

Metamodellbasiertes Konzeptdesign

Dr. Boris Petkoff
AccordSystems[®], Berlin
(petkoff@t-online.de)

Metamodellbasiertes Konzeptdesign

Produktentwicklungen unterliegen heute einem permanenten Innovationszwang. Um im Wettbewerb bestehen zu können, werden die Standardisierungs- und Variationsanforderungen an die Unternehmen immer größer. Kosten müssen reduziert, Entwicklungszeiten verkürzt und immer mehr kundenspezifische Varianten angeboten werden. Der Paradigmenwechsel von klassischen Engineering-Prinzipien hin zum Prozesskettenorientierten virtuellen Produktdesign in digitalen Fabriken ist einer der erfolgversprechenden Wege, diesen Anforderungen des Marktes gerecht zu werden. Die weitere Verkürzung der Entwicklungszeiten für neue Fahrzeuge gehört zu den wichtigen und dringenden Aufgaben der Automobilindustrie. Produkte mit innovativer Technik, gepaart mit ressourcenschonender Herstellung in modernen Werken erfordern die intensivere Nutzung analytischer und mathematischer Methoden. Bereits heute simulieren Ingenieure und Techniker auf Hochleistungsrechnern unter Anwendung spezieller Software neue Fahrzeuge und verbesserte Fertigungsprozesse. Geplante Fahrzeugkomponenten werden unter bisher unbekanntem Bedingungen digital getestet und absolvieren erste Bewährungsproben bereits vor Anlauf der Prototypenfertigung. Komplexe Montagevorgänge lassen sich heute dreidimensional darstellen und Herstellungsverfahren am Bildschirm optimieren und erproben. Die zunehmend simultane Zusammenarbeit aller Bereiche verkürzt die Entwicklungszeit entscheidend, steigert die optimale Nutzung von Prototypen und verbessert die Qualität.

1. Problemstellung und Motivation

1.1 Projektzyklen

Produzierende Unternehmen werden heute zunehmend von den Forderungen nach kurzen Produktentwicklungszeiten bei höchster Qualität bestimmt. Die sichere Beherrschung vorhandener Prozesse sowie die schnelle Integration neuer Fertigungsverfahren sind dabei entscheidende Erfolgsfaktoren. Nur durch eine intensive Betrachtung der Technologien und Methoden können Zielsetzungen wie die Verkürzung der Produktentwicklungszeit oder die Steigerung der Effektivität der Fertigungsprozesse verwirklicht werden. Das ökonomische Umfeld in dem Unternehmen gegenwärtig agieren wird vielfach als „turbulent“ bezeichnet. Gemeint sind damit unter anderem die zunehmende Verkürzung der *Time-to-Customer*, die steigenden Anforderungen der Kunden bezüglich Individualisierung sowie die rasante Entwicklung einiger Technologien, die direkt auf die verschiedenen Produktentstehungsprozesse rückwirken.

Die zentralen Aufgaben der Fabrikplanung und Logistik sind deshalb zum einen die Konzeption wandlungsfähiger Fabriken, welche die neuen Anforderungen hinsichtlich Reaktionsfähigkeit und Anpaßbarkeit an Umfeldveränderungen erfüllen. Zum anderen besteht die Herausforderung, innerhalb des Unternehmens den logistischen Zielsetzungen eine noch stärkere Bedeutung beizumessen. Hierbei sind Termintreue und Lieferzeit neben den klassischen Faktoren wie Preis und Produktqualität die entscheidenden Kriterien, die durch eine flexible und reaktionsfähige Auftragsabwicklung erreicht werden sollen.

Die heute geforderte Flexibilität bei Planung, Entwicklung und Umsetzung von Produktionstechnik kann nur erreicht werden, wenn alle Prozesse durchgängig digitalisiert und miteinander verknüpft werden. Dies kann anhand von Szenarien, die einen realistischen Vorgriff auf die Zukunft erlauben, geschehen, da diese den Entscheider in die Lage versetzen, viele unterschiedliche Faktoren und ihre Beziehung miteinander darzustellen. Die Kommunikationsplattform *Digitale Fabrik* versucht daher, eine Infrastruktur zur Unterstützung des gesamten Planungs- und Entwicklungsprozesses aufzubauen. Der Paradigmenwechsel von klassischen Engineering-Prinzipien hin zu Prozessketten-orientiertem *Virtual-Product-Design* ist der einzige Weg, diesen Anforderungen des Marktes gerecht zu werden. Ziel eines adäquaten Ansatzes muß es daher sein, die Grundideen und Konzepte eines generischen Bewertungs- und Auswertungssystems zur Entscheidungsunterstützung, sowohl für Innovations- wie auch für Risikopotentiale zu definieren.

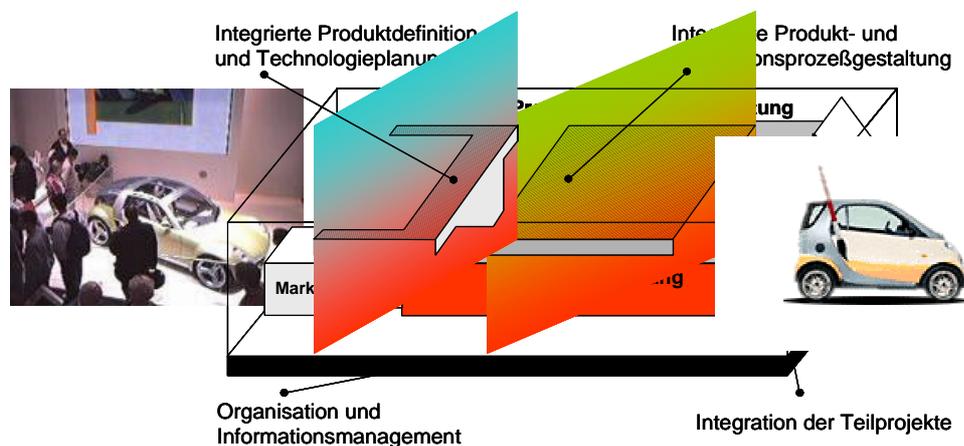


Abbildung 1: Abstrakter Projektzyklus

Die Graphik der Abbildung 1 zeigt, stark abstrahiert, die vier Grundelemente eines Projektzyklus zur Definition eines neuen Produkts. Ausgehend von einer Markt-

beobachtung mit zugehöriger Abschätzung des zu erwartenden Absatzes (die Graphik deutet hier im linken Bildbereich eine Prototypstudie an) wird auf Basis der verfügbaren Technik ein Produktionsprozeß gestaltet. Dieser orientiert sich neben der einzusetzenden Technik auch an der Gestaltung des entgeltigen Produktes. Neben technischen Randbedingungen spielen hier vor allem Zielgruppenorientierung und kundenspezifische Konfigurierbarkeit eine Rolle. Einen weiteren Planungsschritt bildet die Integration der verschiedenen Anforderungen und Möglichkeiten unter Berücksichtigung der organisatorischen Gegebenheiten. Besonderes Augenmerk liegt auch hier auf den IT-Anteilen der Informationslogistik, hierunter fallen insbesondere die zeitnahe Versorgung mit aktuellen Daten sowie die permanente Aggregation produktionsrelevanter Informationen.

Eine Erhebung des Vereins Deutscher Ingenieure verdeutlicht den Zeit- und Kostenvorteil durch den Einsatz von *Datenintelligenz* innerhalb der Prozeßkette. Gerade bei der Bearbeitung und der Variantenbildung spiegeln sich die Vorteile „intelligenter“ Teile als entscheidender Zeitfaktor wieder. Deshalb müssen bereits alle Standard- und Katalogteile „Intelligenz“ in Form beschreibender Daten (sog. Metadaten) vorweisen, um die Fähigkeit zu besitzen, sich bei konstruktiven Eingriffen idealerweise selbsttätig anzupassen. Der klassische Zeichnungs-orientierte Begriff der Geometrie erweitert sich daher zum *Feature*, welches neben der reinen Formgebung Semantik integriert, die sich auf alle Prozeßabschnitte beziehen kann.

Bereits in der Anfangsphase ist der direkte Nutzen solcher Komponenten weitaus höher als bei konventionellen Bibliotheks-Systemen welche ausschließlich Geometrien vorgefertigter Komponenten versammeln.

Dennoch entspricht der skizzierte Vorteil nur einem Bruchteil des Einsatzpotentials, das in späteren Entwicklungsphasen weiter ansteigt, wie z.B.

- Design neuer Fertigungsstätten (Beispiel: Smart-Fertigung in Hambach)
- Dokumentation von Entscheidungsprozessen
- Integration mit existierenden Produktivsystemen
- Bereitstellung von Eingangswerten für Simulationsprozesse
- Erweiterte parametrische Konstruktion auf der Basis von *Features*
- Codeerzeugung für Simulationssysteme, Formal: Template-basierte Instanziierung des Metamodells

Die Struktur der heutigen Automobilzuliefererkette besteht im wesentlichen aus vier Stufen. Auf oberster Ebene sind die *Original Equipment Manufacturer* (OEM) angesiedelt, welche das Endprodukt „Fahrzeug“ herstellen. In den nachfolgenden Stufen sind die Zulieferer angesiedelt, die in unterschiedlicher Komplexität Systeme, Baugruppen und Teile herstellen. Dabei bestehen Beziehungen sowohl zwischen den OEMs und den Zulieferern als auch zwischen den Zulieferern

untereinander. Ziel der Kommunikationsplattform *Digitale Fabrik* ist der Aufbau einer geschlossenen Wertschöpfungskette „Karosserie, Interieur und Exterieur“ mit globaler Ausrichtung sowie Integration der Informations- und Kommunikationstechnologien in allen Prozessen. Weltweit vergleichbar wettbewerbsfähiges Know-how existiert nur in der Flugzeugindustrie wie z.B. im Projekt SAVE - *Simulation Assessment Validation Environment* [Sav99].



Abbildung 2: Projekt SAVE

Die in diesem Zusammenhang zu beobachtenden globalen Entwicklungen sind gekennzeichnet durch:

- Einführung von Informationstechnologie in allen Bereichen der Wertschöpfungskette insbesondere der Übergang zum *Simultaneous Engineering* sowie verstärkte Einbeziehung von Zulieferern in die Entwicklungsprozesse
- Entwicklung und Einsatz von neuen Werkstoffen, Technologien und Prozessen
- damit verbunden Veränderungen in der Struktur von Wertschöpfungsketten
- Zunehmende Lieferantunabhängigkeit und verbreiterte Einkaufsmöglichkeiten (sog. *Global Sourcing*)
- Plattform- und Gleichteilestrategie bei Fahrzeugherstellern
- Modul- und Systemtechnik
- sinkende Fertigungstiefe der OEMs (unter 30%)
- Zwang zur Kostensenkung in allen Bereichen
- Öko-Orientierung in Politik, Wirtschaft und Gesellschaft, damit verbundene Forderungen an umweltorientierte Entwicklung bzgl. Herstellung und Betrieb sowie Lebenszykluskonzepte

SAVE – Simulation Assessment Validation Environment								
	CAD	Factory Simulation	Assembly Planning	Schedule Simulation	Risk Analysis	Cost Analysis	Variation Analysis	Enterprise Optimization
Process Plan WorkInst		X	X	X	X	X	X	X
Geometric Definition	X		X		X	X	X	
Task Duration		X	X		X			
Resource Estimates		X		X				X
Rates and Factors		X	X	X	X	X		
Process Rates		X	X					
Factory layout	X	X		X				
Manufacturing Rules						X	X	X
Time Lines				X				X
Future Definition						X	X	
Cost	X				X	X		
Tolerance Limits					X		X	X
Risk					X		X	X

Die Schwerpunkte des Projektes SAVE zeigen, daß eine Wirtschaftlichkeitsanalyse (*value proposition*) praktisch unmöglich für verschiedene Szenarien ist, da zugrundeliegende Daten nicht einheitlich und durchgängig verfügbar sind.

Ein kombinierter Metaansatz

Ein adäquates System sollte generisch durch die Sichtenunabhängigkeit auf eine Problemdomäne sein, d.h., es sollte die durch eine Menge von Eigenschaften beschriebenen Elemente eines beliebigen Typs bewerten und auswerten. Die Auswertung soll durch in der Praxis bewährte Verfahren erfolgen, die ständig erweitert werden können, z.B. mathematische Berechnungen, logische Inferenzen, etc. Das System soll dem Anwender folgende Dienste zur Verfügung stellen:

1. **Definition von Objektontologien:** Produkt, Fabrik, Prozeß, Technologie, Material, Qualitätskriterien, Wirtschaftskriterien, Kundenanforderungen etc.
2. **Beschreibung von Objektsemantik** für die Auswertungsschritte.
3. **Definition von Vergleichsverfahren** für Objekte.
4. **Durchführung von Bewertungen** und Auswertungen basierend auf Objektontologien.
5. **Bereitstellung von Suchmechanismen** zur Auswertung und Bewertung.
6. **Definition von Metamodellen** als Zusammenführung der modellierten Objektontologien.
7. **Generierung von ablauffähigen Applikationen** aus den Metamodellen.

Die Zielsetzung kann damit als Anwendung der Ideen des Wissensmanagements umrissen werden. Diese sollen praktisch angewendet werden, um automatisiert ausführbare Applikationen (beispielsweise in einer konkreten Programmiersprache) generieren zu können. Hierdurch soll die Applikationserstellung soweit automatisiert werden, daß für die Propagierung von Änderungen am Informationsmodell in Applikationsstrukturen kein zusätzlicher Programmieraufwand erforderlich ist.¹

¹ Die Argumentation der Vorteile eines Systems, welches nicht die fest kodierte Darstellung eines Fachgebietes erfordert, sondern die konkrete Ausgestaltung durch Modelle den Experten überläßt, und so auch ihre Erfahrung miteinbezieht, ist in [Pet98] zusammengefaßt.

2 Technologische Aspekte

2.1 Ontologien

Die Modellierung von Ontologien als fachliche Komponenten und der dazugehörigen Referenzprozesse scheint eine Kommunikationsbasis zur Beschreibung und Konfiguration wissensbasierter Anwendungssysteme zu sein. Die Beschreibung zugehöriger Problemlösungsprozesse nimmt dabei eine besondere Stellung ein, da diese spezifizieren, auf welche Weise Objekte zur Erreichung eines Zieles kooperieren und kommunizieren. Standardisierte *Common Objects* und *Domain Objects* könnten die Basis von betriebswirtschaftlichen Anwendungen bilden. [Pet97]

Bei der modernen Ontologie handelt es sich um eine explizite und formalsprachliche Spezifikation einer gemeinsam verwendeten Konzeptualisierung von Phänomenen der Realität. Daher ist auch das „moderne“ Ontologieverständnis der Forschung aus dem Bereich der künstlichen Intelligenz eine pragmatische Verzahnung von ontologischer und epistemologischer Perspektive: Der epistemologisch „naive“ Standpunkt einer Realität „an sich“ – unabhängig von Verzerrungen durch das erkennende Subjekt – ist überwunden. Statt dessen wird davon ausgegangen, daß Realitätsausschnitte und die darin beobachteten Phänomene nur als Resultate von aktiven Erkenntnisleistungen wahrgenommen werden können, die vom erkennenden Subjekt zur Erfüllung der von ihm verfolgten Erkenntniszwecke vollbracht werden.

Modellierung im Sinne einer Ontologie ist genau die abstrakte Sichtweise auf Phänomene eines Realitätsausschnitts, um daraus Strukturen herauszulesen oder Strukturen hineinzuzinterpretieren, die für vorgegebene Erkenntniszwecke von Interesse sind. Diese Erkenntniszwecke bestimmen, welche Aspekte der wahrgenommenen Phänomene für die erkennenden Subjekte relevant sind. Konzeptualisierung bedeutet daher immer zweck- und subjektabhängige Auszeichnung relevanter Realitätsaspekte. Das Ergebnis eines Konzeptualisierungsprozesses stellen die *Konzepte* dar, mit denen der betrachtete Realitätsausschnitt hinsichtlich seiner für relevant erachteten Aspekte vorstrukturiert wird.

Unter einem Metamodell wird die Spezifikation des Termvorrats und der Syntax einer formalen Sprache verstanden, die ihrerseits zur formalsprachlichen Modellierung von möglichen Welten dient. Entsprechend zum Mehrstufenkonzept der sprachlichen Welterfassung werden Modelle, die einen Realitätsausschnitt repräsentieren, als Objektmodelle bezeichnet, während ein Metamodell die Gesamtheit aller Objektmodelle spezifiziert, die sich mittels einer vorgegebenen formalen Modellierungssprache auf der Objektebene ausdrücken lassen. Diese Spezifikation besteht einerseits aus der Festlegung aller formalsprachlichen Ausdrücke (*Terme*), aus denen ein Objektmodell aufgebaut werden kann, und andererseits aus der Festlegung aller formalsprachlichen zulässigen Ausdrucksverknüpfungen (*Syn-*

tax). Der Begriff des Metamodells läßt es offen, ob die Spezifikation des Termvorrats und der Syntax einer formalen Modellierungssprache entweder auf graphische und/oder formalsprachliche Weise erfolgt. Mit anderen Worten liefern Ontologien einheitliche Begriffswelten und sind Grundlagen von Metamodellen, die diese Begriffswelt aufgreifen und formalisieren.

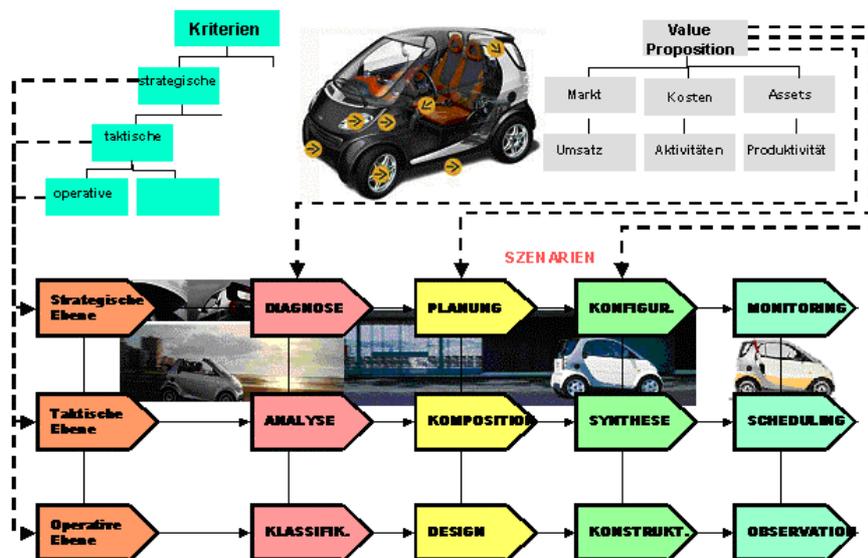


Abbildung 3: Konzeptuelle Sichten und Phasen der Szenarien

2.2 Konzeptuelle Modelle, Metamodelle und Modelltransformationen

Besondere Betrachtung innerhalb der Entwurfs- und Evaluierungsphase neuer Produktions- und Fertigungskonzepte verdient die Darstellung der statischen Informationsstrukturen, welche die Basis aller zeitlich nachgelagerten Aktivitäten bilden.

In den frühen Konzept-Entwicklungsphasen muß die weitestmögliche Paradigmenneutralität als oberstes Ziel angesehen werden. Dies bedeutet in der Praxis die Vermeidung implementationsspezifischer Details (etwa Klassenbibliotheken, Frameworks aber auch gewählte Umsetzungssprache) zugunsten einer neutralen Sichtweise auf das zu lösende Problem selbst. Intuitiv wird hieraus klar, daß diese Forderung die Einsatzmöglichkeit von Beschreibungssprachen logischer Modelle, wie der Unified Modeling Language [UML01], negiert, da diese inhärent bereits eine implementierungsspezifische Sichtweise zu Grunde legen. Überdies legen objektorientierte Vorgehensweisen im allgemeinen komplexe Entwicklungspro-

zesses zugrunde, die einerseits sehr stark auf den Entwurf von Softwaresystemen focussieren, und andererseits nicht alle Projektphasen in gleicher Weise unterstützen (Vgl. u.a. [Kru99, Oes01]).

Daher erweist sich zur Erfassung, Validierung und Weiterverarbeitung der Informationsstrukturen die Orientierung an der klassischen drei-Schema-Architektur [Nij76] als am erfolgversprechendsten. Sie gliedert die im Entwicklungsprozeß verwendeten Modelle in *konzeptuelle*, *logische* und *physische*. Zusätzlich Hierarchisiert Nijssen die Schemata in der angegebenen Reihenfolge. Jeder Modelltyp erfüllt eine ihm zugewiesene Aufgabe, und stellt sein Ergebnis -- das Modell -- der übergeordneten Ebene zur Verfügung. Während sich das konzeptuelle Modell auf die hinsichtlich Paradigmen und Ausführungsmodell neutrale Erfassung der Problemstrukturen konzentriert, weist das logische Modell (bekannte Beispiele hierfür sind das relationale und das objektorientierte Modell) schon auf die gewählte Implementierung hin. Wesentliches Kennzeichen des logischen Modells ist die Paradigmenpezifität. Sie abstrahiert jedoch gleichzeitig von der in einem späteren Entwicklungsschritt zu treffenden Festlegung auf eine konkrete Implementierung. Eine der bekanntesten und meistverbreiteten Modellierungssprachen konzeptueller Modelle stellt sicherlich das *Entity-Relationship-Modell* nach Chen [Che76] dar. Ausgehend von dieser Notation wurden im Laufe der Jahre eine Reihe verschiedenster Erweiterungen vorgeschlagen. Einen jüngeren Vorschlag, der sich neben der technischen Sichtweise auch die Integration der Domänenexperten zum Ziel setzt, markiert das *Semantically Enriched Extended Entity-Relationship Model* [Jec00]. Zwischen den drei erwähnten Modelltypen wird der Übergang idealerweise durch automatisierte Algorithmen (etwa: [KJ98, Jec00]) werkzeuguunterstützt vollzogen. Nur so kann die Modelltransformation informationserhaltend erfolgen.

Neben der Erfassung und Darstellung problemspezifischer Informationsstrukturen ist in der jüngeren Vergangenheit der Ansatz der generischen Abbildung eines Problemfeldes vermehrt in den Blickpunkt des Interesses gerückt. Hierbei wird versucht allgemeingültige Informationsstrukturen zu erheben, die in einer ganzen Klasse von Problemfällen anzutreffen sind. Die so gefundenen und dokumentierten Muster können daher in verschiedenen konkreten Anwendungsfällen zum Einsatz gebracht werden. Naturgemäß variiert jedoch die durch solche Modelle verwendete Begrifflichkeit teilweise massiv von der Terminologie der konkreten Problemstellung. Daher wird -- analog der in der drei-Schichten-Architektur angewandten Vorgehensweise -- das generische Modell als zusätzliche Modellierungsebene aufgefaßt und dient als Modell eines konkreten Modells. Mithin wird ein konkretes Modell durch ein abstrahierteres modelliert. Die sich hieraus ergebenden Modelle-von-Modellen werden in der Literatur allgemein mit dem Begriff *Metamodelle* [Jeck00] belegt.

Metamodelle ermöglichen die Definition generischer Vorgehensweisen, die später auf den konkreten Modellen (d.h. den Ausprägungen des Metamodells) angewandt werden können. Insbesondere lassen sich so identische Interpretationen auf ver-

schiedene Modelle anwenden, ohne die Prinzipien anpassen zu müssen, was u. U. zu Verfälschungen des Ergebnisses führen könnte.

2.3 Neutrale Datendarstellung

Eine wesentliche Rolle im betrachteten Problemkontext kommt der physischen Repräsentation der beteiligten Daten zu. Die geringe Eignung jedes beliebigen nativen Formates der am Prozeß beteiligten Systeme liegt auf der Hand -- wären doch zeitaufwendige und in der Regel fehlerträchtige Konversionen aus den „anderen“ Formaten die Folge.

Die Lösung kann daher nur in einer neutralen Darstellung liegen, welche fähig ist, beliebige Informationen auszudrücken ohne ein bestimmtes Datenhaltungsparadigma zu Grunde zu legen oder ein Ausführungsmodell zu implizieren.

Als Ansatz bietet sich daher die generische Auszeichnungssprache *Extensible Markup Language* (XML) [BPS+00] in natürlicher Weise an. Als Unicode-basiertes Textformat erlaubt sie die vergleichsweise einfache Verarbeitung durch beliebige Rechnersysteme und Plattformen. Sie eignet sich daher sowohl zur Darstellung der Modelldaten, als auch zum Transfer der Simulationseingangsdaten aus den Wirksystemen.

Einen wesentlichen Standard aus der XML-Familie stellt das durch die OMG standardisierte *XML Metadata Interchange* Format XMI [XMI99] dar, welches es gestattet, Modellinformation werkzeugunabhängig darzustellen. Hierdurch kann neben einer -- schon allein aus wirtschaftlichen Gründen gewünschten -- Unabhängigkeit vom Werkzeughersteller auch ein wesentlicher Freiheitsgrad hinsichtlich der möglichen Weiterverarbeitung realisiert werden. Durch die XML-Sprache der *XML Transformations* [XSLT99] ist ein vergleichsweise einfacher Weg zur Transformation der neutralen Modellinformation in physische Zielformate definiert. Hierzu zählt insbesondere die Code-Generierung zum Customizing von Standardsystemen, aber auch die schnelle Erzeugung ausführbarer Prototypen zur frühen Konzeptvalidierung.

Zusätzlich erlaubt diese Technologiekombination auch die Erzeugung eigener XML-Sprachen für beliebige Problemstellungen. Wird hierzu -- wie in [Jec00a, Jec01b] beschrieben -- dasselbe Datenmodell verwendet, welches auch zur System-Umsetzung herangezogen wird, so kann in dieser Prozeßphase der Bruch zwischen Hauptspeicherdatenhaltung und Serialisierungsformat vermieden werden. Auf diesem Wege wird die Übertragbarkeit beliebiger Systeminhalte und -zustände sichergestellt.

3 Realisierungsszenarios

Anwendungssysteme müssen heute als verteilbare Systeme entworfen und realisiert werden. Bestimmte Technologien wie die des Internets, Java-Entwicklungsumgebungen oder die verschiedenen Verteilungstechniken (z.B. CORBA) unterstützen diesen Trend. Selbstverständlich spielt die Technologie eine entscheidende Rolle, d.h., sie ist zwar nicht primär bestimmend, aber vom „state-of-the-art“ und von den im Prozeß umgebenden Unternehmensgegebenheiten abzuleiten.

Als Lösungsansatz konnten sich in der Vergangenheit offene Rahmenwerke (*Framework*) etablieren. Diese bieten die Möglichkeit flexibel zu beliebigen Zeitpunkten neue Systeme in bestehende Prozesslandschaften zu integrieren. Frameworks sind daher nicht nur im Hinblick auf das informationstechnische Architektur- und Anwendungswissen von Bedeutung, sondern auch im Hinblick auf die Zusammensetzung und das Zusammenspiel betriebswirtschaftlich-organisatorischer Abläufe. Hierbei müssen die Abläufe durch geeignete Rahmenwerke eindeutig und durchgängig von der Planung bis zur Implementierung beschrieben werden. Organisationen in turbulenten Märkten benötigen stabile Frameworks, welche die Planung und Steuerung der Organisationsabläufe durchgängig unterstützen und kontinuierlich zur Verbesserung von Produktions- und Geschäftsprozessen beitragen.

Als branchen- und anwendungsübergreifende Inkarnationen solcher Rahmenwerke werden die Ansätze *NEXUS*SoftwareFactory*, *alfabet meta-modelling tool* und die *TREND – Java Intergationsplattform* nachfolgend eingehender betrachtet.

* Die **NEXUS*SoftwareFactory** ermöglicht unter Berücksichtigung der Java-Technologien eine Modell-basierte Anwendungsentwicklung, die auf relationalen und Objektdatenbanksystemen zum Einsatz gebracht werden kann. Aus Struktur- und Geschäftsprozeßinformationen kommerzieller CASE-Tools erzeugt der Anwendungsgenerator *NEXUS*Builder* unter Verwendung von Hilfsinformationen generische grafische Benutzungsoberflächen, die zur Datenerfassung und Datennavigation sofort benutzt werden können.

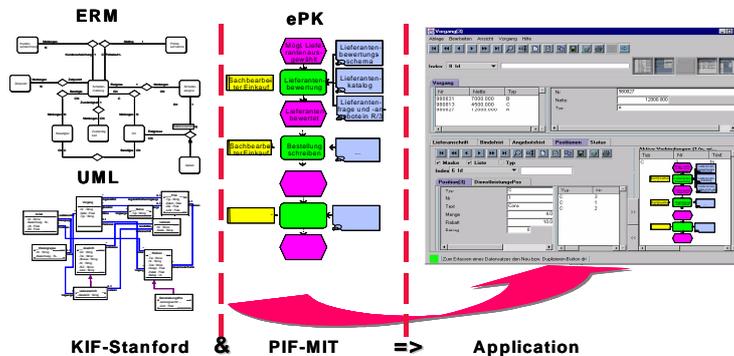


Abbildung 4 : NEXUS*SoftwareFactory-Vorgehen

* **ALFABET** bietet eine C++ Umgebung zur Gestaltung und Visualisierung von Modellen der Organisationsbeschreibung und der operativen Planung bis hin zur Modellierung von Programmabläufen, indem die Integration und dynamische Anpassung unterschiedlicher Modelle durch das Konzept der Metamodellierung realisiert wird. Der Anwender beschreibt „sein“ Metamodell selbst und erhält somit volle Freiheit in der Beschreibung der einzelnen Elemente und deren Beziehungen, die später im konkreten Modell verwendet werden sollen. Hierfür bietet ALFABET dem Anwender eine proprietäre Technologie, (die eine CORBA-Umgebung voraussetzt) für eine breite Palette von Funktionen und Möglichkeiten bei der Definition von GUI-Elementen und der grafischen Visualisierung der Daten.

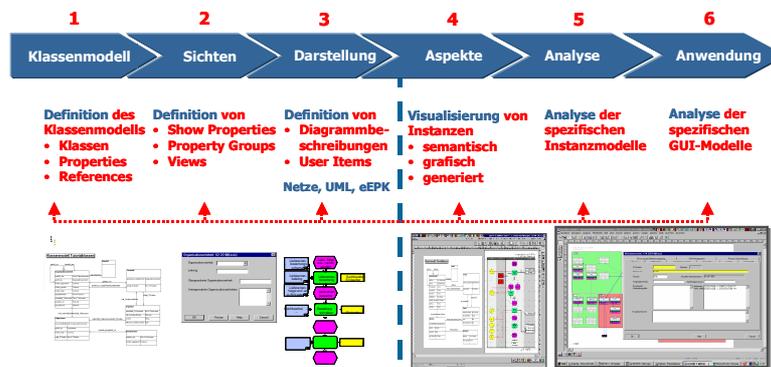


Abbildung 5 : ALFABET-Vorgehen

* **TREND** schafft Brücken zwischen Modell und Implementierung, indem es das erstellte Modell direkt für die Anwendung nutzt. Durch diese Kopplung ist es mit TREND möglich

- ablauffähige Prototypen – Anwendungen direkt aus dem Design-Modell abzuleiten, und zwar ohne Einschränkung der Erweiterbarkeit durch Implementierung
- Konsistenz zwischen entwickeltem Design-Modell und der fertigen Anwendung sicherzustellen
- die Motivation zu erhöhen, sich auf das Design einer Anwendung zu konzentrieren. Dies versucht den klassischen Bruch zwischen Design und Implementierung zu vermeiden.

Die direkt aus dem Design erstellten Anwendungsprototypen können durch schrittweise Verfeinerung des Modells und durch Implementierung bis hin zur fertigen Anwendung verfeinert werden. Die folgende Abbildung zeigt das Designmodell einer Anwendung und TRENDS *DataEntry-Anwendung*, die den direkten Test des Designmodells ohne Implementierungsarbeiten erlaubt.

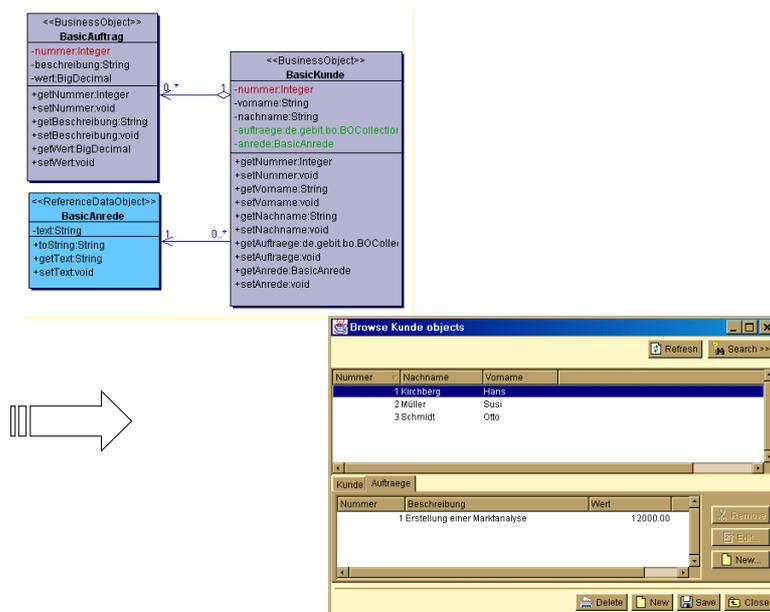


Abbildung 6 : TREND

TREND setzt zur Erreichung dieses Zwecks folgende Mittel ein:

- Voraussetzung ist die Existenz eines in UML modellierten Metamodells. Aufgrund der Komplexität der zu abzubildenden Problemzone scheidet der Einsatz von Modellierungssprachen der logischen Ebene (wie UML) aus naheliegenden Gründen aus; die sprachinhärente Komplexität und Paradigmen-spezifika seien hier nur als zwei offensichtliche Gründe angeführt. Daher sollte im praktischen Einsatz das benötigte logische Modell durch ei-

nen geeigneten Algorithmus aus dem konzeptuellen Schema transformiert werden.

Das resultierende UML-Metamodell kann für die Anwendungsdomäne „Geschäftsanwendungen“ mit den durch die UML vorgesehenen Standardmechanismen erweitert werden um Persistenzverhalten u.ä. abzudecken.

- TREND bietet vorgefertigte sogenannte generische Komponenten. Diese können UML-Modellinformationen zur Laufzeit der Anwendung auswerten. Beispiele für generische Komponenten sind:
 - ein Editor für beliebig geartete Objekte, der zur Laufzeit die zum Editieren eines Objekts notwendigen Komponenten erzeugt
 - ein Rahmen, der den Workflow einer Anwendung gemäß eines Zustandsübergangs-Diagramms steuert
- Zur weiteren Beschleunigung der Prototyp-Erstellung existieren vorgefertigte Schablonen, die über einen Ableitungsmechanismus während des Designs in die Anwendung integriert werden können. Diese Komponenten können später bei Bedarf durch eigene handgeschriebene Komponenten ersetzt werden.

Es stehen Generatoren zur Verfügung, die aus dem UML-Modell Anwendungselemente wie Oberflächenbestandteile oder ein Datenbank-Schema generieren können.

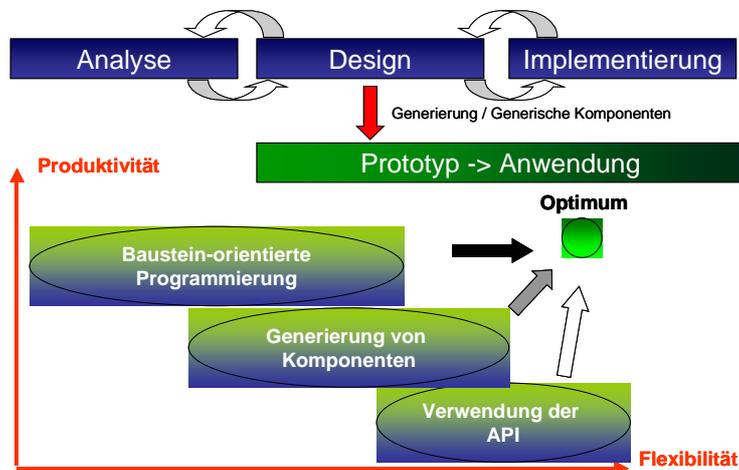


Abbildung 7 : TREND-Vorgehen

Obwohl der Ansatz eine Reihe von konkreten Lösungen für unterschiedliche Anwendungsbereiche im Standard-Lieferumfang beinhaltet, liegt der Focus auf der Integration verschiedener Plattformen und Produkte innerhalb eines UML-basierten Entwicklungsprozesses. Die Darstellung der Abbildung 8 zeigt die Komponentensicht innerhalb der TREND-Umgebung.

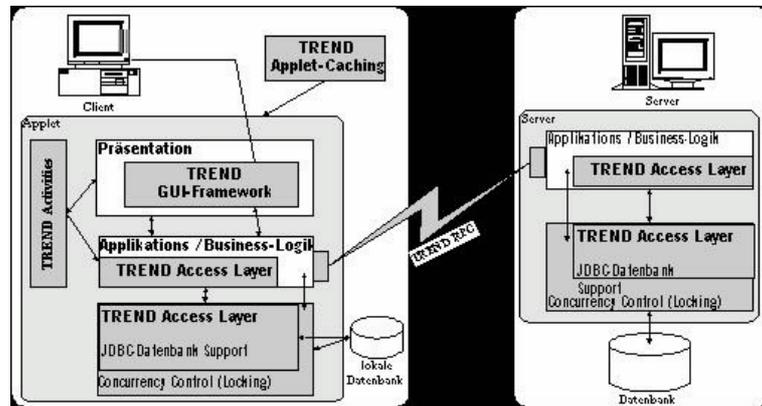


Abbildung 8: Mögliche Einbettung von TREND in die Zielapplikation

4 Wirtschaftlichkeitspotential

Wir wollen in Anlehnung an Mertens [Mer97] die Diskussion um die Digitale Fabrik mit Thesen über einige ökonomische Sachverhalte in Verbindung mit Zielsystemen von Unternehmen konfrontieren. Plakativ ausgedrückt, werden diese Sachverhalte „im Tagesgeschäft“ sowohl von Forschern als auch von Praktikern immer wieder übersehen. Die Ursache hierfür kann in „routine- oder gewohnheitsmäßigen Reaktionen“ liegen oder auch darin, daß aktuelle Werbebotschaften aufgesogen und unreflektiert wiedergegeben werden. In der Wirtschaft führt dies zu Aussagen, denen es an der nötigen Differenzierung mangelt, so daß unterschiedliche Sachverhalte gleich behandelt werden.

1. Ökonomie heißt, Ressourcen bestmöglich auszunutzen.
 2. In der Marktwirtschaft bedeutet das für die Unternehmung: Maximierung der Rentabilität. An die Stelle der Rentabilität können situationsbedingt ihre "Derivate" Return-on-investment (ROI) und Shareholder Value treten.
 3. Da sich der Punkt maximaler Wirtschaftlichkeit (definiert als Ertrag/Aufwand) nicht mit dem maximaler Rentabilität deckt, ist Wirtschaftlichkeitsmaximierung theoretisch nicht exakt.
 4. So genannte „Ingenieurziele“ (minimale Kosten, Termintreue, minimale Durchlaufzeiten, Bestandsminimierung, maximale Kapazitätsauslastung) sind theoretisch nur haltbar, wenn zahlreiche *ceteris-paribus*-Klauseln gelten:
 - a) Trivial ist z. B. die Feststellung, daß sich die digitale Fabrik Kostenminimierung nur dann als Ziel stecken darf, wenn der Ertrag und die Kapitalbindung konstant bleiben.
 - b) Man erinnere sich an Sachverhalte, die teilweise physikalisch bedingt sind. Beschleunigung verlangt einen Aufwand, der überproportional zur erreichten Geschwindigkeit steigt. Dafür gibt es auch viele Beispiele aus der betriebswirtschaftlichen Praxis, etwa bei Verkehrsunternehmen.
- Um die Beziehungen zwischen Sub-Zielen vor Augen zu haben, empfehlen sich hierarchische Darstellungen, wie z. B. der EVA (Abbildung 1).

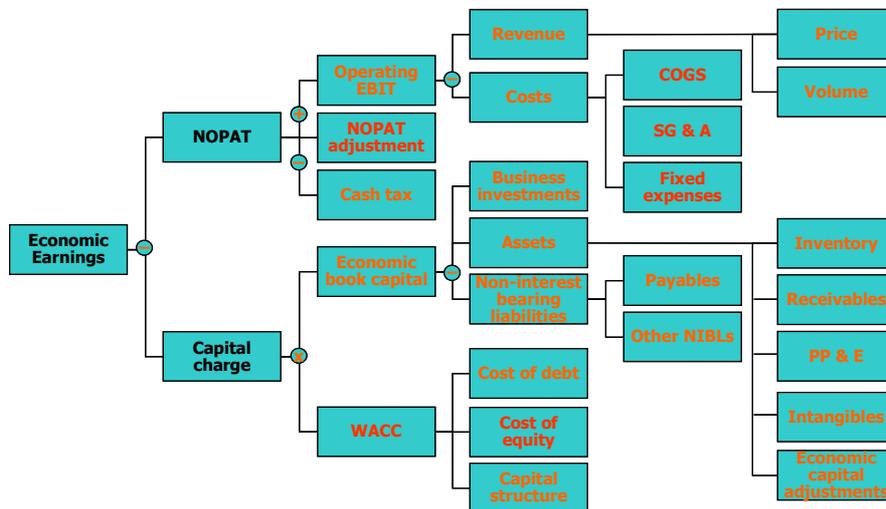


Abbildung 9: Value Proposition

Mit den folgenden Thesen wollen wir uns der Prozeßbetrachtung zuwenden, die wichtige Ergänzung der bisherigen Analysen darstellt.

5. Ein beachtlicher Teil der Prozeßveränderungen führt zu geänderter Kapitalbindung. Daher ist auch bei derartigen Maßnahmen Rentabilitätsmaximierung als Ziel zu beachten.

a) Die veränderte Kapitalbindung kann sich als Kapitalfreisetzung (Beispiel: reduzierte Lagerbestände, Verzicht auf Zahlungsziele), als erhöhte Kapitalbindung (Beispiel: Investition in automatisierte Fertigungsprozesse) oder bei kombinierten Maßnahmen (Beispiel: stärker automatisierte Fließfertigung mit minimalen Puffern) als Saldo von Investitionen in Betriebsmittel und Desinvestitionen in Lagern manifestieren.

b) Oft sind die Wirkungen erst bei genauerer Analyse festzustellen. (Beispiele: Eine neue Arbeitsteilung in der logistischen Kette führt dazu, daß Lager nicht beim Produzenten, sondern beim Logistikdienstleister entstehen. Als Konsequenz re-finanziert der Hersteller diese Lagerhaltung zwangsläufig über höhere Preise. Belastungsorientierte Auftragsfreigabe reduziert je nach Wahl der Freigabeparameter die Werkstattbestände, erhöht aber Bestände vor der Werkstatt.)

6. Vielfach bestehen Zielkonflikte zwischen Prozeß- und Ressourcenökonomie. Die prozeßökonomischen Ziele sind oft nur schwer auf die Rentabilität zu projizieren.

a) Dieser Zielkonflikt ist teilweise schon mit dem Dilemma der Ablaufplanung beschrieben worden.

b) Ressourcenökonomie orientiert sich am Oberziel der Rentabilitäts- bzw. ROI bzw. Unternehmenswert-Maximierung. Übliche Prozeßziele sind Durchlaufzeitminimierung, Kostenminimierung, Fehlerarmut usw.

c) unterschieden werden können:

- **Prozeßeffizienz:** Gegenstand ist die Gestaltung des Leistungsprozesses von seiner Auslösung bis zur Vertragserfüllung gegenüber dem Kunden. Die Verwirklichung von Prozeßeffizienz erfordert, den Leistungsprozeß über alle Stufen auf die Ziele der Gesamtunternehmung auszurichten. Mangelnde Prozeßeffizienz äußert sich u. a. in Form von Zwischenlagern und Terminüberschreitungen.
- **Ressourceneffizienz:** Gegenstand ist die Nutzung von Ressourcen in Form von Potentialfaktoren. Die Realisierung von Ressourceneffizienz erfordert, unter Abwägung von Autonomie- und Abstimmungskosten die Nutzung von Ressourcen über Bereichsgrenzen hinweg auf die Ziele der Gesamtunternehmung hin auszurichten. Mangelnde Ressourceneffizienz äußert sich u.a. in Leerkapazitäten und der problematischen Allokation knapper Ressourcen.
- **Markteffizienz:** Gegenstand ist die Nutzung von Chancen auf dem externen Beschaffungs- und Absatzmarkt. Die Verwirklichung von Markteffizienz erfordert, unter Abwägung von Autonomie- und Abstimmungskosten die Kontakte mit Marktpartnern zur Berücksichtigung von Marktinterdependenzen und zur Nutzung von Marktpotentialen über Bereichsgrenzen hinweg auf die Ziele der Gesamtunternehmung hin auszurichten. Mangelnde Markteffizienz äußert sich auf dem Absatzmarkt vor allem in entgangenen Absatz- und Beschaffungschancen und schlechten Konditionen bei der vertraglichen Regelung von Markttransaktionen.

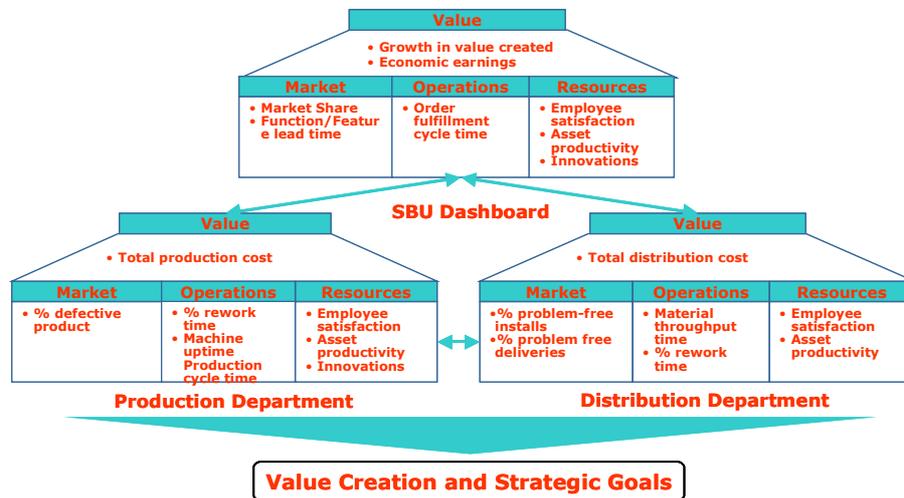


Abbildung 10: Value Creation and Strategic Goals

Werden Ressourcen- und Markteffizienz verbessert, so sinkt die Prozeßeffizienz, nicht zuletzt als Folge der Prognoseungewißheit. Vor diesem Hintergrund wird verständlich, warum Unternehmungen mit ausgeprägter Kundenorientierung Ressourceneffizienz im umfassenden Sinne nicht realisieren können. (vgl. auch These 7b).

7. Es bedarf erheblicher theoretischer und empirischer Forschung, um Behauptungen der „Prozeßlehre“ zu verifizieren oder zu falsifizieren:

a) Erhöhte Geschwindigkeit bedeutet hohen Ertrag, niedrigere Kosten und reduzierte Kapitalbindung.

Abbildung 11 sucht die verschiedenen Ursache-Wirkungs-Beziehung verschiedener zunächst intuitiv als orthogonal interpretierter Optimierungsziele zu veranschaulichen. Die mit (1) gekennzeichneten Pfeile verfolgen in vereinfachter Weise Zusammenhänge bei einer Aktion mit dem Ziel einer Durchlaufzeit-Minimierung. Es bleibt nachzurechnen, ob die Umsatzsteigerung zu einer Gewinnerhöhung führt, die die verstärkte Kapitalbindung mehr als kompensiert, oder ob der Saldo aus erhöhter und verminderter Kapitalbindung negativ (=günstig) ist.

Die mit (2) ergänzten Pfeile markieren die Ausstrahlungen eines Projekts, mit dem die Qualität gesteigert werden soll. Hier entsteht die Frage nach der Amortisation der Investitionen in Hardware und Software sowie in das Projekt selbst und damit nach der Rentabilität.

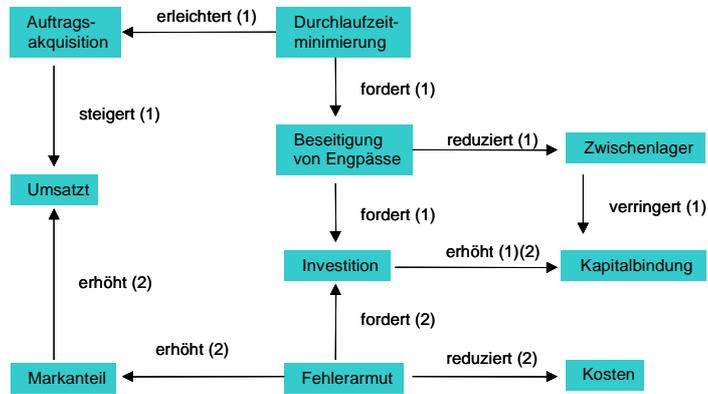


Abbildung 11: Beispielhafte Ursache-Wirkungs-Ketten

b) Prozeßorientierung führt zu besserer Kundenbedienung.

Eine Aufgabe der Digitalen Fabrik ist es, die Kundenbedienung nicht undifferenziert in großer Breite zu verbessern, sondern zur Kundenselektion beizutragen (siehe These 11).

Hier ist als erstes zu artikulieren, was darunter verstanden wird: Will etwa der Kunde vom Autohandel weniger Komfort, die notwendige Ausrüstung aber gleichzeitig Niedrigpreise? Oder gründlich und rasch über Extras die seine Individualität betonen, beraten werden? Bei gleichem Rentabilitätsziel würden im zweiten Fall höhere Preise notwendig.

c) Prozeßfocus führt zu weniger Schnittstellen als alternative Untergliederung von Organisationen.

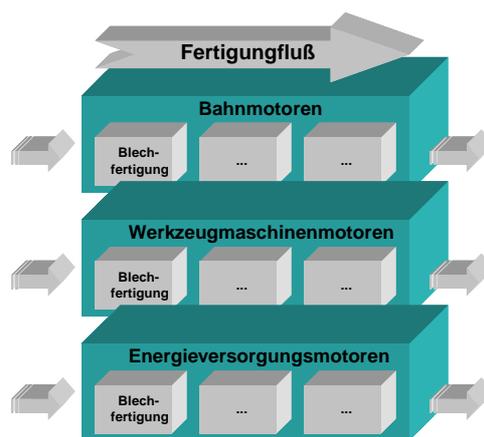


Abbildung 12: Business Process Re-Engineering in einem Motorenfertigungswerk

Abbildung 12 zeigt ein Projekt in einem Werk der Motorenfertigung. Die Strukturen sind strikt nach dem Prozeß bei der Auftrags Erfüllung ausgerichtet. Es ergibt sich hierbei jedoch eine ungünstige Ressourcenökonomie, weil die Maschinen in der Blechbearbeitung ungenügend ausgelastet wurden. Es muß „re-re-engineert“ und die Blechfertigung im Interesse einer besseren Kapitalrentabilität zentralisiert und verselbständigt werden (s. Abbildung 13).

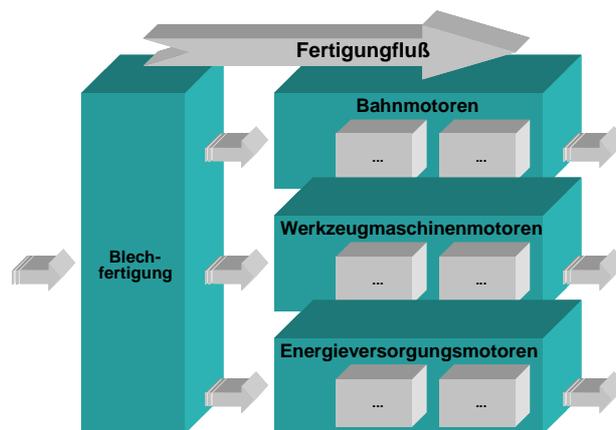


Abbildung 13: Lösung nach Business Process Re-Engineering

Abbildung 14 verdeutlicht das Beispiel einer Prozeßgliederung, wie sie im Zuge einer SAP-Einführung bei einem Hersteller von Bildschirmen anzutreffen sein könnte. Der Prozeß „Auftragsbearbeitung“ (*Order to Invoice*) wird getrennt von dem Prozeß „Reparaturabwicklung“ (*Complaint to Invoice*) geführt. In beiden Prozessen werden teure Komponenten (Bildschirmeinheiten) benötigt, aber innerhalb der Teilprozesse separat disponiert. Die Folge dieser unkoordinierten Beschaffung wären Einbußen bei der Beschaffungslogistik.

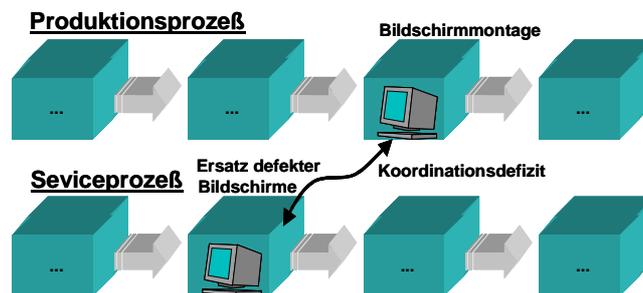


Abbildung 14: Prozeßgliederung bei einem Fernsehgeräte-Hersteller

Letztlich ist jedes Unternehmen ein komplexes Wirkungssystem. Unabhängig von der Art des Prozesseingriffs entstehen unweigerlich Wechselwirkungen und Rückkopplungen auf weitere, zunächst vermeintlich nicht tangierte, Prozesse.

Im Einzelfall bleibt durch konkreten Einsatz zu beweisen, ob eine Prozessumgestaltung sich langfristig bewährt. Zielsetzung der Digitalen Fabrik ist es daher lediglich durch Typisierung Entscheidungshilfen bereitstellen, nicht jedoch pauschalisierte Aussagen allgemeingültigen Anspruchs zu liefern.

8. Im Zweifel ist einer produkt- und / oder kunden- und / oder regionsorientierten Organisationsgliederung leichter ein Ziel vorzugeben, das mit der Rentabilitätsmaximierung des Gesamtunternehmens korreliert, als bei funktions- oder prozeßorientierter Aufspaltung.

Einem Prozeßbeigener ein Rentabilitätsziel vorzugeben ist viel schwieriger, als einem Produktmanager diese Aufgabe zu stellen, weil dem Produkt und damit (bedingt) auch der Tätigkeit des Produktmanagers ein Erlös zugerechnet werden kann.

9. Viele Projekte binden Kapital. Die Rückflüsse solcher Vorhaben erstrecken sich oft aber eine Anzahl von Jahren. Daher sind solche Projekte mit den Werkzeugen der kapitalorientierten dynamischen Investitionsrechnung zu evaluieren, nicht durch Gegenüberstellung von Aufwand und Ertrag.

Die Entscheidung, die bisherigen Anwendungssysteme durch ein großes integriertes Softwarepaket abzulösen, ist eine Investition. Sie besteht aus dem Kaufpreis K der Software und einem Betrag von x mal K (Installation, Schulung etc.), wobei x bei in der Praxis (bei sog. „spitzer Rechnung“) oft größer fünf ist.

10. Mehr als bisher sind bei Automatisierungs- und Entautomatisierungsmaßnahmen, die vorgeschlagen werden, die Ziele anzugeben und Abweichungen von der Rentabilitätsmaximierung zu rechtfertigen.

Während die ältere Literatur versucht, das *Produktivitätsparadoxon der Informationstechnik* mit Meßfehlern, methodologischen Problemen oder Zeitverschiebungen zu erklären, gibt es auch eine Anzahl von neueren qualitativen Erklärungen. In diesem Kontext wird das Paradoxon durch Mißwirtschaft und unzulängliche Verwendung von Technologien erklärt.

11. Nicht-quantifizierbare Ziele („Soft Goals“) sind kritisch zu hinterfragen.

Oft dienen Ziele wie z.B. „strategische Bedeutung“, „Flexibilität“, „Kundenzufriedenheit“ oder „Humanisierung der Arbeitswelt“ als „Fluchtargumente“, die nüchternen Ökonomen suspekt sein und die von daher hinterfragt werden sollten.

12. Einige Unternehmen gehen weg von der Steuerung über die Kostenrechnung, hin zur Lenkung über Bilanzen (Soll-Rentabilität) von (unter Umständen künstlich abgegrenzten) Unternehmensteilen. Die digitale Fabrik muß dieses abbilden.

Die beschleunigte ökonomische, technologische, soziale und umweltabhängige Änderung zwingt Entscheider, sich verstärkt neuer Bewertungsmethoden zu bedienen, während parallel die Komplexität der zu bewertenden Systeme wächst.

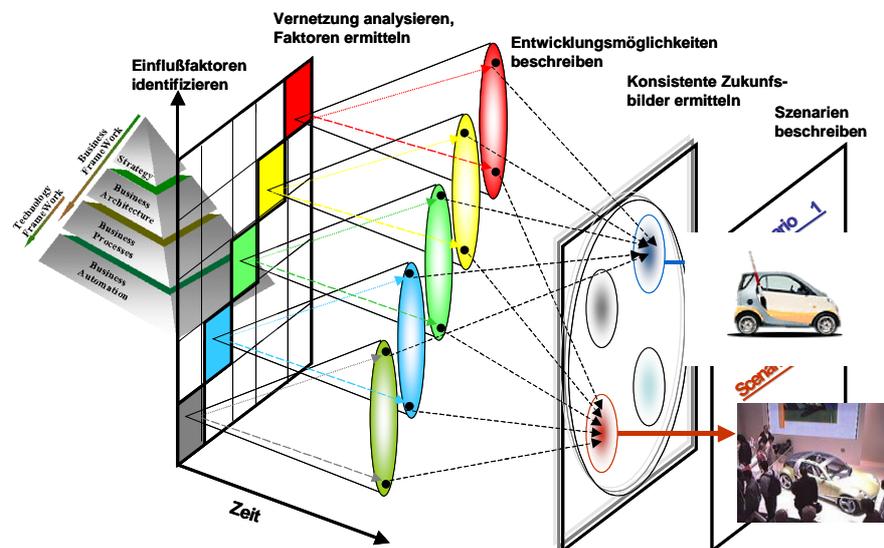


Abbildung 15: Phasen der Szenario-Generierung

5 Zusammenfassung

Die Gestaltung des Produktentstehungsprozesses sowie des IT-Einsatzes ist gegenwärtig primär durch eine starke Autonomie der Funktionsbereiche gekennzeichnet. Die einzelnen Unternehmensbereiche sind bestrebt, ihre Prozesse durch eine individuelle IT-Lösung optimal zu unterstützen. Bei den eingesetzten Systemen handelt es sich in der Regel um Individualsysteme oder weit verbreitete Quasi-Standards externer Anbieter. Die jeweiligen Teillösungen lassen sich vielfach zu keiner optimalen Gesamtlösung verbinden, da entsprechende Schnittstellen fehlen bzw. nicht ausreichend geplant werden. Die Komplexität des Produktentstehungsprozesses hat abgenommen. Die Zahl der Modelle und Produktvarianten ist im heutigen Vergleich gesunken. Entsprechend haben sich die Fertigungsprozesse vereinfacht und die Anzahl der Vorgaben für die Produktionsvorbereitung ist zurückgegangen.

Die Unternehmen kapseln sich von der Außenwelt zunehmend ab und erbringen den Großteil der notwendigen Tätigkeiten im Produktentstehungsprozess eigenständig oder in Zusammenarbeit mit wenigen eng integrierten Partnern. Der Grad der offenen spontanen Kooperationen mit weiteren externen Partnern ist daher gering. Aufgrund der geringen Notwendigkeit eines erweiterten externen Datenaustausches setzen die Hersteller jeweils eigene IT-Infrastrukturen (Individualsysteme) ein, um primär eine hohe interne Integration zu erreichen. Während Daten mit wenig relevanten Partnern auf Dateibasis ausgetauscht werden, halten wichtige Systemlieferanten das System des Herstellers intern vor.

Entsprechend laufen Planung und Entwicklung funktional voneinander getrennt ab. Die einzelnen Bereiche reichen die ausgereiften und dokumentierten Ergebnisse in einem seriellen Produktentstehungsprozess weiter. Der „Workflow“ zwischen den Funktionen wird aus der Organisation heraus definiert. Er ist in der Regel papierbasiert und führt zu einer Vielzahl an Medienbrüchen. Elektronische Workflowsysteme sowie Projektmanagement-Werkzeuge kommen lediglich innerhalb einzelner Abteilungen zum Einsatz. Die mangelnde informationstechnische Kopplung der einzelnen Funktionsbereiche führt dazu, daß keine effektive Werkzeugunterstützung für das Änderungsmanagement verfügbar ist. Änderungen erfolgen in der Regel mit hohem Aufwand manuell, so daß nur substanzielle Änderungen behandelt werden können. Wissen wird als wichtigste Ressource aufgefasst und mit hohem Aufwand gehandhabt. Das Wissensmanagement wird durch spezielle (Software-)Systeme unterstützt. Diese werden jedoch in den einzelnen Funktionsbereichen eigenständig genutzt. Die funktionale Trennung von Produktentwicklung und Produktionsvorbereitung führt dazu, daß die Dauer des Serienanlaufes weiterhin lang ist und ein hoher Aufwand zur Erreichung der geforderten Qualität zu erbringen ist.

Die Softwareunterstützung in der Konstruktion ist durch eine geringe Wiederverwendbarkeit von Standardkomponenten sowie einen geringen Automatisierungsgrad gekennzeichnet. Die Systeme geben dem Konstrukteur somit einen hohen Freiheitsgrad in der Produktgestaltung, der ein entsprechendes Höchstmaß an Kreativität erfordert. Der Entwicklungsaufwand ist aufgrund der geringen Werkzeugunterstützung hoch.

Die Verifikation des Designs erfolgt durch die Kombination von virtuellen und realen Prototypingtechniken. So wird der Aufwand für die Modellierung des digitalen Prototypen auf wichtige Kernbereiche beschränkt, für die sich mit Hilfe der Computersimulation hinreichend realistische Resultate bei geringem Aufwand erzielen lassen. Insbesondere Bereiche, die einen enormen Modellierungsaufwand erfordern oder nicht hinreichend genau simuliert werden können, werden auf Basis weiterentwickelter Rapid-Prototyping-Methoden verifiziert. Viele Fahrzeugtypen liegen sowohl als virtuelle Prototypen als auch als Versuchsfahrzeuge vor.

Für die Planung der Fertigung stehen nur vereinzelt Ergebnisse aus früheren Planungen zur Wiederverwendung zur Verfügung. Es könnten jedoch teilweise Planungselemente aus Objektkatalogen (Norm- und Standardkomponenten) verwendet werden, die den Planungsprozess von Routineaufgaben befreien. Für Bereiche, in denen keine Standardkomponenten vorliegen, beginnt die Planung bei Null. Dies führt zu einem hohen Aufwand für die Informationsbeschaffung und Modellierung. Die Planung wird auf wesentliche Teilbereiche focussiert, in denen eine sehr hohe Genauigkeit bzw. Realitätsnähe der Abbildung erreicht wird. Es entstehen einzelne Planungsinself. Innerhalb dieser Planungsinself existieren für den jeweiligen Anwendungszweck spezielle leistungsstarke Simulationssysteme, die jedoch nicht miteinander gekoppelt werden. So lassen sich die Wechselwirkungen zwischen den Bereichen nicht analysieren und das Gesamtsystem nicht optimieren. Die starke funktionale Trennung der Bereiche führt so trotz eines hohen Simulationsaufwands lediglich zu einzelnen Teiloptima innerhalb der Abteilungen. Das Gesamtoptimum wird dagegen verfehlt. Die Planung in den einzelnen Teilbereichen wird durch den Einsatz von Virtual Reality unterstützt. Der Schwerpunkt des Einsatzes ist die Visualisierung der Fertigungsabläufe (Architektur; Layout; Materialfluß). Aufgrund der hohen Genauigkeit der Modelle können einzelne Fertigungsprozesse bzw. -abschnitte sehr detailliert dargestellt werden. Es fehlt allerdings ein umfassendes Simulationsmodell für den gesamten Fertigungsstandort.

Die Beschaffung der Rohstoffe und Teile erfolgt auf der Basis von B2B-Plattformen von wechselnden Lieferanten. Als Folge unterliegen die Liefer- bzw. Produktionsnetze einer dynamischen Veränderung. Die Prozesse innerhalb der Auftragsabwicklung werden daher weitestgehend standardisiert. Mit den Lieferanten werden lediglich Beschaffungsdaten ausgetauscht. Innerhalb der eigenen Fertigung werden Betriebsdaten nur soweit erfaßt, wie sie für die operative Steuerung des Prozesses notwendig sind. Die direkte Ausregelung von Abweichungen ist eher gering. Die Programmierung und Wartung von Fertigungssystemen erfolgt

manuell und ist wenig automatisiert. Die Bearbeitungspläne und Modelle der Fertigungseinrichtungen werden manuell erzeugt und von Hand aufwendig vor Ort umgesetzt.

Die o.g. Entwicklungen haben zu einer spezifischen Systemlandschaft im Produktentstehungsprozeß geführt. Aufgrund der hohen Autonomie der Funktionen haben sich in den einzelnen Unternehmensbereichen verschiedenste IT-Lösungen durchgesetzt. Während in einigen Abteilungen Individuallösungen entwickelt wurden, werden in anderen Bereichen Systeme großer Hersteller eingesetzt. Bei diesen Systemen handelt es sich in der Regel um Quasi-Standards, die für den jeweiligen Anwendungsfall von nahezu allen Unternehmen der Branche verwandt werden und damit den Markt dominieren. Insbesondere da die Schnittstellen durch die Hersteller nicht offengelegt werden, entstehen geschlossenen Systemwelten, zwischen denen Daten nur mit hohem Aufwand ausgetauscht werden können. Die Benutzungsschnittstellen dieser Systeme sind in der Regel multimedial. Die Interaktion erfolgt insbesondere durch Tastatur und Bildschirm. Datenstandards sind hauptsächlich in der CAx-Kette verbreitet und werden vor allem durch die entsprechenden Systeme innerhalb der Produktentwicklung unterstützt. Die Datenqualität ist grundsätzlich gut. Die Eingangsdaten liegen in den einzelnen Bereichen vollständig vor, müssen aber aufwendig umgewandelt und in die eigenen Modelle überführt werden. Die Datenschnittstellen sind durch spezielle Austauschformate zwischen den CAx-Systemen geprägt. Da die einzelnen Systeme in der Regel unabhängig voneinander sind, sind die Datenstrukturen und -modelle systemabhängig und funktional. Die Datenhaltung findet am Ort der Entstehung statt, wo sie zunächst temporär vorgehalten werden, um dann über einen manuellen Abgleich im zentralen Server gespeichert zu werden. Insgesamt gesehen entsprechen die datentechnischen Voraussetzungen den durch die Organisation des PEP entstandenen Anforderungen. Die Trennung von Produkt und Prozess, Produktentwicklung und Produktionsvorbereitung sowie der entsprechenden Systemlandschaft verhindern ein integriertes Produkt- und Prozessdatenmanagement.

Literatur

- [BPS+00] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler (Hrsg.): *Extensible Markup Language (XML) 1.0 (Second Edition)*, World Wide Web Consortium, Boston, USA, 2000.
- [Che76] Chen, P.-P. S.: *The Entity-Relationship model - toward a unified view of data*, ACM Transactions on Database Systems, 1(1), March, pp. 9-36, 1976.
- [Jec00] M. Jeckle: *Automatische Generierung objektorientierter Strukturen*, in: Proceedings KnowTech, Leipzig, 2000.
- [Jeck00] M. Jeckle: *Konzepte der Metamodellierung -- zum Begriff Metamodell*, S. 29-30, in: GI Softwaretechnik Trends, 20(2), Mai, 2000.
- [Jec01a] M. Jeckle: *Using XSLT to derive schemata from UML*, in: Proceedings XSLTUK01, Oxford, UK, 2001.
- [Jec01b] M. Jeckle: *Practical usage of W3C's XML-Schema and a process for generating schema structures from UML models*, in: Proceedings SSGRR 2001, Rom, Italy, 2001.
- [KJ98] L. Kern-Bausch, M. Jeckle: *From a Semantically Irreducible Formulated Conceptual Schema to an UML Model*, S. 32-44, in: M. Schader, A. Korthaus (Hrsg.): *The Unified Modeling Language - Technical Aspects and Applications*, Physica-Verlag, Würzburg, Wien, 1998.
- [Kru99] P. Kruchten: *The Rational Unified Process. An Introduction*, Addison-Wesley, Reading, MA, USA, 1999.
- [Mer97] P. Mertens: *Operiert die Wirtschaftsinformatik mit falschen Unternehmenszielen? - 15 Thesen*, Wiss'97, 1997.
- [Nij76] G. M. Nijssen: *A gross architecture for the next generation of database management systems*, Proc. 1976 IFIP Working Conference on Modelling in Data Base Management Systems, Nijssen, G. M. (ed.), Freudenstadt, North-Holland Publishing, S. 1-24, 1976.
- [Oes01] B. Oestereich (Hrsg.): *Erfolgreich mit Objektorientierung -- Vorgehensmodelle und Managementpraktiken für die objektorientierte Softwareentwicklung*, 2. Aufl., Oldenbourg, München, 2001.
- [Pet97] B. Petkoff: *Die Kybernetik II. Ordnung: eine methodologische Basis der Wirtschaftsinformatik?* Wiss'97, 1997.
- [Pet98] B. Petkoff: *Wissensmanagement*, Addison Wesley, Bonn, 1998.

[Sav99] Referenzen zum Projekt *SAVE*:

-<http://skipper.mar.external.lmco.com/save/>

-OMG-Dokument mfg/99-07-01

[UML01] UML Partners: *OMG Unified Modeling Language (UML)*, Version 1.5, Object Management Group, Framingham, USA, 2001.

[XMI99] XMI Partners: *XML Metadata Interchange (XMI) Version 1.1*, Object Management Group, OMG Document ad/1999-10-02, Framingham, USA, 1999.

[XSLT99] J. Clark (Hrsg.): *XSL Transformations (XSLT)*, Version 1.0, World Wide Web Consortium, Boston, USA, 1999.