

Eine modellbasierte Software-Architektur zur Unterstützung dezentraler Prozesse in der Produktentwicklung

Patrick D. Stiefel, Jörg P. Müller

*Institut für Informatik, Technische Universität Clausthal
{patrick.stiefel | joerg.mueller}@tu-clausthal.de*

1 Einleitung

Kollaborationsplattformen sind ein moderner Ansatz für die Unterstützung der organisationsübergreifenden Wissensverarbeitung in der Produktentwicklung. Die dadurch entstehenden, kollaborativen Produktdatenmanagement (PDM)-Netze bieten nicht nur weltweiten Zugriff auf alle Daten im Produktlebenszyklus, sondern leisten auch einen entscheidenden Beitrag zur Bewältigung der langfristigen Herausforderungen im digitalen Produktentwicklungsprozess.

Radtke et. al. prognostizieren bis zum Jahre 2015 eine Revolution in der Schlüsselbranche der deutschen Volkswirtschaft, der Automobilindustrie (Radtke 2004). Steigender Kosten- und Innovationsdruck zwingt die Hersteller in eine „Produktivitätszange“ und damit zu einem dritten, revolutionären Einschnitt nach der Erfindung des Fließbands durch Henry Ford und der Lean Production durch Toyota. Wesentliche Veränderungen betreffen die Produktstruktur und Verflachung der Fertigungstiefe und durch Verlagerung der Innovations- und Wertschöpfungsschwerpunkte weg vom OEM hin zu starken Systemzulieferern mit massivem Einfluss auf das Endprodukt („Smart-Modell“).

Die *organisationsübergreifende, kollaborative Produktentwicklung* (engl.: *Collaborative Product Development, CPD*) ist wichtigster Bestandteil dieser Entwicklungstendenz. Besonders bei der Definition und der Ausführung von Produktentwicklungsaufgaben haben sich bi- oder multilaterale Entwicklungskollaborationen schon länger bewährt (vgl. Tietze 2003, S. 195 ff.), da sich durch den Zusammenschluss unterschiedlichster Partner mit heterogenen Kernkompetenzen die individuellen Entwicklungs- und Produktionstechnologien, bzw. Produkt- und Prozess-Know-hows positiv ergänzen (Albers 1998, S. 4 ff).

In der CPD können *synchrone* („gleicher Zeitpunkt – gleicher Ort“) und *asynchrone Kollaborationen* („unterschiedliche Zeitpunkte – unterschiedliche Orte“) und

unterschieden werden. Synchrone Kollaborationen werden häufig eingesetzt für Echtzeit-Kommunikation oder –Datentransfer, während asynchrone Kollaborationen stark das Workflowmanagement fokussieren (Li and Qiu 2006). *CPD-Plattformen* liefern einen entscheidenden Mehrwert insbesondere in den frühen Phasen der Produktentwicklung, da Engineering-Kernkompetenzen im Bereich Entwicklung & Design, sowie Prozesswissen gebündelt werden können. Die Art Kollaborationsplattform, die in unseren Arbeiten im Mittelpunkt steht, arbeitet asynchron auf Basis der „Design-By-Feature“-Technologie, die hauptsächlich in der Produktentwicklung angewendet wird. Eine der Hauptaufgaben besteht darin, hinsichtlich des Volumens besonders große CAD Modelle und deren Modifikationen (engl.: design changes) zwischen den Kollaborationsteilnehmern effizient zu verteilen. (Li and Ong 2006).

Die Zusammenarbeit zwischen einzelnen Bereichen eines Unternehmens und organisationsübergreifende Kollaborationen werden weiter zunehmen. In Krause werden dazu drei aktuelle Trends aufgezeigt, die dabei helfen sollen, die gegenwärtigen Defizite – zu hohe Komplexität, zu geringe Integrationstiefe und Interoperabilität – zu beseitigen (Krause 2007, S.75ff):

1. Ausreifung der existierenden Methoden, Werkzeuge und Systeme,
2. Entwicklung anpassungsfähiger, skalierbarer und offener Integrationsplattformen mit intelligenter Umgebung, sowie
3. Durchsetzung von Standards in Werkzeugen.

Erfolgreiche Kollaborationen in der Produktentwicklung erfordern nach Krause die folgenden Voraussetzungen:

- Nutzung von Ontologien als gemeinsame Sprachbasis einer Domäne, bzw. eines Arbeitsgebiets.
- Öffnung der Unternehmen für eine optimale Zusammenarbeit durch Einbindung externer Arbeitsvorgänge in unternehmensinterne Arbeitsprozesse und einen geeigneten Umgang mit technischen Sicherheitsinfrastrukturen zum Schutz des intellektuellen Firmeneigentums.
- Abstimmung der technischen und organisatorischen Prozesse zwischen den kooperierenden Unternehmen.

Auf der Basis dieser Anforderungen beschäftigt sich unsere Forschung damit, wie mit Hilfe dezentraler Informationstechnologien CPD-Prozesse unterstützt werden können (Stiefel und Müller 2007; Stiefel und Müller 2008). Ein konkretes Ergebnis ist die prototypische Product Collaboration Platform (PCP)¹, eine peer-to-peer(P2P)-basierte Kollaborationsplattform für die dezentrale Verwaltung von Produktmodellen und die Unterstützung dezentral organisierter Kollaborationsprozesse in der Produktentwicklung.

¹ Siehe auch <http://plm.in.tu-clausthal.de/PCP/cebit.php> [01.09.2009]

Wir verfolgen einen modellbasierten Entwicklungsansatz (MBE) für Informationssysteme für die dezentrale und kollaborative Produktentwicklung (wie die PCP). Auf der Basis von IT-unabhängigen Prozess- und Kollaborationsmodellen (Computation Independent Model, CIM) werden in einem iterativen Entwicklungsprozess unterschiedliche Typen IT-basierter Modelle abgeleitet, die sich in ihrem Abstraktionsgrad unterscheiden: Von IT-Architekturmodellen (Platform Independent Model, PIM) über plattformspezifische Modelle (PSM) hin zu konkreten Software-Artefakten.

Eine offene Frage hierbei ist, wie im Rahmen eines MBE-Ansatzes die Anforderungen dezentraler Entwicklungsprozesse einerseits und die Möglichkeiten dezentraler IT-Plattformen (und im speziellen P2P-Umgebungen) berücksichtigt werden können. In diesem Papier stellen wir Methoden und Modelle vor, die im Rahmen der sonst üblichen Entwicklung von zentralen (Kollaborations-)Systemen keine Rolle spielen und daher im MBE-Kontext bisher kaum Betrachtung finden. Wir schlagen Modellierungsansätze auf den verschiedenen Ebenen der MBE vor, die eingesetzt werden können, um – unterstützt durch teilweise automatisierte Modelltransformationen – dezentrale Architekturmodelle und schließlich konkrete Software-Artefakte zu generieren. Unser Ziel ist es hier, wichtige Grundlagen für die nächste Generation lose gekoppelter Kollaborationssysteme für organisationsgreifende Produktentwicklung zu schaffen.

In Abschnitt 2 werden dazu zunächst die Grundlagen der dezentralen, kollaborativen Produktentwicklung diskutiert. Abschnitt 3 stellt den „State-of-the-art“ der modellbasierten Software-Entwicklung vor und diskutiert spezielle Anforderungen bei der Anwendung des MBE-Ansatzes zur Entwicklung dezentraler Kooperationssysteme. Im vierten Abschnitt werden Modelle der unterschiedlichen MBE-Ebenen präsentiert, bevor im letzten Abschnitt eine Bewertung des vorgestellten Ansatzes vorgenommen und dessen Eignung hinsichtlich der Praxistauglichkeit beurteilt wird.

2 Hintergrund

Es gibt aktuell vier Schlagwörter, die zur Beschreibung der Zusammenarbeitsformen zwischen Individuen verwendet werden. In Anlehnung an das 3K-Modell nach Teufel et. al. beschreibt der Begriff *Kommunikation* die Art und Weise, wie Informationsobjekte zwischen Organisationseinheiten ausgetauscht werden (Teufel 1995, Abschnitt 1.2.2). Ohne *Koordination* generieren unterschiedliche Einheiten Redundanz oder stark unterschiedliche, zusammenhangslose Ergebnisse. *Koordination* ist notwendig, um Einheiten darüber zu informieren, zu welchem Zeitpunkt und in welcher Form eine Reaktion notwendig ist, um das gemeinsame Ziel zu erreichen. Dies erfordert in jedem Fall gemeinsame Interessen der Beteiligten, bzw. eine organisatorische Zusammengehörigkeit. Einen noch stärkeren Zusammenhalt erreicht man, wenn alle Einheiten ein und dasselbe Ziel verfolgen und ihnen die

Konsequenz jeder einzelnen Handlung bewusst ist. In diesem Fall spricht man von *Kooperation*.

Kollaboration unterscheidet sich von allen drei K-Wörtern. Im Unterschied zu Kommunikation geht es bei Kollaboration nicht um den reinen Informationsaustausch, sondern um das *Nutzen von Information, um etwas Neues zu schaffen*. Im Gegensatz zu Koordination, hat jede Kollaboration einen *Spontanitätscharakter* und läuft in der Regel nicht harmonisch strukturiert ab. Und ergänzend zur Kooperation sind *Ablehnungen und Widersprüche*, letztendlich gar *Konflikte* durchaus erwünscht. Kollaborationen werden also verwendet um Probleme zu lösen, um neue Lösungsansätze zu generieren, bzw. um neue Produkte zu designen. Dabei sind die folgenden Schritte essentiell (Schrage 1990, Winer and Ray 1994):

1. Definieren des Kollaborationsziels/der Herausforderung (achievement point),
2. Festlegen der Kollaborationsteilnehmer,
3. Definieren des Kollaborationsraums (blackboard, shared screen) und
4. Setzen eines genügend großen Zeitraums zum Finden eines Prototyps.

Die Formen der Produktentwicklung, die wir in unseren Arbeiten betrachten, entsprechen einer unternehmensübergreifenden Kollaboration mit dem Ziel der Erstellung eines Produktmodells, das den Anforderungen einer gemeinsam oder von einzelnen Partnern erstellten Spezifikation genügt. Als *Produktmodell* bezeichnet man „[...] die Spezifikation von Produktinformationen in Form technischer Dokumente oder sonstiger Produktrepräsentationen, die im Laufe des Entwicklungsprozesses als (Zwischen-)Ergebnisse entstehen. Produktmodelle stellen damit formale Abbilder realer oder geplanter Produkteigenschaften dar.“ (Grabowski 1993).

Geschäftsprozesse für die *organisationsübergreifende, modellbasierte Produktentwicklung (OMP)* beziehen sich immer auf einen bestimmten *Entwicklungsprozess* nach Lindemann und arbeiten mit einer konkreten Produktmodellausprägung (Lindemann 2007, S.18 ff.):

- *Zielmodell*: Erfassung, Strukturierung und Dokumentation von gewünschten Systemmerkmalen.
- *Problemmodell*: Generierung eines besseren Problem- oder Systemverständnisses in Bezug auf existierende oder zu entwickelnde Systeme zur Darstellung von Schwachstellen und Optimierungspotenzialen (Belastungen, Realisierbarkeit, Werkstoffeigenschaften, usw.).
- *Entwicklungsmodell*: Zur Spezifikation der Struktur sowie der geometrischen und stofflichen Beschaffenheit eines zu entwickelnden Produktes.
- *Verifikationsmodell*: Erfassung und Analyse wesentlicher Eigenschaften eines Produktes, die für eine Bewertung hinsichtlich der Produktqualität und Anforderungserfüllung relevant sind.

Jeder *OMP-Prozess* beschreibt die verteilte Lösung eines vorgegebenen Problems der Produktentwicklung (*Spezifikation*); unser konzeptueller Lösungsansatz lehnt sich dabei an das ursprünglich aus dem Bereich der Multiagentensysteme stammende Paradigma der *Verteilten Problemlösung* (vgl. Abbildung 1) an (Müller 1996; Smith 1981, S.61-70). Am Ende des OMP-Prozesses steht die Synthese aller gültigen, verteilt unter den Teilnehmern vorliegenden Teillösungen (Produktmodellvorschläge, engl.: *Proposal*) zu einer Gesamtlösung für die durch einen *Initiator* formulierte Spezifikation.

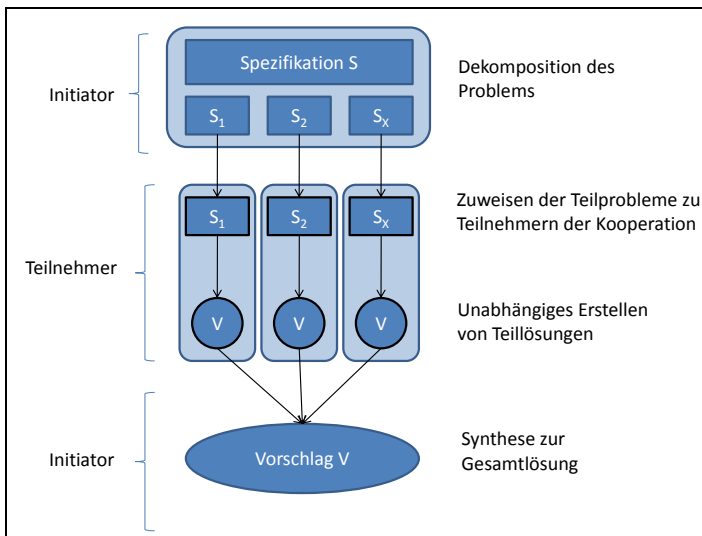


Abbildung 1: Allgemeiner Prozess der OMP

Dezentralität als Konzept zum Entwurf verteilter Systeme hat sich inzwischen als ein anerkanntes Prinzip für das Management globaler und dynamischer Netzwerke etabliert (vgl. Stäber 2008). Eine dezentrale Architektur auf Basis von P2P-Technologie bietet für Kooperationsplattformen einen flexiblen, skalierbaren Ansatz für die Beherrschung der Anforderungen, die bei der OMP entstehen:

- Unterstützung von ad-hoc-Zusammenschlüssen weltweit verteilter Partner, die untereinander zuvor in der Regel keine feste Partnerschaft eingegangen sind.
- Effiziente Verteilung von Produktmodellen unter den beteiligten Produktentwicklern entweder im Sinne einer Lastverteilung oder einer aufgabenorientierten Verteilung und des damit in der Regel einhergehenden, schnelleren Antwortzeitverhaltens.

3 Organisationsübergreifende modellbasierte Produktentwicklung

Die Definition von Modellen auf unterschiedlichen Abstraktionsebenen und damit die Modellierung unterschiedlicher Aspekte ist eine der Grundideen von MBE (Gruhn 2006). Das CIM beschreibt die Funktionalitäten der Kooperationsplattform auf fachlicher Ebene und wird in einer Sprache definiert, die für die Produktentwickler, bzw. Ingenieure als Anwender des Systems verständlich ist. Es dient zum Diskurs zwischen Ingenieur und Softwarearchitekt über die konkreten Anforderungen der OMP. In unseren Arbeiten verwenden wir nach anfänglichen Experimenten mit eEPKs inzwischen als CIM-Sprache die *Business Process Modelling Notation (BPMN)*², da mit Hilfe dieser Notation fachliche als auch technische Modelle (mit Schleifen, Ausnahmebehandlungen und Transaktionen) erstellbar sind, die von der Process Engine eines Workflow- oder Business Process Management Systems (BPMS) interpretiert und ausgeführt werden können (Allweyer 2008). BPMN-Modelle werden in Form eines Business Process Diagramms (BPD) visualisiert.

Der *Plattform*-Begriff in der MBE bezeichnet eine abgeschlossene Softwarekomponente oder –technologie, die über Schnittstellen verwendet werden kann, ohne dass die benutzende Komponente die Implementierung der Plattformfunktionalität kennt (Kempa 2005). Die Plattform stellt technische Dienste bereit, ohne die die Software nicht funktioniert. Im Zusammenhang mit Plattform steht der Begriff *Platform Independent Model (PIM)*. Dieses modelliert – wie sein Name schon sagt – die Funktionalität einer Komponente unabhängig von der Plattform. Es enthält also genau den Teil des Systems, der sich beschreiben lässt, ohne die endgültige Zielplattform zu kennen. Das *Plattform Specific Model (PSM)* dagegen kennt Service-Kompositionen, DRM und Overlay-Ausprägungen und setzt das PIM durch geeignete Service-Kompositionen um (vgl. Konzept in Abschnitt 3.1).

3.1 Konzept

Die vorgestellten MBE-Methoden werden in dieser Arbeit verwendet, um *top-down* ausgehend von fachlichen Beschreibungen kooperativer Geschäftsprozesse auf der Ebene des CIM Architekturen für die OMP mit dezentralem Charakter auf der Ebene des PSM zu entwickeln (Stiefel 2008). Eine Architektur entspricht dabei einer Kombination aus *Service-Kompositionen* im Rahmen einer *Service-orientierten Architektur (SOA)*, dem *Dezentralen Ressourcen Management (DRM)* und einem konkreten P2P-Overlay (vgl. Abbildung 2, rechts). Das DRM ist eine Abstraktionsschicht konkreter P2P-Overlayfunktionen für die OMP und stellt dem Service-Layer Me-

² Die BPMN Spezifikation Version 1.2 vom Januar 2009 kann nachgelesen werden unter: <http://www.omg.org/spec/BPMN/1.2/PDF>. Ein sehr gutes Übersichtsposter von G. Decker et.al. (Hasso Plattner Institut, Universität Potsdam) findet sich unter: http://bpt.hpi.uni-potsdam.de/pub/Public/BPMNCorner/BPMN1_1_Poster_DE.pdf [01.09.2009].

thoden zur Verfügung (insbesondere: *publish*, *search*, *subscribe*, *notify*). Dort werden generische Kollaborationsdienste (z. B. „Verwaltung der Produktmodellstruktur“) durch Service-Kompositionen beschrieben, die dann in lokalen Service-Engines ausgeführt werden können.

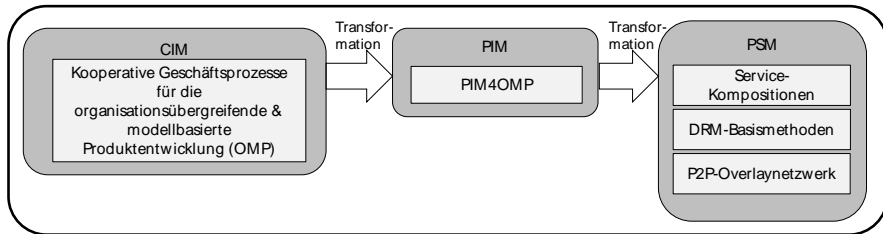


Abbildung 2: MBE-Vorgehen in der OMP

3.2 Anforderungen

Bei der Generierung der Modelle im Rahmen der MBE lässt sich feststellen, dass sich bei der Entwicklung eines dezentral organisierten Informationssystems spezielle, nur auf diese Netzwerktopologie zugeschnittene Anforderungen auf allen Modell-Abstraktionsebenen wiederfinden.

Beim Entwurf des CIM wird zur Design-Time ein *globales Geschäftsprozessmodell (gGPM)* entworfen, das den gesamten Verlauf der Kollaboration durch Verknüpfung von organisationsübergreifenden, globalen Prozessen beschreibt. Ein gGPM definiert damit aus übergeordneter Sicht die Aufgaben-, bzw. Prozessverteilung an Rollen der OMP (Initiator/ Teilnehmer) auf hohem, grobgranularem Abstraktionsniveau. Das gGPM ist weiterhin dafür verantwortlich die Übergangszeitpunkte zwischen den Aktivitäten unterschiedlicher Rollen zu spezifizieren.

Im linken Teil der Abbildung 3 im Abschnitt 4.1 ist das gGPM einer Kollaboration dargestellt, in der beispielsweise ein produktstrukturorientiertes Design eines Entwicklungsmodells gesucht wird. In diesem Fall arbeiten eine beliebige Anzahl an Kollaborations-Teilnehmer isoliert an der Lösung von Teilproblemen und übermitteln ihre Teilergebnisse in der Regel direkt an einen Initiator, der die Ergebnisse auswertet, ein Zwischenergebnis festlegt und über das weitere Vorgehen entscheidet. Jeder *globale Geschäftsprozess (gGP)* des gGPM (zum Beispiel: Spezifikation erstellen) kann durch ein feingranulares, *lokales Geschäftsprozessmodell (IGPM)* detaillierter beschrieben werden (zum Beispiel: „Daten aus Teamcenter Eng. importieren“, „Spezifikationsinhalt festlegen“, „Empfängerkreis bestimmen“, usw.). Die im IGPM beschriebenen, *lokalen Geschäftsprozesse (IGP)* müssen in der festgelegten Reihenfolge zur Run-Time durch einen konkreten Akteur (zum Beispiel dem Initiator „Herr Müller“) abgearbeitet werden.

Das bedeutet, dass mittels einer Transformation im Rahmen des MBE-Vorgehens das IGPM in ein *lokales Workflowmodell (IWFEM)* auf der Ebene des PSM transformiert wird. Jedes IWFEM besteht aus einer Reihe an OMP-Basisdiensten,

die zur Design-Time einer Rolle zugeordnet und zur Run-Time durch einen Akteur ausgeführt werden müssen. Damit ist jedoch noch keine Aussage getroffen, ob die Dienste durch einen Akteur lokal implementiert werden oder durch Dritte bezogen werden. Daraus resultieren konkrete Architekturmodelle und Service-Verteilungen, vgl. Abschnitte 4.2 und 4.3.

Die Transformation erfolgt über ein PIM, so wie im Abschnitt 4.3 dargestellt. Im PIM werden im Wesentlichen Rollen zu konkreten Peers zugeordnet, der Ausführungsort der Basisdienste und die Existenz oder Nicht-Existenz eines Koordinators festgelegt. Das hängt insbesondere davon ab, welches konkretes P2P-Overlay zum Einsatz kommen soll und ob beispielsweise ein strukturierter, unstrukturierter oder ein hybrider P2P-Overlay-Ansatz zur Realisierung sinnvoll sein kann.

4 Modelle eines MBE-Ansatzes

4.1 Geschäftsprozesse für die OMP auf CIM-Ebene

Im linken Teil der Abbildung 3 ist ein gGPM exemplarisch dargestellt. Es spezifiziert den folgenden Verlauf einer OMP. Das Erstellen einer Spezifikation ist ein globaler Prozess, in dem die Vorstellung des Initiators über die Gestalt und Eigenschaften eines zu konstruierenden Entwicklungsmodells durch eine Menge an Parametern in einem Datenelement „Spezifikation“ beschrieben wird. Teilnehmer analysieren vorliegende Spezifikationen. Daraufhin wird ein korrespondierender Lösungsvorschlag generiert und als Datenelement „Vorschlag“ veröffentlicht. Dieser wird wiederum vom Initiator überprüft. Variiert man im dargestellten gGPM einen der folgenden Parameter, so entsteht ein neuer globaler Geschäftsprozess: *Produktmodell(PM-)Verteilung, Hierarchien und Iterationen*. Abbildung 3 (rechts) zeigt die vorgestellten Dimensionen der in dieser Arbeit betrachteten Produktentwicklungsszenarien. Das einfachste Szenario ist durch die Ausprägungen *iterationsfrei/zentrale Modellverwaltung/hierarchielos* beschrieben.

PM-Verteilung: Die Zuordnung der Datenelemente Spezifikation und/ oder Vorschlag zu einer bestimmten Rolle wird im gGPM explizit ausgedrückt. Der gGPM-Designer legt die Verteilungsregeln, insbesondere unter Berücksichtigung von erwartetem Vertrauen³ unter den Entwicklungspartnern, fest. Es gibt drei grundsätzlich voneinander zu unterscheidende Strategien: *Zentrale, hybride und dezentrale PM-Speicherung*.

³ Vertrauen ist eine der wichtigsten Quality-of-Service(QoS)-Variablen in der OMP.

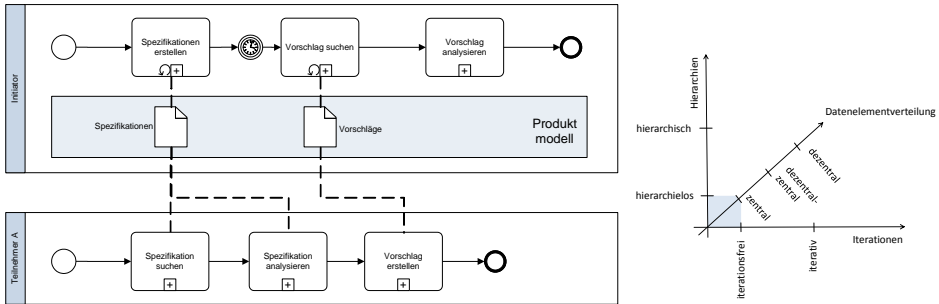


Abbildung 3: Beispiel für ein globales Geschäftsprozessmodell (links) und Kooperations-Dimensionen (rechts)

- Bei der *zentralen PM-Speicherung* wird die Verwaltung einer Spezifikation und der dazugehörigen Vorschläge genau einem Teilnehmer übertragen, im Regelfall dem Initiator. Im gGPM in Abbildung 3 (links) ist dies durch Zuordnung der Datenelemente „Spezifikationen“ und „Vorschlag“ zum Initiator-Pool modelliert. Zusätzlich ist in Abbildung 4 (links) schematisch ein Kollaborationsnetzwerk bei zentraler PM-Speicherung dargestellt. Die Notation $V_z(T_y).x$ bei den Produktmodellen bedeutet, dass die x -te Version eines Vorschlags z durch den Teilnehmer y im Netzwerk abgegeben worden ist.
- Bei der *hybriden PM-Speicherung* verwaltet jeder Teilnehmer seine Modelle in Eigenverantwortung. Entsprechend würde im zugehörigen gGPM das Datenelement Spezifikation beim Initiator vorgesehen, während der Vorschlag auf die Seite des Teilnehmers zu setzen wäre. Diese Variante der PM-Speicherung liefert die Option, bestimmte PM-Vorschläge zur Run-Time in einer (Sub-)Kollaboration mit ausgewählten Akteuren auszutauschen (vgl. Abbildung 4, rechts).
- Bei der *dezentralen PM-Speicherung* werden Spezifikationen und Vorschläge „beliebig“ im Netzwerk verteilt (dezentrales Verfahren). Dazu werden im gGPM weder Spezifikation noch Vorschlag einem konkreten Pool zugeordnet.

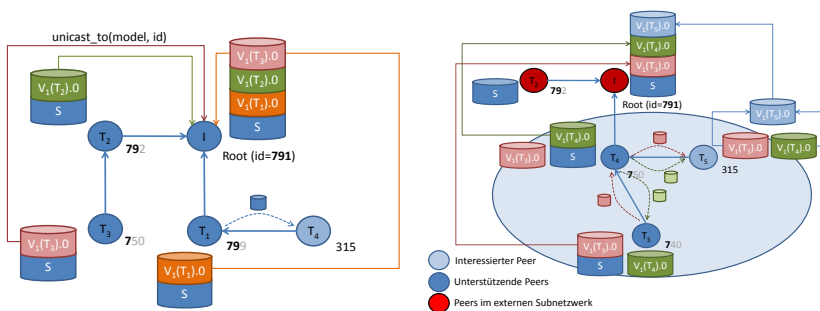


Abbildung 4: Schematische Darstellung der zentralen und hybriden Produktmodell-Verteilung in einem Pastry-P2P-Overlay

Hierarchien: Ein Teilnehmer kann ein durch den Initiator gegebenes Teilproblem weiter zerlegen und mit ausgewählten Partnern in einem Subnetzwerk „separiert“ entwickeln. Im gGPM initiiert ein Teilnehmer A dazu einen Prozess, in der eine Menge von Sub-Spezifikationen nach dem bekannten Verfahren mit weiteren Teilnehmern B,C, usw. ausgetauscht wird. Die Anzahl zusätzlicher Teilnehmer und Hierarchieebenen ist theoretisch unbeschränkt. Hierarchien können grundsätzlich in allen Datenverteilungsformen (zentral/ dezentral/ hybrid) existieren.

Iterationen: Iterationsfreie Kooperationsprozesse entsprechen nicht der Realität. Die verteilte Entwicklung eines Produktmodells benötigt in der Realität für gewöhnlich eine sehr hohe Anzahl an Iterationen. Beispielsweise stellt der Initiator nach Erhalt der ersten Vorschläge fest, dass die von ihm ausgeschriebene Spezifikation zu ungenau, bzw. schlecht formuliert war und muss diese verbessern. Diese Verbesserung wird im Allgemeinfall auch als *PM-Version* bezeichnet. Ebenfalls kann der Initiator ausgewählte Teilnehmer bitten, Vorschläge zu überarbeiten. Dafür werden zwei neue Prozessschritte („Spezifikation überarbeiten“ und „Vorschlag überarbeiten“), sowie entsprechende Entscheidungs-Verzweigungen eingeführt.

4.2 Architekturformen für die OMP

In Anlehnung an (Roser 2008) unterscheiden wir bei der Ausführung eines Dienstes zwischen der

- *lokalen Ausführung* aller kollaborationsrelevanten Dienste und der
- *verteilten Ausführung* der Dienste durch die Kollaborationsteilnehmer.

Lokale Ausführung bedeutet, dass jeder Kooperationspartner über sämtliche für die Kooperation notwendigen OMP-Basisdienste verfügt. Das hat zur Run-Time den Charme, dass zu jedem Zeitpunkt sichergestellt ist, dass die durch einen Teilnehmer zu erbringenden, notwendigen Basisdienste jederzeit lokal bereit stehen und ausgeführt werden können. Sind die Basisdienste verteilt über die Teilnehmer, so muss zur Run-Time immer zuerst sichergestellt werden, dass ein benötigter Dienst auch tatsächlich zur Verfügung steht und in Anspruch genommen werden kann. Für die OMP kommen bei der *verteilten Ausführung von OMP-Basisdiensten* zwei Varianten in Fragen:

- Entweder stellen ausgewählte Knoten (Super-Peers) die Dienste einer Gruppe dauerhaft zur Verfügung und können dort in Anspruch genommen werden oder (*zentrale Ausführung*) oder
- die Dienste liegen vollständig dezentral im Netzwerk und können über eine verteilte Suche bei Bedarf gefunden und zur Nutzung angefragt werden (*dezentrale Ausführung*).

Weiterhin kann unterschieden werden, ob der *globale Workflow (gWF)* zentral durch einen kooperationsübergreifenden Koordinator gesteuert wird, oder ob er dezentral durch mehrere unabhängige Koordinatoren vorangetrieben wird. Im Detail bedeutet dies folgendes:

- In einer *zentral koordinierten Kooperation* wird der globale Workflow (gWF) zentral von einem Koordinator gesteuert. Er ordnet zur Run-Time lokale und/oder verteilt vorliegende Basisdienste in einer Choreographie und kontrolliert deren Ausführung. Diese Variante verlangt, dass zur vor Beginn der Kooperation die Teilnehmer ihr Basisdienst-Angebot an den Koordinator melden.
- In einer *dezentral koordinierten Kooperation* soll auf den zentralen Koordinator verzichtet werden. Jeder Kooperationsteilnehmer ist damit selbst dafür verantwortlich, einen Basisdienst zur richtigen Zeit auszuführen und den im Rahmen des gGPM erwarteten Beitrag zu leisten. Zur Run-Time erfolgt die Entscheidung welcher verteilt vorliegende Dienst ausgeführt werden muss (bzw. kann) über den Zustand des virtuellen Produktmodells (vPM). Verändert sich dieses, so werden beteiligte Produktentwickler darüber informiert und können lokal entscheiden, ob und wann sie einen Basisdienst starten. Bei jedem Teilnehmer gibt es einen speziellen Basisdienst, den sogenannten Kooperationsdienst (engl.: Cooperation Service, CS), der den Status ausgewählter Produktmodell-Partitionen (Spezifikationen und Vorschläge) des vPMs überwacht. Genau dieser CS ist dafür verantwortlich, das Verändern des Produktmodells zu beobachten und die Teilnehmer über das overlayspezifische Nachrichtensystem darüber zu informieren.

Die Fälle der zentral koordinierten Kooperationen werden in diesem Papier nicht weiter betrachtet. Die Abbildung 5 zeigt beispielhaft den Fall der *dezentral koordinierten Kooperation bei lokaler Basisdienst-Ausführung* verdeutlichen. In der Abbildung sind drei Peers dargestellt. Jeder Peer kennt den globalen Workflow (gWF) der Kooperation und implementiert alle für die Kooperation notwendigen Basisdienste (Service S_x). Der gWF ist farbig dargestellt, dabei markiert eine Farbe die durch einen Teilnehmer ausgeführten, lokalen Workflows (OMP-Basisdienste). So führt im Beispiel die grün markierten Basisdienste der Peer1 aus (zum Beispiel $S_1(\text{Peer1})$ = Beispiel das Erstellen und Veröffentlichen einer Spezifikation; $S_2(\text{Peer1})$ = Eingegangene Vorschläge überprüfen), während der rot gefärbte Basisdienst (zum Beispiel $S_1(\text{Peer2})$ = Erstellen und Veröffentlichen eines Vorschlags) durch den Peer2 bereitgestellt wird.

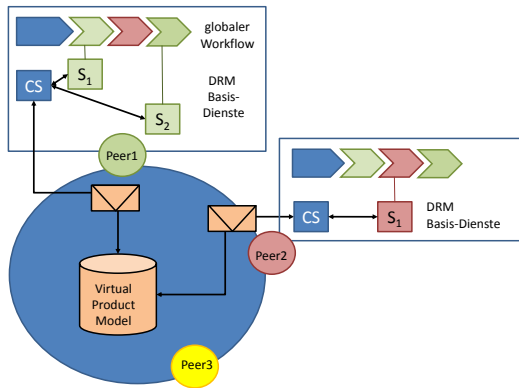


Abbildung 5: Architekturmodell für eine dezentral koordinierte Kollaboration bei lokaler Dienstausführung

Da bei lokaler Dienstausführung jeder Kooperationspartner über sämtliche Basisdienste verfügt (= lokale Dienstausführung), kann er diese direkt lokal nutzen, jedoch nicht zu einem beliebigen Zeitpunkt. Die Basisdienste stehen ähnlich wie in einer Choreographie in einer Reihenfolge, mit dem Unterschied, dass es hier keinen Koordinator gibt, der diese zentral steuert. So definiert der gWF im Beispiel die folgende Reihenfolge: $S_1[\text{Peer1}] \rightarrow S_1[\text{Peer2}] \rightarrow S_2[\text{Peer2}]$. Ein Dienst kann jedoch nur dann ausgeführt werden, wenn das virtuelle Produktmodell (vPM) einen erwarteten Zustand erreicht hat, der Collaboration-Service (CS) dieses registriert und einen passenden, lokalen Dienst aktiviert.

4.3 Plattformspezifische Modelle für die OMP (PSM4OMP)

In unseren Arbeiten haben wir exemplarisch damit begonnen, die in Abbildung 5 dargestellte Architektur im PSM zu modellieren und zu evaluieren (vgl. Abbildung 6). Bei den OMP-Basisdiensten wird unterschieden zwischen fachlichen und technischen Diensten. Beispielsweise kann der fachliche Dienst „CreateSpecification“ bedeuten, dass eine Spezifikation aus einer lokalen Instanz eines Produktdatenmanagementsystems (z.B. Teamcenter Engineering, TCE) erstellt wird. Dieser Dienst ist also speziell durch einen Teilnehmer zu implementieren, der entsprechende Kenntnis über diesen Vorgang hat.

Weiterhin gibt es eine Reihe an technischen Diensten, wie „Publish“, „Search“, „Subscribe“ und „Notify“, die in ihrer Implementierung bei allen Peers gleich und daher auch allen bekannt sind (vgl. Stiefel 2007). Jedem Entwicklungspartner ist somit ein IWF zugeordnet, der die fachlichen und technischen Services so miteinander verknüpft, dass das im Geschäftsprozess definierte Verhalten umgesetzt wird.

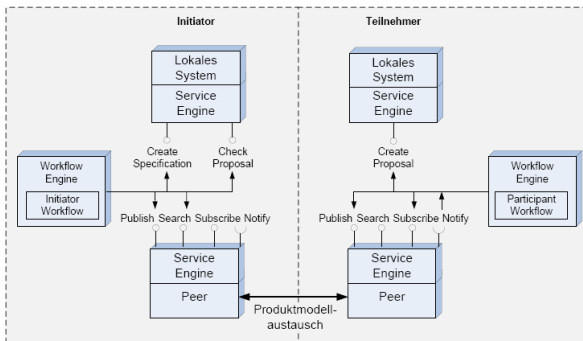


Abbildung 6: Workflowmodell einer dezentral koordinierten Kollaboration bei lokaler Dienstauführung

4.4 Ein plattformabhängiges Modell für die OMP (PIM4OMP)

Die Abschnitte 4.1 - 4.3 zeigen, dass die Modelle auf CIM- und PSM-Ebene variabel sind. Diese Variabilität kann durch Parameter beschrieben werden, deren konkrete Ausprägung durch den Geschäftsprozessdesigner einerseits (top-down) und/oder durch den IT-Experten andererseits (bottom-up) festgelegt wird. Ein Beispiel demonstriert die Top-Down-Variante. Dazu definiert ein Geschäftsprozessdesigner die folgenden zwei Varianten der OMP (A und B) und setzt die in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellten Parameter für die angegebenen Dimensionen.

Tabelle 1: Dimensionen für zwei Varianten der OMP

<i>Dimensionen</i>	<i>OMP Variante „Entwicklungsmodell“</i>	<i>OMP Variante „Zielmodell“</i>
Anzahl max. Teilnehmer (davon Sub-Initiatoren)	100 (10)	1000 (1)
Anzahl Spezifikationen	10	1000
Anzahl an Vorschlägen pro Spezifikation und Teilnehmer (Iterationen)	10	100
Kooperationsunabhängige Teilentwicklungen erlaubt (Sub-Netzwerke)	Ja	Nein
Teilnehmerverhalten	Geheim/ nur kooperativ zum Initiator	Kooperativ

Variante A ist der zuvor eingeführte, bereits bekannte Fall, während in Variante B ein *PM-Austauschs mit dem Ziel der Findung gültiger Zielmodelle* beschrieben wird. Im zweiten Fall handelt es sich nur bedingt um einen zentral steuerbaren Prozess, da eine Reihe an (Sub-)Initiatoren involviert sind, um viele stark iterativ erzeugte Teilergebnisse parallel voranzutreiben.

Die Herausforderung von PIM4OMP ist es nun, basierend auf den durch den Geschäftsprozessdesigner festgelegten Parametern für die einzelnen, kollaborationsrelevante Dimensionen (vgl. Tabelle 1: Dimensionen für zwei Varianten der

OMP.) eine möglichst vollständig automatisierte Transformation in einen passenden Architekturmodell zu ermöglichen. In diesem Fall ist zu erwarten, dass für die OMP-Variante A das Architekturmodell mit lokaler Dienstauführung und für die Variante B ein Modell mit hybrider Dienstauführung gewählt wird. Das Metamodell von PIM4OMP, aus Platzgründen stark vereinfacht dargestellt in Abbildung 7, besteht aus den folgenden vier Teilen: *Organisation, Information, Prozess und Overlay*.

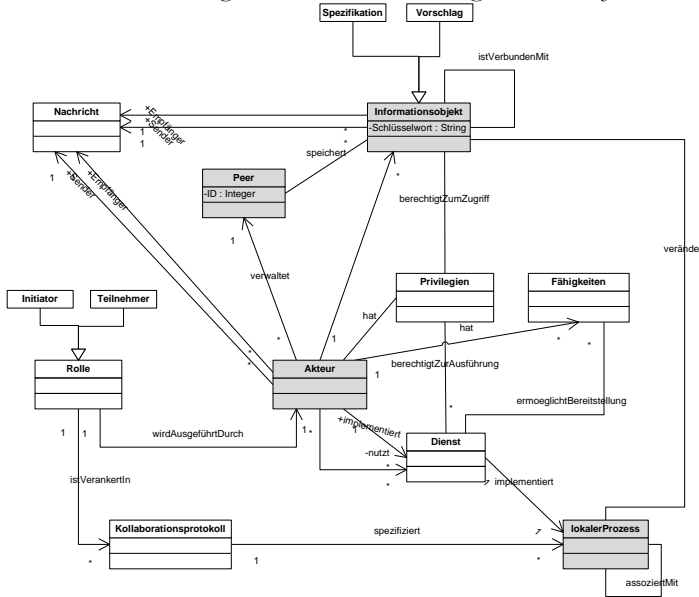


Abbildung 7: PIM4OMP Metamodel

Im Metamodell für die Organisation werden die Formen der Zusammenarbeit zwischen den einzelnen *Akteuren* beschrieben. *Akteure* besitzen Rollen, die Ihre Aufgabe in der Kollaboration über das *Kollaborationsprotokoll* festlegen. Im Informationsmetamodell werden sämtliche für den kooperativen Produktentwicklungsprozess notwendigen *Informationsobjekte* abgebildet. Ein *Informationsobjekt* repräsentiert den produktstrukturbasierten Aufbau eines Produktmodells und Zusammenhänge zwischen *Spezifikation* und korrespondierenden *Vorschlägen*. Jedes Informationsobjekt wird einem bestimmten *Akteur* zugeordnet und auf einem assoziierten *Peer* gespeichert. Der Austausch von Informationsobjekten erfolgt über Nachrichten, die ebenfalls für den herkömmlichen Informationsaustausch zwischen *Akteuren* genutzt werden können. Im Prozessmetamodell werden *lokale Prozesse* abgebildet, die aneinander gereiht einen rollenspezifischen, lokalen Geschäftsprozess (IGP) aus dem gGP abbilden. Alle IGP's werden spezifiziert im *Kollaborationsprotokoll*. Jeder abgeschlossene *lokale Prozess* verändert das *Informationsobjekt*. Das Architekturmetamodell legt fest, welcher konkrete *Akteur* in einer speziellen *Rolle* einen *Dienst* implementiert und dem Netzwerk bereitstellt (sofern die dafür notwendigen Fähigkeiten vorhanden sind) oder aber einen *Dienst* eines Dritten nutzt (sofern er

über die entsprechenden Privilegien verfügt). Jeder Akteur ist Besitzer eines *Peers*. Das Architekturmetamodell ist Ausgangspunkt für die Abbildung spezifischer P2P-Overlays.

5 Bewertung und Zusammenfassung

Wir haben in diesem Papier gezeigt, wie mit Hilfe des Vorgehensmodells der MBE eine dezentral organisierte Kollaborationsplattform für die OMP entwickelt werden kann. Dazu wurden Modellansätze auf den unterschiedlichen Abstraktionsebenen der MBE vorgestellt. Bei der abschließenden Bewertung des MBE-Ansatzes sollen im Wesentlichen zwei Aspekte betrachtet werden:

1. In wie weit eignet sich das modellgetriebene Vorgehensmodell für den Entwurf einer dezentral organisierten Kooperationsplattform. Dabei wird ein Blick auf die erzielte Qualität des Endprodukts geworfen, um darüber zu beurteilen, ob die vorgestellten Modelle auf CIM-, PIM- und PSM-Ebene ausreichend sind. Wir messen Qualität anhand von zwei Kriterien: a) Verhält sich die Software entsprechend der Erwartungen der Modellierer und b) Sind die Modelle flexibel genug, um die dezentrale Infrastruktur abzubilden?
2. Wie gut unterstützt die Kooperationsplattform die vorgestellten Prozesse im Vergleich zu herkömmlichen client-/serverbasierten Kooperationsplattformen?

Zum Punkt 1a:

Ob die Software das erwartete und durch die Modellierer festgelegte Verhalten zeigt, lässt sich durch Tests und Simulationen auf Basis unserer prototypischen Implementierung belegen. Dabei ist unter anderem auf folgende Kriterien zu achten:

- Ist das im OMP definierte Kooperationsziel erzielt wurden? Wenn nein, war der Geschäftsprozess unzureichend modelliert?
- Entspricht die erreichte Produktmodellverteilung dem Verteilungsmodell?
- Sind die Workflows korrekt umgesetzt wurden: Entspricht der tatsächliche Verlauf der Kooperation dem geplanten Verlauf?

Zum Punkt 1b:

Die Frage, ob die Modelle flexibel genug sind, lässt sich im dezentralen Fall nicht einfach beantworten. Die hier entstehende Systemdynamik ist bereits zur Entwurfszeit zu berücksichtigen; sie trägt je nach erzielter Abbildungsgenauigkeit maßgeblich zum Erfolg des Kooperationssystems bei. Während der Modellierung zur Entwurfszeit kann nicht deterministisch festgelegt werden, wie lange die Dienste eines *Peers* dem Netzwerk zur Laufzeit zur Verfügung stehen: Jeder Akteur einer OMP reagiert autonom. Das dadurch entstehende *dynamische Systemverhalten*, das eine Eigenschaft des P2P-Netzwerks ist, muss auf allen Modellierungsebe-

nen der MBE berücksichtigt werden. Wir erreichen das einerseits dadurch, dass unabhängig vom Abstraktionsgrad der Modelle, jede zu einem Peer zugeordnete und von diesem zu erfüllende Aufgabe (sei es ein lokaler Abschnitt eines kooperativen Geschäftsprozesses der OMP oder ein daraus entwickelter, lokaler Abschnitt des entsprechenden Workflows) in sich abgeschlossen modelliert wird. Das Zusammenspiel der einzelnen, unabhängig ausgeführten Teilaktivitäten wird durch P2P-basierte Kommunikation erreicht, die in der Regel über das Produktmodell selbst erfolgt. Daher sprechen wir im Rahmen der OMP auch immer von modellzentrierten Prozessen. Weiterhin modellieren wir Dynamik durch die Abbildung des erwarteten Verhaltens des P2P-Netzwerks bereits im PIM (Quality of Service). Das ermöglicht uns später einerseits passende Overlay-Strukturen zu selektieren und andererseits die den OMP entsprechenden Workflows/ Service-Verteilungen so gut wie möglich darauf anzupassen.

Folgende QoS-Parameter spielen unter anderem dabei eine Rolle: Der Parameter *Verfügbarkeit* sichert zur Run-Time zu, dass Produktmodelle beim Nachfragen mit hoher Wahrscheinlichkeit vorliegen. Dazu sind zum Beispiel spezielle Teams zu definieren, innerhalb derer Replikat ausgetauscht oder die Rollen der Akteure vererbt werden können. Auch Warteschlangenmechanismen können genutzt werden, um temporäre Nicht-Verfügbarkeiten auszuschließen. Der Parameter *Skalierbarkeit* sichert eine effiziente Produktmodellverteilung im Netzwerk in Abhängigkeit der festgelegten Teilnehmerstruktur und bietet bei Bedarf spezielle Content-Delivery-Mechanismen, zum Beispiel durch Erstellen und Verteilen von Datenchunks für besonders große CAD-Dateien. *Lokalisierung* dient dem Bereitstellen effizienter Suchmechanismen zum Auffinden von Teilnehmern und/oder Produktmodellen in Abhängigkeit von der Anzahl der Produktentwickler im Netzwerk (Wildcard-Suche, Exact-Match-Suche, Range-Query, usw.). Mit Vertrauenswürdigkeit sicher ein letzter QoS-Parameter das Vorhandensein von Private-Key-Infrastrukturen (PKI) für sensiblen Produktmodelaustausch oder die Möglichkeit des Bildens voneinander getrennter Subnetzwerke. Wie gut im implementierten Ansatz die QoS-Anforderungen durch das Overlay erfüllt werden, lässt sich durch geeignete Metriken in Bezug auf *Adaptivität (Skalierbarkeit/ Stabilität/ Flexibilität)*, *Effizienz*, *Validität und Vertrauenswürdigkeit (Verlässlichkeit/ Sicherheit)* des P2P-Overlays über Simulationen nachweisen (QuaP2P 2009).

Zum Punkt 2:

Abschließend hängt die Akzeptanz der OMP im Unternehmen von vielen Faktoren ab, denen man sich bei der Entwicklung des Ansatzes kritisch stellen muss und die abschließend noch einmal erwähnt werden sollen. Vorteile des P2P-Ansatzes gegenüber den etablierten Client-/Server-Ansätzen lassen sich in Bezug auf Skalierbarkeit, Flexibilität, Datenverteilung, und Antwortzeitverhalten sehen. Insbesondere in sensiblen Kooperationsumgebungen (wie in der verteilten Produktentwicklung zu finden, reichen diese Vorteile offensichtlich jedoch nicht aus, um eine P2P-basierte Kooperationsplattform im Unternehmen zu rechtfertigen.

Nach der Veröffentlichung des Prototyps durchgeführte Befragungen liefern als Hauptgründe:

- die Angst der Entwicklungsingenieure, durch P2P-Verteilung die Kontrolle über die Produktdaten zu verlieren und
- das noch nicht genügend ausgeprägte Bewusstsein für die Notwendigkeit einer Öffnung der Prozesse für effizientere Zusammenarbeit.

Der in dieser Arbeit vorgestellte Ansatz der MBE soll helfen, diese Barriere zu überwinden. Dazu ist es notwendig, dass sich der Anwender eines dezentralen Systems bereits in der Phase der Modellierung über das Verhalten der Akteure der Kooperation und den daraus entstehenden Konsequenzen Gedanken macht. Die Idee unserer Arbeiten ist es, den Entwickler und Designer organisationsübergreifender Kooperationsprozesse P2P-basierte Produktmodellentwicklung technologieunabhängig auf Geschäftsprozess-Ebene in einer ihm vertrauten Darstellung (z.B. durch BPDs) modellieren zu lassen. Ohne IT-Kenntnisse entscheidet der Designer darüber, wie später ein konkretes P2P-Netzwerk seine individuellen OMP-Prozesse unterstützt.

Literatur

- Albers, Albert; Schweineberger, Dirk (1998): Effektives Kooperationsmanagement in der Produktentwicklung. IIR, Frankfurt.
- Allweyer, Thomas (2008): BPMN - Business Process Modeling Notation. Einführung in den Standard für die Geschäftsprozessmodellierung. Books on Demand, Norderstedt.
- Erl, Thomas (2006): Service-oriented architecture. Concepts, technology, and design. 5th print. Prentice-Hall, Upper Saddle River, NJ.
- Grabowski, Hans; Anderl, Reiner; Polly, Adam (1993): Integriertes Produktmodell. 1. Aufl. Beuth, Berlin.
- Gruhn, Pieper, Röttgers (2006): MDA®. Effektives Software-Engineering mit UML 2® und Eclipse. <http://dx.doi.org/10.1007/3-540-28746-9>.
- Kempa, Martin; Mann, Zoltán Adám (2005/08/01/): Model Driven Architecture. In: Informatik-Spektrum. 28 (4), S. 298–302.
- Krause, Frank-Lothar (2007): Innovationspotenziale in der Produktentwicklung. Hanser, München.
- Li, W.D. and Qiu, Z.M. (2006): State-of-the-Art Technologies and Methodologies for Collaborative Product Development Systems. In: International Journal of Production Research, 44(13): 2525-2559.

- Li W.D., Ong S.K. and Nee A.Y.C. (2006): Integrative and Collaborative Product Development Environment. Technologies and Implementations. World Scientific Publishing Co. Pte. Ltd. Singapore.
- Lindemann, Udo (2007): Methodische Entwicklung technischer Produkte. In: Methoden flexibel und situationsgerecht anwenden. 2., bearb. Aufl. Springer, Berlin.
- Müller, J. P. (1996): The Design of Intelligent Agents – a Layered Approach. Lecture Notes in Artificial Intelligence, Volume 1177, Springer-Verlag.
- QuaP2P (2009): Arbeitsberichte der Forschergruppe QuaP2P über ihr gleichnamiges Forschungsprojekt an der Universität Darmstadt. <http://www.quap2p.tu-darmstadt.de/> [01.09.2009]
- Radtke, Philipp; Abele, Eberhard; Zielke, Andreas E (2004): Die smarte Revolution in der Automobilindustrie. Das Auto der Zukunft, Optionen für Hersteller, Chancen für Zulieferer. Ueberreuter, Frankfurt.
- Roser (2008): Designing and Enacting Cross-organisational Business Processes. Dissertation, Uni Augsburg.
- Schrage, Michael (1990): The New Technologies of Collaboration. Random House, NY, p. 140ff.
- Smith, Reid G. (1981): Frameworks for Cooperation in Distributed Problem Solving. In: IEEE Transactions on systems, man, and cybernetic, Volume 11, Seiten 61-70.
- Stäber (2008): Service layer components for decentralized applications. Dissertation, TU Clausthal.
- Stiefel, Patrick D.; Müller, J. P. (2007): ICT interoperability challenges in decentral, cross-enterprise product engineering. In: Gonçalves; Ricardo J., S. 171–182. http://dx.doi.org/10.1007/978-1-84628-858-6_18.
- Stiefel, Patrick D.; Müller J.P. (2008): Realizing dynamic product collaboration processes in a model-driven framework: Case study and lessons learnt, in K.-D. Thoben, K. S. Pawar, & R. Gonçalves, eds., 14th International Conference on Concurrent Enterprising, , 23-25 June 2008, Lisbon, Portugal.
- Teufel, Stefanie; Sauter, Christian; Mühlherr, Thomas (1995): Computerunterstützung für die Gruppenarbeit. Oldenbourg.
- Tietze, Oliver (2003): Strategische Positionierung in der Automobilbranche. Der Einsatz von virtueller Produktentwicklung und Wertschöpfungsnetzwerken. 1. Aufl. Dt. Univ.-Verl., Wiesbaden.
- Winer M., Ray K. (1994): Collaboration Handbook: Creating, Sustaining, and Enjoying the Journey. Fieldstone Alliance, 1st edition.