

Prozessmodellierung im Web-Zeitalter

Gero Decker und Mathias Weske
Hasso-Plattner-Institut, Universität Potsdam
(gero.decker,mathias.weske)@hpi.uni-potsdam.de

Abstract: Web-Technologien haben mittlerweile einen solch hohen Reifegrad erreicht, dass heute komplexe und hochinteraktive Anwendungen über das Web angeboten werden können. Dadurch werden Anwendungen leichter zugänglich und Installations- und Administrationsaufwand werden drastisch reduziert. Darüber hinaus werden durch die Einbindung von Web 2.0-Konzepten neue Anwendungsfälle unterstützt, die mit herkömmlichen Desktop-Applikationen nur schwer zu realisieren wären.

Dieser Artikel skizziert einige dieser neuen Anwendungsfälle und stellt für den Bereich der Geschäftsprozessmodellierung das webbasierte Modellierungswerkzeug Oryx vor, welches diese neuen Möglichkeiten erschließt. Schließlich werden Einsatz-erfahrungen erläutert, und ein Ausblick auf zukünftige Arbeiten wird gegeben.

1 Motivation und Einführung

Zur Geburtsstunde des World Wide Web (WWW) wurden vorrangig statische Seiten veröffentlicht, die sich gegenseitig referenzieren. Nach Einführung von Formatierung und Einbindung anderer Artefakte, wie z.B. Bildern, erschienen schließlich dynamische Webseiten, auf denen je nach Anfrage dynamisch der Seiteninhalt erzeugt wird. Dies ermöglicht die Umsetzung einfacher Anwendungen, wie z.B. Online-Shops. Technologisch wichtige Systeme sind dabei der Browser zur Anzeige der Seiteninhalte und der Webserver zur statischen oder dynamischen Bereitstellung der Inhalte [BL96].

In einer nächsten Generation wurden clientseitige Skripte populär, die nach dem Laden der Webseite im Browser ausgeführt werden. Während zunächst nur sehr einfache Funktionalität damit umgesetzt wurde, wie z.B. visuelle Effekte oder Validierungen von Formularinhalten, standen mit der Zeit immer mehr Skriptbibliotheken zur Verfügung, mit denen sehr schnell auch komplexere Funktionalität realisiert werden kann. Als Hauptsprache kommt hier JavaScript zum Einsatz. Mit Hilfe von AJAX (asynchronous JavaScript and XML) können Anwendungen implementiert werden, die nicht bei jeder Nutzerinteraktion ein komplettes Neuladen der Webseite erfordern [Min05].

Unter Zuhilfenahme von JavaScript und entsprechender Bibliotheken war es nun möglich, auch solche Anwendungen über das Web anzubieten, die zuvor nur als Desktopanwendungen zur Verfügung standen. Diese *Rich Internet Applications* zeichnen sich vor allem dadurch aus, dass die Daten zusammen mit der Anwendung vom Browser geladen werden. Somit wird eine Installation von Applikationen auf dem Desktop vermieden und mehrere Personen können auf die zentral verwalteten Daten zugreifen. Als prominentes Beispiel

lässt sich Google Apps¹ anführen, die u.a. eine webbasierte Office-Software anbietet.

Einige der Vorteile von Webanwendungen sind (1) einfache Referenzierung von Daten und Anwendungen durch Weiterreichen von Weblinks, (2) Vermeidung von Softwareinstallation auf dem Desktop, (3) höhere Benutzerpartizipation durch niedrige Zugangsbarrieren, (4) Bereitstellung einer aktuellen Softwareversion anstelle von Softwareupdates und deren Rollouts und (5) einfache Integrierbarkeit von Webanwendungen, Stichwort Mashups [VH07].

Aus der Sicht der Geschäftsprozessmodellierung, wo grafische Modelle genutzt werden, um Arbeitsabläufe in Unternehmen darzustellen, zu verbessern und effizienter auszuführen [Wes07], sind grafische Editoren die zentralen Werkzeuge. Diese Werkzeuge als Rich Internet Applications anzubieten, wurde technisch erst durch Standards für Vektorgrafiken im Webbrowser, wie z.B. Scalable Vector Graphics (SVG) ermöglicht. Die erhöhte Rechenkomplexität bei der Manipulation und Anzeige von Vektorgrafiken wird bei modernen Browsergenerationen dadurch kompensiert, dass die Ausführungsgeschwindigkeit von JavaScript drastisch verbessert wurde.

Bei webbasierten Prozessmodellierungswerkzeugen handelt es sich nicht nur um eine neue Technologie, die auf einen gut verstandenen Bereich angewandt wird. Vielmehr sind nun, ähnlich wie bei anderen Webanwendungen, ganz neue Anwendungsfälle möglich, die auch neue Herausforderungen mit sich bringen.

Diese Anwendungsfälle gehen einher mit einschlägigen Trends im Bereich der Prozessmodellierung. Zum einen gibt es den Trend der Dezentralisierung von Prozessmodellierung. Während es bisher üblich war, dass zentrale Organisationsabteilungen Prozesse abbilden und daraus abgeleitete Anforderungen an die IT-Abteilungen weiterreichen, gehen einige Organisationen dazu über, Prozessmodellierung in die Linie zu verteilen, manchmal sogar bis auf Abteilungsebene. Dies erzeugt eine große Anzahl von "Gelegenheitsmodellierern", die sich zusätzlich zu ihren eigentlichen Aufgaben Prozessmodellierungswissen aneignen und die Prozesse abbilden.

Als zweiter großer Trend ist organisationsübergreifende Prozessmodellierung zu beobachten. Diese wird dadurch forciert, dass immer mehr Interaktionen zwischen Organisationen über IT abgewickelt werden und dadurch eine genauere Spezifikation des gewünschten Verhaltens nötig wird. Zum anderen gibt es gerade im öffentlichen Sektor große Prozessmodellierungsinitiativen, die durch gesetzliche Vorgaben befeuert wurden. Z.B. die EU-Dienstleistungsrichtlinie verlangt von Behörden und Ämtern eine engere Zusammenarbeit, welcher wiederum eine Einigung auf Prozessmodelle vorausgeht.

Dieser Artikel wird diese beiden Trends in den Abschnitten 2 und 3 näher beleuchten. Im Anschluss wird das webbasierte Prozessmodellierungswerkzeug Oryx und die dahinter liegenden Design-Entscheidungen und die Architektur vorgestellt. Ein Ausblick auf zukünftige Arbeiten beschließt diesen Artikel.

¹Siehe <http://www.google.com/a/>

2 Verteilte und partizipative Prozessmodellierung

Prozessmodelle dienen vorrangig der Kommunikation zwischen Menschen. Sie stellen eine Kommunikationsbasis dar, die von allen an einem Prozess Interessierten verstanden und nachvollzogen werden kann. Folgende Stakeholder lassen sich identifizieren:

1. *Prozessbeteiligte* führen Aktivitäten in Geschäftsprozessen aus, müssen aber nicht zwingend an der Erstellung des Prozessmodells beteiligt sein. Sie werden zur Validierung von Prozessmodellen herangezogen und können idealerweise bei einem gegebenen Modell urteilen, ob dieses den Arbeitsablauf korrekt widerspiegelt.
2. *Prozessmodellierer* erstellen Prozessmodelle. Sie kommunizieren mit anderen Personen, um die Richtigkeit des Modells sicherzustellen. Ein Prozessmodellierer hat so detailliertes Prozesswissen, dass er fachliche Anforderungen dem Prozessimplementierer gegenüber äußern kann.
3. *Prozessverantwortliche* müssen entweder den reibungslosen Ablauf von Prozessen sicherstellen oder auch die strategische Ausrichtung von Prozessen dirigieren. Sie müssen Modelle lesen können, um Verbesserungspotenziale lokalisieren zu können.
4. *Prozessimplementierer* sind für die IT-Unterstützung von Geschäftsprozessen verantwortlich. Entweder lesen sie die Prozessmodelle und verwenden sie als Anforderungsspezifikationen für die Softwareentwicklung oder sie verfeinern die Prozessmodelle bis hin zur Ausführbarkeit in einer Prozessausführungsebene.

Diese Auflistung von Stakeholdern zeigt bereits, dass sehr viele Personen in Organisationen auf Prozessmodelle zugreifen müssen. Viele davon haben nur lesenden Zugriff, andere sind an der Erstellung beteiligt. Ein Trend zeigt jedoch, dass die Anzahl der prozessgestaltenden Personen in Organisationen eher zu- denn abnimmt. Prozesse werden nicht mehr nur in einer zentralen Organisationsabteilung modelliert, sondern teilweise auch schon auf der Abteilungsebene. Dies hat viele Prozessmodellierer zur Folge, die nur einen Bruchteil ihrer Zeit mit Prozessmodellierung verbringen. Diese Nutzer stellen insofern eine Herausforderung dar, da bislang Werkzeuginstallationen an diese Personen ausgerollt werden musste. Zum anderen musste die Barriere überbrückt werden, dass der Mitarbeiter mit einem zusätzlichen Werkzeug arbeiten muss.

Dies wird durch webbasierte Modellierungswerkzeuge drastisch vereinfacht, da ein Rollout der Modellierungssoftware nicht mehr nötig ist und die Zugangsbarriere für den einzelnen Mitarbeiter sehr viel geringer ausfällt. So verhält sich der vertraute Webbrowser allein durch das Anklicken eines Links wie ein Modellierungswerkzeug. Durch ein vereinfachtes User Interface kann der Prozessbeteiligte sehr schnell einfache Modellierungsaufgaben übernehmen, wie z.B. das Umbenennen von Aktivitäten.

Durch diese technologische Neuerung können mehr Mitarbeiter dazu gebracht werden, ihr Wissen zu dokumentieren. Ähnlich wie Wiki-Systeme zu mehr Partizipation bei der Erstellung von Textdokumenten führen [VH07], führt ein webbasiertes Modellierungswerkzeug zu mehr Partizipation in der Modellierung von Geschäftsprozessen und damit auch zu einer höheren Akzeptanz der Prozessmodellierung in den Fachabteilungen.

Allerdings zeigt sich bei erhöhter Partizipation auch eine Herausforderung: Es muss sichergestellt werden, dass wichtige Prozessmodelle nicht beliebig überschrieben werden können. Hier kommen Freigabeworkflows ins Spiel, durch die entsprechende Änderungen nachverfolgt und genehmigt werden können. Durch die webbasierte Lösung sind auch solche Freigabeworkflows einfach konfigurierbar, weil alle Stakeholder auf einem gemeinsamen Datenbestand arbeiten und die eigentliche Freigabe durch eine zentrale Autorisierung realisiert werden kann.

3 Prozessmodellierung über Organisationsgrenzen hinweg

Bei der traditionellen Prozessmodellierung mit Client-Server-Applikationen stellten sich besondere Herausforderungen, wenn verschiedene Organisationen gemeinsam an Prozessmodellen arbeiten wollten. Dafür gibt es vor allem zwei Gründe.

1. *Getrennte Modellrepositories.* Bei klassischen Modellierungswerkzeugen gibt es zwar ein organisationsweites Repository, in dem alle Prozessmodelle gespeichert sind und auf das alle berechtigten Mitarbeiter der Organisation zugreifen können. Allerdings wird ein solches Repository typischerweise nicht für Mitarbeiter anderer Organisationen zugreifbar gemacht. Hierfür wäre das Hinzufügen der Mitarbeiter in die Benutzerverwaltung nötig, womit womöglich auch Zugriff auf andere, sicherheitskritische Systeme gewährt würde. Daher hat jede Organisation auch bei organisationsübergreifenden Prozessmodellierungsprojekten typischerweise ihr eigenes Modellrepository.

Durch diese redundante Verwaltung von Prozessmodellen entstehen die typischen Probleme. Zum Austausch müssen Modelle exportiert werden und dann per E-Mail oder auf einem physischen Datenträger verschickt werden. Die empfangenen Modelle müssen wiederum in das eigene Modellrepository eingepflegt werden. Dies resultiert in großer zeitlicher Verzögerung bei der Modellierung und kann zu Inkonsistenzen führen. Zum anderen ist der Aufwand typischerweise so groß, dass auf einen Austausch wenn möglich verzichtet wird. Auch existieren manchmal weitreichende Verlinkungen zwischen Modellen (darunter auch zu Modellen, die nicht zu der organisationsübergreifenden Initiative gehören), die bei der Verwendung der Modelle in verschiedenen Repositories verloren gehen können.

Ein weiteres Problem ergibt sich, wenn eine Organisation ihre Prozesse nicht vollständig der anderen Organisation zur Verfügung stellen möchte. In diesem Fall ist lediglich das extern sichtbare Verhalten zu spezifizieren und dann der anderen Organisation zugänglich zu machen. Durch die redundante Datenhaltung müssen Veränderungen jeweils händisch nachgezogen werden, wodurch ein zusätzlicher Wartungsaufwand erzeugt wird.

2. *Inkompatible Werkzeugversionen / Rolloutpläne.* Selbst wenn sich mehrere Organisationen auf das gleiche Werkzeug geeinigt haben, kommt es oft zu inkompatiblen Werkzeugversionen. Da jede Organisation ihre eigene IT-Abteilung hat, werden Entscheidungen zum Erwerb des Werkzeugs oder zum Upgrade auf neuere Versionen

unabhängig voneinander getroffen. Außerdem erstellt jede IT-Abteilung ihre eigenen Softwarerolloutpläne, die bestimmen, wann und für wen neue Versionen installiert werden sollen und wann Softwareupdates eingespielt werden.

Durch den Einsatz unterschiedlicher Werkzeugversionen ist die Anzeige von Modellen anderer Organisationen teilweise nicht möglich oder bearbeitete Modelle können von den anderen nicht weiter verwendet werden.

Eine offensichtliche Lösung für dieses Problem ist ein zentrales System, auf das alle Organisationen zugreifen können. Dies setzt allerdings ein gewisses Vertrauen dem Betreiber dieser Installation gegenüber voraus. Das Problem von Benutzerkonten kann durch moderne Techniken föderierter Authentifizierung gelöst werden: Jeder Benutzer verwendet das Benutzerkonto seiner Organisation, die z.B. als OpenID [ope07] nach außen verwendbar gemacht wird.

Das Problem inkompatibler Werkzeugversionen und Rolloutpläne kann auch durch ein webbasiertes System gelöst werden. Hier werden die Modelle jeweils mit der passenden Version des Editors ausgeliefert. Ein Softwarerollout geschieht lediglich bezüglich der zentralen Installation und in den einzelnen Organisationen muss lediglich ein Webbrowser (und ein Authentifizierungsmechanismus) bereit gestellt werden.

4 Design-Entscheidungen und Architektur von Oryx

Dieser Abschnitt stellt Oryx vor, ein Prozessmodellierungswerkzeug, das im Webbrowser ausgeführt wird. Zentrale Komponenten sind dabei das Modellrepository und die Modellierungskomponente. Abbildung 1 zeigt einen Screenshot von der Modellrepositoryansicht von Oryx. Man sieht dort alle Modelle, auf die der momentan eingeloggte Benutzer Lese- oder Schreibrechte besitzt. Über eine verkleinerte Ansicht sieht der Modellierer bereits, wie das Modell grob aussieht. In dieser Ansicht können auch die Zugriffsrechte auf Modelle eingestellt werden. So lässt sich definieren, ob ein Modell nur vom Modellierer selbst eingesehen werden darf, ob eine Gruppe oder bestimmte andere Nutzer darauf zugreifen dürfen, oder ob das Modell sogar webweit einsehbar sein soll.

Darüber hinaus können noch weitere Informationen für Modelle hinterlegt werden, z.B. können sogenannte *Tags* vergeben werden, über die Modelle dann später leichter gefunden werden können. Es stehen eine Reihe von Such- und Filterfunktionen zur Verfügung, um Prozessmodelle in einem großen Repository schneller lokalisieren zu können.

Für jedes Modell existieren eine Reihe von URLs, über welche man zu den verschiedenen Repräsentationen des Modells gelangt. So stehen mit PDF und PNG rein lesbare Formate zur Verfügung, mit SVG ein Format, das zumindest die in einem Modell hinterlegten Verlinkungen navigierbar macht und schließlich das Format, das zu einem Laden des Editors im Browser führt.

Abbildung 2 zeigt die Modellierungskomponente von Oryx. Auf der linken Seite sind alle Modellierungskonstrukte aufgeführt, die in der gewählten Sprache zur Verfügung stehen. Oryx unterstützt hierbei die Business Process Modeling Notation (BPMN [bpm08]) und

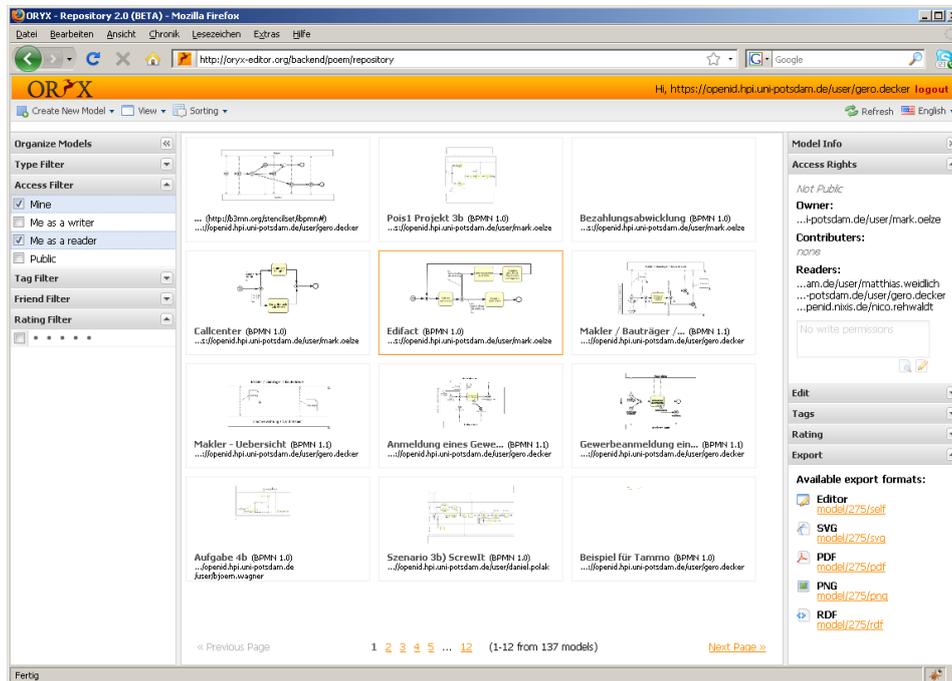


Abbildung 1: Oryx-Modellrepository

Ereignisgesteuerte Prozessketten (EPK, [KNS92]) als Prozessmodellierungssprachen. Auf der rechten Seite können Attributwerte für Modellelemente editiert werden. Die Liste der Attribute bestimmt sich hierbei wiederum aus der gewählten Modellierungssprache.

Im Zentrum befindet sich die Zeichenfläche, auf der Modellelemente angeordnet und editiert werden können. Die hinterlegte Sprachdefinition definiert dabei die Verknüpfungsregeln, die bestimmen, welche Modellelemente mit welchen anderen Elementen verbunden werden dürfen. Im sogenannten *Shape Menu* werden jeweils die Modellelemente angezeigt, welche als Folgeelemente für ein selektiertes Element in Frage kommen. So dürfen z.B. End Events in BPMN keinen ausgehenden Sequenzfluss besitzen.

Die Buttons in der Titelleiste zeigen Funktionen, wie z.B. Speichern von Modellen, Kopieren/Einfügen und Undo/Redo. Auch sprachspezifische Funktionen finden sich hier, wie z.B. schrittweise Simulation von Prozessmodellen.

4.1 Architektur und genutzte Technologien

Die Architektur von Oryx wird in Abbildung 3 illustriert. Dabei besteht Oryx aus einem in den Browser geladenen Client, in dem das Modell editiert wird, und einem Backend, das eine Menge von Prozessmodellen im Repository vorhält.

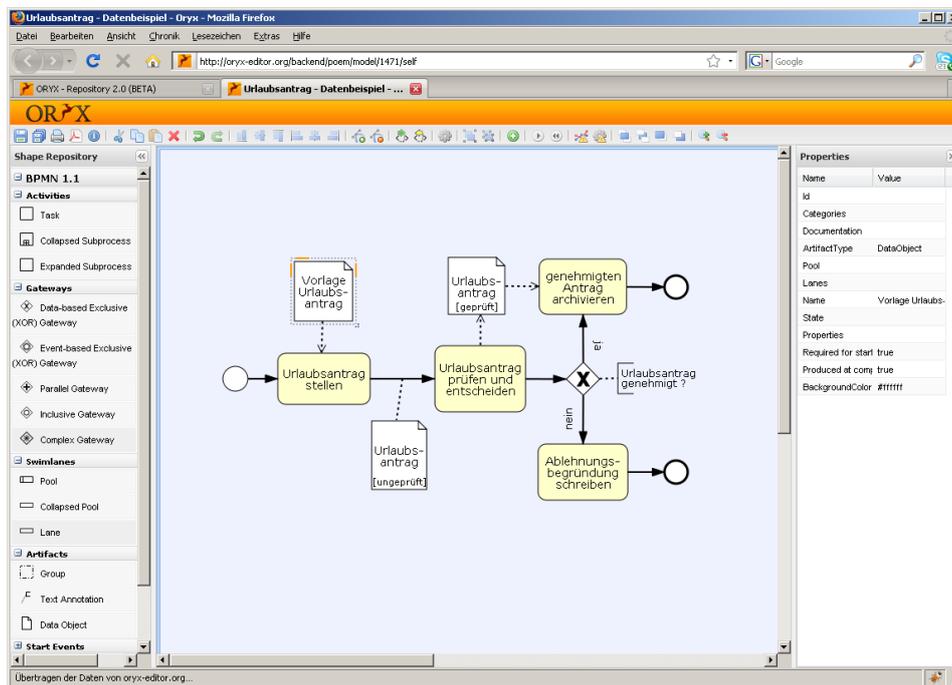


Abbildung 2: Oryx-Modellierungskomponente

Zur Anzeige des Editor-Clients werden vor allem JavaScript-Routinen verwendet, die als Teil eines Dokuments in den Browser geladen werden. Dieses über eine URL adressierbare Dokument enthält ein Prozessmodell, das im Embedded RDF (eRDF, [Dav06]) Format vorliegt. eRDF ermöglicht es, Metadaten direkt in ein HTML-Dokument zu schreiben und daraus RDF zu extrahieren. Die Benutzung von RDF macht Oryx-Modelle einfach portierbar.

Die mit dem Prozessmodell mitgelieferten JavaScript-Routinen sind dafür verantwortlich, die grafische Benutzerschnittstelle zu erzeugen und Modellierungsfunktionalität bereitzustellen. Modellelemente des aktuellen Diagramms finden sich als JavaScript-Objekte im Browser wieder und können so manipuliert werden. Beim Speichern eines Modells werden die Elemente im eRDF-Dokument asynchron an das Backend geschickt und dort in einer Datenbank gesichert.

Die Benutzerschnittstelle, die neben der Zeichenfläche aus *Toolbar*, *Shape Repository* und *Property Pane* besteht, ist mit Hilfe des ExtJS-Frameworks² erstellt. ExtJS ist eine JavaScript-UI-Bibliothek, die Widget-Typen für Rich Internet Applications bereitstellt. Die Zeichenfläche enthält Modellelemente, die als Scalable Vector Graphics (SVG) in den Browser geladen werden.

Jedes Prozessmodell in Oryx ist durch eine URL eindeutig adressiert. Dadurch können

²Siehe <http://extjs.com>

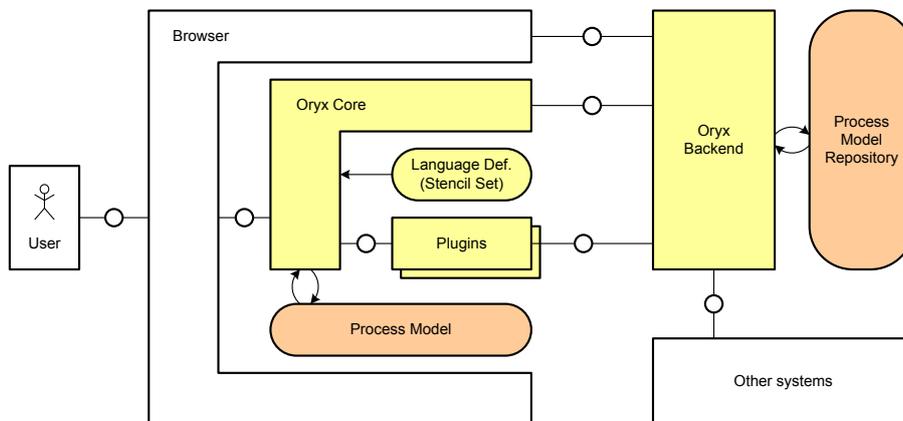


Abbildung 3: Oryx-Systemarchitektur

Modelle durch Weiterreichen der URL ausgetauscht werden. Darüber hinaus können Modelle unterschiedliche Repräsentationen aufweisen. So kann z.B. neben dem im Oryx editierbaren Dokument auch eine PDF-Repräsentation eines Prozesses angefordert werden.

Durch die OpenID-basierte Authentifizierung können Änderungen an Prozessmodellen – ähnlich wie in Wikis – nachverfolgt werden. Auf kritischen Ressourcen können zudem die Rechte, die ein Nutzer hat, eingeschränkt werden. Das System kennt hierbei die drei Rollen *Owner*, *Contributor* und *Viewer*, wobei ein Contributor ein Modell verändern und ein Viewer das Modell lediglich ansehen darf.

4.2 Stencil-Set-Konzept

Die Erweiterbarkeit ist eine der wichtigsten Eigenschaften von Oryx. Um verschiedene Notationen in Modellierungsprojekten zu unterstützen, wurde Oryx mit einem Mechanismus zum Auflösen beliebiger aus Knoten- und Kantentypen bestehender Modellierungssprachen versehen. Dabei wird eine solche Notation im Oryx durch ein so genanntes *Stencil Set* repräsentiert. Stencil Sets werden beim Öffnen eines Prozessmodells in den Editor geladen und erzeugen dort Einträge im Shape Repository und in der Property Pane.

Kernstück eines Stencil Sets ist eine in der JavaScript Object Notation (JSON) verfasste Datei, die das Metamodell der Notation enthält. In ihr werden die Typen aller Modellelemente definiert. Jeder Elementtyp ist entweder Knoten oder Kante und verfügt über eine ID, einen Namen und eine Beschreibung. Zur grafischen Repräsentation existiert für jeden Typ eine SVG-Datei, die als Vorlage für die modellierten Elemente genutzt wird.

Darüber hinaus verfügt ein Typ über eine Menge von Eigenschaften, die bei der Modellierung gesetzt werden können. Beispiel hierfür ist der Name einer BPMN-Aktivität oder die URL einer Subprozess-Verfeinerung. Jede Property hat einen Datentypen und einen Standardwert. Properties können an XML-Knoten innerhalb der SVG-Datei des gleichen

Elementtyps gebunden werden. Dadurch erhalten die Propertywerte während der Modellierung Einfluss auf die grafische Erscheinung des Elements. Durch diesen Mechanismus wird zum Beispiel der Name einer BPMN-Aktivität innerhalb des sie repräsentierenden abgerundeten Rechtecks angezeigt.

Um die Beziehungen, die zwischen Modellelementen bestehen dürfen, auszudrücken, können im Stencil Set Kompositionsregeln angegeben werden. Dabei gibt es Regeln, die festlegen, ob Elemente zweier Typen durch Knoten oder Kanten eines bestimmten Typs verbunden werden dürfen. Außerdem kann geregelt werden, ob Elemente eines Typs in Elementen eines anderen Typs enthalten sein dürfen. Auf Grundlage dieser Bedingungen wird beim Modellieren geprüft, ob neue Beziehungen korrekt sind. Darüber hinaus werden aus den Verbindungsregeln Vorschläge für nachfolgende Elemente erzeugt, die dann durch Benutzung eines Kontextmenüs einfach zu erstellen sind.

Die Erweiterung von Oryx um eine zusätzliche Notation ist sehr einfach: Es muss nur ein neuer Unterordner im Stencil-Set-Verzeichnis angelegt werden. In diesem Ordner werden die JSON sowie die Icon- und SVG-Grafikdateien der neuen Notation gespeichert. Anschließend kann sofort damit begonnen werden, Modelle der neuen Notation zu erstellen.

Oryx eignet sich dabei nicht nur für die Modellierung von Prozessmodellen. Auch wurde über diesen Mechanismus bereits ein grafischer Editor für XForms-Formulare realisiert.³

4.3 Plugin-Konzept

Neben der Erweiterbarkeit durch Stencil Sets kann Oryx auch funktional durch Plugins erweitert werden. Ein Plugin wird beim Laden von Oryx initialisiert und registriert dabei seine Funktionalität im Nutzerinterface. Durch Nutzerinteraktion kann die Funktionalität des Plugins angestoßen werden. Dabei hat es Zugriff auf die JavaScript-Objekte des aktuellen Prozessmodells sowie auf alle anderen geladenen Ressourcen.

Der Großteil der Modellierungsfunktionalität von Oryx ist durch Plugins implementiert: Während der Oryx Core aus Abbildung 3 für den Aufbau des Editor-Layouts verantwortlich ist, werden Funktionen wie das Ausrichten von Modellelementen oder auch das Speichern der Modelle durch austauschbare Plugins realisiert. Zudem sind auch das Shape Repository, die Property Pane und selbst die Toolbar, in der die meisten anderen Plugins vertreten sind, als Plugin implementiert.

Plugins werden in JavaScript programmiert. Außerdem muss ein Plugin in der *plugins.xml*-Datei eingetragen werden. Dabei kann auch spezifiziert werden, dass ein Plugin nur für Modelle eines bestimmten Stencil Sets geladen werden soll.

In den vergangenen Monaten wurden schon zahlreiche Plugins entwickelt: So existieren Im- und Exporter für einfache Petrinetze und ein Transformator, der BPMN in ausführbare Petrinetze übersetzt [DLO⁺08]. Ein weiterer Transformator übersetzt BPMN nach BPEL (Business Process Execution Language) gemäß [ODBtH06]. Auch im Bereich der EPKs werden Plugins genutzt: Ein Syntax-Check überprüft EPK-Modelle, welche wiederum in

³Siehe <http://xforms-editor.org>

EPML exportiert und aus EPML importiert werden können.

5 Ausblick

Dieser Artikel hat zwei Anwendungsfälle aus dem Prozessmodellierungsumfeld beleuchtet, die erst durch webbasierte Prozessmodellierungswerkzeuge angemessen unterstützt werden können. Als Implementierung eines solchen Werkzeugs wurde Oryx vorgestellt, ein erweiterbarer webbasierter Prozessmodelleditor.

Über die bereits genannten Anwendungsfälle hinaus eignet sich Oryx auch sehr gut für einen Einsatz in der universitären Lehre. Ohne Installationsaufwand für die Lehrpersonen können große Gruppen von Studenten die laufende Oryx-Installation unter <http://oryx-editor.org> nutzen, um ihnen gestellte Modellierungsaufgaben mit BPMN, EPK oder Petrinetzen zu lösen. Am Ende der Bearbeitungszeit können dann die Ergebnisse entweder als PDF oder PNG exportiert werden oder den Lehrpersonen kann Zugriff auf die erstellten Prozessmodelle gewährt werden, sodass diese dann die Qualität der Modelle überprüfen können. Diese Lehrform wurde bereits an mehreren Universitäten mit mehreren hundert Studierenden durchgeführt. Ein positiver Nebeneffekt einer solchen zentralen Installation ist, dass die gespeicherten Modelle – deren Freigabe durch den Modellierer vorausgesetzt – wiederum zu Forschungszwecken genutzt werden können.

Während schon heute eine Reihe neuer Anwendungsfälle mit Oryx unterstützt werden, sind weitere Integrationen mit anderen webbasierten Systemen möglich. Durch ein Mashup-Konzept soll es in Zukunft ermöglicht werden, dass die laufenden Installationen von Oryx mit anderen Anwendungen einfach kombiniert werden können. So könnten die grafischen Modelle, die mit Oryx erstellt werden, ähnlich wie bei Kartendiensten (wie z.B. Google Maps, <http://maps.google.com>) mit Daten aus anderen Anwendungen verbunden werden. Auf diese Weise ist beispielsweise eine einfache Verknüpfung von Prozessmodellen mit Wiki-basierten Prozessdokumentationen möglich.

Danksagungen. Oryx ist eine Initiative, zu der zahlreiche Personen beigetragen haben. Wir danken dem gesamten Oryx-Team für die umfangreichen Entwicklungen.

Literatur

- [BL96] Tim Berners-Lee. WWW: Past, Present, and Future. *IEEE Computer*, 29(10):69–77, Oct. 1996.
- [bpm08] Business Process Modeling Notation, V1.1. Bericht, Object Management Group (OMG), Jan 2008. <http://www.omg.org/spec/BPMN/1.1/PDF/>.
- [Dav06] Ian Davis. RDF in HTML. Bericht, Talis Information Limited, 2006. <http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml>.

- [DLO⁺08] Gero Decker, Alexander Lueders, Hagen Overdick, Kai Schlichting und Mathias Weske. RESTful Petri Net Execution. In *Proceedings of the 5th Workshop on Web Services and Formal Methods (WS-FM)*, LNCS, Milan, Italy, Sep 2008. Springer Verlag.
- [KNS92] G. Keller, M. Nüttgens und A.-W. Scheer. Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)". Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany, 1992.
- [Min05] Stefan Mintert. AJAX: die nächste Generation der Web-Anwendungen. Zwei Helden. *iX*, November 2005.
- [ODBtH06] Chun Ouyang, Marlon Dumas, Stephan Breutel und Arthur H.M. ter Hofstede. Translating Standard Process Models to BPEL. In *Proceedings 18th International Conference on Advanced Information Systems Engineering (CAiSE)*, Luxembourg, June 2006.
- [ope07] OpenID Authentication 2.0 - Final. Bericht, OpenID Foundation, December 2007. http://openid.net/specs/openid-authentication-2_0.html.
- [VH07] Gottfried Vossen und Stephan Hagemann. *Unleashing Web 2.0: From Concepts to Creativity*. Morgan Kaufmann, 2007.
- [Wes07] Mathias Weske. *Business Process Management*. Springer, 2007.