

Grundlagen einer feature-orientierten Referenzbibliothek im Virtual Engineering

Stephan Vornholt, Mario Pukall und Ingolf Geist

Fakultät für Informatik,

Universität Magdeburg

{vornholt, pukall, geist}@iti.cs.uni-magdeburg.de

Zusammenfassung

Die Entwicklung von Produkten mit Hilfe von computergestützten Systemen erlangt durch den wachsenden Wettbewerbsdruck immer mehr Bedeutung. Der Prozess des Virtual Engineering umfasst die frühen Phasen der Produktentwicklung vom Entwurf über die Konstruktion bis zur Simulation und Analyse. Dieser Prozess verlangt eine Daten- und Modellverwaltung, die über bisherige Insellösungen hinausgeht. Eine vielversprechende Möglichkeit, verschiedene Modelle physikalischer Spezialbereiche und Phasen miteinander zu verknüpfen, stellt eine Komponentenbibliothek dar. Ein virtuelles Produkt sowie die zugehörigen Modelle werden dort als Komponente gespeichert. Dabei ist die Wiederverwendung der Komponenten von besonderer Bedeutung, um eine änderungsbedingte Neuentwicklung virtueller Produkte zu vermeiden. In diesem Zusammenhang wird eine Zerlegung von Komponenten in feingranulare, unabhängige Bestandteile benötigt, was die geeignete Modellierung der Verbindungen zwischen Komponenten beinhaltet.

Das aus der Softwaretechnik bekannte Verfahren der feature-orientierten Programmierung ermöglicht die Modularisierung von Eigenschaften oder Funktionalitäten, die über verschiedene Klassen verteilt sein können. Die Technik erweitert damit den objekt-orientierten Ansatz um sinnvolle Dimensionen, die auch in Produktbibliotheken des Virtual Engineering ihre Anwendung finden können. In dieser Arbeit sollen die Möglichkeiten und Probleme einer zentralen Komponentenbibliothek für eine durchgängig virtuelle Entwicklung von Produkten aufgezeigt und Lösungsansätze auf der Basis aktueller Methoden der Softwaretechnik beschrieben werden.

1 Einleitung

Durch den wachsenden Wettbewerbsdruck müssen Produkte schnell auf den Markt gebracht werden, gleichzeitig kostengünstig produziert werden und qualitativ hochwertig sein. In der Raum-/Luftfahrt und der Automobilindustrie wurde der Entwicklungsprozess als entscheidender Faktor für die Erfüllung der genannten Kriterien identifiziert. Unterstützung verspricht die virtuelle Produktentwicklung. Der Prozess des Virtuell Engineering (VE) umfasst Design, Modellierung, Tests und Weiterentwicklung virtueller Produkte, ohne ein physisches Produkt herstellen zu müssen. Dabei kapselt ein virtuelles Produkt (VP) verschiedene Modelle, welche das reale Produkt in unterschiedlichen Phasen der Entwicklung aus verschiedenen Blickwinkeln darstellen (vgl. Abbildung 1) [16].

Der Prozess des VE stützt sich dabei auf die Arbeiten von unterschiedlichen Teams in den Bereichen Design, Modellbildung, Simulation und Analyse. Dabei werden sie durch spezielle Werkzeuge unterstützt, welche sich, je nach physikalischem Spezialbereich (z.B.: mechanische/elektrische Modellierung), oft erheblich unterscheiden. Dennoch sind die Modelle der verschiedenen Bereiche und Phasen eng aneinander gekoppelt, so dass ein Informationsaustausch unerlässlich ist [11]. Eine Datenhaltung muss dabei die Zusammenarbeit verschiedener Teams, Bereiche und Phasen, sowie verschiedene Versionen und Varianten flexibel unterstützen.

In dieser Arbeit sollen Anforderungen an eine Datenverwaltung für den VE-Prozess gestellt, die Modellierung einer VE-Referenzbibliothek motiviert und Ansätze zur Umsetzung mit Hilfe aktueller Forschungsansätze der Softwaretechnik dargestellt werden. Ein zentraler Bestandteil

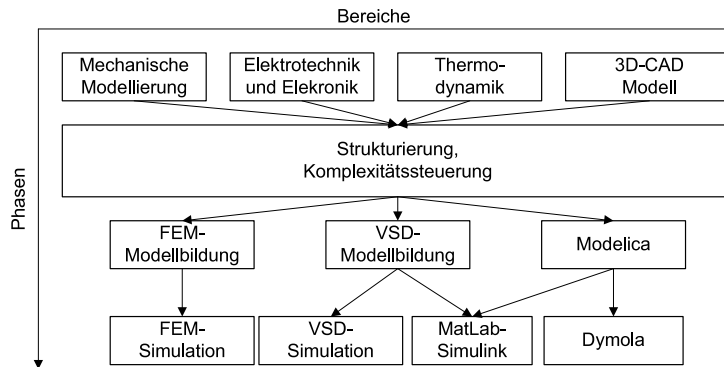


Abbildung 1: VE-Prozess Modell

ist die Entwicklung gemeinsamer Begriffe und ein übergeordnetes Datenmodell. Diese Arbeit stellt einen ersten Schritt zur Entwicklung einer Referenzbibliothek für den Bereich des VE dar.

2 Stand der Technik

Im VE-Prozess ist der Informationsaustausch über verschiedene Phasen und Bereiche hinweg von zentraler Bedeutung. Zur Unterstützung eines durchgängigen VE-Prozesses gibt es eine Vielzahl unterschiedlicher Ansätze.

Kommerzielle CAD-Systeme. CAD-Systeme ermöglichen ein computergestütztes Design von Produkten und bieten Erweiterungen zur Simulation sowie Testumgebungen für virtuelle Produkte. Diese Erweiterungen ermöglichen eine schnelle aber beschränkte Berechnung der Modelle und Tests von virtuellen Prototypen während der ersten Phasen der Entwicklung, bleiben aber produktabhängig und schränken somit die Sichtweise ein. Ein komponentenbasierter Ansatz wird jedoch nicht unterstützt.

Produktdatenmanagement (PDM). PDM-Systeme werden eingesetzt, um eine Datenhaltung in VE-Systemen zur Verfügung zu stellen. Dabei ist die Basis ein Produktmodell, das die produktdefinierenden Daten verwaltet und über den gesamten Produktlebenszyklus speichert. PDM-Systeme nutzen dabei Standardschnittstellen zum Austausch mit CAD/CAE-Systemen. Dennoch können die phasen- und bereichsübergreifenden Anforderungen nur bedingt unterstützt werden [15, 9].

Design-Repositories. Ein Konzept zur Speicherung und Wiederverwendung von Modellinformationen und Modellen für den Entwurf sind Design-Repositories [14, 6]. Dabei können die Metainformationen, Schnittstellen, Datentypen und Genauigkeiten von Modellen erfasst werden. Design-Repositories enthalten keine flexiblen und formalen Beschreibungen für Kopplungsmöglichkeiten verschiedener Modelle.

Schnittstellen. Datenschnittstellen zwischen verschiedenen Systemen spielen eine zentrale Rolle bei der Implementierung von VE-Systemen. Grundlegende Methoden des Datenaustauschs zwischen den verschiedenen Systemen sind neutrale Austauschformate (STEP, IGES) [13, 7], eine direkte Einbindung verschiedener Systeme in ein Gesamtmodell [10] oder spezielle Schnittstellen zwischen verschiedenen Systemen.

Alle Ansätze beschreiben Teillösungen der gegebenen VE-Problematik, scheitern jedoch an einem formalen Weg, über Phasen und Bereiche des VE hinweg, die Modelle zu beschreiben und zu verwalten. Es gibt Insellösungen für das Design und die Modellverwaltung, jedoch handelt es sich dabei um begrenzte Teillösungen, die keine Referenzlösung für das VE darstellen.

3 Referenzbibliothek

Zentraler Gegenstand unserer Forschung ist die Modellierung einer Referenzdatenbank, welche Metainformationen innerhalb des VE-Systems aufnimmt und Schnittstellen zu den einzelnen

Werkzeugen bereitstellt. Die Referenzbibliothek soll Vorteile einzelner Ansätze (CAD, PDM, Design-Repositories) miteinander verbinden und eine Unterstützung für den gesamten Bereich des VE gewährleisten [8].

3.1 Anforderungen

Den ersten Schritt bei der Entwicklung einer Referenzbibliothek stellt die Anforderungsanalyse dar. In dieser Analyse sollen Möglichkeiten einer zentralen Referenzbibliothek betrachtet werden, um den VE-Prozess geeignet zu unterstützen. Eine Referenzbibliothek, welche Modelle verschiedener Sichten in einem einzigen virtuellen Produkt kapselt, soll die Grundlage vielfältiger Möglichkeiten der Unterstützung des VE-Prozesses bilden.

Existierende Ansätze konzentrieren sich auf eine durchgehende Unterstützung der Entwicklung vom Design bis zum Test. Daten und Modelle werden durch Schnittstellen an Werkzeuge der nächsten Phase weitergegeben. Informationen, anhand derer die Entwicklungsschritte vom Test bis hin zu den beteiligten Modellen des Designs nachvollzogen werden können (rückgekoppelter Entwicklungsprozess), werden vernachlässigt. Die Ergebnisse eines Tests und daraus resultierende Änderungen an den Modellen des Designs erfordern eine *Anfrageunterstützung*, sowie Möglichkeiten der *Erweiterung*, *(Re-)kombination* und *Wiederverwendung* von Modellen über verschiedene Phasen der Entwicklung hinweg.

Diese Möglichkeiten müssen bereits in den abgespeicherten virtuellen Produkten und ihren Modellen beachtet und das Metamodell entsprechend konstruiert werden. Um Teilkomponenten erweitern, wiederverwenden und (re-)kombinieren zu können, ist eine feingranulare Modellierung notwendig. Das virtuelle Produkt muss als Kombination verschiedener Modelle gespeichert werden, die wiederum in ihre Basismodelle und deren Verbindungen zerlegt werden können.

Gleiche oder ähnliche Daten werden häufig mehrfach gespeichert. So liegt ein VP oft in verschiedenen anwendungsspezifischen Formaten vor. Diese Redundanz muss vermieden oder verwaltet werden.

Ähnlichkeiten zwischen Modellen bieten die Möglichkeit, Strukturen zu übernehmen und Metadaten sowie Verbindungen zu nutzen, um Vorlagen zu erstellen. Dies betrifft die Varianten eines VP ebenso, wie verschiedene Sichten auf ein VP. Beispielsweise kann das Modell eines Starrkörpers als Grundlage für ein mechanisches Modell genutzt werden.

Auch können Regeln für die Ableitung von Modellen vorgegeben und genutzt werden. Dies kann ein VP, Modelle oder einzelne Verbindungen betreffen. Regeln können auch daraufhin untersucht werden, ob sie von Teilkomponenten "vererbt" werden. Grundlage einer solchen Unterstützung ist eine feingranulare Modellierung und Unterstützung der Prozesse im VE.

Weitere Aspekte sind Ähnlichkeitssuche und Mehrnutzerbetrieb, die durch eine Referenzdatenbank unterstützt werden können [5].

3.2 Struktur der Referenzbibliothek

Aufgabe einer Bibliothek ist die Verwaltung von komplexen Objekten sowie die Speicherung von Vorlagen und Grundelementen. Dies kann Metadaten, Verbindungen und einzelne Komponenten betreffen. Eine grundlegende Betrachtung der abzulegenden Objekte ist im Folgenden gegeben.

Komponenten. Modelle verschiedener Sichten und Phasen eines virtuellen Produkts werden als Basiskomponenten abgelegt. Eine Basiskomponente enthält Metadaten und mindestens ein Modell. Komponenten, die durch Verbindungen aus mehreren anderen Komponenten erzeugt werden, referenzieren auf die Basiskomponenten und die Verbindungen zwischen ihnen (vgl. Abbildung 2). Eine möglichst feingranulare Zerlegung der Komponenten in Basiskomponenten und ihre Verbindungen erleichtert die Wiederverwendung. In vielen Modellierungswerkzeugen bestehen bereits umfangreiche Standardkomponentenbibliotheken (z.B.: Modelica [12, 1]).

Verbindungen. Werden zwei Komponenten zusammengefügt, wird eine Verbindung benötigt. Da verschiedene Sichten auf die gekoppelten Komponenten existieren, müssen für jede Sicht die

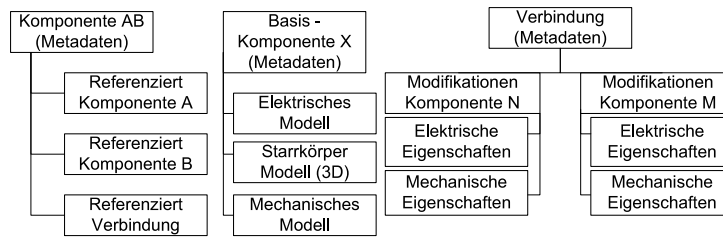


Abbildung 2: Grundlagen des Aufbaus einer Referenzbibliothek

relevanten Verbindungsinformationen gesondert abgelegt werden. Eine Verknüpfung von mehreren Komponenten verlangt oft eine Modifikation der Komponenten selbst. Um Verbindungen flexibler nutzen und Änderungen einfacher gestalten zu können, werden Modifikationen in den Verbindungen abhängig von der Komponente und dem Modell verwaltet.

Modell. Durch verschiedene Entwicklungswerkzeuge erzeugte Modelle werden unverändert, d.h. formaterhaltend, gespeichert. Die Modelle beschreiben Designinformationen aus verschiedenen Sichten (vgl. Abbildung 2).

Instanz/Parameter. Um eine Instanz einer Komponente zu erhalten, muss die Komponente entsprechend ihrer Eigenschaften (z.B.: Länge, Breite, Masse) parametrisiert werden.

4 Feature-orientierte Referenzbibliothek

Wie aus Abschnitt 3.1 hervorgeht, sind die Wiederverwendung von Verbindungen und die (Re-)kombination von Komponenten mittels feingranularer Modularisierung zentrale Aspekte der Unterstützung des VE-Prozesses.

Modularisierung zur Verbesserung von Wiederverwendbarkeit und (Re-)kombinierbarkeit ist ebenfalls eines der Anliegen der objekt-orientierten Programmierung. Im VE-Modell können Komponenten als Klassen und Instanzen einer Komponente als Objekte aufgefasst werden.

Wie in Abschnitt 3.2 beschrieben, beinhalten Verbindungen auch Modifikationen der beteiligten Komponenten. Im Sinne der objekt-orientierten Programmierung können Verbindungen als Eigenschaften, die sich über mehrere Klassen erstrecken, betrachtet werden. Eine effiziente Unterstützung der Modularisierung dieser Eigenschaften bietet die objekt-orientierte Programmierung nicht. Ein aktueller Ansatz zur Lösung dieses Problems stellt die feature-orientierte Programmierung (FOP), als Erweiterung der objekt-orientierten Programmierung, dar.

Abbildung 3 illustriert das Konzept von FOP. Jede Schicht (Layer L_1 - L_3) kapselt eine spezielle Eigenschaft (Feature) und beinhaltet die für die Implementierung der Eigenschaft nötigen Änderungen an den Klassen (C_A - C_C) [3, 4, 2]. Die mittels FOP erstellten Module lassen sich zu unterschiedlichen Programmen zusammenfügen, d.h. ein Modul kann in verschiedenen Programmen Wiederverwendung finden. Zusätzlich können Regeln über Abhängigkeiten zwischen einzelnen Modulen definiert werden, um unzulässige Kombinationen auszuschließen.

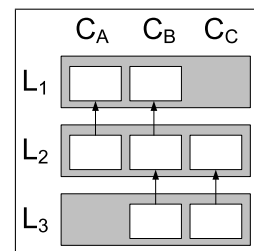


Abbildung 3: Feature-Konzept

Um diesen Ansatz in das Modell einer Referenzbibliothek zu übertragen, werden Verbindungen separat abgelegt (vgl. Abbildung 2). Verbindungen zwischen Komponenten können dann als gekapselte Eigenschaft (Feature) betrachtet werden. Eine Verbindung ist damit eine sich über mehrere Komponenten erstreckende Funktionalität. Auch hier können Regeln definiert werden, um Abhängigkeiten einzubinden. Vorteil der feature-orientierten Modellierung wäre neben der (Re-)kombinierbarkeit, die Wiederverwendung, so dass die Bibliothek um Standardverbindungen erweitert werden kann. Als Beispiel kann hier eine Schweißverbindung zwischen zwei Komponenten dienen. Basisdaten einer solchen Verbindung könnten in verschiedenen Modellen wiederverwendet werden.

5 Zusammenfassung und Ausblick

Die Entwicklung einer Referenzbibliothek für das VE unter der Berücksichtigung aktueller Themen der Softwaretechnik führt zu einer neuartigen Betrachtungsweise existierender Probleme im computergestützten Entwicklungsprozess. Trotz der Stärken kommerzieller CAD-Systeme, PDM, Design-Repositories und neutralen Schnittstellen bieten diese nur Teillösungen und sollten durch ein Datenmodell verbunden werden.

Ausgehend von dem Anspruch einen durchgehenden und rückgekoppelten Entwicklungsprozess im VE zu ermöglichen, wird eine Modellierung der Komponenten, mit einem geeigneten Metamodell, erforderlich. Aus Gründen der (Re-)kombination und der Wiederverwendung ist eine feingranulare Modularisierung und Verwaltung der Strukturen notwendig.

Feature-basierte Modellierung ist ein vielversprechender Ansatz, Verbindungen zu beschreiben und modular zu verwalten.

Ziel der weiteren Forschungsarbeit wird eine Referenzdatenbank sein, die auf bestehenden Systemen (CAD, PDM) und Standards (STEP) basiert und diese erweitert. Dazu muss ein geeignetes Metamodell für die komponentenbasierte Entwicklung erstellt und ein Prozessmodell entwickelt werden.

Literatur

- [1] *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling -Language Specification-Version 2.2.*, 2005.
- [2] Sven Apel. *The Role of Features and Aspects in Software Development*. PhD thesis, University of Magdeburg, 2007.
- [3] Sven Apel and Don Batory. When to Use Features and Aspects? A Case Study. In *Proceedings of ACM SIGPLAN 5th International Conference on Generative Programming and Component Engineering (GPCE'06)*, Portland, Oregon, 2006.
- [4] Sven Apel, Thomas Leich, Marko Rosenmüller, and Gunter Saake. FeatureC++: Feature-Oriented and Aspect-Oriented Programming in C++. Technical report, Department of Computer Science, Otto-von-Guericke University.
- [5] Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, Martin Pfeifle, and Matthias Renz. Der virtuelle Prototyp: Datenbankunterstützung für CAD-Anwendungen. *Datenbank-Spektrum*, 4(10):22–29, 2004.
- [6] M.J. Clayton, P. Teicholz, M. Fischer, and J. Kunz. Virtual components consisting of form, function and behavior. *Automation in Construction*, 8(3):351–367, February 1999.
- [7] R. Dürr, M. Endres, U. Neerpasch, W. Schiehlen, and L. Witte. Standardisierung eines neutralen Datenformats in STEP für die Simulation mechatronischer Systeme. In *VDI-Berichte Nr. 1289: Informationsverarbeitung in der Konstruktion*, page 381, 1996.
- [8] Andreas Dyla. *Modell einer durchgängig rechnerbasierten Produktentwicklung*. PhD thesis, Technische Universität München, Germany, 2001.
- [9] Dimitri Giwerzew, Holger Wierschin, and Georg Paul. PDM-Basisdienste für die integrierte virtuelle Produktentstehung. In *5. Internationale Tagung Wirtschaftsinformatik 2001*, Universität Augsburg, Germany, 2001.
- [10] F.-L Krause, T. Tang, and U Ahle, editors. *Abschlussbericht -Integrierte Virtuelle Produktentstehung*. Carl Hanser Verlag, München, Germany.
- [11] Matthias Kreimeyer, Ulrich Herfeld, Frank Deubzer, and Udo Lindemann. Effiziente Zusammenarbeit von Konstruktions- und Simulationsabteilungen in der Automobilindustrie (In German). *Competence in Design and Development (CiDaD-WPS)*, 02(01), Juni 2006.
- [12] Fritzson P. and Engelson. Modelica - A Unified Object-Oriented Language for System Modelling and Simulation. In *ECCOP'98 - Object-Oriented Programming, 12th European Conference*, page 67, 1998.
- [13] L. Pavez, editor. *STEP Datenmodelle zur Simulation mechatronischer Systeme. Abschlussbericht des Verbundprojekts MechaSTEP*. Forschungszentrum Karlsruhe GmbH, Germany, 2001.
- [14] Simon Szykman, R.D. Sriram, Christophe Bochenek, and Janusz Racz. The NIST Design Repository Project. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 5–19, London, 1999. Springer-Verlag.
- [15] C. Weber, H. Werner, and T. Deubel. A different view on Product Data Management/Product Life-Cycle Management and its future potentials. *Journal of Engineering Design*, 14(4):447–464, December 2003.
- [16] F. Zorriassatine, C. Wykes, R. Parkin, and N. Gindy. A survey of virtual prototyping techniques for mechanical product development. *Proceedings of the I MECH E Part B Journal of Engineering Manufacture*, 217(4):513–530, April 2003.