



7363 - Web-basierte Anwendungen

Eine Vertiefungsveranstaltung
mit Schwerpunkt auf XML-Technologien



XForms

Extensible Forms
(erweiterbare Formulare)



Einführung

Die Nachfolger der Web-/HTML-Formulare



- Vorbemerkungen
 - Die folgenden Folien sind nur ein erster Einstieg ins Thema.
 - Unser Ziel ist, einen ersten Eindruck von den Möglichkeiten und Aufgaben von XForms zu bekommen.
 - Gleichzeitig demonstriert XForms ein Beispiel für die Weiterentwicklung von XHTML.



- *Model-View-Controller-Prinzip*

- Heutige Web-Formulare sind sehr darstellungsorientiert.
- Beispiel: Auswahl per *radio buttons*

```
<form>
  <input type="radio" name="..." value="...">Text</input>
  <input type="radio" name="..." value="...">Text</input>
  ...
</form>
```

- Diskussion

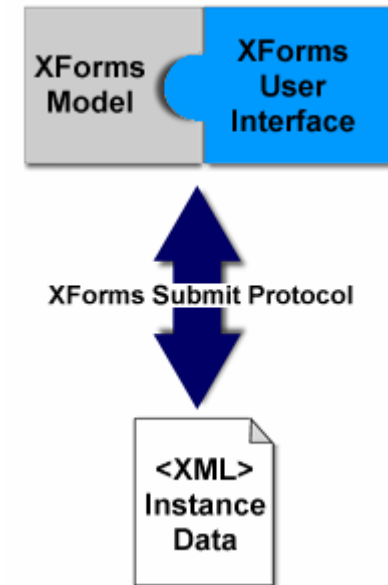
- Die Darstellung als *radio buttons* hat nichts mit der Aufgabe "Auswahl aus einer Werteliste treffen" zu tun!
- Alternative Darstellungen etwa: *Checkboxes*, *Drop-Liste*



XForms: Konzepte und Ziele



- *Model-View-Controller-Prinzip* (Forts.)
 - Bei XForms zu unterscheiden:
 - Logischer Aufbau des Formulars
 - Darstellungsaspekte (*styling*)
 - Inhalt ("Dokumententyp") der Formularfelder (Eingaben)



Quelle: www.w3.org



- XML-Technologie als konsequente Grundlage
 - Formularbeschreibung, Struktur der Inhalte und übertragene Daten sind XML-basiert
 - Dadurch auch bessere Internationalisierung (Unicode-Erbe)
 - Mischung mit Elementen anderer XML-Sprachelemente (Modularisierung) „sauber“ möglich dank Namensraumunterstützung
 - Vereinheitlichung der Werkzeuge (etwa: Parser, XSLT-Prozessoren)
 - Integration von XPath 1.0, plus Funktionserweiterungen



- Unterstützung verschiedener Medien
 - *Handhelds*, TV, Drucker, *Scanner*, ...
 - Barrierefreiheit besser umsetzbar
- Fortgeschrittene Eigenschaften
 - Komplexe Formulare möglich, z.B. für eine vollständige Bestellung
 - Strukturierung der Eingabedaten
 - Mehrere Formulare pro Seite, mehrere Seiten pro Formular
 - Mehr "submit"-Methoden, auch parallel
 - Logik: **switch/case**
 - Schleifen: **repeat** (für jedes Element einer XPath-Knotenmenge)
 - Veränderung des Inheldokuments: **copy, delete, insert**
 - Anwenderschnittstelle mit mehr Möglichkeiten, z.B. "spreadsheet"
 - Neue Eigenschaften (z.B. Rechenmöglichkeiten, vgl. XPath) und neue Anweisungen erübrigen oftmals *Scripting*



- Status:
 - W3C *Recommendation* 1.0, 14.10.2003
 - W3C *Recommendation* 1.0 (*second edition*), 14.3.2006
 - Quelle: <http://www.w3.org/TR/xforms/>
- XForms
 - bildet keine eigenständigen XML-Dokumente!
 - Immer ist eine "*host language*" wie XHTML erforderlich
 - Einbettung der XForms-Elemente per Namensraumkonvention
 - Namensraum-URI:
 - <http://www.w3.org/2002/xforms>
 - Schema:
 - <http://www.w3.org/MarkUp/Forms/2002/XForms-Schema.xsd>



- [1] <http://www.w3.org/TR/xforms/>
 - Die Spezifikation, 13 Kapitel + 10 Anhänge. Lesbar, reich an Beispielen!
- [2] <http://www.w3.org/MarkUp/Forms>
 - Übersichts-Seite mit zahlreichen Links zu weiteren Quellen
 - Gut als zentraler Einstieg in das Thema
 - FAQ dazu: <http://www.w3.org/MarkUp/Forms/2003/xforms-faq>
- [3] <http://www.w3.org/MarkUp/Forms/2003/xforms-for-html-authors>
 - Unstiegshilfe für bisherige Anwender von HTML-Formularen
 - Teil 2: <http://www.w3.org/MarkUp/Forms/2006/xforms-for-html-authors-part2>
- [4] <http://www.w3schools.com/xforms/default.asp>
 - Ein leicht verständliches, kurzes Tutorial
 - Mit Werbebannern finanziert, brauchbar für einen ersten Eindruck
- [5] <http://xformsinstitute.com/>
 - Ein etwas größeres Tutorial mit "*live rendering*"
- [6] **XForms Essentials**. Micah Dubinko, O'Reilly, Sebastopol, CA, 2003.
ISBN: 0-596-00369-2
 - Noch nicht in Augenschein genommen, 2*5*'-'-Bewertungen bei Amazon
 - Unter <http://xformsinstitute.com/essentials/> on-line verfügbar!



- Vorbemerkungen
 - XForms ist noch relativ neu. Von genereller Unterstützung in den üblichen Browsern ist noch keine Rede. Dennoch lässt sich auch heute bereits mit XForms arbeiten:
- Serverseitig:
 - XForm-Technik kann bereits heute auf dem Server eingesetzt werden!
 - XForms werden in traditionelle HTML-Formulare konvertiert
 - Werkzeuge: Z.B. "[XMLForm](#)", eine Apache/Cocoon-Ergänzung
- Clients:
 - Browser mit nativer XForms-Unterstützung
 - Bisher Prototypen oder laufende Projekte, z.B:
 - [X-Smiles](#) (Java-Browser), [Mozilla XForms 0.7](http://www.mozilla.org/projects/xforms/)
 - Plugins
 - Proprietär: Verschiedene IE-Plugins
 - Als Zusatzmodul für die Gecko-Engine (Mozilla), in Vorbereitung
 - Offen (GPL): [DENG](#) - eine Erweiterung zum Flash 6-Plugin

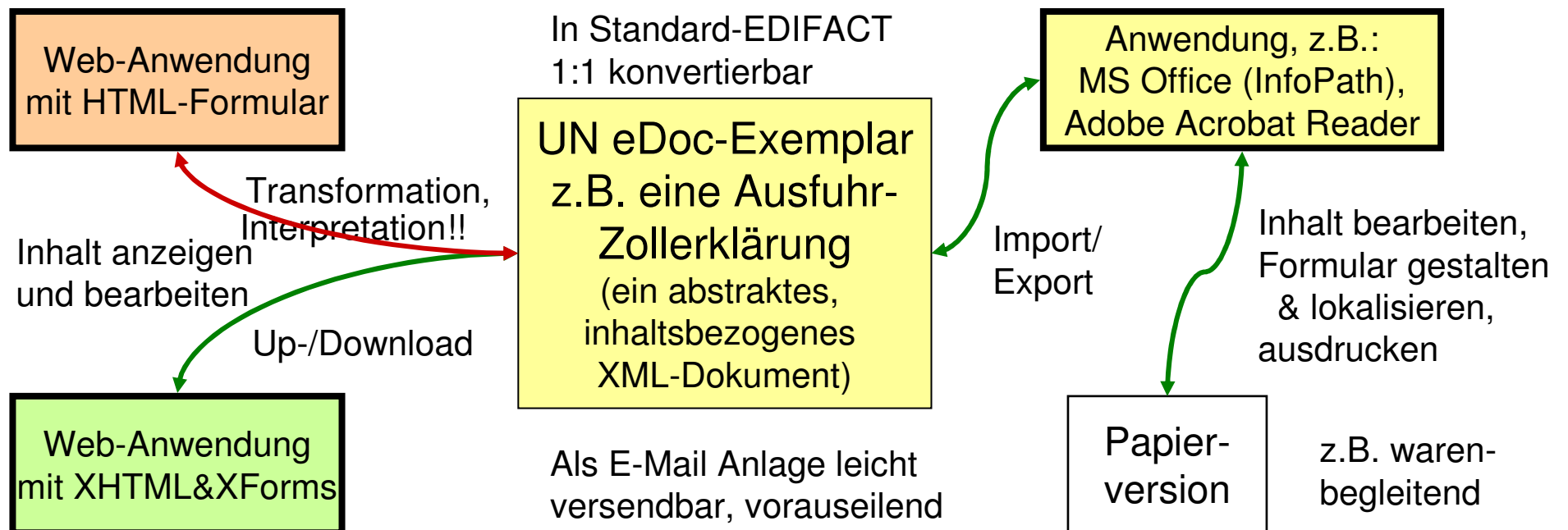


XForms: Proprietäre "Alternativen"



- Microsoft
 - **InfoPath**: Eine neue Komponente von MS Office (*Enterprise Version*), enthält u.a.:
 - Formulareditor
 - Export der Inhalte als XML-Dokumente
 - Weiteres, noch zu recherchieren...
- Adobe
 - **PDF, FDF** (Info von: Demo am Adobe-Messestand)
 - Formulareditor
 - Erzeugung von PDF-Dokumenten mit Eingabemöglichkeiten
 - Erweiterung der Dokumente um Logik (JavaScript)
 - Export der Inhalte als XML-Dokumente
 - Integrierte Authentifizierung, teilweise auch *Workflow*-Aspekte
 - *Off-line* wie auch *On-line* Varianten

- UN-eDocs: *UN Electronic Trade Documents*
 - Initiative zur weltweiten Standardisierung von (im internationalen Handel genutzten) Handelsdokumenten
 - **Brücke zwischen Papier und EDI/XML**
 - Quelle: UNO (UN/ECE), <http://www.unedocs.org>
- Nutzung & Zusammenhang mit XForms:





Ausblick: XForms 1.1



- Status:
 - W3C Working Draft 1.1, 12.12.2006
 - Quelle: <http://www.w3.org/TR/xforms11/>
- Angestrebte Verbesserungen
 - Client/Server-Interaktion
 - **SOAP-Integration: SOAP als weitere "submit"-Methode**
 - Bessere Steuerung der "submissions"
 - Verschiedene kleinere Verbesserungen zu XForms 1.0
 - Verbesserte Einbettung in *host languages*



XForms: Kleine Beispiele



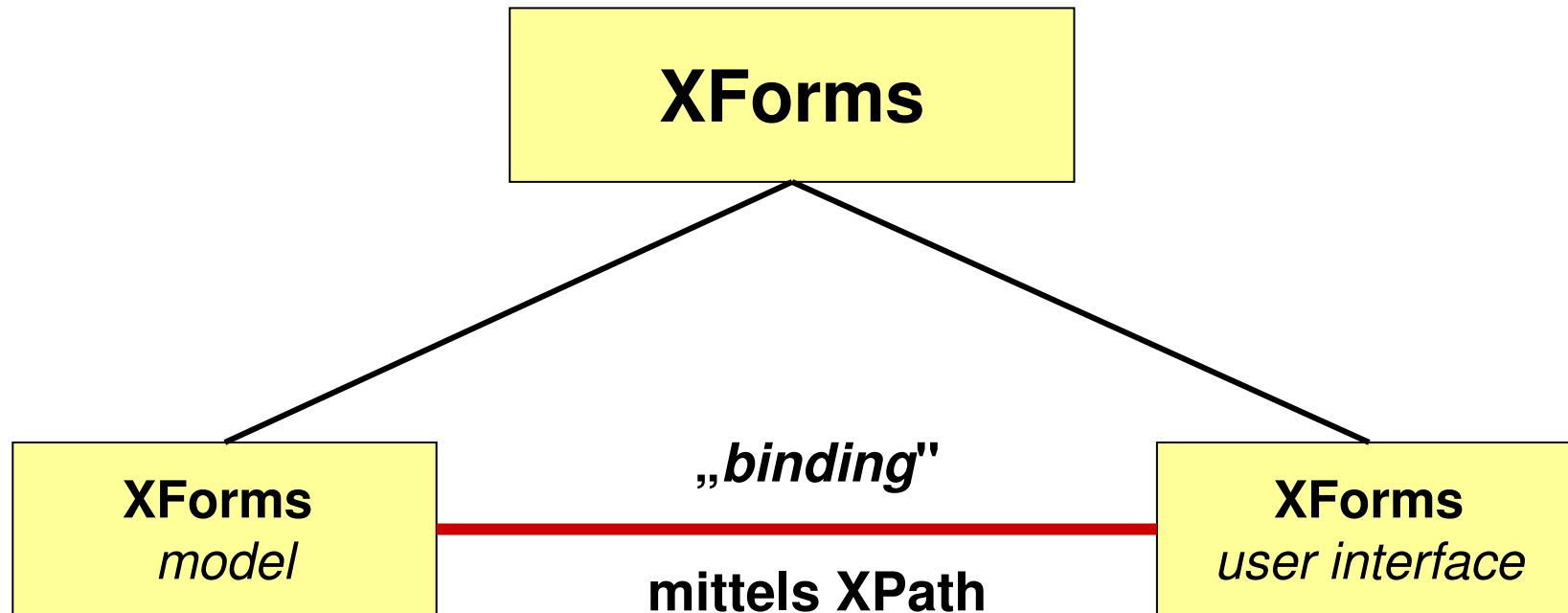
- Einfaches HTML-Formular: "Suchmaschine"

```
<html>
  <head>
    <title>Search</title>
  </head>
  <body>
    <form action="http://example.com/search"
          method="get"> Find
      <input type="text" name="q">
      <input type="submit" value="Go">
    </form>
  </body>
</html>
```




- "Suchmaschine" in XForms-Version

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:f="http://www.w3.org/2002/xforms">
  <head> <title>Search</title>
    <f:model> <!-- Aktionen und ggf. Datenstruktur -->
      <f:submission action="http://example.com/search"
                    method="get" id="s"/>
    </f:model>
  </head>
  <body>
    <p> <!-- Formularkomponenten -->
      <f:input ref="q"><f:label>Find</f:label></f:input>
      <f:submit submission="s">
        <f:label>Go</f:label>
      </f:submit>
    </p>
  </body> </html>
```

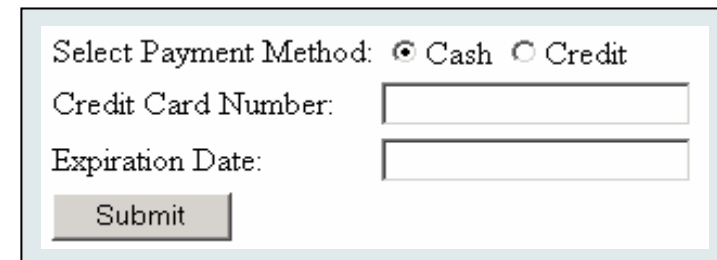


Ggf. Bereitstellung eines Datenexemplars, leer oder mit Defaultwerten befüllt.

Bezugnahme auf Elemente des Datenexemplars mit XPath-Ausdrücken.

- W3C-Beispiel
 - Ausschnitt aus "head" eines XHTML-Dokuments

```
<xforms:model>
  <xforms:instance>
    <ecommerce xmlns="">
      <method>cash</method>
      <number/>
      <expiry/>
    </ecommerce>
  </xforms:instance>
  <xforms:submission action="http://example.com/submit"
    method="post" id="submit"
    includenamespaces="" />
</xforms:model>
```



Select Payment Method: Cash Credit

Credit Card Number:

Expiration Date:



- W3C-Beispiel: Zugeordnete Kontroll-Elemente

```
<select1 ref="method" >                                <!-- Im HTML-Body: -->
  <label>Select Payment Method:</label>
  <item> <label>Cash</label>   <value>cash</value> </item>
  <item> <label>Credit</label> <value>cc</value>   </item>
</select1>
```

select1-Element:
Genau ein "item" soll
ausgewählt werden!

```
<input ref="./number" >
  <label>Credit Card Number:</label>
</input>
```

```
<input ref="child::expiry" >
  <label>Expiration Date:</label>
</input>
```

Werte: XPath-Ausdrücke mit
Kontextknoten=Kindknoten
von "instance"

```
<submit submission="submit" >
  <label>Submit</label>
</submit>
```

Bezug zu Element
"submission",
Attribut "id"



- Labels
 - (Fast) Alle K. müssen ein Element "label" enthalten
 - Damit: Konsistente Beschriftung möglich
- Darstellungsneutral
 - "select1" enthält eine Absicht, aber keine Angaben zur Realisierung
 - Dies gilt hier generell (Ausnahme: Hinweise zur Realisierung)
- Weitere Kontrollelemente
 - **input, secret, textarea**
 - Texteingabe: Einfache Zeile, versteckt (z.B. für Kennworteingabe), mehrzeiliges Textfeld
 - **output**
 - Reines Ausgabeelement, kann auch Werte von XPath's anzeigen!
 - **upload**
 - Löst ggf. Dialogbox zur Auswahl einer Datei aus



- Weitere Kontrollelemente (Forts.)
 - **range**
 - Eingabe eines Wertes aus einem vorgegebenen Wertebereich
 - Umsetzbar z.B. mit *sliders* oder *dial buttons*
 - **trigger, submit**
 - Allgemeine Kontrollfläche bzw. Kontrollfläche zum Absenden des Formulars
 - **select, select1**
 - Elemente zur Bildung von Auswahlgruppen
 - **select** lässt mehrere Auswahlpunkte zu (oder auch keine), **select1** genau eine Auswahl.
- **Demo**
 - Mit Firefox & *XForms extension V 0.7*



- Regelung der Darstellungsfragen
 - *User agent* (z.B. Browser)
 - Eingebaute Default-Regeln
 - Styling
 - CSS ☺ aber: Mindestens CSS3; noch viel zu tun
 - Beispiel in Anhang F der XForms 1.x-Spezifikation zeigt das Prinzip
 - Demo: „payment2“
 - Transformation mit XSLT in konventionelle (X)HTML-Formulare ☹
- Beispiel
 - Eine Telefon-Hotline-Anwendung könnte per Sprachausgabe und Spracherkennung dieselben (!) Formulare bearbeiten wie ein Browser:
 - Auswahlliste: Optionen vorlesen, Entscheidung abfragen
 - Eingabe: Stichworte abfragen, Zahlen wie Kreditkartennummern per Telefon-Tastatur
 - Submit: Per Sprachkommando



W3C: Quo vadis?

Anmerkungen zu aktuellen Entwicklungen



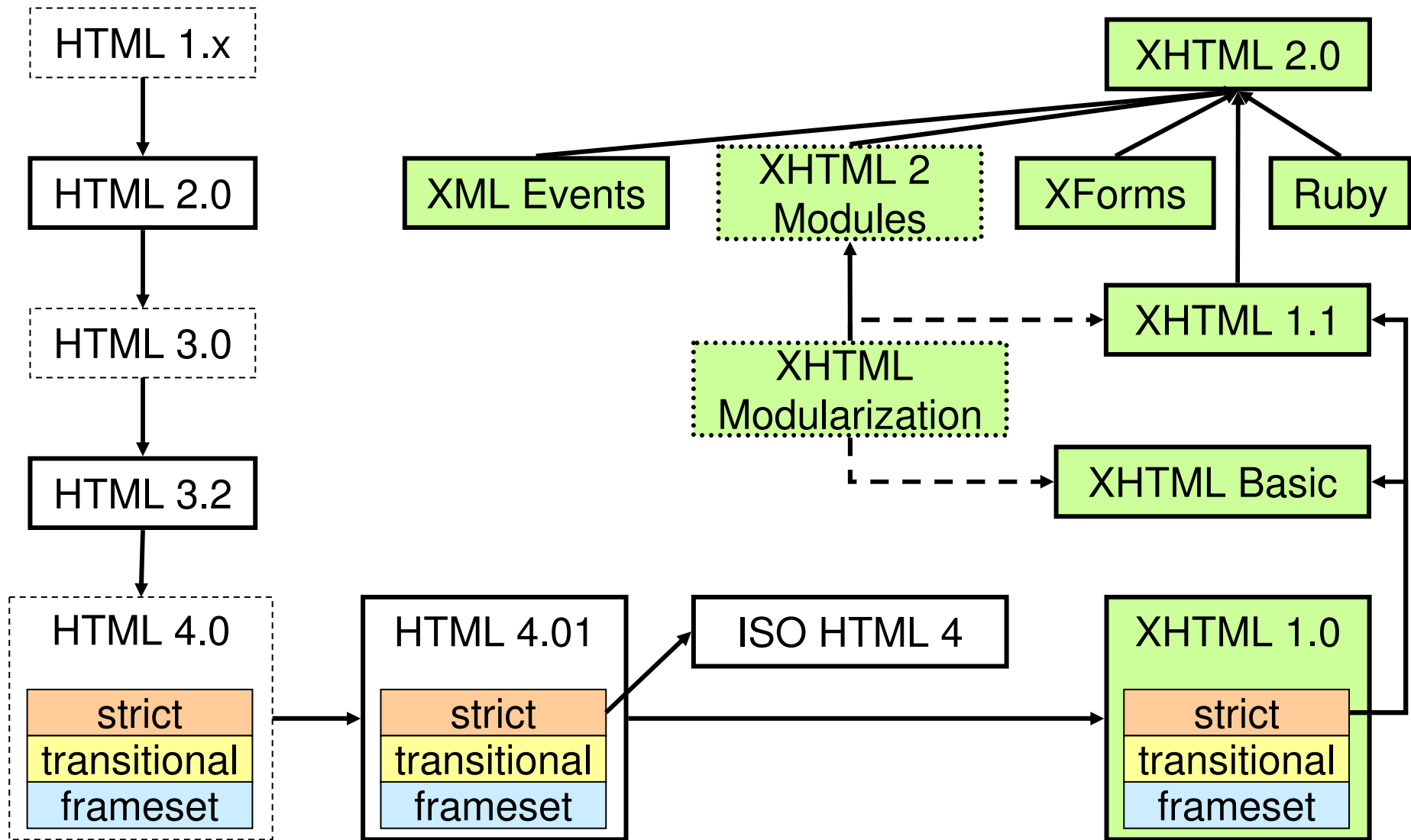
- Aktuelle Befunde zu (X)HTML
 - HTTP header „content-type=application/xhtml+xml“
 - MSIE 6: Abbruch/Fehlermeldung
 - MSIE 7: Aufforderung zum Speichern bzw. Starten einer externen Anwendung, keine korrekte Anzeige
 - Firefox, Opera: Korrekte Anzeige, XML-Betrieb (mit Prüfung auf Wohlgeformtheit)
 - Die Folge: Provider konfigurieren Web-Server so, dass ihre Daten bei der großen Mehrheit der Nutzer (MSIE!) angezeigt werden können, also etwa so:
 - *.html text/html
 - *.xhtml text/html
 - *.xml text/xml



- Aktuelle Befunde zu (X)HTML
 - Konsequenzen
 - Browser werden nicht im echten XHTML-Betrieb genutzt!
 - Webseiten-Entwickler meiden daraufhin XHTML
 - „bringt ja nichts“
 - „macht nur Ärger bei den Kunden“
 - Browser-Unterstützung stockt mangels Nachfrage
 - Die eigentlichen Fortschritte werden nicht erreicht ☹
- Kritik an XHTML 2.0
 - Zu viele Voraussetzungen
 - Zu komplex
 - Zu viel Neues auf einmal
- Die Folge: **Forderung nach HTML-Weiterentwicklung!**

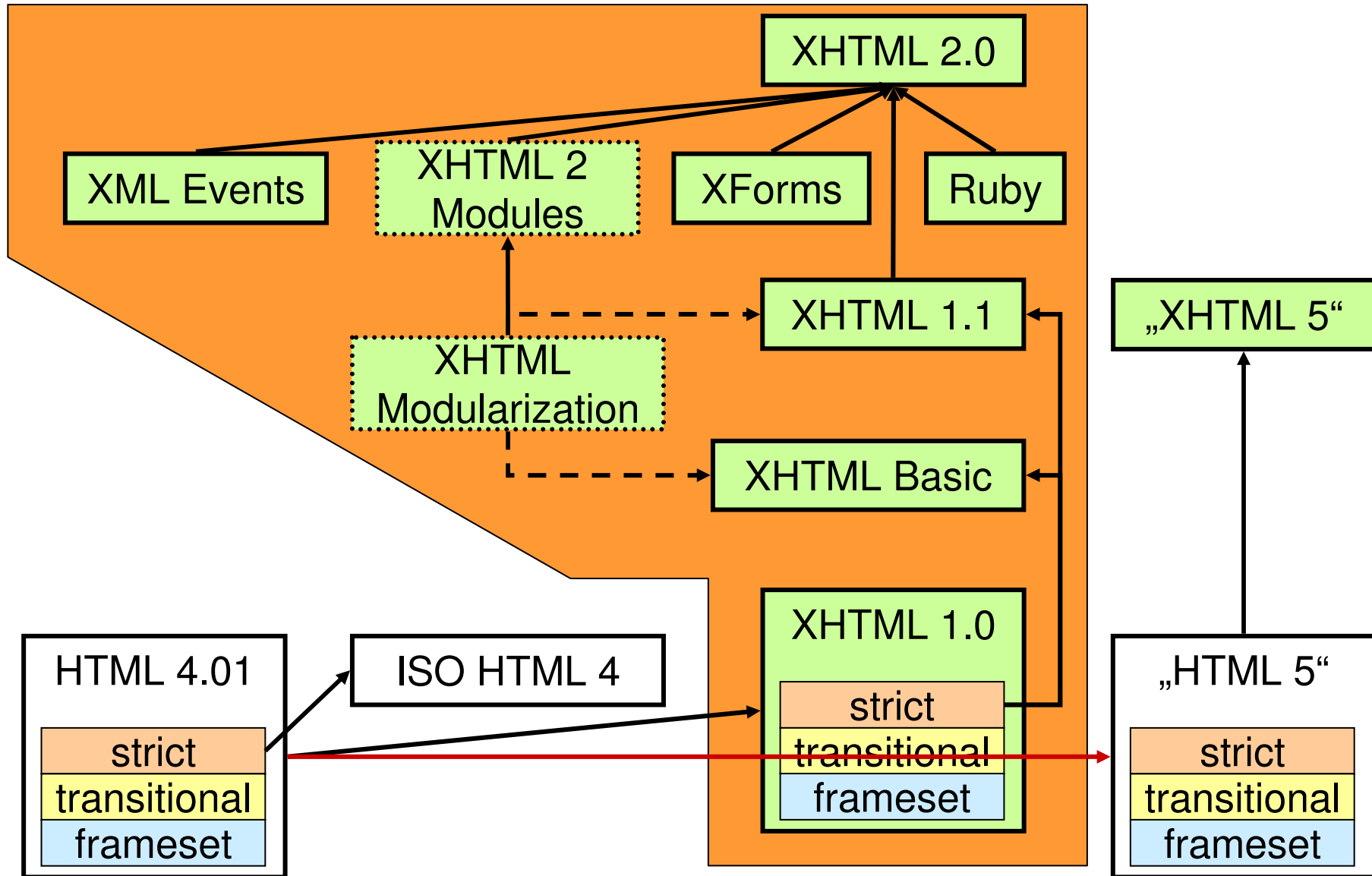


HTML und XHTML: Die bisherige W3C-Sicht





HTML und XHTML: Ziele der WHAT WG





- **Die WHAT WG**

- *WHAT WG = Web Hypertext Application Technology Working Group*

- Forderungen

- Weiterentwicklung von HTML 4.01 zu „HTML 5“
 - XML-Variante dazu namens „XHTML 5“
 - Ignorieren von XHTML 2.0 u.a. W3C-Empfehlungen

- Was soll HTML 5 sein?

- Endlich eine präzise Spezifikation von HTML (interoperable Browser!)
 - Web Forms 2.0 (Übernahme vieler Ideen von XForms!)
 - Web Applications 1.0 (Neue HTML- und DOM-Elemente wie „meter“ , „progress“ etc., mit Elementen / Widgets normaler Anwendungen wie Schieberegler, Drehknöpfe, Fortschrittsbalken auf Web-Seiten ermöglichen.

- Treibende Kräfte: **Apple, Mozilla, Opera (!)**



- **Diskussion**

- Risiken

- Fragmentierung des Web
 - Positionierung von Microsoft?
 - Drohender Verlust integrierender Möglichkeiten wie SVG und/oder MathML in XHTML

- Chancen

- Besser an den Status Quo angepasste, einfachere Standards
 - Auflösung der Entwicklungblockade von (X)HTML

- **Reaktion von Tim Berners-Lee, W3C**

- Nun doch inkrementelle Weiterentwicklung von HTML (aber gemeinsam mit XHTML), dabei Erhalt des XHTML 2.0-Teams
 - Übernahme einiger Positionen der WHAT WG

Mehr zum Thema in: Webstandards im Wandel, Herbert Braun, c't 1/2007, S. 162-169.