

Modellgetriebene Generierung von Webanwendungsprototypen

Eike M. Lang¹, Michael Wissen², Jürgen Ziegler¹

Universität Duisburg-Essen¹, Fraunhofer IAO²

Zusammenfassung

Existierende Entwicklungsmethoden und Werkzeuge für das Software-Engineering unterstützen die Entwicklung informationsgetriebener Webanwendungen nur unzureichend und finden in der Praxis daher kaum Anwendung. Im BMBF-geförderten Projekt „Web Information and Service Engineering“ (WISE) entstand ein werkzeugunterstützter Methodenverbund, der speziell auf die Bedürfnisse bei der Entwicklung webbasierter Informationssysteme zugeschnitten ist. Die Modellierung erfolgt auf Basis von Ontologien und leicht verständlichen Navigationsmodellen. Ein grafischer Editor ermöglicht die Erstellung und Pflege der Modelle und übergibt diese an eine Generatorkomponente, die automatisiert funktionsfähige Webanwendungs-Prototypen erzeugt.

1 Einleitung

Bei der Entwicklung webbasierter Informationssysteme kommen eine Vielzahl unterschiedlicher Systeme wie z.B. Web-Editoren oder Content-Management-Systeme zum Einsatz. Der Fokus dieser Technologien liegt im Wesentlichen auf der Gestaltung und flexiblen Erzeugung der grafischen Präsentation von Inhalten, dem Einstellen und Verwalten der Inhalte innerhalb eines Redaktionsprozesses sowie der Anbindung an Datenbanken und existierende Applikationen. Vorrangig dienen diese Werkzeuge zur Umsetzung von bereits spezifizierten Web-Anwendungen und eignen sich nicht für Konzeptionsaufgaben. Diese werden bislang nur unzureichend methodisch bzw. durch Modellierungswerkzeuge unterstützt: Insbesondere für die Erstellung einer Themenstruktur und eines darauf aufbauenden Navigationskonzepts fehlen geeignete Modellierungstechniken, die die Kommunikation zwischen Kunde und Entwickler auf der Basis ausführbarer Prototypen in den frühen Entwicklungsphasen unterstützen. Gerade im Bereich der Web-Anwendungsentwicklung ist die frühzeitige Präsentation von Konzeptionsschritten gegenüber dem Auftraggeber von hohem Wert.

Im Rahmen des BMBF-geförderten Verbundvorhabens „Web Information and Service Engineering“ (WISE) wurde ein Methoden- und Werkzeugverbund entwickelt, der diese Problematik berücksichtigt, spezifisch die Entwicklung webbasierter Informationssysteme unter-

stützt und aus den erstellten Modellen einen funktionsfähigen Prototyp der modellierten Webanwendung erzeugen kann.

2 Nutzung des Werkzeugs

Der durch die Anwendung unterstützte Methodenverbund sieht drei wesentliche Modelle vor: Ein Informationsmodell für die Erstellung des Informationsraums und die Abbildung einer anwendungsbezogenen Themenstruktur, ein Navigationsmodell für die Spezifikation und strukturelle Anordnung der Seiteninhalte und des Navigationskonzepts sowie ein Aufgabenmodell zur Einbindung von Aufgabenabläufen. Darüber hinaus kann das Navigationsmodell aus einer Adaption- und einer Präsentationssicht betrachtet werden.

Im *Arbeitsbereich* wird das Modell grafisch dargestellt, in der Werkzeugleiste links davon sind zu oberst die im aktuellen Modus verwendbaren *Modellierungselemente* angeordnet, darunter befindet sich der *Parameterbereich* für das aktuell im Modell ausgewählte Element. Zu unterst zeigt der *Vorschaubereich* für bestimmte Modellierungselemente an, welche Konzepte oder Instanzen zur Laufzeit von ihnen dargestellt werden.

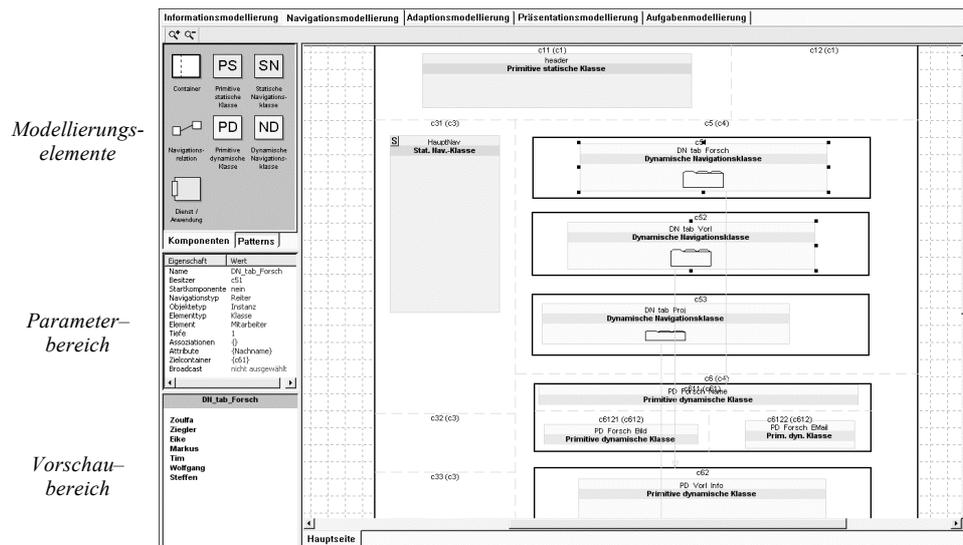


Abbildung 1: Editor

2.1 Informationsmodellierung

Das Informationsmodell entspricht in seiner Grundstruktur einer Ontologie: Es enthält Konzepte, Instanzen und Relationen zwischen diesen. Ein Konzept definiert die Attribute seiner Instanzen und die Relationen, an denen sie beteiligt sein können

Im Werkzeug kann die Inhaltsmodellierung in zwei Darstellungsmodi erfolgen. Die Netzdarstellung, in der Konzepte durch Ovale, Instanzen durch Rechtecke und Relationen durch gerichtete Kanten visualisiert werden, ist besonders für das Anlegen von Konzepten und Beziehungen zwischen Konzepten geeignet. Die tabellarische Darstellung bietet sich vor allem für die Bearbeitung von Instanzen und Instanzbeziehungen an.

Das Informationsmodell enthält die Themen der Web-Anwendung und dient als Grundlage für das Navigationskonzept und die Definition der Inhaltsbereiche. Die Struktur des Informationsmodells vereinfacht die Verknüpfung der Inhalte mit den Navigationselementen der Web-Präsenz und ermöglicht den Informationszugriff anhand der Modellesemantik.

2.2 Navigationsmodellierung

In dieser Sicht können Navigationselemente erzeugt, angeordnet und konfiguriert werden. Bereits an dieser Stelle entsteht ein erster visueller Eindruck vom späteren Erscheinungsbild der Webanwendung. Die grafische Notation des Navigationsmodells basiert auf für den Web-Kontext angepassten Statecharts (Harel 1987), und stellt die folgenden Modellierungselemente zur Verfügung:

Container dienen als strukturierende Elemente. Sie können weitere Modellelemente aufnehmen, auch weitere Container. Sie repräsentieren einen rechteckigen Bereich einer Webseite, können horizontal oder vertikal partitioniert sein und besitzen zur Laufzeit einen Zustand.

Statische Klassen dienen zur Darstellung fest definierter Elemente des Informationsmodells. Eine *primitive statische Klasse* visualisiert ein Attribut eines Elements aus dem Informationsmodell, eine *primitive navigationale Klasse* eine beliebige Auswahl von Elementen des Informationsmodells.

Dynamische Klassen stellen zur Laufzeit berechnete Inhalte dar. Die *primitive dynamische Klasse* präsentiert abhängig vom Zustand des umschließenden Containers ein Attribut eines einzelnen Elements des Informationsmodells, die *dynamische Navigationsklasse* ein Attribut einer Menge von typgleichen Elementen des Informationsmodells.

Eine *Navigationsrelation* von einem Element A zu einem Element B entspricht zur Laufzeit einem Verweis, der die Ersetzung von A durch B auslöst.

In der Präsentations- und Adaptionssicht wird ebenfalls das Navigationsmodell dargestellt, aber nur für die jeweilige Sicht relevante Aspekte können bearbeitet werden.

3 Die Generatorkomponente

Die Generatorkomponente verwendet das Web-Publishing-Framework Apache Cocoon, um im Zuge einer stufenweisen Transformation der Modelldaten die gewünschte Applikation zu erzeugen. Dazu wird die modellierte Navigationsstruktur zunächst in eine JAVA-Objekthierarchie übersetzt und der Zustand der einzelnen Elemente berechnet. Die Berechnung erfolgt auf Basis des bisherigen Zustands und übergebener HTTP-Parameter.

Anschließend wird die Objekthierarchie in einen XML-Strom überführt, der die Schachtelung der Modellelemente syntaktisch abbildet. Nicht aktive Elemente werden ab diesem Punkt in der Bearbeitung nicht berücksichtigt. Aus den Attributen und dem Zustand der Elemente werden dynamisch SPARQL-Anfragen erzeugt und auf dem Informationsmodell ausgeführt, deren Ergebnisse wieder in den XML-Strom eingefügt werden.

Als letzter Schritt werden die XML-Daten in XHTML umgewandelt. Wurde ein CSS-Stylesheet im System hinterlegt, wird dieses von den erzeugten Seiten verwendet, ansonsten sorgt ein Standard-Stylesheet für die modellkonforme Präsentation der erzeugten Seiten.

4 Verwandte Arbeiten

In (Kraus & Koch 2002) wird die halbautomatische Erzeugung von Webapplikationen aus Modellen beschrieben. Im Gegensatz zum vorliegenden Ansatz wird dort UML als Modellierungssprache verwendet und es existiert keine vollständig integrierte Werkzeugunterstützung. WebML (Ceri et. al. 2000) wird durch das kommerzielle WebRatio-Werkzeug durchgängig unterstützt, zielt aber auf die Modellierung produktiver Webanwendungen und ist für das Prototyping häufig unnötig komplex.

5 Bewertung und Ausblick

Das vorgestellte Werkzeug ermöglicht den schnellen Entwurf funktionsfähiger Webanwendungen, ohne auf eine formale Modellierung zu verzichten. Die Auswirkung von Änderungen am Modell wird durch den Generator unmittelbar sichtbar gemacht und ermöglicht somit ein schnelles Prototyping der Anwendung. Die Einbindung von CSS-Stylesheets eröffnet die Möglichkeit, dabei auch Präsentationsaspekte zu berücksichtigen.

In Zukunft soll die Bedienung des Editors weiter vereinfacht werden. Der Generator eignet sich vom Reifegrad derzeit für prototypische Umsetzungen, ist aber architektonisch bereits auf eine Funktionalität als Laufzeitumgebung für Produktivsysteme ausgerichtet.

Literaturverzeichnis

- S. Ceri, P. Fraternali, A. Bongio (2000): Web Modeling Language (WebML): a Modeling Language for Designing Web Sites. In: WWW9 Conference, Amsterdam.
- D. Harel (1987): Statecharts: A visual formalism for complex systems. In: Science of Computer Programming, 8(3):231--274.
- Kraus, A., Koch, N. (2002): Generation of Web Applications from UML Models using an XML Publishing Framework. In: Proceedings of the 6th World Conference on Integrated Design and Process Technology (IDPT'02), Pasadena, USA