

# Formularbasierte Benutzerinteraktion mit Fachkomponenten

Martin Gaedke, Martin Nussbaumer

Universität Karlsruhe (TH), Institut für Telematik, IT-Management und Web Engineering, Postfach 6980  
76128 Karlsruhe, Deutschland, Tel: +49 (721) 608 – 8076, Email: {gaedke | nussbaumer}@tm.uni-  
karlsruhe.de, URL: <http://mw.tm.uni-karlsruhe.de>

**Zusammenfassung:** Das World Wide Web hat sich zu der Plattform für verteilte Anwendungen gewandelt. Eine große Menge dieser Web Anwendungen dient der Realisierung betrieblicher Zielsetzungen. Für die Umsetzung der betrieblichen Abläufe liegt allen Web-Anwendungen ein einfaches, formularbasiertes Implementierungsmodell zugrunde, das es Benutzern ermöglicht mit den Fachkomponenten einer modernen Web Anwendung zu interagieren. Betriebliche Web Anwendungen sind daher durch komplexe Formularstrukturen geprägt. Durch den Fortschritt im Bereich der Darstellungsmedien sowie aus Gründen der Aktualität müssen Web Anwendungen jedoch häufig geändert und erweitert werden. Während Fachkomponenten hiervon zumeist nur selten betroffen sind, erschwert das zugrundeliegende Implementierungsmodell des Web die Wiederverwendung und Evolution der Formulare erheblich, da auf ihre strukturelle Information nach ihrer Abbildung in das grobgranulare Implementierungsmodell nicht mehr zugegriffen werden kann. Ohne einen disziplinierten Ansatz können die Anforderungen an betriebliche Web Anwendungen hinsichtlich der Interaktion mit Fachkomponenten somit nicht erfüllt werden. In diesem Beitrag wird der objektorientierte PetrIX Ansatz als durchgängige Unterstützung für die Modellierung und Realisierung von formularbasierten Web Anwendungen vorgestellt.

## 1 Einleitung

Mit der Standardisierung von HTML 2.0 [1995] für Dokumente im World Wide Web (Web) wurde die Auszeichnungssprache HTML 1995 um Formular-Elemente ergänzt. Die Auszeichnungssprache HTML beschrieb somit nicht nur die Darstellung von Dokumenten und ihrer Verknüpfung mit lokationsunabhängigen Ressourcen im Web [1990b], sondern ermöglichte erstmals die Beschreibung von Interaktion zwischen Benutzer und Web Anwendung mittels Formularen. Mit dem Fortschritt des Formulars als integraler Bestandteil der Dokumentbeschreibungssprache, wurde einem wachsenden Bedarf an betrieblichen Web Anwendungen Rechnung getragen [1999e].

Diesen Web Anwendungen ist gemeinsam, dass Interaktionstechnologien zum Einsatz kommen, die über bekannte Textein-/ausgabe-Verfahren und Mensch-Maschine-Schnittstellen herkömmlicher Software hinausgehen und nicht durch einfache softwaretechnische Maßnahmen am Anwendungscode handhabbar sind. Unter dem Einfluss des zunehmenden Wettbewerbs unterliegen diese Anwendungen einem ständigen Wandel; sei es bezüglich Funktionalität, Anwendungsschnittstellen oder der verfügbaren Informationen [1999a]. Infolgedessen werden die Änderungszyklen von Web Anwendungen immer kürzer. Die Anwendungen befinden sich ständig in einem Prozess der Wartung und Weiterentwicklung, welcher als *Evolution* bezeichnet wird [2000b].

Die kurzen Änderungszyklen an diesen komplexen Strukturen werden insbesondere durch die grobgranularen Beschreibungsmechanismen der Interaktionseinheiten und den damit verknüpften Programmkonstrukten erheblich beeinträchtigt. *Interaktionseinheiten* bezeichnen eine abgeschlossene Einheit, innerhalb der Interaktion zwischen Benutzer und Web Anwendung

ermöglicht wird. Solche Einheiten werden in HTML beispielsweise durch Formulare realisiert, die grobgranulare Beschreibungseinheiten darstellen, da sie nur als Ganzes wiederverwendet werden können. Eine *Interaktionsstruktur* bildet einen Kontext innerhalb einer solchen Interaktionseinheit, in der interaktive Vorgänge zueinander in Beziehung gesetzt werden. So können komplexe Interaktionseinheiten in einzelne, für den Benutzer überschaubare Gruppen von *Interaktionselementen*, wie z.B. Eingabefelder, Auswahllisten, etc., partitioniert werden. So können beispielsweise Gruppen von *Interaktionselementen*, wie z.B. Eingabefelder, Auswahllisten, etc., der Reihe nach navigiert werden, um komplexere Interaktionseinheiten in einzelne, für den Benutzer überschaubare zu partitionieren.

Eine feingranulare Betrachtung von Interaktionseinheiten mit Interaktionsstrukturen ist aber im WWW nicht möglich, da die aktuellen Auszeichnungssprachen HTML und XHTML [2000c] auf der Basis von Formularen eine solche Trennung nicht unterstützen. Durch die steigende Zahl der darstellenden Medien, wie z.B. Web-Browser, Personal Digital Assistant oder Mobiltelefon, muss jede einzelne Interaktionseinheit einer Web Anwendung zumeist in mehreren Ausprägungen realisiert werden, da sich diese Medien in ihren technologischen Möglichkeiten erheblich unterscheiden [1998b]. Eine Änderung am Interaktionsentwurf oder an der Fachkomponente, welche die Formulareingaben verarbeitet, hat mehrere Änderungen an den Implementierungen zur Folge. Somit wird die Evolution von Interaktionseinheiten insgesamt zu einer aufwändigen und fehleranfälligen Aufgabe, da eine entworfene Interaktionseinheit mehrere unterschiedliche, plattformabhängige Realisierungen ihrer Interaktionsstrukturen impliziert.

Trotzdem setzt sich nur langsam die Erkenntnis durch, dass eine disziplinierte Anwendung von Softwaretechnik für Entwicklung und Evolution von Benutzerinteraktion in Web Anwendungen erforderlich ist, um hierdurch die Qualität der Benutzerschnittstelle zu steigern und die Kosten der Evolution zu verringern. In den folgenden Abschnitten werden zunächst die Benutzerinteraktion mit Fachkomponenten im Web untersucht sowie Ansätze zur Modellierung und Realisierung von Benutzerinteraktion für Web Anwendungen analysiert. Anschließend wird das in der Forschungsgruppe IT-Management und Web Engineering entwickelte Vorgehensmodell PetrIX vorgestellt, das die Benutzerinteraktion mit Fachkomponenten durchgängig objektorientiert modelliert. Am Beispiel der Web Anwendung [webengineering.org](http://webengineering.org) Community wird der Ansatz exemplarisch für eine komplexe Fachkomponente verdeutlicht. Der Beitrag schließt mit einer Zusammenfassung und einem Ausblick auf aktuelle Arbeiten der Forschungsgruppe.

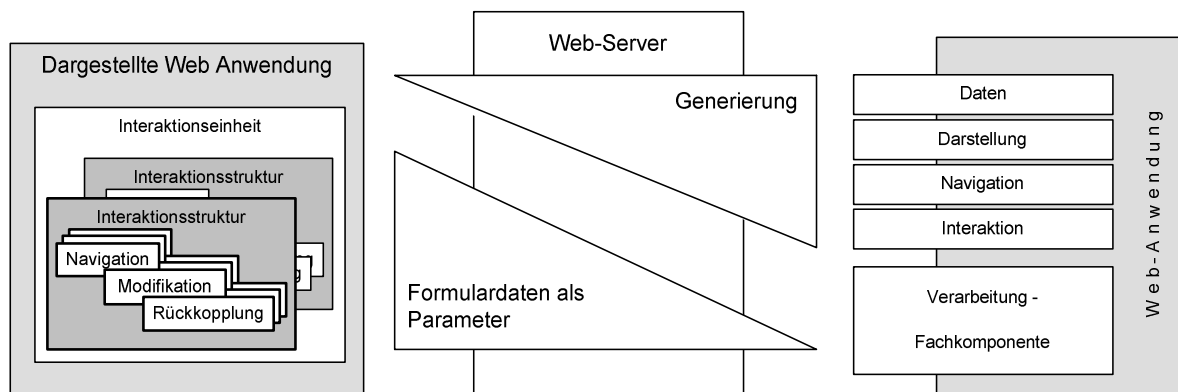
## **2 Interaktion mit Fachkomponenten in Web Anwendungen**

Der Begriff der Interaktivität ist in der Literatur weit gestreut. So gibt es viele unterschiedliche Auffassungen darüber, was Interaktion eigentlich bedeuten soll. Oftmals wird das bloße Vorhandensein von Navigationsstrukturen in einem Dokument als „interaktiv“ bezeichnet. In [1990a] werden für die Interaktion zwischen zwei und mehr Personen, *zustandsorientierte Aktionen* und *Kommunikation* gefordert. Während die Kommunikation zwischen Mensch und Web Anwendung durch das darstellende Medium implizit unterstützt wird, muss der Forderung nach zustandsorientierten Aktionen softwaretechnisch Rechnung getragen werden. Hierauf aufbauend wird in [2002b] folgende Definition abgeleitet:

*In einer formularbasierten Web Anwendung hat ein Benutzer die Möglichkeit durch eine Interaktionseinheit zu navigieren (Navigation), Daten zu modifizieren (Modifikation) und diese Änderungen innerhalb der Interaktionseinheit wieder sichtbar zu machen (Rückkopplung).*

Die in dieser Definition angeführten Ausprägungen von Interaktion, im folgenden *Interaktionsprimitive* genannt, sind Navigation, Modifikation und Rückkopplung. Diese Interaktionsprimitive, insbesondere Navigation, beziehen sich lediglich auf den formularbasierten Teil der Web Anwendung, also auf die Generierung der Parameter des durch die Fachkomponente

zur Verfügung gestellten Dienstes. Aufbauend auf dieser Definition bildet eine *Interaktionsstruktur* somit einen Kontext innerhalb dessen Interaktion auf Basis von Interaktionsprimitiven (Navigation, Modifikation und Rückkopplung) gebildet wird. Diese Interaktionsstrukturen werden von den Schnittstellen der Fachkomponente impliziert, welche die Daten und den Kontext zur Verfügung stellen. Interaktion wird hier nicht als die Interaktion zwischen Fachkomponenten verstanden. Vielmehr handelt es sich bei Interaktionsstrukturen um die abstrakte Definition einer Benutzerschnittstelle, welche die Parametrisierung eines von einer Fachkomponente zur Verfügung gestellten Dienstes vornimmt.



**Bild 1:** Zusammenhang von formularbasierter Benutzinteraktion und Fachkomponenten

Die Interaktion von Benutzern einer Web Anwendung mit verarbeitender Fachkomponente wird basierend auf dieser Definition in Bild 1 dargestellt. Das Interaktionsprimitiv *Rückkopplung* symbolisiert insbesondere die Möglichkeit, ein vom Benutzer *modifiziertes* Datum, sofort, bei allen Verwendungen innerhalb der Interaktionseinheit zu aktualisieren.

Die *dargestellte Web Anwendung* führt Daten sowie Darstellungs-, Navigations- und Interaktionscode in Web-Dokumenten zusammen und stellt die generierte Benutzerschnittstelle dem darstellenden Medium zur Verfügung. *Navigation*, *Modifikation* und *Rückkopplung* wird in der dargestellten Web Anwendung durchgeführt bzw. durch den ausführenden Browser ermöglicht.

Hierbei kann jede einzelne Modifikation innerhalb einer Modifikationssequenz bzw. Formularbearbeitung zu Rückkopplungen führen. Die Daten der finalen *Formularmodifikation* werden abschließend an die *Fachkomponente* der Web Anwendung zur Verarbeitung des implementierten Betriebsprozesses gesendet.

Die Komposition der fünf in Bild 1 dargestellten Komponenten werden als Service bezeichnet und stellen die Web-Fachkomponente einer betrieblichen Web Anwendung dar [1999b]. Ein Service beinhaltet einmal die Informationen, die eine Organisation mit der entsprechenden betrieblichen Dienstleistung einer definierten Anwendungsdomäne assoziiert. Ferner sind Darstellung und Navigation, mit denen diese Informationen zur Verfügung gestellt werden sollen, sowie Anweisungen darüber, welche Interaktionsmöglichkeiten die Benutzerschnittstelle anbieten soll, Bestandteil eines Services. Die Beschreibung der Umsetzung von (Geschäfts-)Prozessen durch eine Fachkomponente ist ein weiterer wichtiger Bestandteil. Er trennt die logische Verarbeitung von der konkreten Ausprägung der dargestellten Web Anwendung. Durch den Einsatz von Fachkomponenten zur Verarbeitung der Datenströme lassen sich Web Anwendungen somit in unterschiedlichen Ausprägungen darstellen ohne die eigentliche betriebliche Verarbeitungslogik ändern zu müssen.

Darstellung, Navigation und Interaktion eines Services sorgen dafür, dass die Dienstleistung als Ressource über das WWW zugänglich gemacht werden kann. Dabei braucht es sich nicht notwendigerweise um eine HTML-Ressource zu handeln. So lassen sich durch die Trennung von

Modell und Darstellung unterschiedliche Mensch-Maschine-Schnittstellen konkret Mensch-Fachkomponenten-Schnittstellen realisieren. Die Komposition solcher Services zu komplexen Web Anwendungen können durch den in [2000a] beschriebenen WebComposition-Ansatz diszipliniert entworfen und wiederverwendungsorientiert mit dem Evolutionsbus-Framework realisiert werden.

Für die abstrakte Betrachtung von Design und Implementierung von Benutzerinteraktion mit Fachkomponenten stellt die effektive Umsetzung auf eine Zielplattform somit die zentrale Problemstellung dar. Eine wichtige Anforderung, die an ein abstraktes Modell zur Beschreibung von solchen Interaktionseinheiten gestellt werden muss, ist daher die Möglichkeit der Abbildung auf ein vom Web unterstütztes Laufzeitsystem. Ferner muss eine abstrakte Beschreibung von Interaktionseinheiten folglich durch erweiterbare und partiell wiederverwendbare Interaktionsstrukturen unterstützt werden.

### 3 Ansätze zur Beschreibung von Interaktionsstrukturen

In der *Softwaretechnik* kommen zur Modellierung von Interaktionsstrukturen hauptsächlich zustandsorientierte Modelle zum Einsatz. Die Modellierung von Interaktion wird häufig als die Modellierung von interagierenden Objekten, wie im Interaktionsdiagramm der UML [1997a], aufgefasst. Benutzerspezifische Modelle wie die UML Use Cases berücksichtigen zwar den Benutzer, allerdings fokussieren sie eher auf die geforderte Funktionalität (Anforderungsanalyse), als dass sie interaktive Vorgänge als konkrete Abläufe beschreiben [1997b]. Daher eignen sich diese Modelle weniger, um die hier vorliegenden Interaktionsstrukturen zu spezifizieren, da der Benutzer hierbei explizit mit in Betracht gezogen werden muss. Für die Beschreibung von Interaktionsstrukturen im *Web Engineering* [1999c] muss somit einerseits der Benutzer explizit berücksichtigt werden, andererseits die Benutzerschnittstelle *abstrakt* bzw. darstellungsunabhängig modelliert werden können. Darüber hinaus muss Parallelität im Sinne von Benutzerinteraktion explizit in der Modellierung Berücksichtigung finden. Das bedeutet die Modellierung von gleichzeitig aktiven Benutzer-Optionen innerhalb eines Formulars muss auch als parallel zueinander erkennbar sein.

Weitergefasstere Modelle wie *Endliche Automaten* und Petri Netze bieten sich daher für die Modellierung der Interaktionsstrukturen eher an, da sie keine starre Auffassung über die zu modellierenden Elemente und deren Zusammenwirken haben. Vielmehr lassen sie sich aufgrund ihrer generischen Struktur besser auf die Bedürfnisse zur Modellierung von Benutzerinteraktionen anpassen.

Endliche Automaten sind prinzipiell gut geeignet, allerdings erfüllen sie nicht die Voraussetzung für das gewünschte Maß an Parallelität. Daher eignen sich Endliche Automaten nur begrenzt zur Modellierung von Benutzerinteraktionen.

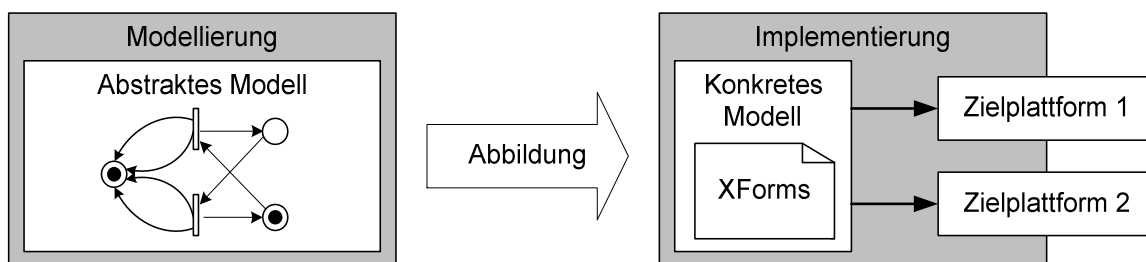
*Petri Netze* [1962] hingegen sind für die Modellierung von Parallelität ausgelegt. Sie sind übersichtlich und lassen sich einfach benutzen und bedienen. Im Gegensatz zu endlichen Automaten ermöglichen sie auch das grafische Simulieren von interaktiven Abläufen. Allerdings erfüllen sie nicht die Anforderungen, die an ein erweiterbares Modell gestellt werden müssen. So können Teile eines Petri Netzes nur schwer wiederverwendet werden, da das Zusammenfassen von Stellen zu Gruppen bei Petri Netzen standardmäßig nicht unterstützt wird, und so ein Netz als atomare Einheit aufgefasst werden muss. Jedoch ermöglichen Petri Netze aufgrund der Objekte Stelle und Transition im Gegensatz zu endlichen Automaten einen einfachen Mechanismus für Assoziationen. Eine Verbindung zwischen Stellen und Transitionen hat lediglich eine Richtung und ist ansonsten unabhängig von einem Typ oder Wert, so dass semantische Informationen auf die Objekte Stelle und Transition verlagert werden können. Dadurch können Erweiterungen jeweils an diesen Objekten vorgenommen werden, und nicht, wie bei endlichen Automaten, an den Übergängen zwischen ihnen.

Um die Interaktionsstrukturen auf eine geeignete *Zielplattform* abzubilden, bedarf es darüberhinaus noch einer unterstützenden Implementierungstechnologie wie beispielsweise HTML [1999d], XHTML [2000c] oder WML [2000d]. Die in HTML wie auch XHTML und WML enthaltenen Mechanismen, um Formulare zu definieren sind nicht geeignet, da sie keine Entkopplung von ihrer Darstellung im Browser und ihrer Semantik zulassen. Vielmehr werden die Formularelemente nahtlos in die grobgranulare Dokumentstruktur des World Wide Webs eingefügt und vermischen sich so mit Teilen des Inhalts und der Darstellung. Dedizierte Implementierungen von Formularen unter Einsatz der Technologie DHTML ermöglichen zwar eine teilweise Entkopplung zwischen den Interaktionselementen und deren Semantik, erhöhen allerdings den Aufwand, um Änderungen an den Formularen durchzuführen.

Eine feingranulare Betrachtung von Interaktionselementen ist mit Hilfe von XForms [2002a] möglich. Aktuell befindet sich XForms noch im Standardisierungsprozess des W3C, der die Lücke zwischen der eXtensible Markup Language (XML) [1998a] und den bisherigen HTML Formularen schließen soll. XForms bildet durch die Trennung zwischen Inhalt (Model) und Layout (Form Control) eine geeignete Ausgangsbasis für die Implementierung von Interaktionsstrukturen. Ferner verfügt XForms über die Möglichkeit dynamische Teile zu definieren, die über zustandsorientierte Benutzeraktionen aktiviert werden können. Als Anwendung der XML erbt XForms zugleich die Vorteile von XML, wie beispielsweise die Möglichkeit der einfachen Überprüfung auf Korrektheit und Validität. Damit lassen sich Gruppen von Interaktionselementen Web-konform zusammenfassen und es wird eine Strukturierung von Formularen, also den Interaktionseinheiten, ermöglicht. Durch die Unterstützung von XML als Datencontainer (*XForms instance*) lassen sich Web Anwendungen mit XForms bis auf die Ebene der Interaktion in XML spezifizieren. Dedizierte Programme, die nur dem Zweck der Abbildung oder des Auslesens von Formulardaten in ihrer XML-Repräsentation dienen, werden damit obsolet. Das XForms-Modell ist plattformunabhängig und kann für unterschiedliche Darstellungsmedien wie PDA oder Mobiltelefon wiederverwendet werden.

#### 4 PetrIX – Design und Abbildung von Interaktionsstrukturen

Um eine Trennung von abstraktem Design und konkreter Implementierung zu erhalten, muss eine Trennung zwischen einem *abstrakten Modell* und einem *konkreten Modell* durchgeführt werden. Durch eine solche Trennung werden die abstrakten Modelleigenschaften der Interaktionsstrukturen bezüglich ihrer Semantik hervorgehoben und explizit von einer konkreten Implementierung getrennt. Dadurch ergibt sich unter anderem auch der Vorteil der plattformunabhängigen Betrachtung von Interaktionsstrukturen.



**Bild 2:** Zusammenhang zwischen abstraktem und konkretem Modell

Das PetrIX Vorgehensmodell unterstützt durchgängig die Benutzerinteraktion mit Fachkomponenten von der Modellierung bis hin zur Abbildung auf eine Zielplattform. Es stellt hierfür ein konkretes und abstraktes Modell zur Verfügung. Die abstrakte Modellierung von Interaktionsstrukturen basiert hierbei auf erweiterten Petri Netzen. Für das konkrete Modell wird

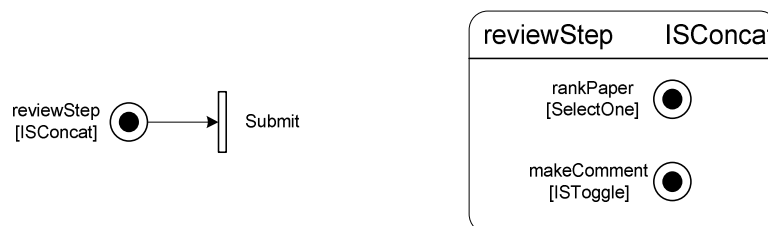
XForms verwendet. Bild 2 stellt die Abbildung des abstrakten auf das konkrete Modell dar. Der Abbildungsmechanismus nutzt die Kontextinformationen der Interaktionsstruktur, um die Generierung für die entsprechende Zielplattform zu ermöglichen.

Ein wesentlicher Bestandteil dieser konkreten Implementierung ist dabei die Umsetzung der Interaktionssemantik auf darstellende Formularelemente und die Bindung von Formularelementen an die Attribute des zu manipulierenden Datenobjekts einer Fachkomponente.

## 5 Objektorientierte Modellierung und Termersetzung mit PetrIX

Im ersten Schritt von PetrIX werden die Interaktionsstrukturen mit Hilfe von Petri Netzen dargestellt. Aufgrund der Komplexität dieser Strukturen werden die Petri Netze in einen objektorientierten Kontext gebettet, der es ermöglicht Strukturen verschachtelt darzustellen. Mit Hilfe dieser Objektstruktur lassen sich die Petri Netze serialisieren. Das bedeutet, dass die instantiierten Objekte als lineare Sequenz aufgefasst werden, die in einen korrespondierenden Term einer Termalgebra übersetzt werden. Ein wesentlicher Vorteil einer Termalgebra ist neben der Möglichkeit der algorithmischen Optimierung auch der Einsatz eines Termersetzungssystems. Mit Hilfe eines Termersetzungssystems lassen sich zuvor generierte Terme, die serialisierten Petri Netze repräsentieren, leicht in eine konkrete Implementierungssprache abbilden.

Bild 3 zeigt zwei Petri Netze, die Interaktionsstrukturen darstellen. Die üblichen Konventionen eines Petri Netzes werden beibehalten: eine Stelle repräsentiert einen Zustand, eine Transition einen Zustandsübergang. In PetrIX werden Zustände als Werte und Zustandsübergänge als Übergänge zwischen diesen Werten behandelt. Eine Stelle wird durch einen eindeutigen Namen und Typ dargestellt. Ist die Stelle vom Typ *InteractionStructure*, so kann sie wieder weitere Stellen aufnehmen. In Bild 3 sind zwei Interaktionsstrukturen dargestellt. Links dargestellt ist die Stelle `reviewStep` vom Typ *ISConcat*, einer Interaktionsstruktur, verbunden mit einer Transition `Submit`, welche die Beendigung einer Interaktionseinheit beschreibt. Rechts davon dargestellt ist die Verfeinerung von `reviewStep` mit den darin enthaltenen Stellen `rankPaper` (Typ *InteractionElement*) und `makeComment` (Typ *ISToggle*). Da es sich bei `rankPaper` um ein Interaktionselement handelt, kann diese Stelle nicht weiter verfeinert werden. Die Interaktionsstruktur *ISToggle* wird an späterer Stelle vertieft.

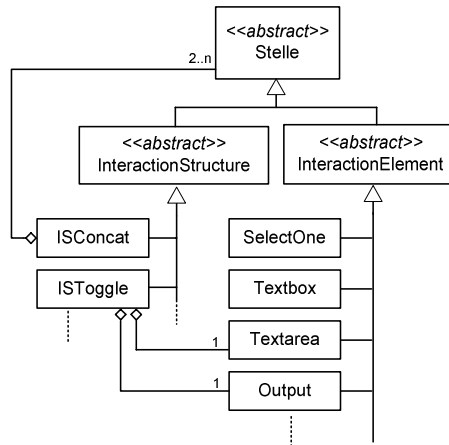


**Bild 3:** Abstrakte Darstellung einer Interaktionsstruktur als Petri Netz

Aufgrund des objektorientierten Kontextes, in den die Petri Netze gebettet werden, können sie besser an die sich ständig ändernden Bedingungen angepasst werden. Dadurch können änderungsrelevante Teile leichter identifiziert und modifiziert werden. Ferner wird in PetrIX durch die objektorientierte Modellierung von Petri Netzen eine „Netz-im-Netz“ Struktur, also die Schachtelung von Interaktionsstrukturen, ermöglicht. Die so definierten hierarchischen Petri Netze gewährleisten den Entwurf von Interaktionsstrukturen mittels schrittweiser Verfeinerung. Der Grad an Wiederverwendung kann bis auf die Granularität einer Stelle reichen. Zudem können Stellen gruppiert werden und solche Gruppen dynamisch aktiviert werden.

Bild 4 zeigt exemplarisch wie das Objekt Stelle als Basisklasse für den objektorientierten Entwurf benutzt werden muss. Auf diese Weise lassen sich beliebige Interaktionsstrukturen definieren und hinzufügen.

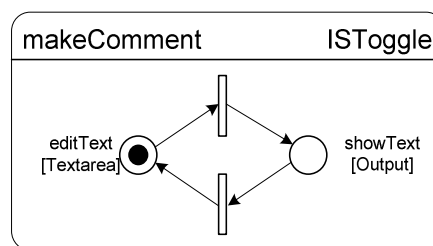
Für die von *InteractionStructure* und *InteractionElement* abgeleiteten konkreten Klassen existieren jeweils korrespondierende Transitionsklassen. Im Zusammenspiel zwischen diesen Objekten können die Interaktionsprimitive in den von der Interaktionsstruktur vorgegebenen Kontext gesetzt werden.



**Bild 4:** Das Objektmodell Stelle eines Petri Netzes

Zusätzlich müssen neben diesen strukturellen Beziehungen zwischen den Objekten auch die semantischen Beziehungen betrachtet werden. Die semantischen Informationen können beispielsweise zur Fehlerüberprüfung genutzt werden. Aufgrund der „Netz-im-Netz“-Struktur muss zudem noch die Erreichbarkeit von verschachtelten Stellen berechnet werden.

Bild 5 zeigt die Verfeinerung der Stelle *makeComment* aus dem Petri Netz in Bild 3. Die Interaktionsstruktur *ISToggle* ermöglicht das Hin- und Herschalten zwischen den Zuständen Editieren eines Kommentars (Interaktionselement *Textarea*) und Ansicht des Kommentars (Interaktionselement *Output*).



**Bild 5:** Verfeinerung der Stelle *makeComment*

Tabelle 1 verdeutlicht das Vorgehen, wie das in Bild 5 dargestellte Petri Netz (abstraktes Modell), das dem Objektmodell in Bild 4 genügt, in ein XForms-Programm (konkretes Modell) überführt werden kann. Aufgrund des objektorientierten Kontextes, in den die Petri Netze eingebettet sind, lassen sich die Petri Netze als Objekthierarchien auffassen. Mit Hilfe einer kontextfreien Grammatik läßt sich die Struktur dieser Objekthierarchien beschreiben. Dabei werden die objektorientierten Strukturen sukzessive auf die Regeln einer kontextfreien Grammatik abgebildet (Tabelle 1, Schritt 1 und Schritt 2). Durch Erweitern der Regeln dieser Grammatik um die Operationen einer Termalgebra können sie als formale Beschreibung eines Übersetzers zwischen

Petri Netzen und einer konkreten Zielplattform verwendet werden. In Tabelle 1 wird das Hinzufügen der termalgebraischen Operation (toggle) als Erweiterung für die Regel der kontextfreien Grammatik *ISToggle* verdeutlicht (vgl. Tabelle 1, Schritt 3).

Für die Abbildung auf die konkrete Plattform muss abschließend noch ein regelbasiertes Termersetzungssystem für diese Plattform spezifiziert werden. Interaktionsstrukturen können auf diese Weise leicht an das Implementierungsmodell einer Fachkomponente angepasst werden. In Tabelle 1 (Schritt 4) wird das Implementierungsmodell von XForms benutzt, um den produzierten Term zu ersetzen.

<b>1</b>	Objektorientiertes Muster	<pre> classDiagram     class ISToggle     class Textbox     class Output     ISToggle "2..n" o-- "2..n" Textbox     ISToggle "2..n" o-- "2..n" Output     </pre>
<b>2</b>	Regel der kontextfreien Grammatik	ISToggle := Textbox Output
<b>3</b>	um termalgebraische Operation erweiterte Regel	ISToggle := <u>toggle</u> (Textbox, Output)
<b>4</b>	Termersetzungsregel	<pre> toggle(t1,t2) → &lt;switch&gt;     &lt;case id="t1"&gt; eval(t1)&lt;/case&gt;     &lt;case id="t2"&gt; eval(t2)&lt;/case&gt; &lt;/switch&gt;     </pre>

**Tabelle 1:** Abbilden der Objektstruktur in eine Regel einer kontextfreien Grammatik

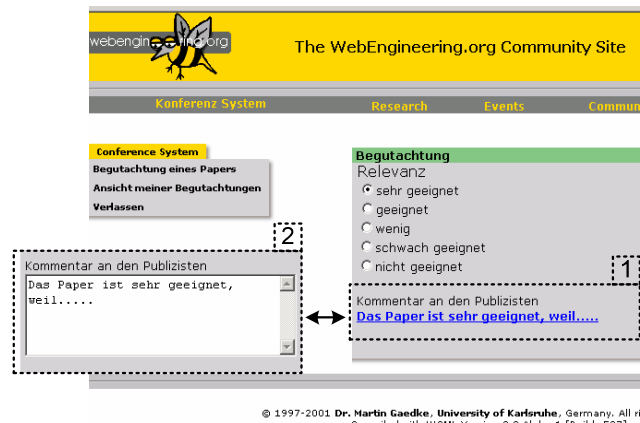
## 6 Beispiel – WebEngineering.org

Im Folgenden wird der Einsatz von PetrIX an dem realen Beispiel der Web Anwendung WebEngineering.org Community aufgezeigt. WebEngineering.org bietet Dienste zur Unterstützung der Web Engineering Community. Diese Dienste sind als Fachkomponenten realisiert und werden als Services über die Web Anwendung zugänglich gemacht. Eine solche komplexe und interaktionsreiche Fachkomponente ist ein *Peer-Review-System*, das zur Unterstützung der Begutachtung und Verwaltung von Konferenzbeiträgen im Rahmen von WebEngineering.org entwickelt wurde. Die Fachkomponente kann in unterschiedlichen Szenarien eingesetzt werden, wobei im Wesentlichen der Begutachtungsprozess bzw. die Interaktionsmöglichkeiten für den Benutzer konfiguriert werden müssen.

Die in Bild 6 dargestellte Benutzerschnittstelle zeigt einen Ausschnitt eines solchen konfigurierten Konferenz Systems. Die abstrakten Strukturen der dargestellten Benutzerinteraktion sind das *Bestimmen der Relevanz eines Beitrages* sowie dessen *Kommentierung*. Die Petri Netze in Bild 3 und Bild 5 entsprechen diesen Strukturen. Während das *Bestimmen der Relevanz* ein Interaktionselement darstellt, lässt sich die Struktur der Beitragskommentierung noch weiter verfeinern. Wenn der Text für den Kommentar eingegeben worden ist, soll das dafür vorgesehene Interaktionselement durch den Text des Kommentars in der Ausprägung als Hyperlink ersetzt werden (vgl. Bild 6: Schritt 1 und 2). Durch Aktivieren dieses Links soll wieder das Interaktionselement angezeigt werden, usw. Auf diese Weise wird ein Hin- und Herschalten (*engl. to toggle*) zwischen den beiden Interaktionselementen realisiert, das sich in der grafischen Entsprechung als Petri Netz wiederfindet (Bild 5). Der Vorteil einer solchen Struktur ist, dass schon



bearbeitete Teile eines (umfangreichen) Formulars als solche sofort erkannt werden und die Aufmerksamkeit des Benutzers auf die noch zu bearbeitenden Teile des Formulars gelenkt wird.



**Bild 6:** Benutzerschnittstelle des Peer-Review-Systems in WebEngineering.org

Aus der grafischen Darstellung der Interaktionsstrukturen lässt sich der Interaktions-Term, wie in Tabelle 1 beschrieben, bilden:

$ISConcat(IE_{SelectOne}, ISToggle(IE_{textarea}, IE_{output}))$ .

Durch die regelbasierte Termersetzung kann dieser Term nun weiter in die Zielsprache, im vorliegenden Fall Xforms, übersetzt werden.

```
<xforms_program>
  <xform id="model">
    <instance>
      <ranking>1</ranking>
      <comment>please write a comment</comment>
    </instance>
  </xform>
  <selectOne xform="model" ref="ranking" selectUI="radioGroup">
    <item value="1">sehr geeignet</item>
    ...
  </selectOne>
  <switch xform="model">
    <case id="show">
      <output xform="model" ref="comment"/> ...
    </case>
    <case id="edit">
      <textarea xform="model" ref="comment"/> ...
    </case>
  </switch>
</xforms_program>
```

Das Beispiel zeigt den übersetzten Term als XForms-Programm. Aufgrund der Aktualität von XForms existieren augenblicklich nur wenige Implementierungen von XForms-Prozessoren. Im Rahmen der Arbeiten wurde ein prototypischer XForms-Prozessor entwickelt, der zur Validierung für den erzeugten XForms-Programmcode benutzt wurde.

## 7 Zusammenfassung und Ausblick

In diesem Beitrag wurde das objektorientierte Vorgehensmodell PetrIX zur Unterstützung von Entwurf und Realisierung von formularbasierter Benutzerinteraktion mit Fachkomponenten vorgestellt. Durch die Abstraktion mittels darstellungsunabhängiger Interaktionsstrukturen, können diese für unterschiedliche Formuldarstellungen wiederverwendet werden. Durch den objektorientierten Ansatz lassen sich auch Teile von Interaktionsstrukturen gezielt wiederverwenden und an die durch Evolution geprägten Web Anwendungen und Fachkomponenten anpassen. Dadurch steigt sowohl die Qualität der Benutzerschnittstelle als auch die Produktivität im Softwareprozess. Durch die Abbildung auf XForms des World Wide Web Consortiums kann der Entwurfsprozess durchgängig mit Einsatz von XML gestaltet werden.

Zur Zeit wird in der Forschungsgruppe IT-Management und Web Engineering der Universität Karlsruhe ein Unterstützungswerkzeug für die grafische Modellierung von abstrakten Interaktionsstrukturen mit Petri Netzen entwickelt. Auf diese Weise können Interaktionsstrukturen grafisch entworfen, automatisch validiert und übersetzt werden. Aufgrund der semantischen Beziehungen zwischen Stellen und Transitionen eines Petri Netzes in PetrIX können Fehler schon während des Entwurfs lokalisiert werden. Dadurch soll die Qualität des Entwurfprozesses weiter verbessert und die Produktivität erhöht werden.

## 8 PetrIX-Werkzeuge

Der PetrIX XForms-Compiler sowie die WebComposition Laufzeitumgebung und weitere Informationen zum Evolutionsbus stehen im Web zur Verfügung:

<http://webe.tm.uni-karlsruhe.de/>

## 9 Literatur

- [1990a] Bahrtdt, H. P.: Schlüsselbegriffe der Soziologie. Eine Einführung mit Lehrbeispielen. C.H. Beck, München 1990a.
- [1990b] Berners-Lee, T. Information Management: A Proposal. <http://www.w3.org/Proposal.html>, Abruf am 1998-10.10.1998.
- [1995] Berners-Lee, T.; Connolly, D. Hypertext Markup Language (HTML) 2.0. RFC Nr. 1866, 1995.
- [1997a] Booch, G.; Jacobson, I.; Rumbaugh, J. The Unified Modeling Language for Object-Oriented Development. <http://www.rational.com/uml>, Abruf.
- [1998a] Bray, T.; Paoli, J.; Sperberg-McQueen, C. M. Extensible Markup Language (XML) 1.0. Recommendation Nr. REC-xml-1998021, World Wide Web Consortium (W3C) 1998a.
- [1999a] Cusumano, M. A.; Yoffie, D. B.: Software Development on Internet Time. In: IEEE Computer 32 (1999a) 10, S. 60-69.
- [2002a] Dubinko, M.; Dietl, J.; Merrick, R.; Raggett, D.; Raman, T. V.; Welsh, L. B. XForms 1.0: W3C Working Draft 08 June 2001. Draft Nr. WD-xforms-20010608, World Wide Web Consortium (W3C) 2002a.
- [1997b] Fowler, M.; Scott, K.: UML distilled: applying the standard object modeling language. Addison Wesley Longman, Reading, Mass. 1997b.
- [2000a] Gaedke, M.: Komponententechnik für Entwicklung und Evolution von Anwendungen im World Wide Web. Shaker Verlag, Aachen 2000a.
- [1998b] Gaedke, M.; Beigl, M.; Gellersen, H.-W.; Segor, C.: Web Content Delivery to Heterogeneous Mobile Platforms. In: Lecture Notes in Computer Science (LNCS), Springer Verlag 1552 (1998b) Advances in Database Technologies, S. 205-217.
- [2000b] Gaedke, M.; Graef, G.: WebComposition Process Model: Ein Vorgehensmodell zur Entwicklung und Evolution von Web-Anwendungen. In: R. G. Flatscher; K. Turowski (Hrsg.): 2. Workshop Komponentenorientierte betriebliche Anwendungssysteme (WKBA 2). Wien, Austria 2000b, S. 21-38.
- [1999b] Gaedke, M.; Turowski, K.: Framework for Maintaining Evolution of E-Commerce Applications in the Web. In: J.-C. Rault (Hrsg.): 12th International Conference - Software and Systems Engineering and their Applications (ICSSEA '99). Bd. 5, Paris, France 1999b, S. 18.14.11-18.14.10.

- [1999c] Muragesan, S.: Web Engineering. In: SIGWEB Newsletter 8 (1999c) 3, S. 28-32.
- [2002b] Nussbaumer, M.: Einsatz von Petri Netzen und Xforms zur Modellierung und Unterstützung von interaktiven Web-Anwendungen. Diplomarbeit, Universität Karlsruhe. Karlsruhe 2002b.
- [2000c] Pemberton, S.; Altheim, M.; Austin, D.; Boumphrey, F.; Burger, J.; Donoho, A. W.; Dooley, S.; Hofrichter, K.; Hoschka, P.; Ishikawa, M.; ten Kate, W.; King, P.; Klante, P.; Matsui, S. i.; McCarron, S.; Navarro, A.; Nies, Z.; Raggett, D.; Schmitz, P.; Schnitzenbaumer, S.; Stark, P.; Wilson, C.; Wugofski, T.; Zigmond, D. XHTML 1.0: The Extensible HyperText Markup Language - A Reformulation of HTML 4 in XML 1.0. Recommendation Nr. REC-xhtml1-20000126, World Wide Web Consortium (W3C) 2000c.
- [1962] Petri, C. A.: Kommunikation mit Automaten. Dissertation, Technischen Universität Darmstadt. Darmstadt 1962.
- [1999d] Raggett, D.; Le Hors, A. HTML 4.0 Specification. <http://www.w3.org/TR/html4>, Abruf am 1999-24. December.
- [1999e] Turowski, K. (Hrsg.): Tagungsband des 1. Workshops Komponentenorientierte betriebliche Anwendungssysteme (WKBA 1). Otto-von-Guericke-Universität Magdeburg, Magdeburg 1999e.
- [2000d] WAP Forum. Wireless Application Protocol Wireless Markup Language Specification Version 1.3. Specification Nr. SPEC-WML-20000219, 2000d.