

EIN INSTRUMENTARIUM ZUR PLANUNG VON PRODUKTENTWICKLUNGSPROZESSEN

Alexander Redenius, Daniel Steffen

Zusammenfassung

Technische Systeme beruhen mehr und mehr auf der synergetischen Verbindung von Mechanik, Elektronik, Software- und Regelungstechnik, was durch den Begriff Mechatronik zum Ausdruck kommt. Der hohen Produktkomplexität und den zahlreichen Abhängigkeiten im Produkt und im Entwicklungsprozess soll mit einem Instrumentarium zur Planung von Produktentwicklungsprozessen Rechnung getragen werden. Es beruht auf einer Fallbasis von ausgeprägten Entwicklungsprozessen und einem Methodenbaukasten von etablierten Entwicklungsmethoden.

1 Instrumentarium

Künftige Systeme des Maschinenbaus werden aus Konfigurationen von intelligenten Systemelementen bestehen. Das Verhalten des Gesamtsystems wird dabei durch die Kommunikation und Kooperation intelligenter Systemelemente geprägt. Solche Systeme erlauben, sich an veränderte Umweltbedingungen anzupassen und gehen über die im Maschinenbau bekannten Regel- und Adaptionsstrategien wesentlich hinaus. Daraus resultiert eine höhere Komplexität der Produkte wie auch der Produktentwicklungsprozesse [1]. Im Sonderforschungsbereich 614 „Selbstoptimierende Systeme des Maschinenbaus“ wird ein Instrumentarium erarbeitet, das die Entwicklung selbstoptimierender Systeme unterstützt. Es umfasst u.a. die folgenden zwei Bestandteile zur Planung von Produktentwicklungsprozessen: 1) eine Fallbasis für erprobte Entwicklungsprozesse auf Basis von Prozessbausteinen und 2) einen Methodenbaukasten. Die Fallbasis und der Methodenbaukasten sind Teil eines umfangreichen Wissenspools, der langfristig Dritte in die Lage versetzen soll, selbstoptimierende Systeme zu entwickeln.

1.1 Fallbasis

Der Entwickler soll bei der Planung eines neuen Entwicklungsprozesses Hilfestellung durch Erfahrungswissen in Form von durchgeführten Prozessen erhalten. D.h. es soll anhand der Aufgabenstellung für ein zu entwickelndes Produkt ein Prozess gesucht werden, mit dessen Hilfe bereits ein ähnliches Produkt erfolgreich entwickelt wurde. Dieser dient dann als Vorlage für den neuen Entwicklungsprozess.

Um dies zu realisieren wurde eine Fallbasis entwickelt, in der detailliert beschriebene Prozesse abgelegt und gesucht werden können. Für die Ablage und Suche wurde eine fallbasierte Klassifikation (Fallbasiertes Schließen) gewählt, d.h. Problem (Beschreibung der Entwicklungsaufgabe und der Randbedingungen) und Lösung (erfolgreicher Prozess) werden als Kombination zusammengefasst und mit charakterisierenden Merkmalen (Attribute und Ausprägung) versehen.

Für die Beschreibung der Entwicklungsaufgabe wurden drei Bereiche definiert, Aufgabenstellung, Randbedingungen und Zielsystem. Innerhalb dieser Bereiche wurden Attribute und deren mögliche Ausprägungen definiert. Attribute des Bereichs Aufgabenstellung sind z.B.: Komplexität, Funktionalität und Stückzahl. Für die Randbedingungen sind die Attribute bei-

spielsweise benötigte Ressourcen sowie die Projektart und für das Zielsystem sind es Kosten, Lebensdauer, Leistungsmerkmale, etc.

Es wurden zunächst Entwicklungsprozesse für mechatronische und selbstoptimierende Produkte aus Forschungs- und Industrieprojekten dokumentiert und entsprechend dieser Attribute und deren Ausprägungen klassifiziert und in die Fallbasis eingestellt. Dadurch wurde es möglich mit Hilfe der Attribute, deren Ausprägungen und einer entsprechenden Gewichtung nach dem Entwicklungsprozess zu suchen, der am besten geeignet ist. In Bild 1 ist ein Suchergebnis (oben rechts) einer solchen Anfrage abgebildet. Demnach würde "Prozess 13" gewählt werden, da sein Ähnlichkeitswert (0,87) am größten ist. Der zur Berechnung des Ähnlichkeitswertes verwendete Algorithmus stammt aus der bekannten, ähnlichkeitsbasierten Suche des fallbasierten Schließens [2].

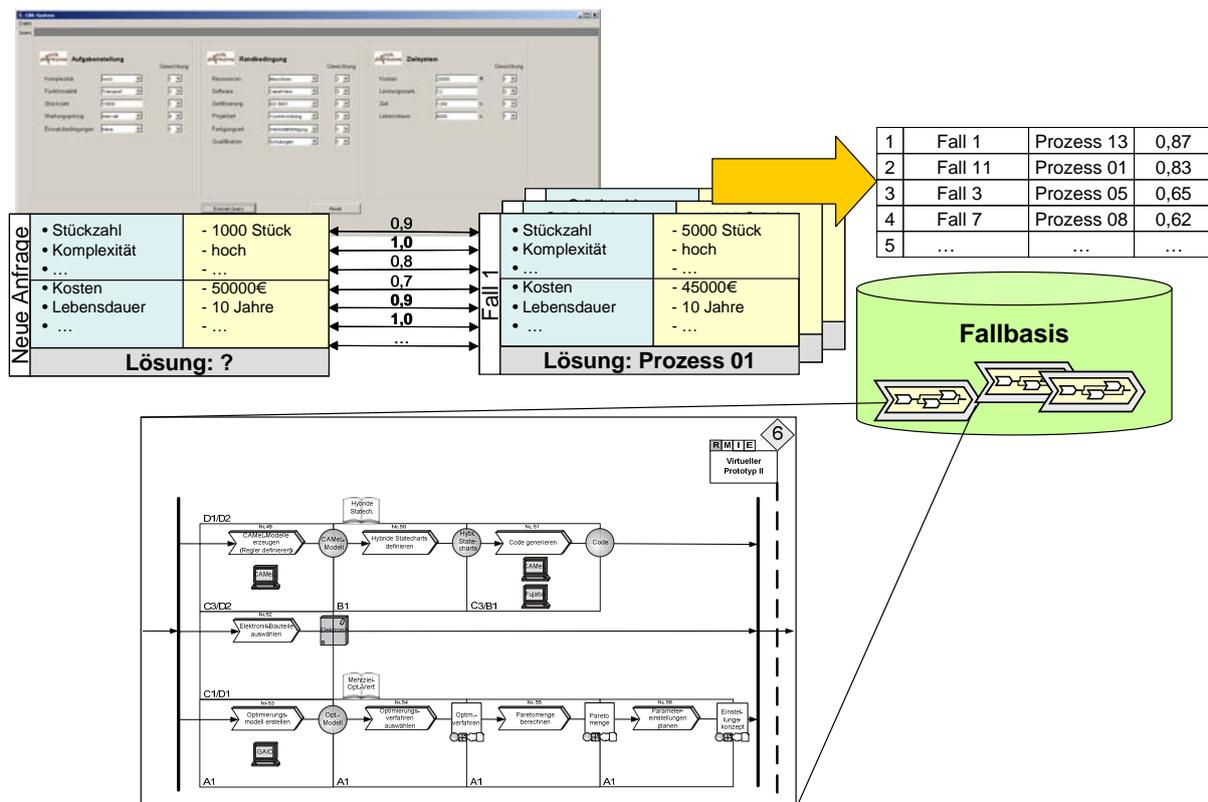


Bild 1: Prinzip der Fallbasis für erprobte Entwicklungsprozesse

Zu Beginn der Entwicklung komplexer Systeme sind in der Regel noch nicht alle Merkmale des Systems bekannt. Es findet ein mehrstufiges Verfahren statt. Zunächst werden die frühen Phasen der Entwicklung (Planen und Klären der Aufgabe, Konzipierung) detailliert durch einen ausgeprägten Entwicklungsprozess und die weiteren Phasen der Ausarbeitung nur durch die zu durchlaufenden Phasen und Meilensteine beschrieben. Damit ist zwar eine grobe Planung des Gesamtvorhabens (Kosten, Dauer) möglich, die Beschreibung der späten Phasen ist jedoch nicht detailliert genug, um sie auf dieser Basis zu durchlaufen, da die zu erbringenden Aktivitäten und benötigten Ressourcen nicht beschrieben sind. Ausgehend von der Prinziplösung, mit der bei komplexen Systemen die Aufteilung des Produktes in parallel zu entwickelnde Module erfolgt, können detaillierte Suchanfragen an Fallbasis gestellt werden. Die Prozessplanung wird revolvierend durchlaufen, indem die Merkmale zur Anfrage an die Fallbasis entweder für die späten Phasen des Gesamtprozesses oder für die Entwicklungsstränge der Module schrittweise konkretisiert werden. Sie werden aus den Anforderungen abgeleitet. Wenn für diese Module bereits Entwicklungsprozesse bzw. Teilstränge bekannt sind, können diese in den Gesamtprozess integriert werden (Bild 2).

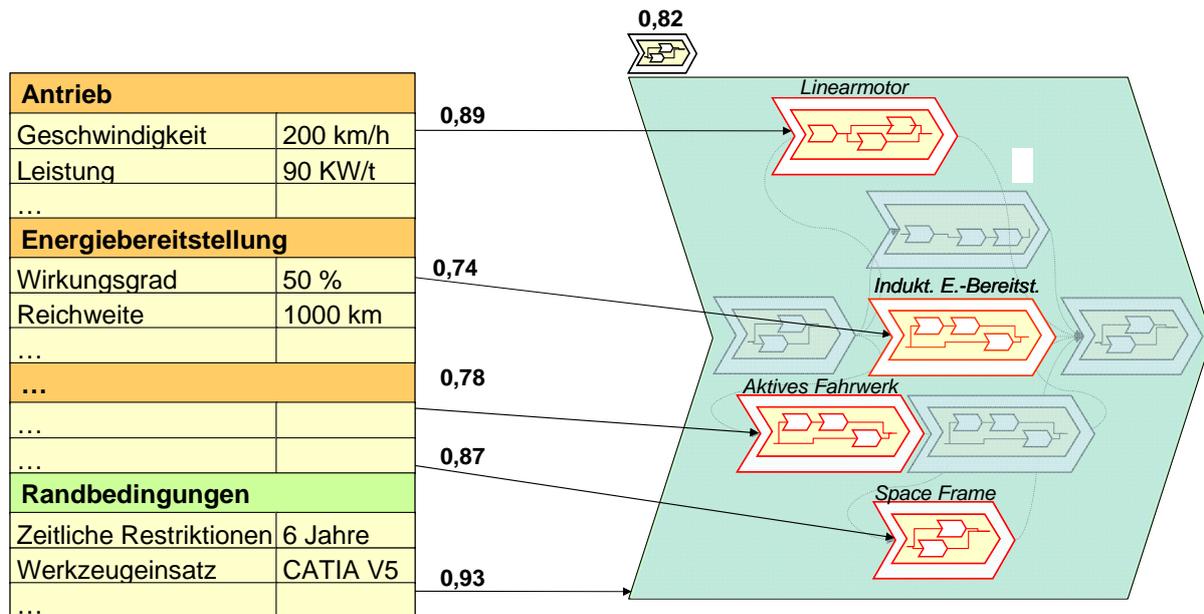


Bild 2: Detaillieren des Entwicklungsprozesses durch Integration von Teilsträngen

1.2 Prozessbeschreibung mit OMEGA

Zur anschaulichen Beschreibung der Entwicklungsprozesse wird die Methode OMEGA verwendet. OMEGA wurde am Lehrstuhl für Rechnerintegrierte Produktion entwickelt und bereits in vielen Industrieprojekten erfolgreich eingesetzt. OMEGA wurde anfangs zur Beschreibung von Geschäftsprozessen verwendet. Die Beschreibung von Entwicklungsprozessen für komplexe mechatronische und selbstoptimierende Systeme stellt jedoch weitere Anforderungen an die Prozessmodellierungsmethode. Daher war zunächst die Erweiterung der Syntax und Semantik von OMEGA erforderlich. Im Folgenden wird die Syntax erklärt, die sowohl alte [3], als auch neue Konstrukte enthält [4]. Bild 3 zeigt einen Ausschnitt aus einem mit OMEGA beschriebenen Entwicklungsprozess.

Die einzelnen Prozessschritte werden in ihrer logischen Reihenfolge von links nach rechts angeordnet und durch einen Pfeil dargestellt. Untereinander angeordnete Prozessschritte stellen eine parallele Abarbeitung dar. Einen solchen Prozessschritt stellt "Tragstruktur konstruieren" (siehe Bild 3) dar. Die zugehörige Organisationseinheit, z.B. Entwicklerteam „Mechanik“ wird unten links in dem umschließenden Rechteck vermerkt. Der bei der Abarbeitung erforderliche Papierspeicher "Ablage" wird unterhalb des Prozessschrittes dargestellt und das erforderliche Dokument „Handskizze“ wird auf der Verbindungslinie zwischen der Ressource und dem Prozessschritt angeordnet. Neben dem Papierspeicher können ebenfalls IT-Ressourcen, Betriebsmittel sowie Materialspeicher durch entsprechende Konstrukte dargestellt werden. Links des Prozessschrittes wird das erforderliche Eingangsobjekt, hier "Prinzip-lösung", und rechts als Ergebnis das Ausgangsobjekt "CAD Modell" dargestellt. Bei den so genannten Bearbeitungsobjekten wird zwischen IT-basierten, papierbasierten, mündlichen und gemischten Objekten unterschieden. Wenn zur Abarbeitung eine Methode verwendet wird, kann sie durch das aufgeschlagene Buch dargestellt werden. Die dicken Splitting- und Synchronisations-Linien links und rechts der Prozessschritte drücken die intensive Zusammenarbeit zwischen den Prozessbeteiligten aus, die durch iteratives Vorgehen und gegenseitige Abstimmungen geprägt ist. Das Konstrukt in der Mitte und am rechten Rand symbolisiert einen Meilenstein.

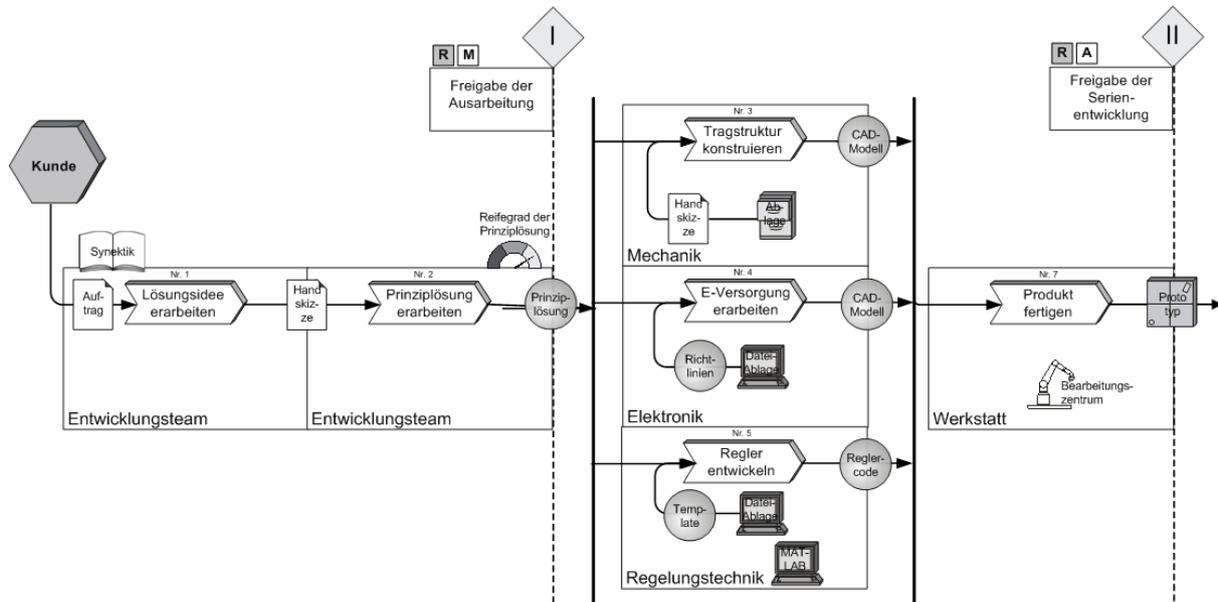


Bild 3: Ausschnitt aus einem Entwicklungsprozess

1.3 Entwurfsmethoden

Die beschriebenen Prozessschritte können durch Zuordnung von Methoden konkretisiert und unterstützt werden. Methoden stellen ein bewährtes Vorgehen zur Lösung einer bestimmten Aufgabe dar, z.B. zur Bewertung von mehreren Alternativen oder, im Falle der Selbstoptimierung, der Planung von Rekonfigurationsvorgängen. Die in mehreren Studien belegte mangelnde Akzeptanz des Methodeneinsatzes in der Praxis ist auf eine schlechte Zugänglichkeit von Methoden sowie unzureichende Beschreibungen und Auswahlhilfen für den Entwickler zurückzuführen [5], [6].

Es wurden eine Reihe bekannter Ansätze untersucht, die die Methoden in ganz unterschiedlicher Form aufbereiten und zur Verfügung stellen. Dabei wurden sowohl theoretische Ansätze [7], [8], [9], als auch praktische Implementierungen [10], [11] untersucht. Auf Grundlage der untersuchten Ansätze wurde ein Methodenbaukasten erstellt, der Methoden zum Entwurf s.o. Systeme aufnimmt und aufbereitet. Aus den untersuchten Ansätzen wurden Kernpunkte übernommen. Müller ordnet beispielsweise die Methoden in einer Zuordnungsmatrix spaltenweise Entwurfsobjekten (z.B.: Funktionsstruktur, Anforderungen, etc.) und zeilenweise objektbezogenen Arbeitsprogrammen (z.B. suchen, präzisieren, formulieren) zu. Die Zuordnung der Methoden zu Entwurfsobjekten wurde aus diesem Ansatz übernommen. Es besteht darüber hinaus ein enger Zusammenhang zwischen den Entwurfsobjekten, den Arbeitsprogrammen und den in der Prozessbeschreibung genannten Entwurfsschritten. Der Ansatz von Ambrosy, Elementartätigkeiten zur Strukturierung zu verwenden, kann an dieser Stelle zur Zuordnung der Methoden zu Entwurfsschritten genutzt werden. Er stellt den Bezug zur Prozessmodellierung mit OMEGA und den spezifischen Prozessschritten her.

Ausgehend von den beschriebenen Entwicklungsprozessen wurden die einzelnen Prozessschritte analysiert und entsprechende Entwurfsobjekte (zu erzeugende Dokumente) extrahiert. Ebenso wurden Elementartätigkeiten definiert. Diese stammen ebenfalls sowohl aus den Prozessschritten als auch aus den im Vorfeld untersuchten Ansätzen. Elementartätigkeiten sind beispielsweise "modellieren", "suchen", "kombinieren", etc. Analog zu [12], [13] wurde ein Methodenmodell erstellt, das die Klassifizierungsmerkmale der Methoden (Entwurfsobjekt, Elementartätigkeit, Input, Output, Rahmenbedingungen Werkzeuge, Erklärung) anschaulich darstellt (Bild 4). Parallel zu diesen Arbeiten wurde eine Reihe von bekannten Methoden zusammengetragen. Diese wurden den untersuchten Ansätzen und der Literatur ent-

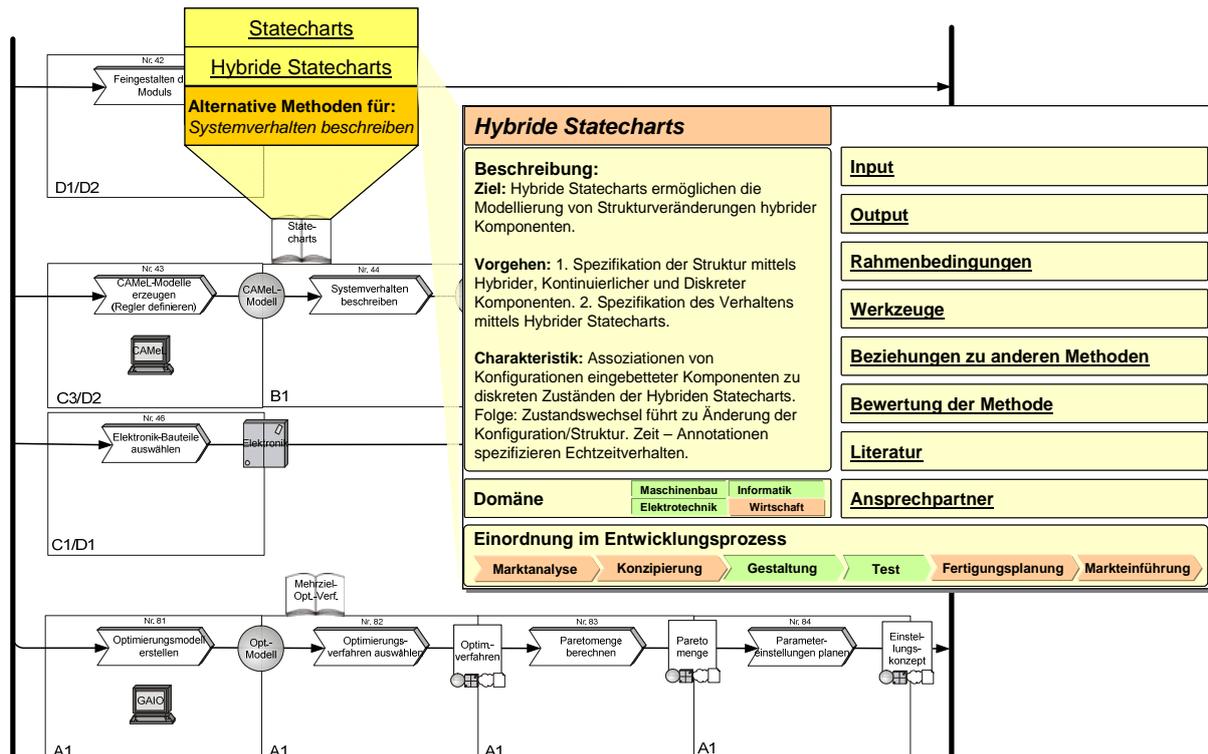


Bild 5: Verknüpfung der Prozessmodellierung mit dem Methodenbaukasten

2 Zusammenfassung und Ausblick

Die optimale Unterstützung des Prozessplaners und Entwicklers stellt eine große Herausforderung dar, da die Vorgaben auf der einen Seite nicht zu strikt sein dürfen, auf der anderen Seite jedoch den nötigen Rahmen vorgeben müssen. Hierzu stellt die Fallbasis in Kombination mit dem Methodenbaukasten ein geeignetes Instrumentarium dar. Zunächst kann mit Hilfe der Fallbasis die Planung des Entwicklungsprozesses vorgenommen werden. Die Zuordnung von Methoden zu Prozessschritten leitet den Entwickler bei der Durchführung zusätzlich und verbessert die Anwendbarkeit der Prozesse.

Im Rahmen des SFB 614 „Selbstoptimierende Systeme des Maschinenbaus“ sind beide Werkzeuge in eine Entwurfsumgebung eingebunden. Eine prozesstechnische Integration auf Grundlage der in den einzelnen Prozessschritten erzeugten Dokumente steuert den Prozessablauf. Dieses Konzept wird ausgebaut: Durch Methoden der Prozesssynthese und -evolution sollen die Entwicklungsprozesse zukünftig teilautomatisch generiert sowie zur Laufzeit an wechselnde Rahmenbedingungen und Kenntnisstände angepasst werden. Ebenso soll die Zuordnung von Methoden automatisiert erfolgen, wobei auf Grundlage der Methodeigenschaften, wie Input und Output, eine durchgängige und schlüssige Methodenkette entsteht.

3 Literatur

- [1] Gausemeier, J.; Frank, U.; Redenius, A.; Steffen, D.: Development of Self-Optimizing Systems. MechRob 2004 (IEEE), Mechatronics & Robotics 2004, 13.-15. September 2004
- [2] Aamodt, A.; Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations and Systems Approaches. AI Communications, IOS Press, Vol.7, S. 35-59, 1994

- [3] Fahrwinkel, U.: Methode zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des Business Process Reengineering. Dissertation im Fachbereich Maschinentechnik, Universität Paderborn, HNI Verlagsschriftenreihe Bd.1, Paderborn, 1995
- [4] Gausemeier, J.; Michels, J. S.; Redenius, A.: Modellierung und Planung von Produktentstehungsprozessen. In: Gausemeier, J.; Wallaschek, J. (Hrsg.): 2. Paderborner Workshop Intelligente mechatronische Systeme, 25.-26. März 2004, S. 33-42, Heinz Nixdorf Institut, Universität Paderborn, 2004
- [5] Berliner Kreis - Wissenschaftliches Forum für Produktentwicklung e.V. (Hrsg.): Kurzbericht über die Untersuchung Neue Wege zur Produktentwicklung. 2. Aufl., Paderborn, 1998
- [6] Grabowski, H.; Geiger, K. (Hrsg.): Neue Wege zur Produktentwicklung. Raabe Verlag, Stuttgart, 1997
- [7] Müller, J.: Arbeitsmethoden der Technikwissenschaften – Systemtechnik, Heuristik, Kreativität. Springer-Verlag, Berlin, 1990
- [8] Ambrosy, S.: Methoden und Werkzeuge für die integrierte Produktentwicklung. Dissertation an der Fakultät für Maschinenwesen, TU München, München, 1996
- [9] Zanker, W.: Situative Anpassung und Neukombination von Entwicklungsmethoden. Dissertation an der Fakultät für Maschinenwesen, TU München, München, 1999
- [10] Grabowski, H.; Paral, T.: Integrierter Methodeneinsatz im Produktinnovationsprozess. In: VDI-Z 10/2001, Springer Verlag, Berlin, 2001
- [11] Ganzheitliche Innovationsprozesse in modularen Unternehmensnetzwerken, www.gina-net.de, Stand: September 2004
- [12] Birkhofer, H.; Kloberdanz, H.; Berger, B.; Sauer, T.: Cleaning Up Design Methods – Describing Methods Completely and Standardised. Proceedings of the 7th International Design Conference DESIGN 2002, May 14-17, 2002, Dubrovnik Croatia, Vol. 1, Zagreb, 2002
- [13] Lindemann, U.: Flexible Adaptation of Methods within the Design Process. Proceedings of the 7th International Design Conference DESIGN 2002, May 14-17, 2002, Dubrovnik - Croatia, Vol. 1, Zagreb, 2002

Dipl.-Wirt.-Ing. Alexander Redenius
Dipl.-Wirt.-Ing. Daniel Steffen
Heinz Nixdorf Institute
Universität Paderborn
Fürstenallee 11, D-33102 Paderborn
Tel: +49-5251-60-6497
Fax: +49-5251-60-6268
Email: alexander.redenius@hni.uni-paderborn.de
daniel.steffen@hni.uni-paderborn.de
URL: <http://www.hni.upb.de/rfp>

