Die Abbildung 4 beschreibt als Objektdiagramm anfallenden Objekte der Systemverwaltung und die Beziehungen zwischen diesen. Dabei wurden die folgenden CAx-System Versionen angelegt:

- Pro/ENGINEER Wildfire 5.0 vom Hersteller Parametric Technology Corporation.
- RobotMax 2.0 vom Fraunhofer IFF.

Dafür wurde jeweils eine Instanz der Klasse *CAx-System* und *CAx-System\_Version* mit den entsprechenden Beschreibungen der Systeme angelegt. Weiterhin wurde eine Export- bzw. Importmöglichkeit zwischen den CAx-System-Versionen definiert. Pro/ENGINEER kann VRML und die benötigte XML-Datei exportieren.

Zu diesen Dateiformaten wurde ein beispielhafter Austauschvorgang definiert, der für das XML-Format genau eine Datei benötigt, während die Datei-Anzahl bei dem VRML-Format von der Anzahl der Einzelteile abhängt. Durch die Definition von Systemen, Schnittstellen, Einstellungsmöglichkeiten, Dateiformaten können nun sehr komplexe Systeme verwaltet werden.

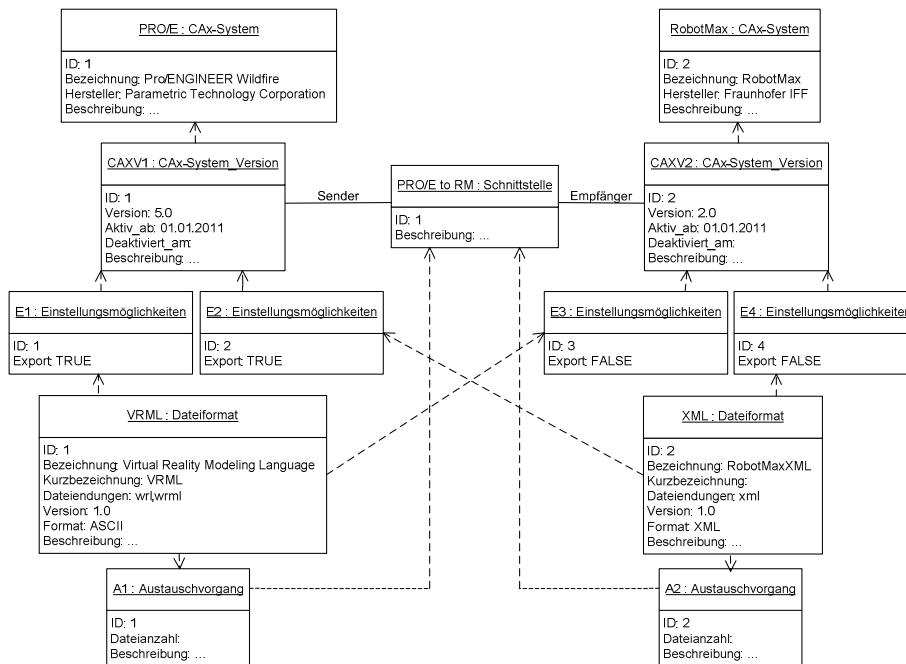


Abbildung 4: Objektdiagramm zur Systemverwaltung (Beispiel)

## 4.2 Prozessverwaltung

Das Objektdiagramm zu einem beispielhaften Prozessmodell (Abbildung 5) illustriert die Definition von Datenaustauschprozessen und deren Parametern. Das Modell (vgl. Abbildung 7) ermöglicht die Festlegung von Prozessabläufen, gekoppelten Schnittstellen und die Abbildung von Alternativen zu bestehenden Prozessabläufen. Zusätzlich zu Eigenschaften von Prozessschritten und Importalternativen werden Daten zur Beschreibung von bereits durchgeführten Schritten und Datenaustauschvorgängen (Instanzen) verwaltet. Durch die Belegung der Prozessabläufe oder -schritte mit Parametern werden feste Abläufe definiert (Instanzen). Zu festgelegten Prozessen können auch alternative Prozessschritte oder Importalternativen vorgeschlagen werden. Als Beispiel kann hier wieder die Alternative des STEP-Formats anstelle von Pro/ENGINEER für den Datenaustausch genutzt werden.

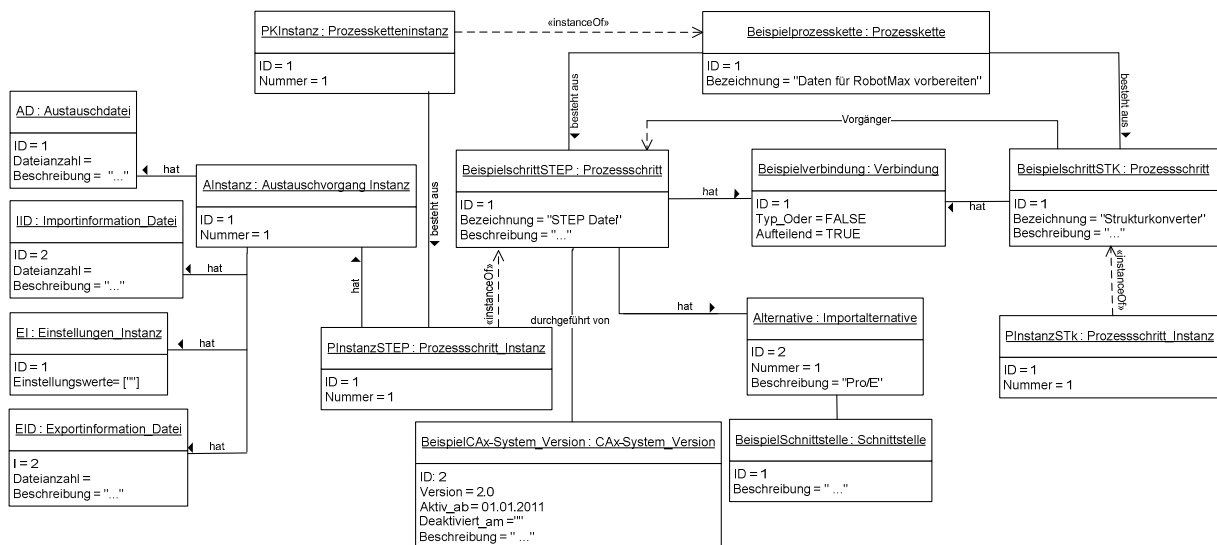


Abbildung 5: Objektdiagramm zur Prozessverwaltung (Beispiel)

## 5. Implementierung

### 5.1 Architektur

Das vorgestellte Metadatenmodell wurde anhand einer prototypischen Implementierung evaluiert. Dazu wurde das konzeptuelle Schema der Datenhaltung in einer relationalen PostgreSQL-Datenbank umgesetzt. Die Daten werden über einen Apache-Tomcat-Webserver, welcher auch die Anwendungslogik beinhaltet, an einen Web-Client weitergereicht. Über diesen können die Anwender auf die Daten zugreifen und neue Systeme, Schnittstellen und Prozesse definieren. Um weitere Funktionalitäten sowie einen automatischen Zugriff auf weitere Metadaten von virtuellen Produktmodellen zu erhalten, wird eine spätere Umsetzung innerhalb eines PDM-Systems vorgeschlagen. Diese Umsetzung würde die Stärken von Workflow-Management-Systemen (z.B.: Prozesse, Versions- und Variantenverwaltung) mit einer neuartigen Form zur Verwaltung und Nachvollziehbarkeit des Datenaustausches kombinieren.

Zur Realisierung des Prototyps wurde eine webbasierte „Client-Server-Architektur“ gewählt. Die Präsentationsschicht wird vom Client realisiert. Er stellt die Benutzeroberfläche für den Anwender bereit und nimmt Interaktionen entgegen. Es werden Hypertext-Markup-Language (HTML), Cascading-Style-Sheets (CSS) und Java-Script-Dateien unterstützt, um so den Browser von Betriebssystem zu entkoppeln.

Auf der Serverseite kommt Apache Tomcat und PostgreSQL als relationale Datenbank zur Anwendung. Dabei kommuniziert der Webbrowser mit dem Webserver per Hypertext-Transfer-Protocol (HTTP). Die Datenbank speichert alle persistenten Daten der Webanwendung, wobei die Kommunikation mit Hilfe der Schnittstelle Java-Database-Connectivity (JDBC) realisiert wird.

### 5.2 Anwendungsfälle

Zur Nutzung des Systems wurden einzelne Anwendungsfälle spezifiziert. Diese lassen sich in Konfiguration und Anwendung unterteilen. Dabei beinhalten einzelne Anwendungsfälle andere Anwendungen (z.B.: CAX-System anlegen enthält automatisch eine Versionsdefinition) oder Erweiterungen (z.B.: ein neues Dateiformat kann direkt vorhandenen Import- und Exportmöglichkeiten zugeordnet werden). Zu den umgesetzten Anwendungsfällen der Konfigurationsmöglichkeiten gehören das Anlegen und Deaktivieren von Einträgen (CAX-Systemen, CAX-System\_Versionen, Dateiformaten, Export- oder Importmöglichkeiten einer CAX-System\_Version, Schnittstellen). Anwendungsfälle hingegen ermöglichen eine Anzeige der Systemlandschaft, Schnittstellen und Export- und Importmöglichkeiten einer CAX-System\_Version. Abbildung 7 zeigt den Anwendungsfall „Prozesskette anlegen“.

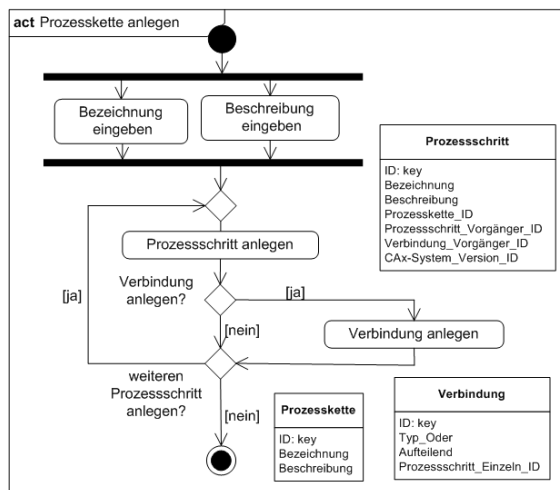


Abbildung 6: Aktivitätsdiagramm zum Anwendungsfall: "Prozesskette anlegen"

### 5.3 Datenbankschema

Das Ergebnis des Datenbankentwurfs ist ein relationales Schema, in welchem die Daten zur Prozess- und Systemverwaltung angelegt sind. Die Pfeile stellen Referenzschlüssel (abhängige Attribute) und damit Verbindungen zu anderen Tabellen dar. Die Beschreibungen der CAX\_Version, sowie die Beschreibung der Austauschvorgänge lassen sich beispielsweise durch die in Abschnitt 2.3 genannten Attribute verfeinern. Dabei werden die Fremdschlüssel in der Abbildung 7 zusätzlich durch Verbindungen gekennzeichnet.



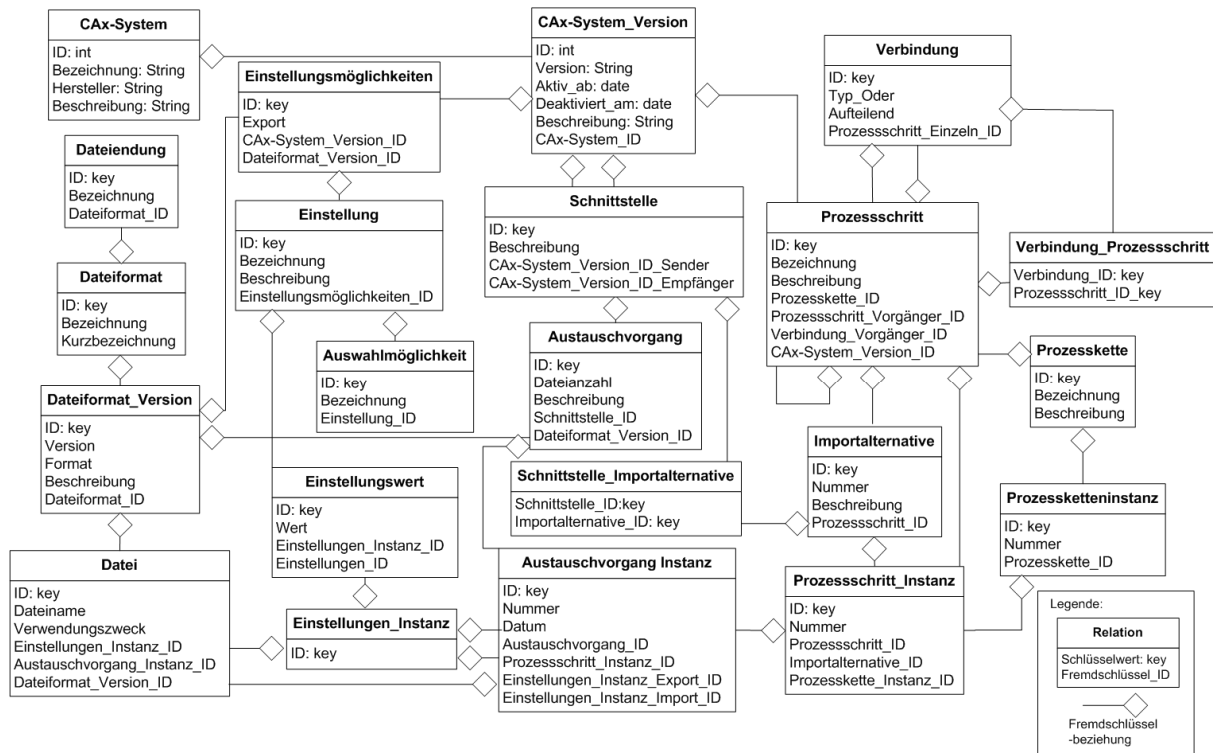


Abbildung 7: Relationales Schema der Datenbank nach [17]

## 6. Zusammenfassung und Ausblick

In der virtuellen Produktentwicklung entstehen immer komplexere Systemlandschaften und zwischen diesen ebenso komplexe Datenaustauschprozesse. Die Nachvollziehbarkeit dieser Prozesse kann nur sehr eingeschränkt durch herkömmliche Workflow-Management-Systeme unterstützt werden, da diese zumeist auf festen Beziehungen und Abläufen beruhen, welche jedoch für flexible Entwicklungsprozesse nicht geeignet sind.

Eine Möglichkeit solche komplexen Systemlandschaften und Abläufe zu verwalten, stellen die Sammlung der Metadaten dieser Analyse dar. Diese können geeignete Informationen über den Ablauf erfassen und so eine Verwaltung flexibler, komplexer Systemlandschaften ermöglichen. Alternativen beim Datenaustausch werden ebenso beschrieben, wie sich ändernde Systeme und Austauschvorgänge. Als Ergebnis entsteht ein Ablauf für Datenaustauschprozesse und dabei anfallende Daten, ebenso wie Beschreibungseigenschaften für diese Formate im Bereich der virtuellen Produktentwicklung.

Die beiden Teilkonzepte, System- und Prozessverwaltung, wurden in einer gemeinsamen Datenbank implementiert und können mit Hilfe einer Webanwendung genutzt werden.

Mit dem entwickelten Konzept können Systeme erfasst, verwaltet und gesteuert werden. Es können verschiedene Entwicklungsstadien der genutzten Systemlandschaft, mit ihren jeweiligen Import-, Export- und Austauschmöglichkeiten sowie deren Auswirkungen abgebildet werden. Es wurde gezeigt, dass dabei den Schnittstellen und ihren Vor- und Nachteilen eine besondere Bedeutung zukommt. Des Weiteren ist es für Prozessverwalter nun möglich, Informationen über einen Entwicklungsprozess weitergehend zu überprüfen.

Der Prototyp wurde auf der Basis freier Software entwickelt, kann aber in bestehende Workflow- und PDM-Systeme eingebunden werden. Dabei kann die Metadatenverwaltung eines PDM-Systems soweit angepasst werden, dass eine Integration von Prozessverwaltung und weiterführender Prozessunterstützung ermöglicht wird. Anpassungen an weitere Domänen und Austauschformate der virtuellen Produktenwicklung sind Thema zukünftiger Forschung und Entwicklung.

### Danksagungen

Diese Veröffentlichung entstand in Kooperation mit dem Verbundprojekt Competence in Mobility (COMO C1-3201201 und C3-320120) und wurde durch das deutsche Ministerium für Bildung und Forschung innerhalb des Projektes ViERforES-II (Nr. 01IM10002B) unterstützt. Der Inhalt dieser Veröffentlichung steht in alleiniger Verantwortung der Autoren und spiegelt somit in keiner Weise die Meinung der Europäischen Union oder des Ministeriums für Bildung und Forschung wieder.

## Literaturverzeichnis

- [1] Gümbel, Christoph: Wie digitale Prototypen durch Virtual Reality lebendig gemacht werden. In: Schenk, Michael (Hrsg.): Virtual Reality - Mensch und Maschine im interaktiven Dialog: Gastvortragsreihe 2006 Bd. 3. Magdeburg : Fraunhofer Institut für Fabrikbetrieb und -automatisierung IFF, 2006, S. 11–63.
- [2] Köppen, Veit; Saake, Gunter: Einsatz von Virtueller Realität im Prozessmanagement. In: Industrie Management (2010), Nr. 2, S. 49-53.
- [3] Swieczek, B. ; Arnold, F. ; Kilb, T. ; Janocha, A. T. ; Sartiono, R.: Online-Kopplung von CAX-Systemen für die virtuelle Produktentwicklung - ein Vergleich mit dem dateibasierten Datenaustausch. In: Verein Deutscher Ingenieure (Hrsg.): Prozessketten für die virtuelle Produktentwicklung in verteilter Umgebung: Tagung München (VDI-Berichte 1435). Düsseldorf: VDI-Verlag, 1998. S. 219–238.
- [4] Grabowski, Hans ; Anderl, Reiner ; Polly, Adam: Integriertes Produktmodell. 1. Aufl. Berlin : Beuth Verlag GmbH, 1993.
- [5] Stekolschik, Alexander: Ein Beitrag zum ganzheitlichen Qualitätsmanagement von CAD-Modellen in der Produktentstehung. Bochum, Ruhr-Universität Bochum, Diss., 2007.
- [6] Sauter, J.; Fricke, H.; Güngör, Z.; Himmler, G.; Hougardy, P.; Lauber, B.; Müller, O.; Neithardt, W.; Schirmmayer, R.: Integrierte Topologie- und Gestaltoptimierung im virtuellen Produktentstehungsprozess – Einbindung in die iViP Architektur und industrielle Anwendung. In: Internationaler VDI-Kongress, 14.-15. September, Würzburg, 2000.
- [7] Vornholt, Stephan; Geist, Ingolf; Li, Yuexiao: Categorisation of Data Management Solutions for Heterogeneous Data in Collaborative Virtual Engineering. In: Saake, Gunter; Köppen, Veit (Hrsg.): IWDE 2010: Proceedings of the 1st International Workshop on Digital Engineering; Otto-von-Guericke-Universität Magdeburg, ACM, 2010, S. 9–16.
- [8] Grabowski, Hans; Anderl, Reiner: Produktdatenaustausch und CAD-Normteile. Ehningen bei Böblingen : expert-Verlag, 1990.
- [9] Anderl, Reiner (Hrsg.); Trippner, Dietmar (Hrsg.): STEP - STandard for the Exchange of Product Model Data: Eine Einführung in die Entwicklung, Implementierung und industrielle Nutzung der Normenreihe ISO 10303 (STEP). Stuttgart : Teubner, 2000.
- [10] VDI 2209:3-D-Produktmodellierung - Technische und organisatorische Voraussetzungen - Verfahren, Werkzeuge und Anwendungen - Wirtschaftlicher Einsatz in der Praxis. Berlin : Beuth Verlag GmbH, März 2009.
- [11] Dyla, Andreas: Modell einer durchgängig rechnerbasierten Produktentwicklung. München, Technische Universität München, Diss., 2002.
- [12] Schuhmann, Sören: Methoden zur Optimierung des Austausches von Geometrie- und Parametrikdaten. Magdeburg, Otto-von-Guericke-Universität Magdeburg, Diss., 2001.
- [13] Pratt, Michael J.: Introduction to ISO 10303 - the STEP Standard for Product Data Exchange. In: Journal of Computing and Information Science in Engineering (2001), Nr. 1, S. 102–103.
- [14] Sandler, Ulrich: Entscheidend ist die Qualität der Daten - JT, führendes Format für die Verbreitung und Visualisierung von CAD-Daten, wird zum ISO Standard. In: Interface (2009), Nr. 02, S. 10–11.
- [15] ISO 14772-1: Information technology - Computer graphics and image processing - The Virtual Reality Modeling Language - Part 1: Functional specification and UTF-8 encoding, 1997.
- [16] ISO 19775-1: Information technology - Computer graphics and image processing - Extensible 3D (X3D) - Part 1: Architecture and base components, 2004.
- [17] Stoye, Michael: Entwicklung eines Datenmodells zur Unterstützung des dateibasierten Datenaustauschs in der Produktentwicklung, Diplomarbeit, Magdeburg, 2011.
- [18] Juhász, Tamás; Schmucker, Ulrich: Automatic Model Conversion to Modelica for Dymola-based Mechatronic Simulation. In: Bachmann, Bernhard (Hrsg.): Modelica 2008: Proceedings of the 6th International Modelica Conference; Bielefeld, 2008, S. 719–726.
- [19] Reichert, Manfred; Dadam, Peter: Geschäftsprozessmodellierung und Workflow-Management Konzepte, Systeme und deren Anwendung. In: Industrie Management (2000), Nr. 3, S. 23–27.
- [20] Meerkamm, Harald (Hrsg.); Henrich, Andreas (Hrsg.); Jablonski, Stefan (Hrsg.); Krcmar, Helmut (Hrsg.); Lindemann, Udo (Hrsg.); Rieg, Frank (Hrsg.): Flexible Prozessunterstützung in der Produktentwicklung: Prozesse – Daten - Navigation: Bayerischer Forschungsverbund FORFLOW für Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung; Abschlussbericht; Aachen : Shaker, 2009.
- [21] VDI 2219: Informationsverarbeitung in der Produktentwicklung – Einführung und Wirtschaftlichkeit von EDM/PDM-Systemen. Berlin : Beuth Verlag GmbH, Mai 1993.