

XForms

Wechseln zu: [Navigation](#), [Suche](#)

Inhaltsverzeichnis

- [1 Definition](#)
- [2 Designziele](#)
- [3 Formularstruktur nach MVC](#)
 - [3.1 Anschauungsbeispiel](#)
 - [3.2 Model](#)
 - [3.3 View](#)
- [4 Absichtsorientierte Formularelemente](#)
- [5 Eventgetriebene Aktionen](#)
- [6 Standardisierungsgeschichte](#)
- [7 Quellen](#)

1 Definition

Die Beschreibungssprache **XForms** bietet eine deklarative Entwicklungsart für reichhaltige Webformulare, die XML als Transferformat akzeptieren und den Browser als Prozessknoten ausnutzen. Sie verspricht offensichtliche Schwächen der alten HTML-Formulargeneration zu beheben, ist nahtlos in gängige XML-Formate wie *XHTML 1.0* oder *SVG* integrierbar und wird nativer Bestandteil von *XHTML 2.0*.

XForms wird im Rahmen des [W3C](#)-Konsortiums durch die *Forms Working Group* als Standard spezifiziert. Die aktuelle Version *XForms 1.0* trägt den finalen Status einer *W3C Empfehlung* und greift massiv auf [XPath](#) für Adressierung und Berechnungen zurück.

2 Designziele

Bei XForms wurden aus den analysierten Schwächen von herkömmlichen HTML-Formularen einige Designziele abgeleitet:

Eine einfachere, barrierefreie Interpretation auf heterogenen Geräten durch Absichts- statt Präsentationsorientierung.

Eine strukturierte Entwicklungsform über die Trennung von Formulardaten und Präsentation nach dem MVC-Pattern.

Ein reichhaltigeres Datenmodell für die Speicherung und den Transfer von Formulardaten durch typisiertes XML.

Eine dynamischere, interaktive Bedienoberfläche ohne Scripting-Aufwand durch eventgekoppelte Aktions- und Widgetelemente.

Eine Entlastung für Server und Netz, indem der Browser als Prozessknoten bei Validierung und Berechnungen mitarbeitet.

3 Formularstruktur nach MVC

Bei XForms werden die Formulardaten (*Model*) von den korrespondierenden Formularfeldern (*View*) separiert beschrieben - die Steuerlogik (*Controller*) realisiert ein XForms-Prozessor im Browser. Mit XHTML als Trägerformat kann folgende repräsentative Struktur befüllt werden:

Beschreibung des XForms Model

Beschreibung der XForms View

3.1 Anschauungsbeispiel

Als einführendes Beispiel wird nun für die Erklärung von Model- und View-Elementen eine primitive **Preisauskunft** für einen *Produktkatalog* konstruiert. Sie zeigt zu einer eingegebenen *Produktnummer* nach Betätigen des Formular-Buttons den *Preis* des Produktes an:



3.2 Model

Das XForms **Model** (`<xf:model>`) beschreibt Daten und Aktionen, die zur Erbringung der Grundfunktion nötig aber unabhängig von einer konkreten Darstellung sind. Beim Modell zur *Preisauskunft* sind folgende Elemente von zentraler Bedeutung:

```
<xf:model>
  <xf:instance xmlns="" id="transfer">
    <product num="1337">
      <price>99.99</price>
    </product>
  </xf:instance>
  <xf:bind id="num" nodeset="/product/@num" required="true()" />
  <xf:submission id="fetch" action="prices.xql" method="get"
    ref="instance('transfer')" replace="instance" instance="transfer"
  />
</xf:model>
```

Instanzen (`<xf:instance>`): dienen als XML-Templates, die im Zuge der Formularbearbeitung gefüllt, dem Server zugeschickt oder von ihm empfangen und präsentiert werden. Der Server liest aus obigen Transferdaten später z.B. das gesetzte *num*-Attribut aus, befüllt entsprechend das *price*-Element und schickt alles zurück zum Formular.

Übermittlungen (`<xf:submission>`): legen Transferaktionen fest, die im Laufe der

Formularbearbeitung initiiert werden. Attribute konfigurieren die Übermittlung: angesprochene URI (*action*), verwendete HTTP-Methode (*method*), mitzusendende XML-Daten (*ref*), was durch die Serverantwort ersetzt wird (*replace*) und wie hier bei Instanzersetzung die Zielauswahl (*instance*). **Bindings** (`<xf:bind>`): erlauben es mittels XPath (*nodeset*) Knoten aus den *Instanzen* auszuwählen und über Attribute an Sonderleistungen zu knüpfen: Festlegen eines Identifizierers (*id*), Ausweisen als verbindlich (*required*), Ausweisen als schreibgeschützt (*readonly*), Zuweisen eines Datentyps zur Validierung (*type*) oder Berechnung des Knoteninhalts über XPath (*calculate*).

3.3 View

Die XForms **View** (`<xf:view>`) bedient sich am vorbereiteten Datenmodell und setzt es in Steuerelemente um. Für das Modell zur *Preisauskunft* könnten dies mit einem Input-Feld, einem Output-Feld und einem Submit-Button geschehen, mit einigen Besonderheiten:

```
<xf:group>
  <xf:label>Preisauskunft</xf:label>
  <xf:input bind="num">
    <xf:label>Produktnummer</xf:label>
  </xf:input>
  <xf:output ref="/product/price">
    <xf:label>Produktpreis</xf:label>
  </xf:output>
  <submit submission="fetch">
    <xf:label>Abfrage</xf:label>
  </submit>
</xf:group>
```

Visuelle **Gruppierung** (`<xf:group>`): Steuerelemente lassen sich einfach als Gruppe hervorheben.

Labels (`<xf:label>`): Feldbezeichnungen sind in XForms verpflichtend und direkt den Feldern als Kindelement zugeordnet.

Widgets: Referenzieren die Modelldaten entweder über XPath-Ausdrücke mit (*ref* bei `<xf:output>`) oder über im Modell vorgegebene Identifizierer (*bind* & *submission* bei `<xf:input>` und `<xf:submit>`). Die erste Variante ist weniger sauber, da damit die referenzierte Instanzstruktur bekannt sein muss (engere Kopplung).

4 Absichtsorientierte Formularelemente

XForms interpretiert alle Steuerelemente streng absichts- und nicht präsentationsorientiert, um möglichst viele heterogene Endgeräte barrierefrei zu erreichen. Das bedeutet, dass ein Endgerät über die tatsächliche Darstellung entscheidet und so ein *input*-Feld für einen Datumsknoten z.B. in Browsern einen Kalender anbietet oder in blindengerechten Systemen einen Audiodialog formuliert. Das *appearance*-Attribut steht in allen Feldern zur Verfügung, um die Darstellungsabsicht abstrakt zu präzisieren: Eine Einzelauswahl (*select1*) wird damit in Browsern z.B. mit dem Wert *full* als Radio-Buttons, mit *compact* als List-Box oder mit *minimal* als Drop-Down-List dargestellt. Alle weiteren Details im Erscheinungsbild sind per CSS-Formatierung zu regeln. Die verfügbaren **primitiven** Widgets umfassen:

input: Eingabe von einzeiligem Text, Zahlen, Datum, ...
secret: Eingabe von verdecktem Text wie Passwörter
textarea: Eingabe von mehrzeiligem Text
output: Ausgabe von Text, Zahlen, Datum, ...
upload: Auswahlfeld für den Dateiupload
range: Slider-Auswahl für einen Wertebereich
trigger: Entspricht einem beliebigen Knopf
submit: Entspricht einem Übermittlungsknopf
select1: Liste zur Einzelauswahl
select: Liste zur Mehrfachauswahl

5 Eventgetriebene Aktionen

Als **Events** kann XForms alle Ereignisse des [Document Object Model \(DOM\)](#) auswerten wie *DOMActivate*, das von der Auswahl eines Kontrollelements z.B. per Mouse-Click berichtet. XForms feuert auch eigene Ereignisse wie *xforms-ready* nach der Initialisierung, *xforms-submit* vor dem Ausführen einer Übermittlung oder *xforms-value-changed* vor dem Schreiben von Feldinhalten in das Datenmodell. Die Idee ist nun, dem Entwickler für solche Ereignisse Eingriffsmöglichkeiten als **Aktions**-Elemente bereitzustellen, wie z.B. `<xf:message>` zum Anzeigen einer Nachricht, `<xf:setvalue>` zum Setzen eines Instanzknotenwerts oder `<xf:send>` zum Starten einer Übermittlung. Mehrere Aktionen können in einer `<xf:action>` gruppiert und fortan wie eine Einzelaktion behandelt werden. Um eine Aktion an ein Event zu binden, setzt man schlicht ihr Attribut `ev:event` auf den Ereignisnamen. Eine Erweiterung zum *Model* der *Preisauskunft* setzt z.B. nach der Initialisierung Nummer und Preis auf 0 und zeigt einen modalen Dialog:

```
<xf:model>
  ...
  <xf:action ev:event="xforms-ready">
    <xf:setvalue ref="/product/price" value="0.00" />
    <xf:setvalue bind="num">0</xf:setvalue>
    <xf:message level="modal">
      Tragen Sie bitte eine Produktnummer ein!
    </xf:message>
  </xf:action>
</xf:model>
```

Mit einigen speziell entworfenen dynamischen **Widgets** wie einem *Tab-Control*-Mechanismus (`<xf:switch>`) lässt sich damit zusammen mit dem Einsatz von Aktionen, Events und XPath-Berechnungen der Scripting-Einsatz oft komplett vermeiden.

6 Standardisierungsgeschichte

10/2001: *XML Events - W3C Working Draft*
08/2002: *XForms 1.0 - W3C Working Draft*
11/2003: *XForms 1.0 - W3C Recommendation*
01/2004: *XForms 1.1 - W3C Requirements*
11/2004: *XForms 1.1 - W3C Working Draft*

10/2005: *XForms 1.0 (2nd Edition)* - W3C Proposed Recommendation

03/2006: *XForms 1.0 (2nd Edition)* - W3C Recommendation

10/2007: *XForms 1.0 (3rd Edition)* - W3C Recommendation

7 Quellen

Dubinko, Micah (2003): *XForms Essentials*. O'Reilly Media Inc., ISBN 0596003692.

[W3C - XForms 1.0 \(Third Edition\)](#)

[W3C - The Forms Working Group](#)

[W3Schools.com - XForms Tutorial](#)

[W3C - Document Object Model Activity Statement](#)

[Wikipedia - XForms](#)

Dieser Artikel ist [GlossarWiki-konform](#).

In diesem Artikel sollten die Quellenangaben überarbeitet werden.
Bitte die Regeln der [GlossarWiki-Quellenformatierung](#) beachten.

Kategorien:

[XML](#)

[Web-Programmierung](#)

[Content-Management](#)

[Glossar](#)

Diese Seite wurde zuletzt am 6. März 2014 um 12:31 Uhr bearbeitet.

Inhalt verfügbar unter [CC BY-SA 4.0](#).

