

Fallstudie zur Modellierung von Software-Entwicklungsprozessen auf Basis von SPEM 2.0

Max Brunner^{1,2}, Martin Jung¹, Detlef Kips^{1,2}, Karsten Schmidt³

¹ develop group, Am Weichselgarten 4, 91058 Erlangen,
{brunner | jung | kips}@develop-group.de

² Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Informatik 2,
Martensstraße 3, 91058 Erlangen

³ Audi Electronics Venture GmbH, Sachsstraße 18, 85080 Gaimersheim,
karsten.schmidt@audi.de

Abstract: Das SPEM (Software Process Engineering Metamodel) der OMG ist ein UML-basiertes Metamodell zur Spezifikation von Software-Entwicklungsprozessmodellen. Die vorliegende Fallstudie beschreibt die Erfahrungen, welche die Autoren beim praktischen Einsatz des SPEM gewonnen haben. Ziel der Fallstudie war es, einen in der Praxis etablierten Entwicklungsprozess aus dem Bereich der Automobilindustrie auf Basis des SPEM 2.0 möglichst präzise zu beschreiben.

1 Einleitung

Ein Software-Entwicklungsprozessmodell (SWEPM) dient zur präzisen Beschreibung eines Software-Entwicklungsprozesses (SWEP). Im Mittelpunkt der Beschreibung stehen dabei typischerweise die am Prozess beteiligten Rollen, ferner die Artefakte, die als Ergebnisse des Prozesses entstehen sollen, sowie die Abläufe und Aktivitäten, die im Rahmen einer prozesskonformen Projektabwicklung durchgeführt werden müssen.

In den letzten Jahren wird sowohl im wissenschaftlichen als auch im industriellen Umfeld ein wachsendes Augenmerk auf eine exakte Beschreibung derartiger Modelle gerichtet. Speziell im Automotive-Bereich wird die präzise Beschreibung von SWEPMen zunehmend als Herausforderung erkannt. Zum einen steigt die Vernetzung und damit die Komplexität der Zielsysteme. Zum anderen existieren hohe Anforderungen an Sicherheit, Verfügbarkeit und Fehlertoleranz, deren Nichtbeachtung extrem hohe Kosten in der Serie verursachen kann. Ein weiteres Problemfeld stellt die komplexe Interaktion zwischen dem jeweiligen OEM und seinen Lieferanten dar, die einer präzisen Beschreibung der Schnittstellen und Synchronisationsmechanismen bedarf. Die Tendenz zur Präzisierung und Standardisierung der Software-Entwicklungsprozesse wird außerdem durch den AUTOSAR-Standard [Au07] und die DIN-Norm EN 61508 [Di02] verstärkt.

Diesen Anforderungen kann durch die bislang weit verbreitete Beschreibung der Prozesse in Form von informellen (Hyper-)Textdokumenten, Handbüchern und Task-Listen nur unzureichend Rechnung getragen werden. Dagegen ermöglicht eine (semi-)formale Spezifikation von SWEPMen eine weitgehend automatisierte Prüfung der Modellkonsistenz sowie die Zusicherung bestimmter kritischer Prozesseigenschaften, z.B. im Hinblick auf die Konformität zu Referenzmodellen wie CMMI [CKS06] oder SPICE [HDM07].

Darüber hinaus wird dadurch sowohl die projektspezifische Anpassung (das "Process Tailoring") als auch die eigentliche Ausführung des Prozesses (das "Process Enactment") deutlich erleichtert. Schließlich schafft eine Formalisierung des SWEPMs auch eine präzise Grundlage für die (zumindest partielle) Automatisierung der Prozessabläufe.

In der Literatur finden sich diverse Ansätze zur (semi-)formalen Beschreibung von SWEPMen, z.B. [FKN94] [Gr02] [Ar02] [He02]. Dabei handelt es sich zumeist um Prozessbeschreibungssprachen, die sich jedoch in der Praxis nicht dauerhaft etablieren konnten. Aus dem Bereich des *Business Process Modelling* sind ferner Ansätze zur Modellierung von Geschäftsprozessen mittels Workflows bzw. Ereignisgesteuerter Prozessketten (EPKs) bekannt [St06]. Auf diesen Ansätzen beruht u.a. auch das im Bereich betriebswirtschaftlicher Anwendungen verbreitete Prozessmodellierungswerkzeug ARIS [Sc01]. Diese Ansätze sind jedoch nur bedingt zur Modellierung realistischer SWEPMen geeignet, da sie aufgrund ihres grobgranularen Detaillierungsgrades nur eingeschränkte Ausdrucksmöglichkeiten für komplexe Aktivitäten und logische Abhängigkeiten bieten.

Eine Grundlage für die formalisierte Darstellung speziell von Software-Erstellungsprozessen bietet das sog. SPEM ("*Software Process Engineering Metamodel*"), das von der OMG (*Object Management Group*) derzeit in der Version 2.0 zur Verfügung gestellt wird. Der vorliegende Erfahrungsbericht beschäftigt sich mit der Anwendung des SPEM auf einen in der Praxis etablierten Entwicklungsprozess aus der Automobilindustrie.

2 Das "Software Process Engineering Metamodel" der OMG

Mit dem Ziel, eine einheitliche, präzise und möglichst allgemein gültige Modellierungsnotation für SWEPMen zu schaffen, arbeitet die OMG seit dem Jahr 2001 an der Definition des SPEM. Nach den offiziellen Versionen 1.0 [Ob02] und 1.1 [Ob05b] befindet sich derzeit (August 2007) die Version 2.0 [Ob07] kurz vor dem Abschluss.

Formal basiert SPEM 2.0 auf der UML (*Unified Modeling Language*) 2.0 [Ob05a] und nutzt deren Metamodellierungsmechanismen im Sinne einer konsistenten Erweiterung des UML-Sprachumfangs. Das SPEM 2.0 stellt dem Prozessmodellierer somit die volle Mächtigkeit der UML 2.0 zur Verfügung, ergänzt um spezifische Konzepte (Klassen, Beziehungen) zur Beschreibung von SWEPMen.

Abb. 1 zeigt einen kleinen Ausschnitt der SPEM-Definition, in dem die Beziehungen zwischen einigen "typischen" SPEM-Metaklassen spezifiziert werden. Eine *RoleDefinition* beschreibt die Rolle eines am Prozess mitwirkenden Akteurs, wobei eine Person mehrere Rollen repräsentieren kann. Ein Arbeitspaket, das von einer oder mehreren Rollen ausgeführt wird und ein bestimmtes Entwicklungsziel verfolgt, wird durch eine *TaskDefinition* charakterisiert. Eine *WorkProductDefinition* beschreibt ein Artefakt bzw. Arbeitsergebnis, welches bei der Ausführung von Tasks produziert, konsumiert oder verändert werden kann und dessen Erstellung in der Verantwortung einer Rolle liegt. Eine *Activity* beschreibt einen Prozessschritt innerhalb eines SWEP, wobei Aktivitäten hierarchisch ineinander verschachtelt sein dürfen und wiederum aus mehreren Tasks bestehen können. Die diversen *Map*-Metaklassen repräsentieren Abbildungen zwischen den Instanzen der durch sie verknüpften Metaklassen, z.B. setzt die *TaskDefinitionPerformerMap* einzelne Tasks zu den für ihre Ausführung zuständigen Rollen in Beziehung.

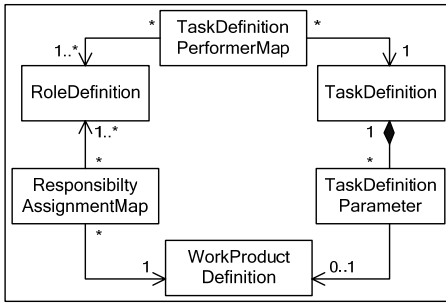


Abb. 1: Ausschnitt der SPEM-Definition

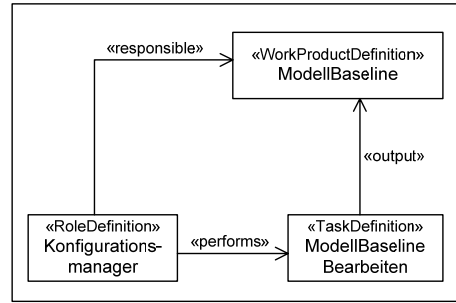


Abb. 2: Ausschnitt eines SWEPM

Abb. 2 zeigt einen Ausschnitt eines konkreten Prozessmodells, der auf dem in Abb. 1 dargestellten Metamodellausschnitt basiert. Beispielsweise wird durch die mit dem Stereotyp *«performs»* annotierte Kante ausgedrückt, dass die Rolle *Konfigurationsmanager* für die Ausführung des Arbeitspakets *ModellBaselineBearbeiten* zuständig ist. Dies entspricht einer Instanzierung der Beziehungen zwischen den Metaklassen *RoleDefinition*, *TaskDefinition* und *TaskDefinitionPerformerMap* aus Abb. 1.

Ein entscheidender Vorzug des SPEM besteht zweifellos darin, dass durch die UML-Einbettung eine hohe formale Präzision und eine beliebig skalierbare Granularität der Prozessmodellbeschreibung erreicht werden kann. Allerdings setzt diese Art der Prozessmodellierung sowohl fundierte UML-Kenntnisse als auch ein solides Verständnis der UML-basierten Metamodellierungskonzepte voraus. Die daraus resultierende Komplexität für den Modellierer kann jedoch durch geeignete Werkzeuge erheblich reduziert werden, und nachdem die OMG das SPEM als offenen Standard entwickelt, ist bei entsprechender industrieller Akzeptanz eine gute Toolunterstützung mittelfristig absehbar.

Die SPEM-Version 1.x hat allerdings in der Industrie eher geringen Anklang gefunden, da einige Konzepte hinsichtlich ihrer Praxistauglichkeit noch nicht ausgereift waren. [Al05] beschreibt sehr detailliert die Mächtigkeit und Grenzen von SPEM 1.x. Das *Eclipse Process Framework (EPF)* orientiert sich zwar an SPEM 2.0, jedoch ist dessen Umsetzung derzeit noch nicht abgeschlossen, so dass eine Evaluation des EPF im Hinblick auf SPEM 2.0 im Rahmen unserer Fallstudie nicht möglich war [Ha07].

3 Hintergründe der Fallstudie

Die in der Einleitung skizzierten spezifischen Anforderungen an Entwicklungsprozesse im Bereich der Automobilindustrie haben dazu geführt, dass die AUDI AG als großer, weltweit tätiger Automobilhersteller einer präzisen Modellierung ihrer SWEPE verstärkte Aufmerksamkeit widmet. Die bei Audi praktizierten Lösungsansätze beruhten bislang teils auf eher grobgranularen Konzepten zur Geschäftsprozessmodellierung (wie etwa ARIS), teils auf textuellen Beschreibungen, die sich im Hinblick auf geforderte Konsistenzprüfungen sowie im Hinblick auf die Nachweisfähigkeit kritischer Prozesseigenschaften als schwer handhabbar erwiesen haben. In Kooperation mit der Universität Erlangen-Nürnberg und dem Beratungshaus develop group wurde daher die hier beschriebene Fallstudie durchgeführt. Die Aufgabenstellung des Projektes bestand darin,

einen SWEP aus dem Bereich der Steuergeräteentwicklung der AUDI AG mit den Ausdrucksmitteln des SPEM 2.0 hinreichend präzise zu modellieren und das resultierende SWEPM mit einem gängigen UML-Modellierungswerkzeugs zu erfassen. Ferner sollte eine Möglichkeit geschaffen werden, aus dem erstellten Prozessmodell eine rollenrechte Prozessdokumentation für die einzelnen Prozess-Akteure zu generieren.

Ausgehend von dieser praktischen Aufgabenstellung wurden mit der Fallstudie die folgenden Zielsetzungen verfolgt:

- Untersuchung der Praxistauglichkeit des SPEM 2.0, insbesondere im Hinblick auf einen Einsatz zur Modellierung von SWEPen in der Automobilindustrie
- Zusammenstellung von "Best Practices" für die SWEP-Modellierung mit SPEM
- Ermittlung von Schwächen und Verbesserungspotentialen der SPEM-2.0-Definition
- Ermittlung von etwaigen Schwächen, Inkonsistenzen und Verbesserungspotenzialen des real gelebten Prozesses anhand der SPEM-basierten Prozessbeschreibung

Darüber hinaus erhofften sich die Projektbeteiligten weitere Aufschlüsse über

- die Möglichkeiten einer modellbasierten Definition geeigneter Prozess-Schnittstellen zu den am Entwicklungsprozess beteiligten Lieferanten,
- die prinzipiellen Möglichkeiten eines modellbasierten Konformitätsnachweises zu einschlägigen Referenzmodellen, z.B. zu Automotive SPICE [HDM07]
- sowie die prinzipiellen Möglichkeiten einer werkzeuggestützten, weitgehend automatisierten Prozessausführung anhand von SPEM-basierten Prozessbeschreibungen.

4 Umsetzung der Fallstudie

Als Modellierungsgegenstand wurde im Rahmen der Fallstudie ein repräsentativer Ausschnitt des bei Audi etablierten Prozesses zur *Seriencodegenerierung* gewählt. Dabei handelt es sich um die Erzeugung von Steuergeräte-Software aus zuvor erstellten Funktions- und Regelungsmodellen. Der Prozess folgt einem Phasenmodell, das die einzelnen Prozessstufen (z.B. Planung oder Spezifikation) genauer beschreibt, wobei die Phasen teilweise auch parallel durchgeführt werden können. Als Grundlage für die SPEM-basierte Modellierung des Prozesses diente die vorhandene Prozessdokumentation. Sie besteht i.w. aus Schablonen für die zu erstellenden Artefakte sowie aus Task-Listen mit einer groben Beschreibung der einzelnen Phasen und Aktivitäten des SWEPes.

Zur SPEM-basierten Modellierung des gewählten Prozessausschnitts wurden i.w. drei Diagrammtypen der UML 2.0 eingesetzt. Deren Anwendung wird in Abb. 3 bis 6 exemplarisch anhand von (stark vereinfachten) Ausschnitten des SWEPMs veranschaulicht.

Die Grobstruktur des Prozesses sowie der Ablauf der einzelnen Prozessphasen wird mit *Aktivitätsdiagrammen* beschrieben (vgl. Abb. 3). Durch ihre hierarchische Struktur bieten sie eine flexibel detaillierbare Sicht auf die einzelnen zu erledigenden Tasks (vgl. Abb. 4). Der Bearbeitungszustand von Artefakten sowie deren Verwendung als Input oder Output der Tasks kann in UML durch Objektflüsse und Pins modelliert werden.

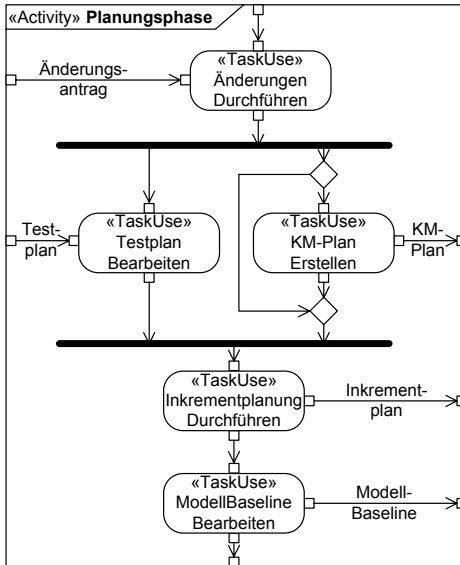


Abb. 3: Ablauf einer Prozessphase als Aktivitätsdiagramm (Ausschnitt)

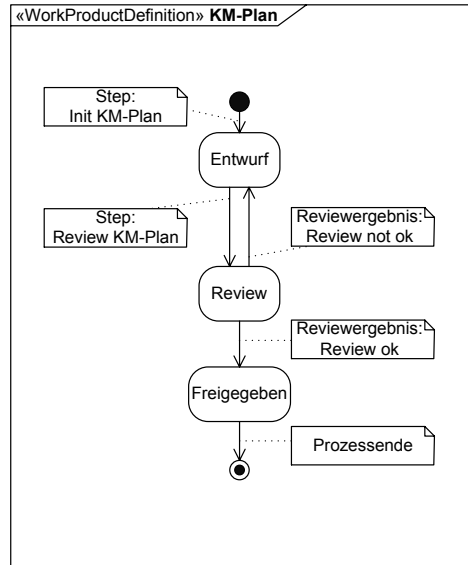


Abb. 5: Lebenszyklus eines Artefakts als Zustandsdiagramm

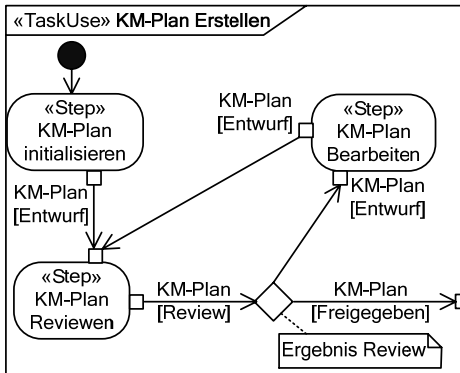


Abb. 4: Detaildarstellung eines Task als Aktivitätsdiagramm

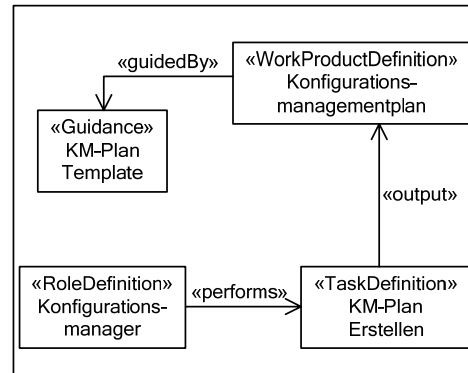


Abb. 6: Beziehungen zwischen Rollen, Tasks und Artefakten als Klassendiagramm

Mit *Zustandsdiagrammen* wird der Lebenszyklus von Artefakten dargestellt (vgl. Abb. 5). Die Zustandsübergänge erfolgen dabei typischerweise als Ergebnis der Durchführung einzelner Arbeitsschritte im Prozessablauf und sind entsprechend annotiert.¹

Ferner werden *Klassendiagramme* verwendet, um die Beziehungen zwischen Rollen, Artefakten und Tasks zu spezifizieren (vgl. Abb. 6). Dabei wird die Semantik der einzel-

¹ Dass dies laut SPEM-Spezifikation [OMG07] nicht durch die Angabe entsprechender Trigger-Ereignisse, sondern in Form von UML-Notes geschehen muss (vgl. Abb. 5), wurde im Rahmen unserer Fallstudie als eine der Schwächen der aktuellen SPEM-Version identifiziert.

nen Beziehungskanten durch entsprechende Stereotypen (im Beispiel etwa «*performs*») gemäß der SPEM-Spezifikation näher charakterisiert.

Wie bei jedem komplexeren UML-basierten Modell gibt es auch bei nichttrivialen SWEPMen in der Regel zahlreiche Eigenschaften, die durch die graphischen Ausdrucksmittel der verwendeten Diagrammtypen nicht adäquat abgebildet werden können. So erwies es sich auch in unserer Fallstudie als erforderlich, die graphische Spezifikation des Prozessmodells durch zusätzliche Restriktionen, Integritätsbedingungen und Invarianten zu ergänzen. Um auch für derartige kontextsensitive Prozesseigenschaften eine hinreichend präzise und standardisierte Repräsentation bereitzustellen, wurden sie als Constraints in der First-Order-Logic-Spezifikationssprache OCL formuliert [Ob06].

Nachfolgend wird an einem einfachen Beispiel gezeigt, wie die Eigenschaften der Klassen des modellierten Prozessausschnittes durch zwei in OCL formulierte Invarianten restringiert werden. Durch die erste Invariante (für die SWEPM-Klasse *WorkProductDefinition*) wird festgelegt, dass alle im Prozess verwendeten Artefakte ein Typ-Attribut haben, durch das sie entweder als *intern* oder als *extern* klassifiziert werden. Die zweite Invariante (für die SWEPM-Klasse *TaskUse*) sagt aus, dass für Aktivitäten, die vor einem bestimmten Synchronisationspunkt ablaufen ("*pre*"), nur externe Artefakte als Input verwendet werden dürfen, da die prozesskonforme Bereitstellung von internen Artefakten vor dem Synchronisationspunkt nicht gewährleistet ist.

```
① CONTEXT WorkProductDefinition INV:
    self.type->notEmpty
    AND (self.type = "intern" OR self.type = "extern")
② CONTEXT TaskUse INV:
    self.scheduletype = "pre"
    IMPLIES self.input.forAll(w:WorkProductDefinition
        | w.scheduletype = "extern")
```

Zur toolgestützten Erstellung des SWEPMs wurde im Rahmen der Fallstudie das Modellierungswerkzeug *Rational Software Architect (RSA)* von IBM eingesetzt [Ib07]. Als Basis für die Prozessmodellierung im RSA wurde das "SPEM 2.0 Profile" der OMG genutzt, welches die erforderlichen UML-Stereotypen zur Verfügung stellt.

Da die Architektur des RSA auf der Eclipse-Plattform und dem *Eclipse Modelling Framework (EMF)* basiert, konnte deren Plugin-Mechanismus verwendet werden, um die Funktionalität des RSA zu erweitern. Zum einen wurde der Integritätsprüfungsmechanismus durch ein Plugin zur Erweiterung des EMF-Validation-Plugins implementiert. Zum anderen wurde durch die Realisierung eines Pluglets eine prototypische Generierung der rollengerechten Dokumentation in Tabellenform ermöglicht.

5 Ergebnisse

Die Ergebnisse der hier vorgestellten Fallstudie sind ausführlich in [Br07] dargestellt.

Zusammenfassend lässt sich festhalten, dass die detaillierte und präzise Abbildung eines realistischen, in der Praxis etablierten SWEPes mit den Ausdrucksmitteln des SPEM 2.0 durchaus möglich ist. Allerdings ist die konkrete Modellierung derzeit noch mit diversen

Problemen behaftet, die in erster Linie auf den unzureichenden Reifegrad der aktuellen SPEM-Version zurückzuführen sind. Weitere Schwierigkeiten bei der praktischen Arbeit ergaben sich aus der Verwendung des für die Umsetzung des Metamodells nicht optimal geeigneten UML-Profiles sowie aus dem Umstand, dass zur Zeit noch kein Werkzeug existiert, das speziell auf eine SPEM-basierte Prozessmodellierung ausgerichtet ist.

Neben derartigen praktischen Problemen sind im Rahmen der Fallstudie auch einige konzeptionelle Schwächen und semantische "Lücken" der aktuellen SPEM-Definition deutlich geworden, die eine Erweiterung bzw. Ergänzung des Metamodells sinnvoll erscheinen lassen. Beispielsweise existieren zu einigen Assoziationen keine Stereotypen, was zur Folge hat, dass die Semantik entsprechender Beziehungen im erstellten Prozessmodell unklar bleibt. So musste z.B. der in Abb. 6 dargestellte Stereotyp «*guidedBy*» hinzugefügt werden, um die Assoziation zwischen *KM-Plan* und *KM-Plan Template* mit einer spezifischen Semantik zu versehen. Als Hindernis für eine einfache und intuitive Modellierung erwies sich ferner die - konzeptionell durchaus begründete - Unterscheidung zwischen den wiederverwendbaren Prozess-Bausteinen im sogenannten *Method Content* und deren Verwendung im Kontext einer konkreten Prozessausführung. Schließlich enthält die verwendete SPEM-Spezifikation bis auf wenige Ausnahmen keine "Wellformedness Rules" für die syntaktische Korrektheit von SPEM-2-basierten Modellen. Diese Lücke, die der Prozessmodellierer derzeit durch Modellierungserfahrung und gesunden Menschenverstand überbrücken muss, soll aber nach Auskunft der OMG in einer der nächsten offiziellen Versionen des SPEM geschlossen werden.

Dennoch hat die Durchführung der Fallstudie gezeigt, dass durch eine konsequente Verwendung der SPEM-Modellierungskonzepte ein erheblicher Grad an Detailpräzision und ein wesentlich besseres Verständnis komplexer Zusammenhänge und Abhängigkeiten innerhalb eines SWEPMs erreicht werden kann. Dies hat sich bereits während der eigentlichen Modellierungsarbeit als ausgezeichneter "Katalysator" für die Aufdeckung von Schwächen, Inkonsistenzen und Verbesserungspotenzialen des modellierten Prozesses erwiesen. Einige grobe Konzeptüberlegungen, die während der Fallstudie angestellt wurden, haben darüber hinaus gezeigt, dass ein SPEM-basiertes SWEPM eine geeignete Grundlage für automatische Modellprüfungen und, davon ausgehend, für einen möglichen modellbasierten Nachweis der Konformität zu einschlägigen Referenzmodellen, z.B. zu Automotive SPICE, darstellt.

6 Ausblick

Ausgehend von den Ergebnissen der Fallstudie sollen in nächster Zukunft weitere Arbeitspakete in Angriff genommen werden. Auf Seiten der AUDI AG ist zunächst die SPEM-basierte Beschreibung und Analyse weiterer Prozessausschnitte geplant, um die daraus resultierenden Potenziale zur Prozessverbesserung zu nutzen. Mittel- und langfristig ist daran gedacht, die Definition von Prozess-Schnittstellen zu den Prozessen der Lieferanten mittels SPEM auf eine präzise gemeinsame Basis zu stellen.

Auf Seiten der wissenschaftlichen Kooperationspartner sollen die Ergebnisse der Fallstudie dazu verwendet werden, unter dem Aspekt der Praxistauglichkeit und der Modellierungsvereinfachung sinnvolle Erweiterungen und Ergänzungen der aktuellen SPEM-Definition zu erarbeiten und in den Standardisierungsprozess der OMG einzubringen.

Dies betrifft insbesondere auch die derzeit größtenteils noch fehlenden "Wellformedness Rules" für die kontextsensitive SPEM-Syntax. Mittelfristig ist die prototypische Implementierung eines SPEM-spezifischen Modellierungswerkzeugs mit integrierten Model-Checking-Mechanismen geplant. Als Fernziel ist schließlich auch an eine werkzeuggestützte, weitgehend automatisierte Prozessausführung anhand von SPEM-basierten Prozessbeschreibungen gedacht.

Literaturverzeichnis

- [Al05] Al-Hilank, S.: Fallstudie zur Modellierung von SWEP mit SPEM. Diplomarbeit am Institut für Informatik der Friedrich-Alexander-Universität Erlangen-Nürnberg, 2005.
- [Ar02] Arbaoui, S. et al.: A Comparative Review of Process Centered Software Engineering Environments. In: Wang, Y.; Bryant, A. (Hrsg.): Annals of Software Engineering 14. Springer, London, 2002; S. 311-340.
- [Au07] AUTOSAR Konsortium: AUTOSAR - Technical Overview, 01/2007. Abruf unter www.autosar.org
- [Br07] Brunner, M.S.: Fallstudie zur Modellierung von Software-Entwicklungsprozessen auf Basis des Software Process Engineering Metamodel 2.0. Diplomarbeit am Institut für Informatik der Friedrich-Alexander-Universität Erlangen-Nürnberg, 2007.
- [CKS06] Chrissis, M.B., Konrad, M., Shrum, S.: CMMI - Guidelines for Process Integration and Product Improvement. Addison-Wesley, New York, 2. Auflage 2006.
- [Di02] DIN EN 61508, Berlin: DIN, 2002.
- [FKN94] Finkelstein, A., Kramer, J., Nuseibeh, B.: Software Process Modelling and Technology. John Wiley & Sons Inc., New York, 1994.
- [Gr02] Gruhn, V.: Process Centered Software Engineering Environments, A Brief History and Future Challenges. In: Wang, Y.; Bryant, A. (Hrsg.): Annals of Software Engineering 14. Springer, London, 2002; S. 363-382.
- [Ha07] Haumer, P.: Eclipse Process Framework Composer – Part1: Architectural Overview, 2007, www.eclipse.org/epf, Abruf: 15.08.2007.
- [He02] Henderson-Sellers, B.: Process Metamodelling and Process Construction - Examples Using the OPEN Process Framework (OPF). In: Wang, Y.; Bryant, A. (Hrsg.): Annals of Software Engineering 14. Springer, London, 2002; S. 341-362.
- [HDM07] Hörmann, K., Dittmann, L., Müller, M.: Automotive SPICE in der Praxis. Interpretationshilfen für Anwender und Assessoren. dpunkt, Heidelberg, 2007.
- [Ib07] IBM, Rational Software Architect 7, www.ibm.com, Abruf: 15.08.2007.
- [Ob02] OMG Konsortium, OMG SPEM 1.0 Spezifikation, 02-11-14, 2002.
- [Ob05a] OMG Konsortium, OMG UML 2.0 Spezifikation, 05-07-04, 2005.
- [Ob05b] OMG Konsortium, OMG SPEM 1.1 Spezifikation, 05-11-06, 2005.
- [Ob06] OMG Konsortium, OMG OCL 2.0 Spezifikation, 06-05-01, 2006.
- [Ob07] OMG Konsortium, OMG SPEM 2.0 Spezifikation, 07-03-03, 2007.
- [Sc01] Scheer, A.-W.: ARIS - Modellierungs-Methoden, Metamodelle, Anwendungen. Springer, Berlin, 4. Auflage 2001.
- [St06] Staud, J.: Ereignisgesteuerte Prozessketten und Objektorientierte Geschäftsprozessmodellierung für Betriebswirtschaftliche Standardsoftware. Springer, Berlin, 3. Auflage 2006.