

# Projektbezogenes Produktdatenmanagement

Ralph Stelzer,  
Dirk Petermann,  
Bernhard Saske und  
Matthias Gubsch, Dresden

Das hier beschriebene Projekt ist ein Verbundvorhaben der xPLM Solution GmbH und des Lehrstuhls für Konstruktionstechnik/CAD der TU-Dresden im Rahmen der F&E-Verbundprojektförderung des Freistaates Sachsen. Dieser Artikel beschreibt ein Konzept zur Unterstützung der unternehmensübergreifenden Zusammenarbeit bei der Entwicklung von Produkten. Neben der Konzeption eines Projektdaten-Repository erfolgt die Beschreibung eines Werkzeugs zur Modellierung unternehmensübergreifender Prozesse. Mit diesem Werkzeug wird durch die Bereitstellung systemunabhängiger Funktionen eine flexible Kommunikation zwischen Anwendern und Anwendungen während der Zusammenarbeit erreicht. Mit der Nutzung dieser Funktionen wird ein konsistentes Produktdatenmodell über Anwendungsgrenzen hinweg realisiert, was in einem Projekt- und Systemumfeld von essenzieller Bedeutung ist. Dies unterstützt die kollaborative Entwicklung neuer Produkte, da Projektpartner (z. B. Zulieferer) translokal in den Entwicklungsprozess eingegliedert werden können.

## Problemstellung

Die fortschreitende Globalisierung von Unternehmen und Geschäftsprozessen führt zu einem fortlaufenden Wandel der Arbeitsweisen in der Produktentwicklung. Neue Produkte werden immer häufiger verteilt entwickelt. Dabei ist von untergeordneter Bedeutung, ob unterschiedliche Abteilungen einer Firma oder mehrere Unternehmen miteinander kooperieren.

Ein Projektraum, in dem Produktdaten und Prozesse in einem Projektdaten-Repository gespeichert werden, unterstützt die Beurteilung des erreichten Entwicklungsstands und die Planung weiterführender Aktivitäten zwischen den Projektpartnern. Ausschlaggebend ist hierfür eine anwendungsneutrale Speicherung der den Prozessen zugrundeliegenden Produktdaten innerhalb einer Datenschicht. Die zur Manipulation der Daten notwendige Prozessschicht ist Bestand-

teil einer Integrationsplattform, die am Entwicklungsprozess beteiligte Applikationen (z. B. CAD, PLM, VR) eingliedert (vgl. [1]).

Verteilte Entwicklerteams werden durch einen translokal zu erreichenden und gemeinsam nutzbaren Projektraum unterstützt, wenn er:

- projektbezogene Produktdaten (z. B. Geometrie-, Simulations- und Berechnungsdaten) und
- unternehmensübergreifende Prozesse (z. B. Freigabe und Verteilung der Produktdaten)

in einem Projektdaten-Repository verwalten kann. Die schematische Beschreibung eines Szenarios aus verteilten Entwicklerteams und die Umsetzung der daran gestellten Anforderungen sind Inhalt des folgenden Abschnitts.

## Projektdaten-Repository

Bild 1 zeigt ein Kollaborationsszenario zwischen einem Originalteilehersteller (OEM) und zwei Zulieferern (Supplier).

Die gemeinsame Entwicklungsumgebung ist ein Projektraum, der ein Projektdaten-Repository enthält. Mittels ei-

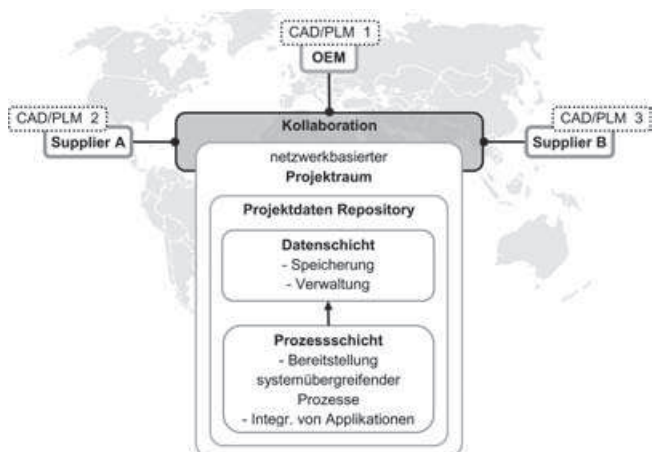


Bild 1. Kollaborationsszenario mit einem Projektdaten-Repository

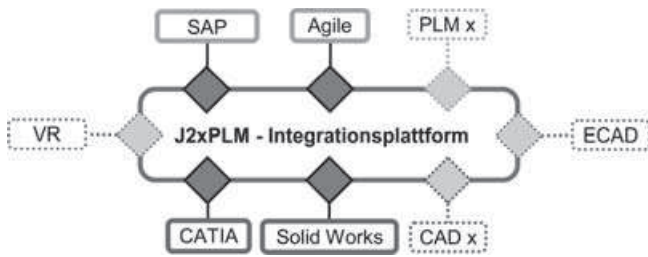


Bild 2. Schnittstellen der J2xPLM-Integrationsplattform

ner netzwerkbasierter Architektur wird die translokale und permanente Erreichbarkeit des Projektraums gewährleistet.

Wie in Bild 1 gezeigt, ist das Projektdaten-Repository aus zwei Schichten aufgebaut. Es beinhaltet eine Datenschicht als Speicher für Projektdaten sowie eine Prozessschicht, die mit Hilfe einer Integrationsplattform Daten externer Applikationen in der Datenschicht speichert. Zudem ist die Prozessschicht in der Lage, transferierte Produktdaten durch systemübergreifende Prozesse zu verwalten. Die Modellierung und die Ausführung systemübergreifender Prozesse bilden Teil der Prozessschicht, die Speicherung ist aber Teil der Datenschicht.

In den folgenden Abschnitten werden die Integration von Produktdaten, die Produktdatenverwaltung sowie die Modellierung und Bereitstellung systemübergreifender Prozesse beschrieben.

**Integrationsplattform**

Die Speicherung unterschiedlicher Produktdaten innerhalb der Datenschicht wird durch die Einbeziehung einer Integrationsplattform realisiert. Die J2xPLM-Integrationsplattform der Firma xPLM Solution GmbH besitzt eine generische Schnittstellenarchitektur zur Integration verschiedener Software-Werkzeuge und eignet sich deshalb für die CAD/PLM-Integration. Sie dient sozusagen als Kommunikator zwischen den verschiedenen CAD- und PLM-Anwendungen, die bei der verteilten Arbeit zum Einsatz kommen. Die vereinfachte Architektur der J2xPLM-Integrationsplattform mit angebotenen Anwendungen gibt Bild 2 wieder:

Die J2xPLM-Integration besteht grundlegend aus drei Komponenten. Kernstück der Integration ist die J2xPLM-Plattform selbst. An diese werden dann die Applikationen mit Hilfe eines entsprechenden Interfaces eingebunden. Diese Schnittstellen werden unter Verwendung der API (Application Programming Interface) erstellt, wodurch die J2xPLM-Integrationsplattform eine Er-

weiterung der jeweiligen Systeme darstellt. Sie bietet dabei die Flexibilität, unter Einsatz verschiedener Interfaces auch mehrere Client-Applikationen und PLM-Lösungen zu integrieren [2].

**Datenverwaltung**

Neben der Integration von Entwicklungswerkzeugen sind die zentrale Speicherung und Verwaltung der generierten Daten innerhalb der Datenschicht notwendig. Hier bietet sich beispielsweise die Nutzung einer vorhandenen PDM/PLM-Lösung an. Entscheidend ist, dass die bestehenden Eigenschaften so erweitert werden, dass eine systemneutrale Verwaltung der Dokumentdaten unterstützt wird. Nur so ist die Replikation der Produktdaten aller Kollaborationspartner innerhalb des Projektraums konsistent.

Die Vergabe von Zugriffsrechten ist ein weiterer Teilaspekt der Datenverwaltung. Sie soll dem Missbrauch vorbeugen und die Rechte der Projektpartner an ihren Produkten sichern. Ein Projektdaten-Repository bzw. der Projektraum ist deshalb als Informationssystem für ge-

schlossene Benutzergruppen zu konzipieren, sodass nur zugriffsberechtigte Projektbeteiligte Dokumente und Informationen verwalten können. Die Zugriffseinschränkung kann zum Beispiel über sichere Authentifizierungsverfahren sowie rollenbasierte projekt- und personenspezifische Zugriffsmechanismen realisiert werden (vgl. [3]).

Da Struktur- und Geometriedaten mit Hilfe der CAD-PLM-Integration auf Basis der J2xPLM-Integrationsplattform gespeichert werden, empfiehlt sich als zentraler Speicherort die Verwendung einer PDM/PLM-Lösung, deren Interface bereits Teil der Integrationsplattform ist. So stehen sämtliche verwaltende Funktionen der modernen Produktentwicklung zur Verfügung und ein zusätzlicher Implementierungsaufwand entfällt.

**Systemübergreifende Prozesse**

In der Wirtschaft existiert eine Vielzahl von CAD-PDM-Systemlandschaften zur Verwaltung von produktbeschreibenden Daten und unternehmensspezifischen Prozessen. Sobald Abteilungen bzw. Unternehmen kooperieren, ist auch die Beschreibung von unternehmensübergreifenden Prozessen erforderlich. Da sich diese Prozesse mit dem Verlauf der Zusammenarbeit dynamisch verändern, müssen sie möglichst flexibel und unabhängig von einer spezifischen PDM/PLM-Lösung beschrieben werden.

Systemübergreifende Prozesse sind die Summe von verknüpften Funktionen

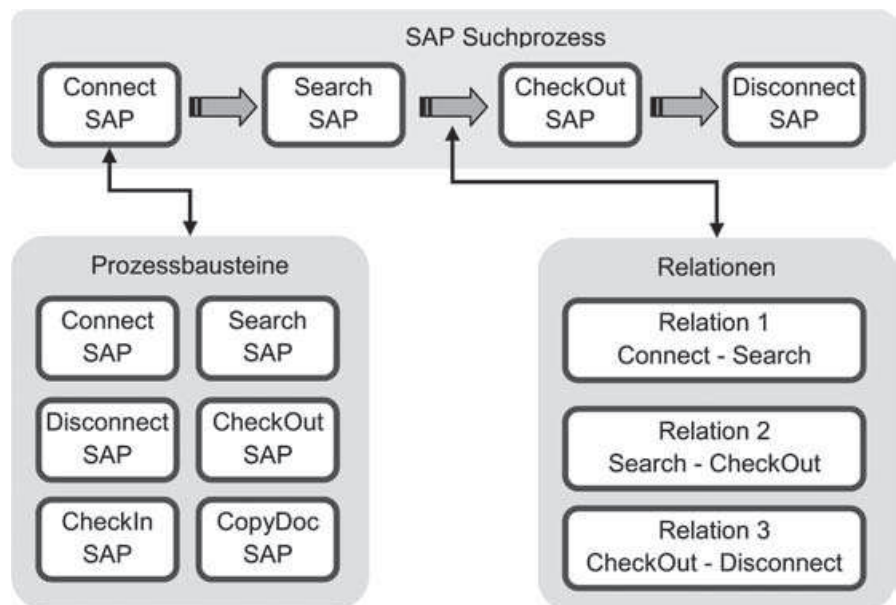


Bild 3. Modulare Prozessdefinition

der Prozessschicht. Um diese Funktionen flexibel und effizient für die Modellierung von Prozessen einzusetzen, ist die Nutzung vorgefertigter Prozessbausteine ein möglicher Ansatz. Der Funktionsumfang eines solchen Prozessbausteins ist auf einem elementaren Level zu begrenzen, da dadurch die Möglichkeit zur Wiederverwendung einzelner Prozessbausteine erhöht wird. Prozessbausteine werden zentral im Projektdaten-Repository verwaltet und ermöglichen so die Bildung einer universellen Bibliothek. Bild 3 beschreibt beispielhaft die Bildung eines Suchprozesses im SAP.

Zur Definition eines zusammenhängenden Prozesses werden notwendige Prozessbausteine aus der Bibliothek entnommen und zu einem Prozess verbunden. Der Prozess bleibt dabei an neue Randbedingungen anpassbar, denn Prozessbausteine können entnommen, hinzugefügt oder neu verknüpft werden.

Die Verwendung einer zentralen projektbezogenen Prozessbausteinbibliothek ist sowohl mit vorkompilierten Prozessbausteinen als auch mit solchen, die zur Laufzeit interpretiert werden, möglich. Inwiefern sich dieser Unterschied auf die Modellierung und Verwendung der systemübergreifenden Prozesse auswirkt, soll durch zwei Prototypen unter-

sucht werden. Vorlage für die prototypisch implementierten Prozessbausteine sind die Funktionen der CAD-PLM-Integration, wie z.B. das Einchecken, Auschecken und Suchen von Dokumenten bzw. das Verbinden und Trennen von PDM/PLM-Anwendungen.

### Prototypen

Im folgenden Kapitel werden zwei Prototypen zur Modellierung und Verwendung systemübergreifender Prozesse vorgestellt.

Der erste Prototyp, dessen Prozessbausteine zur Laufzeit interpretiert werden, basiert auf der standardisierten Prozessbeschreibungssprache BPEL (BPEL-Prototyp). Der BPEL-Standard selbst definiert auf Grund seiner XML-basierten Beschreibung jedoch keine grafische Darstellung der modellierten Prozesse. Aus diesem Grund wurde der BPEL-Process Manager von Oracle für die Darstellung und Implementierung der Prozesse ausgewählt.

Der zweite Prototyp benutzt vorkompilierte Prozessbausteine. Diese werden mit Hilfe einer höheren Programmiersprache (C++) implementiert und nach der Kompilierung in die Software EON-Studio registriert. EON-Studio ist eine

Autoren-Software für das Entwickeln von interaktiven 3D-Anwendungen. Bestandteil ist ein leistungsfähiger Editor zur Modellierung von Signalflüssen zwischen einzelnen Elementen einer Anwendung.

### BPEL basierter Prototyp

Die im Jahr 2002 von IBM, BEA Systems und Microsoft eingeführte Sprache BPEL (Business Process Execution Language) wird zur Beschreibung von so genannten Web Service-Orchestrierungen verwendet. BPEL ist eine XML-basierte Prozessbeschreibungssprache, die über alle Möglichkeiten einer imperativen Programmiersprache verfügt, jedoch speziell im Hinblick auf Web Services entwickelt wurde.

Web Services nutzen Remote Procedure Calls (RPC) bzw. eine nachrichtenbasierte Kommunikation, um die Interoperabilität von in einer Firma oder über mehrere Firmen verteilten Prozessen zu erleichtern und zu verbessern. Web Services reichen jedoch nicht aus, um logische Abläufe zu steuern. Genau da setzt BPEL an, denn es bietet die Möglichkeit, Web Services durch Komposition bereits bestehender Web Services als abstrakte und ausführbare Prozesse zu definieren (vgl. [4]).

Um einen systemübergreifenden Prozess mit BPEL zu erzeugen, müssen zuvor die elementaren Einzelprozesse in Form von Web Services implementiert werden. Die anschließende Kombination der Web Services bildet dann den BPEL-Prozess. Da die Implementierung eines Web Services sowie eines BPEL-Prozesses auf XML basiert, sind zur Modellierung der Prozesse Programmierkenntnisse erforderlich.

Bild 4 zeigt die schematische Darstellung eines systemübergreifenden BPEL-Suchprozesses. Nach der Eingabe eines Suchparameters wird im SAP ein Dokument gesucht und für das Auschecken im Agile E6 bereitgestellt. Mit dem hier exemplarisch aufgeführten BPEL-Prozess können Dokumente über das WEB in unterschiedlichen PDM-Umgebungen gefunden und auch zur weiteren Bearbeitung zwischen Unternehmen ausgetauscht werden.

### EON basierter Prototyp

Für die Modellierung von projektspezifischen Prozessen wurde EON-Studio um spezielle Prozessbausteine erweitert, die in einer universellen Bibliothek zur Ver-

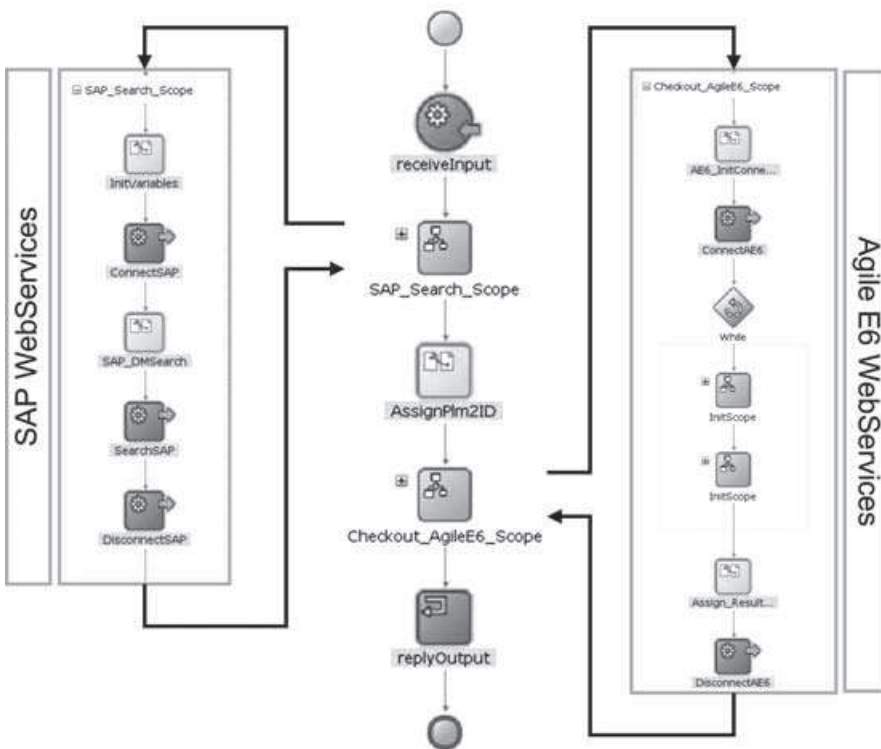


Bild 4. BPEL-Prozess

fügung stehen. Diese können Anwender innerhalb des Software-Werkzeugs zu einem geschlossenen Prozess verknüpfen. Das hat den Vorteil, dass komplexe und wechselwirkende Prozesse schnell und leicht erstellt werden können. Der Anwender benötigt hierfür keinerlei Programmiererfahrung, sondern nur das Wissen über den umzusetzenden Gesamtprozess.

Bild 5 zeigt exemplarisch die Modellierung eines EON-Prozesses, indem Prozessbausteine aus der Bibliothek entnommen und einem Prozess zugeordnet werden. Anschließend werden innerhalb der Routes-Simulation, einer Art virtuellen Arbeitsfläche, die Signalflüsse modelliert. Ein Signalfluss ist eine Verbindung zwischen Objektattributen zweier Prozessbausteine. Es können nur Prozessbausteine verkettet werden, deren Objektattribute den gleichen Datentyp besitzen bzw. zurückgeben

Der EON-Prototyp speichert Prozesse und Prozessbausteine lokal auf den Festplatten der Prozessverantwortlichen. Änderungen und neu erstellte Prozessbausteine müssen folglich zwischen den beteiligten Projektpartnern ausgetauscht werden. Mit steigender Anzahl an Partnern nimmt der Verwaltungsaufwand zu, und es werden geeignete Werkzeuge für die konsistente Verteilung der Daten benötigt.

**Auswertung**

Die beschriebenen Ansätze für die Ausführung systemübergreifender Prozesse wurden in den beiden Prototypen evaluiert. Um eine spätere Anwendung praktikabel und anwenderfreundlich zu gestalten, wurden die in Tabelle 1 aufgeführten Merkmale zur Entscheidungsfindung herangezogen. Die abgegebenen Wertungen beschreiben, inwiefern sich die Ausprägung eines Merkmals auf eine spätere Nutzung positiv (+) oder negativ (-) auswirkt.

Der EON-Prototyp unterstützt die Modellierung systemübergreifender Prozesse sehr gut. Mit Hilfe des grafischen Prozesseditors können Prozesse durch das Verketteten ausgewählter Bausteine aus einer Prozesssteinbibliothek einfach erstellt werden. Die Abbildung erfolgt mittels des Prozesseditors. Die Analyse des Programmcodes vor Anwendungsbeginn unterstützt zudem die Modellierung systemübergreifender Prozesse ohne Programmierkenntnisse. Dadurch wird erreicht, dass der Prozessverantwortliche nur das Wissen über den umzuset-

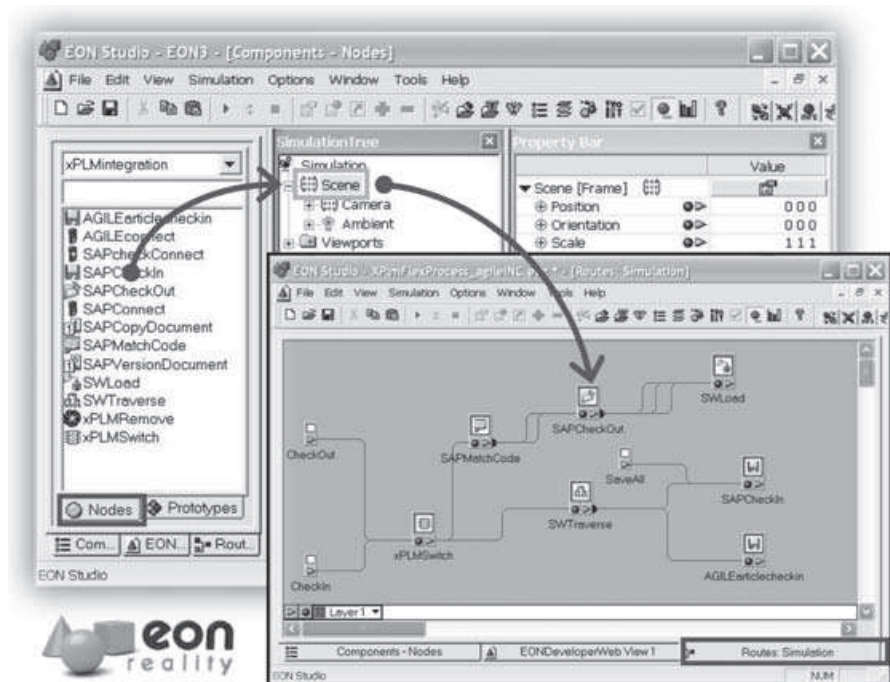


Bild 5. Erstellen eines Prozesses mit dem EON-Prototyp

zenden Gesamtprozess benötigt. Nachteilig wirkt sich die lokale Speicherung der Prozesse und Prozessbausteine bei einer verteilten Nutzung aus.

Der BPEL-Prototyp speichert und verwaltet im Gegensatz zum EON-Prototyp Prozesse und Prozessbausteinbibliothek zentral. Durch diese Zentralisierung können Prozesse serverorientiert gespeichert und abgearbeitet werden. Das ist sowohl für die Installation als auch für die Wartung der Software auf dem zentralen Web Server von Vorteil.

Prozesse und Prozessbausteine können jedoch nur grafisch visualisiert und nicht modelliert werden. Die Modellierung basiert auf der standardisierten Sprache XML und setzt Programmierkenntnisse voraus, was die Anwenderfreundlichkeit reduziert.

Durch eine partielle Fusion werden die Vorteile beider Prototypen bei gleichzeitiger Vermeidung ihrer spezifischen Nachteile vereinigt. Es entsteht ein hybrider Prototyp.

**Strukturkonzept eines hybriden Prototypen**

Die Prozessmodellierung mit dem beschriebenen grafischen EON-Prozesseditor ist flexibel und transparent. Die Wiederverwendung bereits implementierter Software-Bausteine wird hervorragend unterstützt und ermöglicht somit eine schnelle Erarbeitung kundenspezifischer PLM-Lösungen durch das Systemhaus. Gerade im Bereich der unternehmensübergreifenden Kollaboration muss schnell und flexibel auf sich än-

Tabelle 1. Bewertung der Prototypen

Merkmale	BPEL -Prototyp	EON-Prototyp
Analyse des Programmcodes	zur Laufzeit	vor Anwendungsbeginn
universelle Prozesssteinbibliothek	ja (+)	ja (+)
grafischer Prozesseditor	nein (-)	ja (+)
grafische Prozessvisualisierung	ja (+)	ja (+)
Programmiererfahrung erforderlich	ja (-)	nein (+)
Verwendung der Prozessbausteine	zentral (+)	lokal (-)
anwenderfreundliche Modellierung	eher nein (-)	Ja (+)

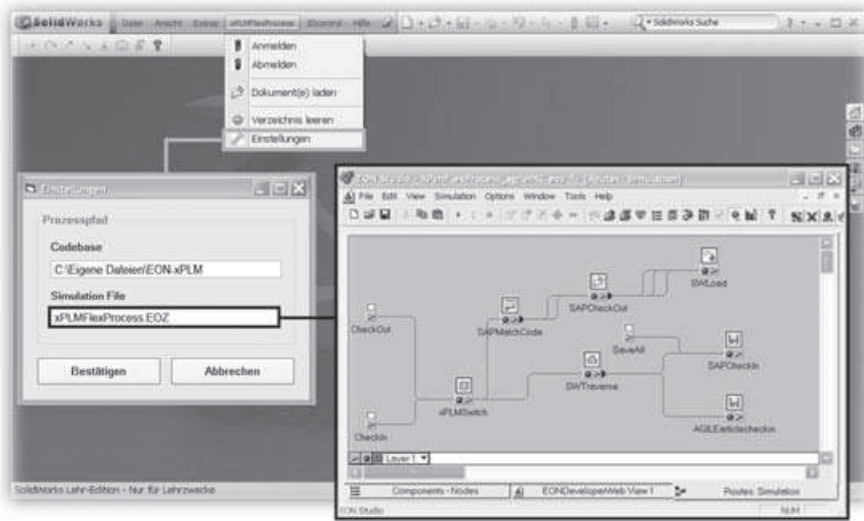


Bild 6. Einbinden einer EON-Prozessdefinition

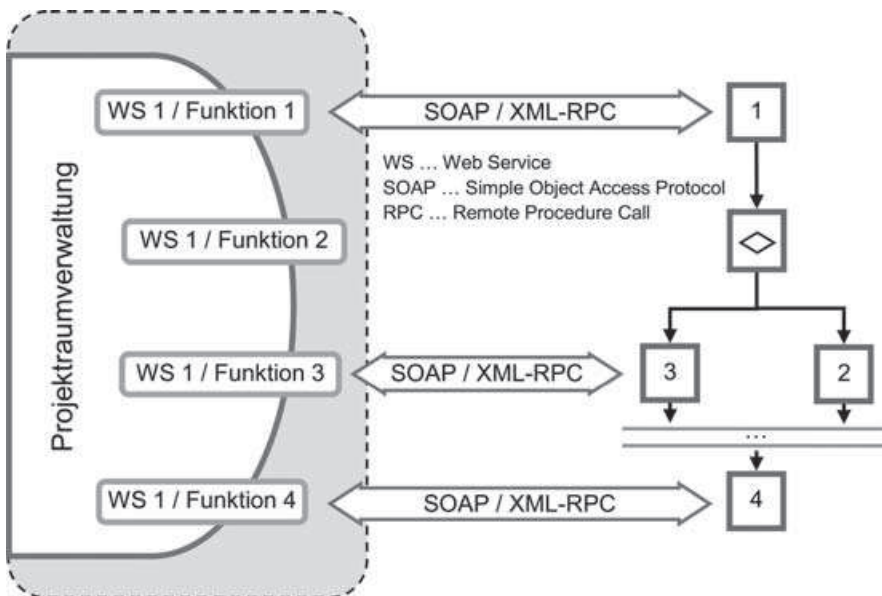


Bild 7. Beziehungen zwischen Web Services und der EON-Prozessbeschreibung

dernde Anforderungen, beispielsweise die Struktur eines Konsortiums, reagiert werden. Darüber hinaus können Prozessverantwortliche innerhalb der Unternehmen auch ohne weitreichende Programmierkenntnisse definierte Datenflüsse innerhalb des Prozesses erkennen und gegebenenfalls anpassen.

Die systemunabhängige, zentrale Bereitstellung von Funktionen mittels Web Services bietet immenses Potenzial für eine flexible Kommunikation mit und innerhalb des Projekttraums. Hier kann zum Beispiel die konsistente Datenverarbeitung mit der Nutzung einheitlicher Funktionen über die Systemgrenzen hin-

weg realisiert werden, was in einem heterogenen Projekt- und Systemumfeld von essenzieller Bedeutung ist.

Die Erstellung sowie die Anpassung der Prozessdefinition erfolgen, wie beschrieben, innerhalb des grafischen Prozesseditors. Im Ergebnis liegt eine Prozessbeschreibung vor, welche alle Prozessschritte enthält und zentral im Projektraum abgelegt wird.

Bild 6 zeigt, wie eine Prozessdefinition vor der Nutzung durch den jeweiligen Client geladen und in der Laufzeitumgebung ausgeführt wird. Die Laufzeitumgebung kann dabei als eigenständige Software-Komponente oder aber wie in Bild 6

als Plug-n des CAD-Systems betrieben werden. Muss beim Durchlaufen des Prozesses beispielsweise ein Zugriff auf das Projektdaten-Repository erfolgen, kommuniziert der Prozessbaustein mit dem Web Service des Projekttraums (Bild 7).

Über den Web Service wird die jeweils notwendige Funktion (im Bsp. 1...4) eines Prozessschritts auf dem Projektdaten-Repository ausgeführt. Die Kommunikation mit den Web Services ist durch etablierte Protokolle, wie z. B. SOAP oder XML-RPC, realisiert. Die Daten werden an den Web Server über das https-Protokoll übertragen, um die Datenflüsse vor unbefugtem Zugriff zu sichern. Die Authentifizierung des Web Services selbst erfolgt mittels Zertifikat.

Mit der Ergänzung des Funktionsumfangs werden die bisherigen Nutzungsmöglichkeiten erhalten und eine größere Flexibilität des EON-Prototypen erreicht.

### Zusammenfassung und Ausblick

Mit dem vorgestellten hybriden Prototyp ist die Kopplung unterschiedlicher Entwicklungswerkzeuge zur Modellierung systemübergreifender Prozesse leicht möglich. Mit Hilfe des grafischen Editors sind Prozesse sowohl vollständig neu modellierbar als auch dynamisch und flexibel an neue Strukturen adaptierbar. Dafür ist lediglich die Kenntnis über den umzusetzenden Prozess von Bedeutung. Durch die Zentralisierung der Prozesse und Prozessbausteine ist eine serverorientierte Speicherung und Abarbeitung der Programme möglich.

Möglichkeiten zur Weiterentwicklung des hybriden Ansatzes liegen in der Schaffung eines speziellen Prozesseditors, mit dem Prozesse anwenderfreundlich definierbar sind, und der gleichzeitig die Implementierung neuer Prozessbausteine ermöglicht. So kann EON-Studio als Demonstrator entfallen.

Die zusätzliche Integration einer Modellierungshilfe, welche nur richtige Verbindungen von Prozessbausteinen zulässt, verhindert inkompatible Prozesse und erhöht somit die Prozesssicherheit.

### Literatur

- 1 Stelzer, R.; Petermann, D.; Saske, B.; Steger, W.: Kollaborationsumgebung in einer heterogenen PDM-CAD-VR Systemlandschaft. In: Gausemeier, J.; Grafe, M. (Hrsg.): Tagungsband zum 8. Paderborner Workshop Augmented & Virtual Reality in der