

DER PROZESSNAVIGATOR – FLEXIBLE PROZESSUNTERSTÜTZUNG FÜR PRODUKTENTWICKLUNGSPROJEKTE

Matthias Faerber

Zusammenfassung

In diesem Beitrag wird gezeigt, wie Produktentwicklungsprozesse durch eine prozessbasierte Ausführungsumgebung unterstützt werden können. Dazu werden zuerst die Anforderungen, die an ein solches System gestellt werden, vorgestellt. Diese Anforderungen werden anschließend über ein Prozessmodell in eine Ausführungsumgebung abgebildet. In diesem Zusammenhang wird speziell die Verknüpfung von Prozessmodell und Ausführungsumgebung betrachtet. Es wird gezeigt, wie Konzepte existierender Workflow-Management-Systeme adaptiert werden können, um die in Produktentwicklungsprozessen geforderte Flexibilität zu unterstützen.

1 Einleitung

Die schnelle Entwicklung fehlerfreier und technologisch anspruchsvoller Produkte ist ein wesentlicher Faktor der maßgeblich zum wirtschaftlichen Erfolg von Firmen beiträgt. Sie ist ein wichtiger Wettbewerbsvorteil im Vergleich mit Wettbewerbern. Nur wer in der Lage ist, Produkte frühzeitig und mit hoher Qualität auf den Markt zu bringen, wird im internationalen Wettbewerb bestehen und langfristig erfolgreich arbeiten können. Gerade moderne Produkte stellen durch ihre Komplexität jedoch große Herausforderungen an Produktentwickler [8]. Die Arbeitsschritte, die zur Entwicklung dieser Produkte nötig sind, werden immer umfangreicher und sind durch die Nutzung einer Vielzahl an Werkzeugen und unterstützenden Systemen charakterisiert. Ein weiterer Faktor, der zur Komplexität von Produktentwicklungsprojekten beiträgt, ist die Arbeit in über verschiedene Standorte verteilten Entwicklungsteams. Die Arbeit im Projekt muss zwischen diesen Teams koordiniert und die Ergebnisse müssen synchronisiert werden.

Zur Unterstützung der Mitarbeiter in Unternehmen wurden in den letzten Jahren eine Vielzahl an Möglichkeiten untersucht und entwickelt. So hat sich gezeigt, dass Workflow-Management-Systeme (WfMS) bei gut strukturierbaren und sich häufig wiederholenden Tätigkeiten (z.B. die Bearbeitung einer Reisekostenabrechnung in der Administration eines Unternehmens) einen wichtigen Beitrag zu Unterstützung der Mitarbeiter leisten können [5]. Dort erhalten Mitarbeiter in sog. Worklists (Aufgabenliste) einen Überblick über die aktuell zu bearbeitenden Vorgänge. In dieser Liste wird für jeden Vorgang ein Eintrag erstellt und Mitarbeiter können diese aufrufen. Nach dem Aufruf der Aufgabe aus der Worklist werden die zugehörigen Dokumente automatisch vom WfMS geladen und dem Nutzer zur Verfügung gestellt. Nach Beendigung einer Aufgabe wird der Vorgang direkt an den nächsten Mitarbeiter weitergereicht und in die dort vorhandene Worklist eingetragen. Dies wird solange wiederholt bis der komplette Vorgang abgeschlossen ist. Auf diese Weise wird vom WfMS sicher gestellt, dass bei der Bearbeitung eines Vorgangs keine Aufgaben vergessen werden. Grundlage für die Unterstützung von Geschäftsvorgängen mit WfMS ist die Definition eines Geschäftsprozesses. Nur die dort definierten Aufgaben können später auch durch ein WfMS unterstützt werden.

Obwohl auch für die Produktentwicklung Prozessdefinitionen in Form von Vorgehensmodellen (z.B. VDI 2221 [1]) vorhanden sind, die die zur Entwicklung eines Produkts nötigen Arbeitsschritte beschreiben, existiert noch keine systematische Unterstützung durch WfMS oder Prozessmanagementsysteme. Die Charakteristik von Produktentwicklungsprozessen kann nicht mit der von administrativen Prozessen, wie etwa einer Reisekostenabrechnung, verglichen werden. Bei administrativen Prozessen ist von Beginn an klar geregelt, welche Arbeitsschritt im Rahmen eines Vorgangs zu bearbeiten sind und welche Dokumente jeweils erstellt werden müssen. Bei Produktentwicklungsprozessen hingegen wird dies häufig erst im Prozessverlauf klar. Aufgaben in der Produktentwicklung sind in aller Regel auch deutlich umfangreicher als Aufgaben in administrativen Prozessen. Ihre Bearbeitung verlangt von den Ingenieuren deutlich mehr Wissen und Kreativität. Dies drückt sich auch in einer höheren Flexibilität der Prozesse aus. Somit muss ein System, das Mitarbeiter bei der Produktentwicklung unterstützen soll, auch diese Flexibilität direkt umsetzen können. Konventionelle WfMS sind hierzu jedoch insbesondere durch die feste Vorgabe der zu bearbeitenden Schritte nicht in der Lage.

Um einen Produktentwickler bei der Entwicklung von Produkten zu unterstützen und ihn immer mit den richtigen Informationen zu versorgen, wird im Rahmen des Forschungsverbundes FORFLOW ein Prozessmanagementsystem (Prozessnavigator) konzipiert. Dieser dient als zentrales Informationssystem zur Koordination von Abläufen in der Produktentwicklung und ist zur Steuerung der Kooperation zwischen den Beteiligten vorgesehen.

In diesem Beitrag soll der grundsätzliche Lösungsansatz zur Erstellung einer prozessbasierten Ausführungsumgebung für Produktentwicklungsprozesse vorgestellt werden. Dazu werden in Abschnitt 2 die an den Prozessnavigator gestellten Anforderungen vorgestellt und vor allem auf die Unterschiede zu existierenden WfMS hingewiesen. Abschnitt 3 skizziert kurz den gewählten Ansatz und zeigt wie eine basierend auf einem Prozessmodell eine flexible Ausführungsumgebung für Produktentwicklungsprozesse erstellt werden kann. In den beiden Abschnitten 4 und 5 wird detailliert sowohl auf das genutzte Prozessmodell und die Ausführungsumgebung selbst eingegangen.

2 Anforderungen

Die Anforderungen, die an den Prozessnavigator gestellt werden, unterscheiden sich teilweise deutlich von denen eines konventionellen WfMS. In diesem Abschnitt soll kurz ein Überblick über die wichtigsten Anforderungen gegeben werden und vor allem auf Unterschiede sowie Gemeinsamkeiten hingewiesen werden.

Ziel des Prozessnavigators ist es Mitarbeitern in Entwicklungsprojekten eine systematische Unterstützung bei der Bearbeitung ihrer Aufgaben zukommen zu lassen. Im Gegensatz zu konventionellen WfMS steht jedoch nicht die direkte Integration der Anwendungen im Mittelpunkt. Vielmehr ist eine umfassende Information der Mitarbeiter, wie ein bestimmter Arbeitsschritt umgesetzt werden kann, gefragt. Dazu muss der Prozessnavigator vor allem die folgenden Informationen bereitstellen:

1. *Informationen über den Arbeitsschritt:* Der Prozessnavigator muss den Mitarbeitern Informationen über den aktuellen Arbeitsschritt bereitstellen. Diese Informationen beinhalten insbesondere eine ausführliche Beschreibung der umzusetzenden Arbeitsschritte. Neben der eigentlichen Aufgabenbeschreibung müssen zusätzlich noch weitere Informationen bereitstellen gestellt werden, wie z.B. zu berücksichtigende DfX-Aspekte.
2. *Bereitstellen von Dokumente und Produktmodellen:* Der Prozessnavigator muss die in einem Prozessschritt zu bearbeitenden Eingangs- und Ausgangsdokumente be-

reitstellen. Mitarbeiter sollen direkt im Prozessnavigator Zugriff auf die Eingangsdokumente erhalten. Dies erspart den Mitarbeitern einerseits Dokumente erst anfordern zu müssen bevor sie bearbeitet werden können; andererseits wird so gewährleistet, dass immer mit den aktuellen Dokumenten gearbeitet wird und diese in einem System archiviert sind. Für die zu erstellenden Ausgangsdokumente muss der Prozessnavigator Dokument-Templates bereitstellen, die die Mitarbeiter nutzen können. Außerdem können Beispieldokumente hinterlegt sein, die den Mitarbeitern aufzeigen, was als Ergebnis eines Arbeitsschritts erwartet wird.

3. *Werkzeuge und Systeme*: Der Prozessnavigator soll dem Mitarbeiter Informationen über die im Prozessschritt verwendeten Werkzeuge und Systeme anbieten. Dies umfasst insbesondere erprobte Verfahren, wie die Werkzeuge zu benutzen sind und wie damit besonders gute Ergebnisse erzielt werden können.
4. *Unterstützung von Entscheidungen*: Sind im Prozessmodell Alternativen vorgesehen wie die Entwicklung fortgesetzt werden kann, so sollen dem Entwickler Hinweise gegeben werden, die ihn bei der Entscheidungsfindung unterstützen. Nachdem der Mitarbeiter eine Entscheidung getroffen hat, wie der Prozess fortgesetzt werden soll, muss diese im Prozessnavigator dokumentiert werden. Somit kann später nachvollzogen werden, warum eine bestimmte Entscheidung getroffen wurde und der Entwicklungsprozess wird insgesamt transparenter.
5. *Unterstützung von Meilensteinen*: Um den Entwicklungsprozess in verschiedenen Phasen unterteilen zu können muss der Prozessnavigator Meilensteine unterstützen. Meilensteine markieren dabei das Ende einer Phase und damit einen Punkt im Entwicklungsprozess an dem die Dokumente einen bestimmten Reifegrad erreicht haben. Nach Abschluss einer Phase sind sie gegenüber Änderungen genügend stabil um in nachfolgenden Phasen auf den dort beschriebenen Ergebnissen aufbauen zu können. Nachdem ein Meilenstein im Prozess abgeschlossen wurde, dürfen die Dokumente der abgeschlossenen Phase nicht mehr verändert werden.

Ein besonders wichtiger Aspekt bei der Unterstützung von Entwicklungsprojekten mit Prozessmanagementsystemen ist die Ausführungssemantik des Systems. Diese legt fest in welcher Reihenfolge Mitarbeitern Arbeitsschritte zur Ausführung angeboten werden und wie weit von ihrer im Prozessmodell definierten Reihenfolge abgewichen werden darf. Die in Produktentwicklungsprozessen benötigte Ausführungssemantik unterscheidet sich dabei grundlegend von der in konventionellen Workflowmanagement-Systemen [2, 3, 9]. Die Ausführungssemantik dieser Systeme ist klar durch die Reihenfolge der Prozessschritte vorgegeben. Ein Beispiel soll dies verdeutlichen:

Gegeben ist folgender Prozess: $A \rightarrow B \rightarrow C$. In konventionellen WfMS darf in diesem Fall der Schritt B erst ausgeführt werden, nachdem Schritt A beendet wurde; analog muss B abgeschlossen sein, bevor C gestartet wird. Ein mehrmaliges Ausführen eines bereits abgeschlossenen Arbeitsschrittes ist ebenfalls nicht vorgesehen. □

Diese Ausführungssemantik kann jedoch nicht auf den Bereich der Produktentwicklung übertragen werden. Sie ist viel zu strikt und würde die Mitarbeiter und Ingenieure zu sehr bei der Bearbeitung der Arbeitsschritte einschränken. Aus diesem Grund müssen die oben genannten Punkte um die folgenden Anforderungen im Bezug auf die Ausführungssemantik ergänzt werden:

6. *Vorschlag der nächsten Arbeitsschritte*: Der Prozessnavigator muss Mitarbeitern die jeweils nächsten Prozessschritte vorschlagen. Diese Anforderung deckt sich mit der aus konventionellen WfMS.

7. *Wiederholen von Arbeitsschritten*: Die Entwicklung eines Produktes ist kein linearer Prozess. Vielmehr ist er gekennzeichnet durch Iterationen und Schleifen. Da diese kurzen Iterationen ad hoc auftreten und nicht von Anfang an im Prozessmodell eingeplant werden können, muss der Prozessnavigator die erneute Ausführung auch bereits abgeschlossener Arbeitsschritte zulassen.
8. *Überspringen von Arbeitsschritten*: Die Kreativität der Ingenieure ist ein entscheidender Erfolgsfaktor bei der Entwicklung von Produkten und darf nicht durch ein Prozessmanagementsystem eingeschränkt werden. Aus diesem Grund sollen Entwickler die Möglichkeit haben Arbeitsschritte zu überspringen und mit einem Schritt, der erst später im Entwicklungsprozess vorgesehen ist fortzufahren. Übersprungene Schritte dürfen jedoch nicht als abgeschlossen markiert werden und müssen den Entwicklern erneut zur Bearbeitung angeboten werden.
9. *Auslassen von Arbeitsschritten*: In Ausnahmefällen kann es vorkommen, dass Schritte, obwohl sie im Prozessmodell eingeplant sind, im Verlauf des Prozesses nicht mehr bearbeitet werden müssen. Die Entscheidung, ob die Ergebnisse eines Prozessschritts nicht mehr benötigt werden, ist komplex und kann nur von Mitarbeitern getroffen werden. Der Prozessnavigator muss diesen Fall jedoch zulassen, allerdings müssen Mitarbeiter in diesen Fällen eine Begründung im System angeben. Dies ist vor allem im Hinblick auf eine spätere Nachvollziehbarkeit der im Projekt getroffenen Entscheidungen wichtig.

Insgesamt ist mit dem Prozessnavigator ein System gefragt, das den Entwickler bei der Bearbeitung der Schritte unterstützt und ihm Hinweise gibt, welche Schritte als nächstes bearbeitet werden müssen. Zu diesen Schritten sollen Informationen gegeben werden, wie diese umzusetzen sind und was besonders zu beachten ist. Allerdings darf das System die Mitarbeiter durch zu strikte Vorgaben nicht in ihrer Kreativität einschränken und muss ihnen genügend Freiraum einräumen, um das Projekt erfolgreich zu Ende zu führen.

3 Lösungsansatz

Um die oben genannten Anforderungen zu erfüllen, soll ein prozessbasiertes System entwickelt werden, das eine durchgängige Unterstützung der Nutzer gewährleistet. Durchgängig bedeutet in diesem Zusammenhang, dass die im Unternehmen vorhandenen und bei der Ausführung zu berücksichtigenden Aspekte in einem Modell abgebildet werden, welches die Grundlage für den Prozessnavigator bildet. Aspekte des Produktentwicklungsprozesses können somit über das Prozessmodell bis in den Prozessnavigator abgebildet werden.

In Bild 1 ist dieser Ansatz dargestellt. Auf der linken Seite werden die verschiedenen Aspekte, die in der Produktentwicklung vorkommen, abgebildet. Im Rahmen des Forschungsverbundes FORFLOW wurden dabei insbesondere der Prozess mit seinen *Arbeitsschritten*, *Daten und Produktmodelle*, *Werkzeuge und Systeme* sowie die am Prozess beteiligten *Mitarbeiter (Organisation)* betrachtet. Diese müssen in ein Modell (Prozessmodell) abgebildet werden, welches die Grundlage für die spätere Ausführung im Prozessnavigator bildet. Die Begrenzung auf die genannten Aspekte stellt jedoch keine Einschränkung des Ansatzes an sich dar. Vielmehr kann dieser durch eine Erweiterung der im Modell enthaltenen Aspekte einfach an die Erfordernisse des jeweiligen Unternehmens angepasst werden.

Das Prozessmodell gibt dabei den Rahmen der Prozessentwicklung vor. Alle dort enthaltenen Elemente können dem Entwickler im Prozessnavigator angezeigt werden. Dieser interpretiert die Inhalte und stellt sie verständlich für den Anwender dar. Entsprechend den in Abschnitt 2 geforderten flexiblen Ausführungssemantik müssen die Mitarbeiter über die nächsten im Prozess vorgesehenen Arbeitsschritte informiert werden. Außerdem sollen Ein-

gangs- und Ausgangsdokumente bereitgestellt werden und Informationen zu den Methoden (z.B. DfX-Methoden) gegeben werden, die bei der Bearbeitung eines Arbeitsschrittes zu beachten sind.

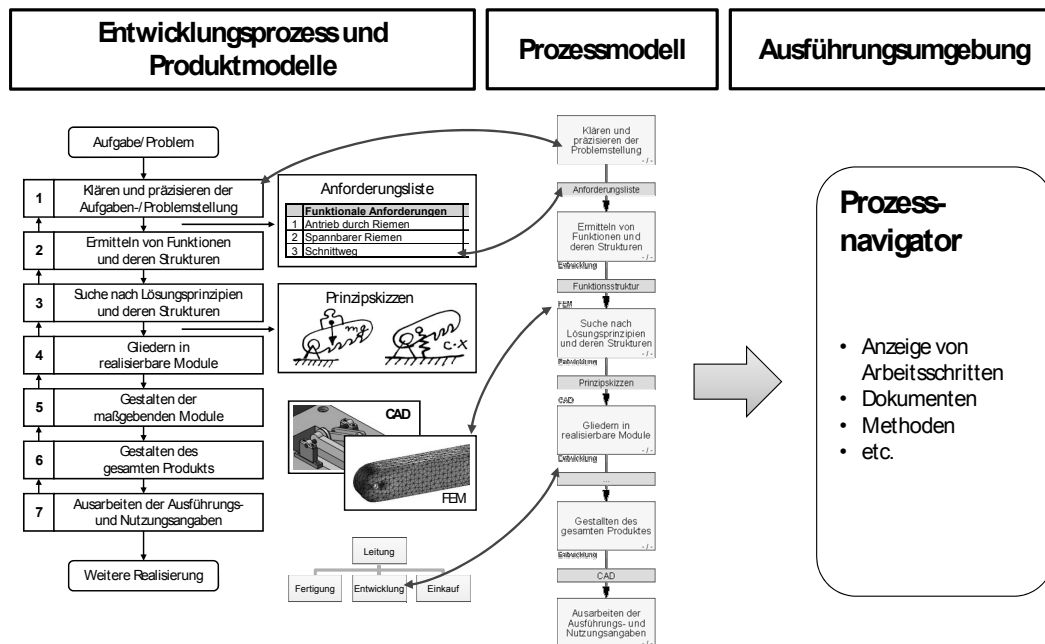


Bild 1: Lösungsansatz

Diese modellbasierte Ausführungsumgebung bringt einen weiteren Vorteil mit sich. Da der Prozessnavigator alle angezeigten Informationen aus dem Prozessmodell bezieht, kann dieser einfach an neue Erfordernisse angepasst werden, indem die im Prozessmodell hinterlegten Inhalte angepasst werden. Somit können Änderungen im Unternehmen (z.B. ein neues Entwicklungsverfahren oder geänderte Werkzeuge) über eine Anpassung des Prozessmodells in den Prozessnavigator übertragen werden.

4 Prozessmodell

Das Prozessmodell bildet die Grundlage für die Ausführung des Prozesses im Prozessnavigator. Zur Modellierung der Produktentwicklungsprozess wird die Aspektorientierte Modellierungsansatz genutzt [6, 7]. Dieser zeichnet sich vor allem dadurch aus, dass in ihm alle Anforderungen an einen Unternehmensprozess in einem integrierten Modell abgebildet werden können und dieser auch durch seine Strukturierung einfach erweitert werden kann. In dem Modellierungsansatz werden die verschiedenen Sichten, die sich auf einen Prozess ergeben, jeweils als eine eigenständiger Aspekt oder Perspektive dargestellt. Diese können so miteinander kombiniert werden, dass sich ein einheitliches Gesamtbild ergibt. Die Basis des Modells bilden die folgenden fünf Aspekte:

- **Funktionaler Aspekt:** Im Funktionalen Aspekt werden die im Prozess zu bearbeitenden Arbeitsschritte definiert. Jeder Arbeitsschritt beschreibt dabei eine genaue umrissene Aufgabe, die zur Erstellung des gesamten Produktes einen Beitrag leistet (Anforderungen 1 und 4).
- **Verhaltensorientierter Aspekt:** Dieser definiert die kausale Abfolge der Arbeitsschritte. Im Wesentlichen werden also die Vorgänger und Nachfolgebeziehungen zwischen den Arbeitsschritten definiert (Anforderungen 4 und 6).

- **Datenorientierter Aspekt:** Dieser Aspekt definiert die Daten oder Produktmodelle, die im Produktentwicklungsprozess genutzt werden. Hierunter werden alle Daten oder Informationen (z.B. Angebotsdokumente, Skizzen, CAD-Dokument etc.) verstanden, die im Entwicklungsprozess genutzt werden. Außerdem wird beschrieben in welchen Arbeitsschritten ein Dokument erstellt wird und wo es als Eingangsdokument benötigt wird. Somit definiert der Datenorientierte Aspekt auch den Datenfluss im Prozess (Anforderung 2).
- **Operationaler Aspekt:** Dieser Aspekt beschreibt die im Prozess zum Einsatz kommenden Werkzeuge und System. In der Produktentwicklung sind dies beispielsweise CAD-Programme, FEM-Programme etc. (Anforderung 3)
- **Organisatorischer Aspekt:** Er beschreibt, wer in einem Prozessschritt für die Ausführung der darin beschriebenen Arbeiten verantwortlich ist.

Obwohl die einzelnen Aspekte grundsätzlich isoliert betrachtet werden können, ergibt sich erst durch ihre Kombination ein vollständiges Bild des Gesamtprozesses. In Bild 2 ist ein Ausschnitt eines Produktentwicklungsprozesses abgebildet. Dieser beschreibt die Ausarbeitung eines Bauteils beginnend mit der *Modellierung des Berechnungssystems* über eine *Optimierung der Topologie* bis hin zur *Absicherung der Kosten* im letzten Schritt:

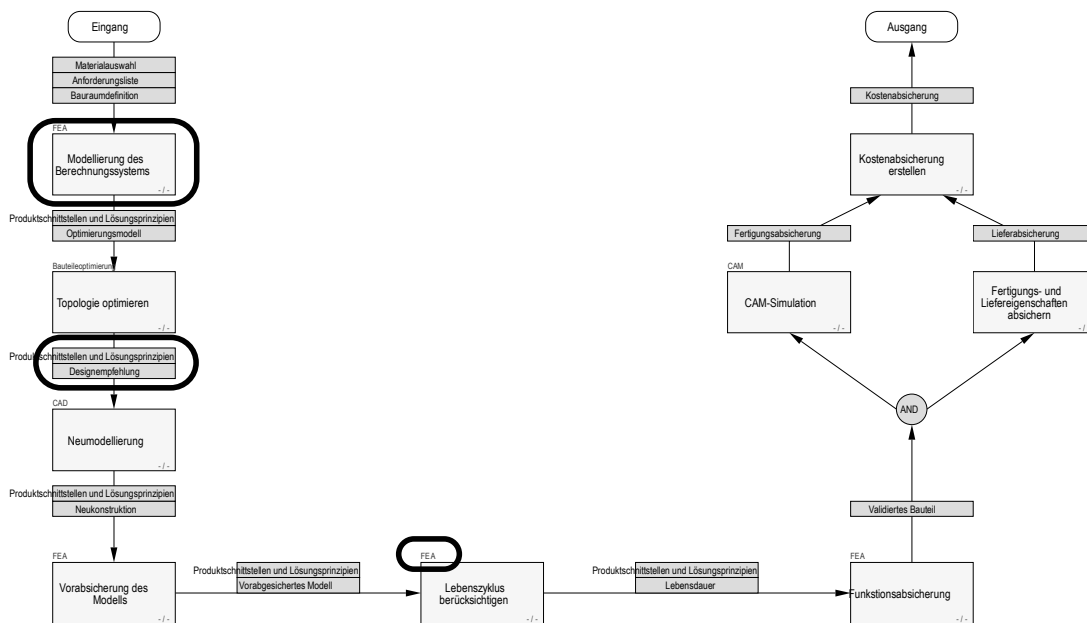


Bild 2: Ausschnitt eines Prozessmodells

Die Arbeitsschritte (Funktionaler Aspekt) werden im Prozessmodell durch Kontrollflusskanten verbunden, die die Reihenfolge der Schritte vorgeben und entsprechend im Prozessnavigator ausgewertet werden. Für jeden Prozessschritt sind die Ein- und Ausgabedaten angegeben. Auf Basis der in den Eingangsdaten vorhandenen Informationen werden mit Hilfe der Werkzeuge die Ausgangsdaten erzeugt.

Zusätzlich zu den abgebildeten Elementen können im Entwicklungsprozess *Meilensteine* und *Entscheider* vorhanden sein. Typische Meilensteine der Produktentwicklung sind: *Klärung der Aufgabenstellung abgeschlossen*, *Konzept abgeschlossen* etc. Sie markieren Stellen im Prozess, an denen die erstellten Dokumente einen gewissen Reifegrad vorweisen und im Entwicklungsprozess mit der Bearbeitung einer neuen Phase begonnen werden kann.

Entscheider hingegen kennzeichnen Verzweigungen im Entwicklungsprozess. Sie sind immer dann erforderlich, wenn ein Prozessablauf in mehrere Äste verzweigt und der Entwickler entscheiden muss, wie im Prozess fortgefahren werden soll. So ist beispielsweise im Anschluss an eine Review die Entscheidung nötig, ob die Ergebnisse der Dokumente mit den Vorgaben in den Anforderungen übereinstimmen und ob im Prozess fortgefahren werden kann, oder ob die erzielten Ergebnisse nochmals überarbeitet werden müssen.

Obwohl das Prozessmodell von jedem Unternehmen auf die dort vorhandenen Gegebenheiten angepasst werden muss, definiert es doch nur ein idealisiertes Abbild der Produktentwicklung. Insbesondere die geforderten Flexibilität (vgl. Abschnitt 2; Anforderungen 7 und 8) sind noch nicht im Modell enthalten und müssen erst durch den Prozessnavigator bei der Ausführung des Prozesses implementiert werden. Hingegen sind die benötigten Informationen über die Arbeitsschritte, Dokumente und eingesetzte Werkzeuge schon komplett vorhanden und brauchen vom Prozessnavigator nur noch ausgelesen werden.

5 Ausführungsumgebung

Analog zu klassischen Workflow-Management-Systemen ist der Prozessnavigator unterteilt in eine Worklist-Komponente, in der die nächsten Arbeitsschritte gezeigt werden, und in eine Anwendungskomponente (genannt FORFLOW-Desktop) mit Informationen zu Ausführung des Arbeitsschrittes. Die einzelnen Komponenten (Worklist und Desktop) unterscheiden sich auf Grund der besonderen Anforderungen (vgl. Abschnitt 2) von konventionellen WfMS. Bild 3 zeigt eine Abbildung des Prozess navigators.

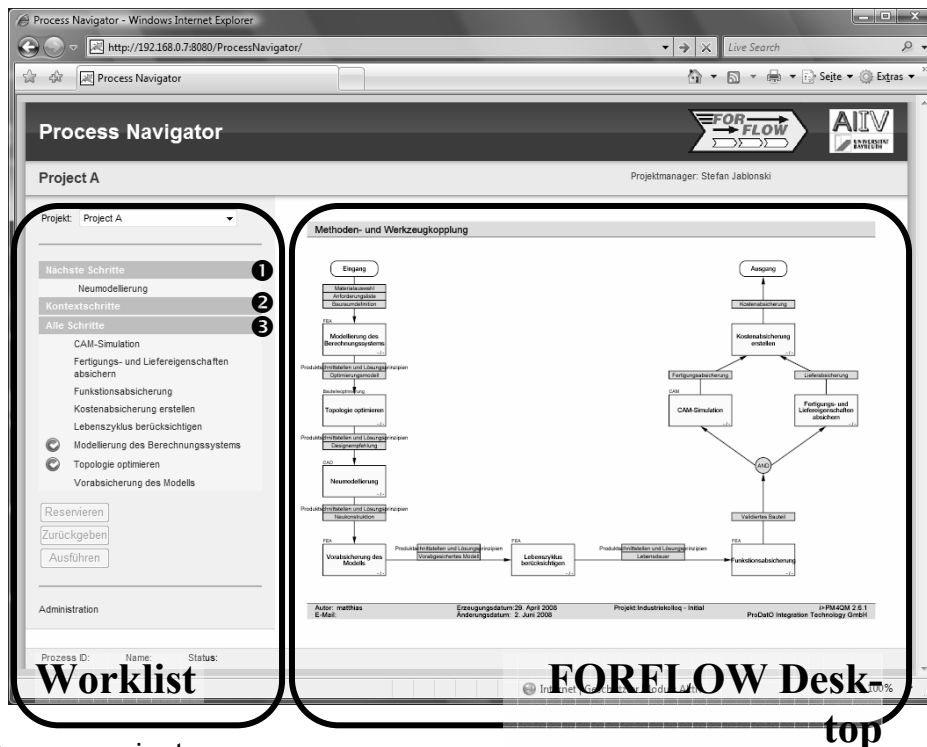


Bild 3: Prozessnavigator

Die Arbeitsliste stellt die für den Anwender ausführbaren Arbeitsschritte dar. Diese können hier ausgewählt und gestartet werden. Anders als in klassischen WfMS wird hier jedoch nicht nur der jeweils nächste Arbeitsschritt angezeigt, sondern alle vorhandenen Arbeitsschritte sind grundsätzlich zur Ausführung bereit, um die erforderliche Flexibilität bei der Prozessausführung zu gewährleisten. Um dennoch dem Nutzer eine Überblick zu geben, welche Arbeitsschritte im Prozessmodell als nächstes eingeplant wurden, wurde die Worklist in drei Abschnitte unterteilt. Jeder dieser Abschnitte erfüllt eine unterschiedliche Funktion und zeigt

eine jeweils andere Menge an Arbeitsschritten mit verschiedenen Prioritäten an. Die Inhalte der einzelnen Abschnitte können für jedes Unternehmen und für jeden Anwendungsfall individuell konfiguriert werden; aktuell sind die Inhalte der Abschnitte wie folgt festgelegt:

- *Nächste Schritte* (❶): In diesem Abschnitt werden im Bezug auf einen gerade bearbeiteten Prozessschritt die jeweils nächsten Prozessschritte angezeigt. Im Fall des oben abgebildeten Beispiels wurden die beiden Schritte *Modellierung des Berechnungssystems* und *Topologie optimieren* abgeschlossen (durch einen Haken markiert). Entsprechend wird in diesem Abschnitt die *Neumodellierung* als nächster auszuführender Schritt angezeigt. Dieser Abschnitt der Worklist ist somit dafür verantwortlich dem Nutzer die jeweils nächsten Arbeitsschritte vorzuschlagen (Anforderung 6).
- *Kontext Schritte* (❷): In diesem Abschnitt werden alle Prozessschritte angezeigt, die sich im Kontext (also Umfeld) des aktuellen Arbeitsschrittes befinden. Hier werden neben den noch offenen zukünftigen Arbeitsschritten auch schon bearbeitete Schritte angeboten, um dem Entwickler die Möglichkeit zu geben bisherige Ergebnisse erneut einzusehen und gegebenenfalls zu überarbeiten (Anforderungen 7 und 8).
- *Gesamter Prozess* (❸): In diesem Abschnitt sind alle Arbeitsschritte des gesamten Prozesses enthalten. Die hier enthaltenen Arbeitsschritte werden dann von einem Entwickler genutzt, wenn der gewünschte Arbeitsschritt nicht in einer der beiden anderen Abschnitte enthalten ist (Anforderungen 7 und 8). Im Regelfall wird ein Nutzer aber auf einen Arbeitsschritt aus der Liste der *Nächsten* oder *Kontext Schritte* und nur im Ausnahmefall auf den *Gesamten Prozess* zugreifen.

Entwickler können einen beliebigen Arbeitsschritt aus einer der drei Abschnitte der Worklist auswählen. Die Entscheidung, wie der Prozess fortgesetzt werden soll ist damit auf dem Mitarbeiter übertragen und somit ist auch die wichtigste Forderung an den Prozessnavigator – die flexible Ausführungssemantik – umgesetzt. Durch den Abschnitt *Nächste Schritte* ist jedoch auch sichergestellt, dass Mitarbeiter immer darüber informiert sind, welche Schritte als nächstes zur Ausführung vorgesehen sind. Durch die Aufteilung der Worklist in verschiedene Abschnitte kombiniert der Prozessnavigator eine systematische Navigation im Prozess mit flexibler Prozessausführung.

Nach der Auswahl eines Arbeitsschrittes aus der Worklist schaltet der Prozessnavigator auf die Anzeige des FORFLOW-Desktops um und zeigt dort Informationen zum aktuellen Arbeitsschritt an. Hier findet der Entwickler alle Informationen, die er zur Bearbeitung des ausgewählten Prozessschrittes benötigt (Anforderungen 1 bis 3).

In Bild 4 ist der Prozessnavigator nach der Auswahl des Arbeitsschrittes *Neumodellierung* dargestellt. Der Desktop-Bereich selbst ist in mehrere Unterabschnitte unterteilt. Diese können durch die Auswahl der Reiter im oberen Bereich (❹) aktiviert werden und die dort hinterlegten Inhalte werden dargestellt. In der aktuellen Version sind im Prozessnavigator Abschnitte zu den Themen Produktmodelle (Dokumente), Werkzeuge, Methoden, CAX-Datenformate und zwei Abschnitte zur Suche in Wissensmanagement Systemen vorhanden. Damit erhält der Mitarbeiter einen umfassenden Überblick über den aktuelle Arbeitsschritt und Informationen, die ihn bei der Bearbeitung unterstützen.

Im Methoden-Abschnitt des FORFLOW-Desktops (Bild 4) findet der Mitarbeiter einen Überblick über die DfX-Aspekte, die er bei der Bearbeitung des Prozessschrittes zu beachten hat. Mit dem *Design for X (DfX)* steht ein umfangreiches Wissenssystem zur Verfügung, in dem Erkenntnisse, wie einzelne Eigenschaften technischer Systeme beim Konstruieren zu erreichen sind, gesammelt und geordnet sind. Diese Informationen liefern Ingenieuren einen wichtigen Beitrag bei der Bearbeitung der Erstellung der Ergebnisse eines Arbeitsschrittes.

Eine ausführliche Beschreibung dieses Abschnitts und seiner Integration in den Prozessnavigator ist in [4] zu finden.

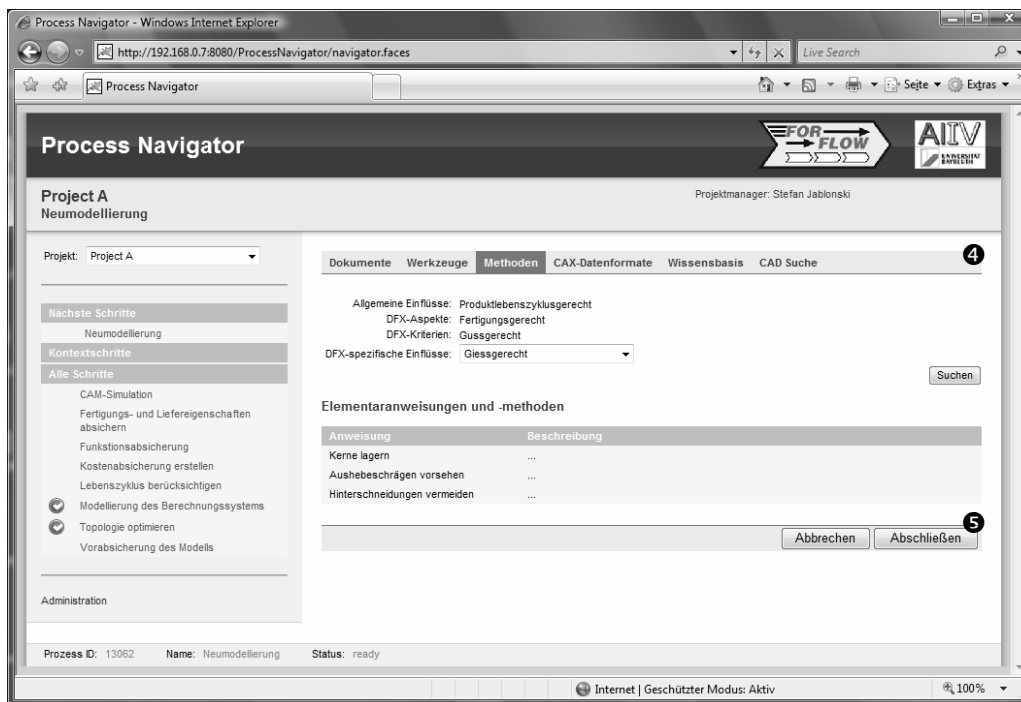


Bild 4: Prozessnavigator mit DfX Hinweisen

Nachdem ein Mitarbeiter die im Arbeitsschritt geforderten Dokumente erstellt hat und diese im System gespeichert hat kann er den Arbeitsschritt abschließen (5). Im Anschluss werden die in den Abschnitten der Worklist angezeigten Arbeitsschritte aktualisiert. Insbesondere werden im Abschnitt *Nächste Schritte* jeweils die nächsten im Prozessmodell eingetragenen Schritte angezeigt. Schon abgeschlossene Schritte werden in der Worklist entsprechend markiert, können aber durch den Benutzer jederzeit erneut aufgerufen werden um die Ergebnisse zu überarbeiten.

Entscheider und Meilensteine (Anforderung 4 und 5) werden in der Arbeitsliste ähnlich den Prozessschritten angezeigt. Um sie von normalen Arbeitsschritten unterscheiden zu können sind sie durch ein „E“ bzw. „M“ für Meilensteine gekennzeichnet. Wird eines dieser beiden Elemente aus der Worklist ausgewählt, so werden die normalen Inhaltsabschnitte im FORFLOW-Desktop durch einen speziellen Abschnitt für Entscheider bzw. Meilensteine ersetzt. Bei Entscheidern werden Informationen angezeigt, die dem Mitarbeiter bei der Entscheidung über den Fortgang im Prozess unterstützen. Außerdem muss vom Entwickler eine Begründung angegeben werden, warum ein Ast im Prozessmodell gewählt wurde, um die Auswahl später nachvollziehen zu können. Wird ein Meilenstein aus der Worklist gewählt, so werden alle Dokumente, die bis dorthin erstellt werden müssen, angezeigt. Der Mitarbeiter kann diese nochmals auf ihre Vollständigkeit überprüfen und anschließend den Meilenstein abschließen oder die Überarbeitung der Dokumente beauftragen.

6 Fazit

Ziel dieses Beitrags ist es den im FORFLOW Projekt entwickelten Prozessnavigator vorzustellen. Dazu wurden zuerst die Anforderungen, die sich an ein System zur Unterstützung von Produktentwicklungsprozess ergeben, betrachtet. Anschließend wurde gezeigt wie diese Anforderungen in einer prozessbasierten Ausführungsumgebung umgesetzt werden. Ein besonderes Interesse liegt dabei auf der Gestaltung der Benutzerschnittstelle. An die dort

vorhandene Worklist werden spezielle Anforderungen gestellt, da diese eine flexible Ausführungssemantik unterstützen muss. Dies wurde durch eine Aufteilung der Worklist in mehrere Abschnitte mit unterschiedlicher Priorität erreicht. Durch die Kombination aus unterteilter Worklist und dem Informationsangebot im FORFLOW Desktop ist eine systematische Benutzerführung im Prozess bei gleichzeitig maximaler Flexibilität gewährleistet.

7 Literatur

- [1] VDI Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, V. Verlag, Ed. Düsseldorf 1986.
- [2] *Web Services Business Process Execution Language Version 2.0*, OASIS, 2007; <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>, retrieved 2008-03-20.
- [3] Casati, F.; Ceri, S.; Paraboschi, S. and Pozzi, G.: Specification and implementation of exceptions in workflow management systems, *ACM Transactions on Database Systems (TODS)*, vol. 24, pp. 405-451, 1999.
- [4] Faerber, M.; Jochaud, F.; Stöber, C.; Jablonski, S. and Meerkamm, H.: Knowledge oriented Process Design for DfX, in *10th International Design Conference*, Dubrovnik, 2008.
- [5] Georgakopoulos, D.; Hornick, M. and Sheth, A.: An overview of workflow management: from process modeling to workflow automation infrastructure, *Distributed and Parallel Databases*, vol. 3, pp. 119-153, 1995.
- [6] Jablonski, S.: MOBILE: A Modular Workflow Model and Architecture, in *4th International Working Conference on Dynamic Modelling and Information Systems*, Noordwijkerhout, The Netherlands, 1994.
- [7] Jablonski, S. and Bussler, C.: *Workflow Management – Modeling Concepts, Architecture and Implementation*. London: International Thomson Computer Press, 1996.
- [8] Meerkamm, H.: Prozesse: eine aktuelle Herausforderungen in der Produktentwicklung, *Konstruktion*, vol. 9, p. 1, 2007.
- [9] van der Aalst, W. M. P.; ter Hofstede, A. H. M.; Kiepuszewski, B. and Barros, A. P.: Workflow Patterns, *Distributed and Parallel Databases*, vol. 14, pp. 5-51, 2003.

Dipl.-Inf. Matthias Faerber
Lehrstuhl für Angewandte Informatik IV
Universität Bayreuth
Universitätsstraße 30, D-95447 Bayreuth
Tel: +49-921-55-7626
Fax: +49-921-55-7622
Email: matthias.faeber@uni-bayreuth.de
URL: <http://www.ai4.uni-bayreuth.de>