

# XForms——下一代的Web窗体

张 昱

(中国科学技术大学 计算机科学技术系, 安徽 合肥 230027)

**摘 要:** 综述了下一代Web窗体——XForms技术对XForms的窗体定义思想以及处理模型给予了阐述;说明了XForms与其它XML技术之间的关系和现状;调研总结了XForms实现所依赖的浏览器和API接口;最后简要分析、评述了XForms的不足以及待研究的问题

**关键词:** 窗体; XForms; 模式; 模型; 用户界面

**中图分类号:** TP393.092

**文献标识码:** A

**文章编号:** 1000-1220(2003)09-1658-07

## XForms--the Next Generation of Web Forms

ZHANG Yu

(Department of Computer Science & Technology University of Science & Technology of China, Hefei 230027, China)

**Abstract** Research on the next generation of Web forms--XForms starts from the observation. The idea of XForms and XForms Processing model are expounded. The relationships between XForms and other XML related specifications, and their current situation are explained. Browsers and APIs which the implementation of XForms depends on are summarized. In the end, the deficiencies of XForms are analyzed and several open problems are put forward.

**Key words:** forms; XForms; schema; model; user interface

### 1 引言

自1993年HTML中引入了窗体(Forms)以来,窗体已成为Web的一个关键部分,是众多Web站点用来与用户交互的主要手段之一。随着Web及其应用的蓬勃发展,Web窗体也受到了一些挑战:如何提供日益复杂的事务处理(如实时监控、电子商务等)窗体?如何有效地降低各种因被管理对象、管理体制、软件体系、界面表现等的变更而带来的大量Web窗体脚本的维护?如何将同一类服务展现给持有不同输入输出设备(如桌面浏览器、掌上机、电视机、打印机、扫描仪、声音浏览器等)的用户?

对于一个窗体来说,它包含功能、表现和数据三部分。现有的HTML窗体将这三部分混杂在一起;并且用户对HTML标记语言的使用经常是不严密的;再者,各浏览器厂商对HTML规范的实现并不完全相同,有的还有自己的扩展。这些造成HTML窗体结构的混乱,以及对HTML使用的不规范、不统一。另外,现有的窗体是以界面为中心的,创作人员往往将大量精力放在如何使网页美观;数据和界面的语义处理仅作为控件的属性,对数据的获取、设置和语义检查只能通过控件来进行。这些进一步造成了现有HTML窗体的编码质量存在严重的问题。

随着XML成为Web应用之间的数据交换标准<sup>[1]</sup>,重新考虑窗体以及它们如何在XML内容中工作逐步引起人们的关注。一些组织和研究人员纷纷提出了与Web窗体有关的新技术:如用分离的XSLT(XML Stylesheet Language Tree)变

换器<sup>[2]</sup>来表现图形界面;采用Form sheet<sup>[3]</sup>增加任意现存tag的交互性,指出界面中的哪一元素可以编辑,并且将变化的值提交回服务器;运用XFDL(XML Forms Description Language)<sup>[4]</sup>和XFA(XML Forms Architecture)<sup>[5,6]</sup>可视化地反映窗体页面,并且增加了复杂的公式、逻辑和数字签名安全等;使用FML(Form Markup Language)<sup>[7]</sup>在现存的HTML窗体中增加一些新的模块,如可复用模板,可以产生程序性的窗体。这些技术在某些方面解决了用户界面中存在的问题,但是都没有给出一个彻底的解决方案<sup>[8]</sup>。

在同一时期,W3C的HTML工作组(Working Group, WG)制定出了“研制一种更好地提供与 workflow、数据库应用相匹配的窗体体系结构(Form architecture),增加越来越多的能在Web上使用的设备种类,构造基于XML的Web的工作宪章<sup>[9]</sup>”。1998年3月,HTML WG提出了下一代Web窗体——XForms的需求草案(Working Draft, WD),指出“XForms应被设计成清楚地区分窗体数据(form data)、逻辑(logic)和表示(presentation)”<sup>[10]</sup>。1999年,XFA、XFDL等纷纷被提交到W3C,作为WG的候选技术,它们的研制人员也参加到XForms的讨论制订中。2000年4月,第一个有关XForms数据模型的WD发布了<sup>[11]</sup>,同年,XForms WG也从HTML WG中独立出来。在XForms WG成立的一年中,其工作进展很大,根据WG的计划,其1.0的建议将很快推出。XForms的研究内容对于Web窗体的自动化构造具有深远的意义。

## 2 XForms 简介 1<sup>[12]</sup>

和以往的窗体技术相比, XForms 最显著的特征是: 将一个窗体的逻辑、表示和数据分离开来, 形成一个与平台和设备无关的、基于 XML 的抽象窗体定义。逻辑层包含窗体中各域的填充、多页、Form 次序等的控制管理以及合法性检查等。表示层描述窗体的表现, 即采用何种窗体控制部件, 响应何种事件; 这种表现是抽象的, 可以借助于不同的表现处理器形成在不同设备上的不同表现, 如桌面浏览器上的 XHTML 窗体、手机上的 WML 窗体、面向盲人用户的浏览器窗体等。数据层则是针对实际的数据源, 通过应用更丰富的模式 (schema) 或类型 (type), 为窗体增加更多的数据结构以及数据一致性的维护。

### 2.1 XForms 的基本组成

图 1 是 XForms 的基本组成。其中, XForms 模型 (XForms Model) 属于窗体的逻辑层, XForms 用户界面 (XForms User Interfaces) 属于表现层, 实例数据 (Instance Data) 属于数据层; 窗体上的界面数据通过 XForms 提交协议 (XForms Submit Protocol) 提交到后台的数据层中。

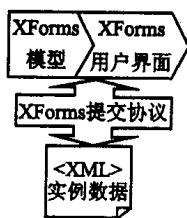


图 1 XForms 的基本组成

Fig. 1 Architecture of XForms

为使得 XForms 能在各种特定的平台、设备上工作, 必须提供专门的 XForms 处理器进行处理变换, 包括窗体与 Web 服务器之间、以及窗体与用户之间的各种交互。

2.1.1 XForms 模型 XForms 模型是与平台、设备无关的 XML 窗体定义, 它包含以下内容:

- 构成 XForms 模型的各个模型项 (model item);
- 模型项相关的实例数据结构和实例数据引用;
- 模型项的数据类型及该类型的可复用定义;
- 模型项自身的限制和约束, 如必需性、可见性、是否只读、合法性等;
- 模型项之间的依赖和约束关系

XForms 模型与以往窗体定义模型不同的是: 它除了提供实例数据的数据结构和类型描述外, 还引入了涉及窗体更为丰富的语义描述的扩展部分, 包括: 实例数据附加的约束、依赖关系、计算表达式、相关性等。这些扩展的组件丰富了窗体语义的描述手段, 可以取代以往窗体定义中所引入的大量脚本, 规范了语义的描述方法。

2.1.2 XForms 用户界面 XForms 用户界面定义窗体的抽

象表现 (abstract presentation), 它包括:

- 窗体中的用户接口控件 (widgets, controls) 的定义;
- 控件与 XForms 模型项之间的绑定机制;
- 控件之间的组织关系: 构造控件、单页、多页;
- 窗体的布局信息和显示风格描述;
- 控件的事件响应

2.1.3 XForms 实例数据 实例数据是指窗体收集的数据。通过使用实例数据, 可以支持工作流、自动填充、预先填充等窗体应用。XForms 模型与实例数据密切相关, 它描述了实例数据的结构。

2.1.4 XForms 提交协议 XForms 提交协议定义 XForms 如何发送和接收实例数据, 包括提交或恢复一个窗体的完成。传统的 HTML 窗体通过 HTTP、以单纯的 URL 编码名字/值对发送实例数据; 而 XForms 将允许更丰富的、国际化的、层次 XML 实例数据通过 XForms 提交协议传输。XForms 提交协议同样也是在 XForms 模型中定义。

由此可见, XForms 的窗体定义实际由 XForms 模型和 XForms 用户界面两部分组成。模型包括窗体自身的语义、涉及的实例数据的结构以及提交的协议的定义、描述; 而用户界面则是窗体的抽象表现。

### 2.2 一个 XForms 例子

```
<? xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1" EN>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:xform="http://www.w3.org/2001/08/xforms"
xmlns:pay="http://www.example.com/payment" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<head>
<xform:xform>
<xform:submitInfo action="http://example.com/submit" method="post" enctype="xml"/>
<xform:instance> <pay:paymentas="credit"> <pay:cc/>
</pay:payment</xform:instance>
<xform:model href="payschema.xsd">
<xform:bindings>
<xform:bind ref="pay:payment/cc" relevant="pay:payment/@as='credit'" required="true" type="xsd:integer"/>
</xform:bindings>
</xform:xform>
</head>
<body>
<xform:selectOne ref="pay:payment/@as">
<xform:caption> Select Payment Method</xform:caption>
<xform:choices>
<xform:item value="cash"> <xform:caption> Cash
</xform:caption> </xform:item>
<xform:item value="credit"> <xform:caption>
Credit</xform:caption> </xform:item>
</xform:choices>
</xform:selectOne>
<xform:input ref="pay:payment/cc"> <xform:caption>
Credit Card Number</xform:caption> </xform:input>
</body> </html>
```

图 2 一个 XForms 例子

Fig. 2 An example of XForms

本文中 XForms 定义引用反映 XForms 2001 年 8 月 28 日的工作草案的最新成果

图2是一个基于XHTML+XFoM s的窗体定义实例 整体的框架符合XHTML的定义,在< head>元素中包含XFoM s模型描述信息,而在< body>元素内则是XFoM s用户界面的描述信息 当前的XFoM s W G并没有提供一个专门的XFoM s根元素,而是致力于将XFoM s与HTML集成,其当前的工作核心在于XFoM s模型以及用户界面本身的规范化

从图2可看出, XFoM s模型由一个< xfoM>元素定义,其中包含< submitInfo>、< instance>、< model>和< bindings>四个子元素 < submitInfo>描述提交所采用的协议以及事件处理; < instance>描述该窗体的实例数据结构和初始的实例数据; < model>是窗体所需的可复用数据类型的定义,这里引用一个外部的Schema定义<sup>[4]</sup>文件payschema.xsd; < bindings>是窗体中各模型项的容器,每一个模型项通过一个< bind>子元素来描述,描述的内容包括该模型项的数据类型(type属性)、所绑定的实例数据(ref属性)、是否必需项(required属性)、与其它模型项如何关联(relevant属性)等 在< body>体中的部分是XFoM s用户界面描述,这里包含一个单选控件< selectOne>和一个输入控件< input>,它们通过ref属性绑定到相应的实例数据上 图2定义的是支付方法的选择窗体,当用户选择的支付方式是"credit"时,需进一步输入信用卡号

从这个例子中可看出,窗体的定义已从HTML的以界面为中心,转变为以数据为中心;对于数据,分别从其表现和语义两方面描述

### 2.3 XFoM s模型说明

如上所述, XFoM s模型由< xfoM>元素定义 在一个窗体定义实例中,可以包含多个< xfoM>元素表示多个子窗体的模型定义 每个子窗体包含逻辑上相互独立的实例数据以及作用于这些数据上的一些约束规则,这些约束规则体现了子窗体的功能

2.3.1 XFoM s数据类型 窗体语义的基础是其所含模型项的数据类型 在XFoM s中,引用了Schema中的所有内建类型<sup>[5]</sup>,包括其值空间(value space)、词法空间(lexical space)和所有的约束面(constraining facets);并且, XFoM s使用Schema定义语言<sup>[4]</sup>扩展子窗体所需的数据类型,这些类型定义收集在相应子窗体< xfoM>的< model>元素中,可以在子窗体中重复使用

2.3.2 XFoM s实例数据 < xfoM>的< instance>子元素描述了子窗体所需的实例数据的XML结构和初值,可以以内嵌的方式,也可以是外部URL引用XML结构中可引用任意名字空间中的tag;允许在其中出现空元素充当占位符,表示该数据将由用户提供或按一定规则从动态的数据源中获取;允许通过给出属性值或元素内容为相应的窗体控件提供初始值 这些实例数据的值会随着用户填充子窗体而被更新,最后被提交

目前的< instance>描述允许实例数据的初始来源是静态的或动态的,提供了这两种情况的实例数据XML结构的描述;但是没有提供动态数据源的来源描述

2.3.3 XFoM s模型项描述 这一部分是XFoM s模型的核心,对应于< xfoM>的< bindings>元素 其所含的各< bind>元素用来描述各模型项的约束信息,包括其绑定的实例数据(ref属性,值为XPath表达式)及其语义属性 这些语义属性定义了模型项所绑定的实例数据的以下特征:数据类型(type属性,其值为XML Schema内建的数据类型或在< model>中定义的类型)、只读性(readOnly属性,布尔表达式)、必需性(required属性,布尔表达式)、与其它实例数据的相关性(relevant属性,布尔表达式)、动态计算方法(calculate属性, XPath表达式)、合法性检查谓词(isValid属性,布尔表达式)、实例数据结点的最多/最少出现次数(maxOccurs/minOccurs属性). 其中readOnly, required, relevant, isValid属性是可以动态计算的,即对数据的约束与用户的当前输入密切相关 如图2中定义的一个模型项:

```
< xfoM: bind ref = "pay: payment/cc" relevant = "pay: payment/@as = 'credit'" required = "true" type = "xsd: integer"/>
表示绑定到实例数据"payment/cc"结点,当< payment>的属性as的值为"credit"时,其值是必需的,值类型为"xsd: integer".
```

### 2.4 XFoM s用户界面

XFoM s定义有自己的抽象控件

#### 2.4.1 XFoM s的原子控件

XFoM s的原子控件包括< input>、< textarea>、< secret>、< output>、< upload>、< range>、< submit>、< button>、< selectBoolean>、< selectOne>、< selectMany>等 其中,< input>、< textarea>、< secret>为输入控件,分别用来支持单行、多行以及密码的输入; < output>为输出控件,旨在将实例数据的值按需要的格式输出,它不提供数据的输入或改变; < upload>用于将本地文件系统中的文件上载到Web站点; < range>则允许从一个连续范围的值中选择一个值; < submit>用于将其绑定的全部或部分界面实例数据提交到Web站点; < button>用于触发动作; < selectBoolean>、< selectOne>、< selectMany>是选择控件,分别表示真/假或是/否的选择、多选一、多选多.

表1 XFoM s控件通用属性

Table 1 Common attributes of XFoM s controls	
属性	含义
xml:lang	为控件指定一种语言
D	定义控件 D
class	指明控件所属的CSS[16]显示风格类
navIndex	指明控件在foM上的浏览次序
accessKey	定义访问此控件的快捷键

这些控件并不指明其具体的表现方式 如多选一控件< selectOne>可以表现为一组radioBox、一个单选listBox、或者一个comboBox 这给浏览器的实现带来很大的灵活性

#### 2.4.2 控件的公共属性和子元素

XFoM s控件具有一些公共的属性和子元素

##### 1. 公共属性

公共属性可分为通用属性(表1)和实例数据结点绑定属性(表2)两类

XForms 控件的表现风格由 CSS<sup>[16]</sup> 属性控制; XForms WG 正与 CSSWG 合作, 由 CSSWG 开发一套缺省的适应于 XForms 的 CSS 样式单

表 2 XForms 控件绑定属性

属 性	含 义
ref/nodeset	选择一个数据结点/数据结点集合的 XPath 表达式
xform	指定相应的子 form, 此属性的值应为 <xform> 元素的 ID
bind	指定控件关联的 <bind> 元素, 此属性的值应为所指定 xform 的一个 <bind> 元素的 ID

数据结点绑定属性提供了一种将控件绑定到实例数据结点的方法 绑定方法有两种: 一种是通过 ref 或 nodeset 属性直接指定所绑定的实例数据结点; 另一种是通过 bind 属性指明关联的 <bind> 元素, <bind> 元素的 ref 属性值即为此控件所绑定的实例数据结点

### 2 公共子元素

公共子元素定义了一个控件的标题 <caption>、详细的帮助信息 <help>、帮助提示 <hint>、解释信息 <alert> (用于在控件所绑定的实例数据不合法时提示用户) 以及动作 <action>。它们的定义简化、规范了界面创作人员对界面的各种辅助提示及事件响应处理的开发

2.4.3 控件容器元素 在 XForms 中, 还提供了控件容器元素用于将多个控件包含进来, 表示它们之间的语义关系 这种做法有助于将一个连贯的用户界面发送到小的设备中 如, 若用户界面需要分裂成多屏, 则在一个控件容器中的控件将被表现在同一屏或页上

容器元素有 <group>、<switch> 和 <repeat> 三种 <group> 用于定义控件之间的层次关系 <switch> 是支持动态用户界面的条件构造, 其中可包含一个或多个 <case> 子元素; 每个 <case> 有属性 id, 它可出现在控件的事件处理中以激活或解除该 <case> 所对应的用户界面部分的活动 <repeat> 支持循环结构的定义, 如一个定单中的多个定单项; 它表示一个同质集合上的用户界面映射; 它一般绑定到诸如列表、集合等的数据结构上

2.4.4 控件的动作 <action> XForms 用 <action> 提供了一致的描述控件行为的规范, 并附加了一组通用的行为元素 结合 XHTML Events<sup>[17]</sup> 提供的事件绑定机制, 这些行为元素为窗体创作者提供了一种在 XForms 用户界面的适当位置详细说明事件处理的灵活手段

```

<xform: button>
  <xform: caption> Reset /xform: caption>
  <xform: action evt="dom-activate">
    <xform: resetInstance/>
  </xform: action>
</xform: button>

```

图 3 action 示例

Fig. 3 An example of <action>

一个控件的 <action> 定义包括要响应的事件和响应的行为两方面 前者通过 <action> 的属性 event 指定, 后者则

是通过若干行为子元素描述 图 3 为一个复位控件的定义, 其 <action> 响应的事件为 "dom-activate", 响应的行为是 <resetInstance>, 即恢复实例数据的值

XForms 中附加的一组行为元素包括:

- <dispatch>: 描述触发一个事件的行为;
- <refresh>: 描述刷新用户界面的行为;
- <recalculate>: 描述重新计算整个界面的实例值的  
行为;
- <revalidate>: 描述重新检查当前界面各元素的值的  
合法性的行为;
- <setFocus>: 描述设置界面的焦点行为;
- <setValue>: 描述设置实例数据项的当前值的  
行为;
- <submitInstance> 和 <resetInstance>: 分别描述  
需要提交或复位实例数据项的行为;
- <insert>, <delete>, <scroll>, <setRepeatCursor>  
>: 一组用来操纵与某 <repeat> 相关的用户界面, 包括插入、  
删除记录, 在记录间的滚动, 设置 <repeat> 的当前项等;
- <toggle>: 描述激活某 <switch> 中的某个 <case>  
对应的用户界面;
- <script>: 这与 HTML 中的类似, 即凡是不能由上面  
的行为元素描述的行为, 均可用该元素写脚本处理

### 2.5 XForms 的处理模型

XForms 除了要规范化窗体的定义外, 还应给软件厂商提供有关处理器构造的参考模型, 即 XForms 处理模型 该模型是处理器的组件、谓词行为和工作机制的标准化解释 处理模型包括核心数据结构的描述、事件、关键处理及算法 下面简要介绍当前草案中的处理模型方案

2.5.1 数据结构 处理器中为每一个子窗体维护一组读写属性 利用这些属性可以控制当实例数据发生变化时, 是否立即更新用户界面、是否触发一次合法性检查, 以及是否触发一次重新计算 此外, 还有反映 XForms 版本等的只读属性

整个处理器是以实例数据为核心的, 在草案中提到“实例数据的内容被复制在实例数据树中, 它基于在 XPath<sup>[19]</sup> 数据模型中定义的信息集 (infoSet) 映射”<sup>[12]</sup> 对于每一实例数据结点, 需要引入 "dirty" 标志反映其值的修改更新状态, 以便维护实例数据几种值 (界面表现值、客户端内部存档的实例数据值、后台的实例数据值) 的一致性

2.5.2 事件 XForms 使用 DOM 2 Events<sup>[18]</sup> 中定义的事件系统 它为子窗体 <xform> 引入了以下事件: xform-submit, xform-reset, xform-value-changing, xform-interactive-value-changing, xform-help, xform-hint, xform-alert, xform-construct, xform-destruct, xform-initialize, xform-instance-changed, xform-exception 此外, WG 目前沿用 XHTML Events<sup>[17]</sup> 的预定义的一般事件处理

2.5.3 处理及算法 草案给出了处理器涉及的主要处理算法的自然语言描述, 包括初始化/恢复算法、实例数据构造算法、操作次序算法、交互性算法、重计算次序算法、用户界面刷新算法、合法性检查算法、提交和复位算法等 这些算法的陈述还只是个初步的框架, 有待于 WG 的进一步细化

WD-xforms-20010828 在算法的陈述上有了突破,它引入了主依赖有向图(Master Dependency Directed Graph)和相关依赖子图(Pertinent Dependency Subgraph)用于重计算算法,并且给出了如何创建相关依赖子图的伪码算法。主依赖有向图以实例数据结点作为顶点,其中的弧反映通过解析实例数据结点的计算表达式(如 calculate, relevant, readOnly 或 required 属性)得到的实例数据结点之间的依赖关系。相关依赖子图反映当前的重计算所必须的顶点和弧。对于第一次重计算,即窗体的装载来说,相关依赖子图就是主依赖图;否则,它由自上次重计算起发生变化的实例数据结点集对应的主依赖图的顶点及其可达顶点和弧组成。这样,重计算算法依赖于两个图的创建算法,以及对相关依赖子图的拓扑排序。

除了讨论主要流程的处理算法,处理模型还要考虑实例数据的序列化格式,以支持更丰富的实例数据提交。

此外,在草案中还对 XForms 处理器、文档和创作工具规定了两种顺应的级别(conformance levels):基本的(basic)和完全的(full)。

### 3 与 XForms 有关的标准

从上面对 XFormsWD 的介绍中,可以看出 XFormsWG 不是孤军奋战,而是采取和其它 W3C 工作组合作的方式。它将自己的研究重心定位在窗体本身的语义和抽象表现的描述上,这样,通过协作可以在较短的时间内取得突破性的成果。下面列举与 XForms 有关的标准、相关性和发展状况。

- HTML WG: XForms 工作组将与 HTML 工作组密切合作,以确保向下一代 Web 窗体的过渡。如,当前 XForms 没有提供自己的根元素,而是使用 XHTML 的框架;在事件绑定机制上结合 XHTML 事件模块的处理。

- XML Schema<sup>[13-15]</sup>: XForms 的语义建立在 XML Schema 数据模型上。XML Schema WG 的目的是提供 XML 语义的一般解决方案,而 XForms 则集中在规范窗体的数据建模需求上。XML Schema 旨在替代 XML DTD,它被相对少的高技能的从业者所创建;而 XForms 则易学,创建简单。XForms 不会受限于 XML Schema 的文法,XFormsWG 会和 XML Schema WG 保持联系以确保对需求的有效相互理解,确保 XML Schema 为 XForms 数据建模需求的核心提供充分的依据。

2001年5月 XML Schema 建议的发布,将影响和推动其它 XML 标准的开发。其中的 Part 0<sup>[13]</sup>提供了对 XML Schema 的易读的描述,使用户能快速理解如何用 XML Schema 语言创建模式。Part 1<sup>[14]</sup>详细地说明了 XML Schema 定义语言,该语言提供了支持传统和面向对象(继承、多态、重载等)的简单和复杂类型定义、数据之间的同一性约束定义(Identity-constraint Definitions)、模型组的定义和使用、元素与属性的声明和使用等一系列组件;其重心是为组件的实现者提供一组实现的规范。Part 2<sup>[15]</sup>给出了可以用在 XML Schema 中的数据类型的定义,包括值空间、词法空间和约束面。

- DOM (Document Object Model)<sup>[20]</sup>: DOM 是平台、语言中立的文档结构的标准 API,可以使程序员容易地访问文

档中的组件,删除、编辑组件的内容、属性和风格,它使得程序员可以在所有浏览器、服务器、平台上编写应用。对 XForms 用户界面、实例数据和数据模型的访问依赖于 DOM。

1998年的 DOM Level 1 提供了对 XML 和 HTML 的支持;2000年的 Level 2 对 Level 1 扩展了对带名字空间(namespace)的 XML 的支持,增加了对 CSS、事件(用户接口事件和树操作事件)的支持,改进了树操作(树范围和遍历机制);Level 3 正在开发,将完成带名字空间的 XML 1.0 的支持(结合 XML Infoset,支持 XML Base),扩展用户接口事件(键盘),增加对抽象模式(如 DTD, XML Schema 等)和 XPath 的支持。

- CSS<sup>[16]</sup>和 XSL<sup>[2]</sup>: 分别由 CSS WG 和 XSL WG 制定,均属于 W3C 的风格活动(Style Activity)。XFormsWG 将结合 CSS 或 XSL 的表现解决方案,描述用户界面的表现风格。

1996年的 CSS1 关心诸如颜色、字体、背景图象等简单样式单功能;1998年的 CSS2 包含了处理基于页的布局,支持可下载的字体系列、定义矩形区显示不同的文档等特征;正在制定的 CSS3 将引入诸如多列布局等新特征以及规范的模块(modules)化,这有助于实现器的处理和设备 profiles 的创建。

XSL 被设计成对 XML 数据的变换,它应用一个样式单将一个文档变换到另一个;XSL T 1.0 建议已于 1999 年发布,而 XSL 规范当前还是一个候选建议。

- XPath<sup>[19]</sup>: XPath 是用于在一个 XML 文档内的寻址语言。XForms 的各种计算表达式和绑定表达式均采用 XPath 表达式表示。

1999年的 XPath 1.0 建议提供了结点或结点集的寻址方法,并提供了一些操作串、数字、布尔值的基本设施。但是随着 XML Schema 建议的逐步制定,如何寻址和变换基于 XML Schema 类型化的 XML 文档成为一个新的问题。2001年2月发布了 XPath 2.0<sup>[21]</sup>和 XSL T 2.0<sup>[22]</sup>的需求草案,标志着新一轮的标准制定开始了。

- XML Protocol<sup>[23]</sup>: XForms 的提交、挂起和恢复的协议由 W3C 的 XML Protocol 支持。W3C 的 XML Protocol 活动与 WG 于 2000 年创建,同年 5 月 W3C 接受了简单对象访问协议 SOAP 1.1 的提交。XML Protocol WG 正致力于开发基于如下内容的建议:

- 1) 包装要传输的 XML 数据的包层;
- 2) 和 IETF (Internet Engineering Task Force) 合作,开发操作系统中立的用于远程过程调用(RPC, Remote Procedure Call)应用的包层的内容;
- 3) 提供序列化基于 XML Schema 数据类型的数据的机制;
- 4) 和 IETF 合作,开发在 HTTP 传输层上的非独占机制。目前正在开发 SOAP 1.2,2001年10月发布了有关 SOAP 1.2 的消息框架和附件的 WD。

### 4 XForms 的实现

这里从两个角度讨论 XForms 的实现,一是支持 XForms 的浏览器,二是支持 XForms 处理器实现的 API。

#### 4.1 支持 XForms 的浏览器

由于 XForms 本身仍在开发,其 1.0 的建议还没有发布,因此并不存在商用的支持 XForms 的浏览器。目前正在努力将包括 XForms 等 XML 规范变成能在浏览器上直接浏览的。有芬兰赫尔辛基技术大学通讯软件与多媒体实验室开发的基于 Java 的 X-Styles 浏览器<sup>[24]</sup>,以及 Mozilla 技术组织开发的 XML WebAccess 2.0<sup>[25]</sup>。

4.1.1 X-Styles X-Styles 浏览器的主要优势是支持与 XML 有关的规范,适合在支持 Java 环境的嵌入式设备中使用。它不是一个 HTML 浏览器,尽管将来可以支持 XHTML。它是由赫尔辛基技术大学通讯软件与多媒体实验室开发的非盈利项目,最初在 1998-1999 年构思于一个学生软件项目,以后由实验室的工作人员开发。在 2001 年初, X-Styles 被作为开放源码发布。2001 年 8 月 31 日发布的 X-Styles 0.4 实现了同年 6 月份的 XForms 草案的一个子集,且 XForms 可以嵌入到 X-Styles 的 XSL FO (XSL Formatting Objects), SML (Synchronised Multimedia Integration Language) 和 SVG (Scalable Vector Graphics) 文档中。

4.1.2 XML WebAccess 2.0 Mozilla Technologies AG 是第一个创建将后端的 XML 数据取出送给用户的 XML 用户界面,是全球基于 XML 的用户界面软件的领袖,成立于 1998 年,是 W3C 的成员组织。在 2001 年秋发布的 XML WebAccess 包括服务器端的软件和许多通用开发环境的工具箱插件,它支持在现有浏览器上直接访问服务器侧的 XML-UI 文件,完成动态的交互。XML-UI 是一个基于 XForms 的窗体描述文件,其中窗体的模型部分遵循 XForms 的定义(包括模型项、实例数据以及提交信息等的描述),而用户界面部分则是用其 1999 年推出的窗体标记语言 (Forms Markup Language, FML)<sup>[7]</sup>定义的。而支持 XML-UI 访问的 Web 服务器必须装有服务器软件 XML WebAccess 2.0,它以插件形式安装在 Apache 或 IIS 的 Web 服务器上,它将 XML-UI 文件转换成 HTML + JavaScript 返回给客户端的浏览器。另外,其工具箱插件是指支持“所见即所得”的 XML-UI 的开发环境 XML WebFactory,它提供面向 ColdFusion, Dreamweaver, Adobe, FrontPage, XML Spy, Wordpad 等工具的 XML-UI 编辑插件。这些工具均是用 Java 实现的。

#### 4.2 支持 XForms 实现的 API

从前面对 XForms 介绍可以看出, XForms 处理器的实现依赖于对 XML 文档以及 Schema 文档的访问、分析、计算。当前的很多 XML 解析器 (Parser) 一般都能提供对 XML 文档的基于 DOM 或 XML 的简单应用编程接口 (SAX, Simple API for XML) 的访问。根据 DTD 的合法性验证。但是,由于 XForms 的语义描述是以 XML Schema 为基础的,因此需要这些解析器能提供支持 Schema 的验证和访问分析的 API。笔者对当今主流的两个平台——Java 和 Microsoft Visual Studio 对 XML 及 Schema 等的支持进行了调研。

Microsoft 公司的 MSXML 3.0 提供了对 schema 的支持,但是它所依据的规范是 XDR (XML Data Reduced)。XDR 基于 1998 年提交给 W3C 的 XML Data Note<sup>[61]</sup>和 XML 的文

档内容描述 (DCD, Document Content Description)<sup>[62]</sup>; 它的实现集中在 schema 的文法上,不支持继承及其它面向对象的设计特征。2001 年 7 月 21 日, Microsoft 发布了 MSXML Parser 4.0 Beta 2<sup>[26]</sup>。该版本提供了基于 SAX 和 DOM 的 XML Schema 的合法性验证; 并且它提供了 Schema 对象模型 (Schema Object Model, SOM) 接口。SOM 使得开发者可在内存中构造 XML Schema 文档,然后编译、验证 XML 文档是否违背该 schema。目前 MSXML 4.0 已对 XSD Schema 的大部分组件提供了内存访问的接口,这为 XForms 的实现创造了基础。

基于 Java 的 XML 工具包很多,主要有 JDOM<sup>[27]</sup>、Electric XML<sup>[28]</sup>、JAXP 1.1<sup>[29]</sup>、BM XML 4J<sup>[30]</sup>等。其中 2001 年 7 月 12 日发布的 BM XML 4J 3.2.1 提供了对 XML Schema 的支持。

#### 5 结束语

对 XForms 的跟踪调研源于笔者的研究项目“Web 网管界面的统一描述”,其目的旨在提供支持自动化构造的基于 XML 的界面描述。在对 XForms 近一年的跟踪和项目研究实践中,我们认为 XForms 的“以数据为中心,数据、逻辑和表现分离”的思想是值得肯定的,但是由于处在开发阶段,它存在以下不足:

1) 多子窗体特性: 在 XForms 中,子窗体的模型定义在 <xform> 中描述,而它们的用户界面则放在一起,容易造成混乱,不便于明确表现各子窗体的功能;且各子窗体有自己的实例数据,如何在子窗体中共享实例数据以及子窗体之间的关联性尚无描述。

2) 实例数据的动态初始化: XForms 允许使用空元素反映实例数据的结构,但是未提供实例数据的动态初始获取方法的描述。

3) 模型项之间的关系: XForms 将模型项之间的相关性通过模型项 <bind> 元素的 relevant 属性描述, relevant 的值为 XPath 表达式。这种描述功能较弱,无法直接描述当一个数据项改变的时候,如何改变和此数据项相关的其他数据项的状态(比如只读、隐藏等)。

4) XForms 文档之间相关性: Web 网站的各网页之间通常存在着一定的关联,它们之间往往需要传递一些参数,这些参数不能简单地视为某子窗体的实例数据,常常会影响整个网页的各种动态性质。

此外由于本着和其它工作组合作的思想,其界面的风格表现、计算表达式的描述能力、与服务器的交互协议等都存在一些问题。

针对这些不足,我们也尝试着提出了另一种基于 XForms 思想的窗体定义方案(另文叙述)。随着研究的不断深入细化,新的问题会不断的产生,但是下一代的窗体定义也将会不断地明朗。除了考虑窗体的描述模型外,有必要同时考虑其实现,因为在实践中会检验所定义模型的可行性、合理性和性能。我们应积极参与到 XForms 或其它类似的解决方案的开发与实现中,为规范 Web 窗体的创作、简化 Web 窗体软件的

## 编制与维护, 实现软件的自动化贡献一份力量

### References

- 1 Bray T, Paoli J, Sperberg-McQueen C M, Extensible Markup Language (XML) 1.0, W3C Recommendation: REC-xm1-19980210 [EB/OL] February 1998, <http://www.w3.org/TR/REC-xm1>
- 2 Clark J, XSL Transformations(XSLT) 1.0, W3C Recommendation: REC-xslt-19991116 [EB/OL] November 1999, <http://www.w3.org/TR/xslt>
- 3 Kristensen A. Form sheets and the XML forms language [J] Computer Networks, 1999, 31, 1189~ 1201.
- 4 Boyer J, Bray T, Gordon M, Extensible Forms Description Language (XFDL) 4.0, W3C Note: NOTE-XFDL-19980902, September 2 [EB/OL] 1998, <http://www.w3.org/TR/1998/NOTE-XFDL-19980902>
- 5 McKenzie G F. XFA-Template 1.0, JetForm [EB/OL] 1999, <http://www.w3.org/1999/05/XFA/xf-template.html>
- 6 Tardif M. XFA-FormCalc 1.0, JetForm [EB/OL] 1999, <http://www.w3.org/1999/05/XFA/xf-formcalc.html>
- 7 Schnitzenbaumer S, Wedel M, Gunatilake M, Dietl J, Greer C, XHTML-FML 1.0: Form markup language—XHTML modules for dynamic Web interfaces, Mosquito Technologies AG [EB/OL] 1999, [http://www.mosquito.org/sources/spec\\_xhtml-fml.html](http://www.mosquito.org/sources/spec_xhtml-fml.html)
- 8 Heaven S. Can XFom Transform the Web [J] IEEE Internet Computing, March-April 2000, 4(2): 103~ 106
- 9 <http://www.w3.org/MarkUp/2000/Charter>
- 10 Dubinko M, Schnitzenbaumer S, Wedel M, Raggett D, XFom s requirements [EB/OL] Working Draft, 1998, <http://www.w3.org/TR/xhtml1-fom s-req>
- 11 Dubinko M, Silvester S, Schnitzenbaumer S, Raggett D, XFom s 1.0: Data Model, Working Draft: WD-xfom s-data-model-20000406 [EB/OL] 2000, <http://www.w3.org/TR/xfom s-datamodel>
- 12 Dubinko M, Dietl J, Merrick R, Raggett D, Raman T V, Welsh L B, XFom s 1.0, Working Draft: WD-xfom s-20010828 [EB/OL] 2001, <http://www.w3.org/TR/xfom s>
- 13 Fallside D C, XML Schema Part 0: Primer, W3C Recommendation: REC-xmIschema-0-20010502 [EB/OL] 2001, <http://www.w3.org/TR/2001/REC-xmIschema-0/>.
- 14 Thompson H S, Beech D, Maloney M, Mendelsohn N, XML Schema Part 1: Structures, W3C Recommendation: REC-xmIschema-1-20010502 [EB/OL] 2001, <http://www.w3.org/TR/2001/REC-xmIschema-1/>.
- 15 Biron P V, Malhotra A, XML Schema Part 2: Datatypes, W3C Recommendation: REC-xmIschema-2-20010502 [EB/OL] 2001, <http://www.w3.org/TR/xmIschema-2/>.
- 16 Bos B, Lie H W, Lilley C, Jacobs I, Cascading Style Sheets, level 2 (CSS2) Specification [EB/OL] W3C Recommendation: REC-CSS2-19980512, 1998, <http://www.w3.org/TR/REC-CSS2>
- 17 McCarron S, Raman T V, XHTML Events - An updated events syntax for XHTML and friends, W3C Working Draft: WD-xhtml-events-20010608 [EB/OL] 2001, <http://www.w3.org/TR/xhtml-events>
- 18 Pixley T, Document Object Model (DOM) Level 2 Events Specification, W3C Recommendation: REC-DOM-Level-2-Events-20001113 [EB/OL] 2000, <http://www.w3.org/TR/DOM-Level-2-Events/>.
- 19 Clark J, DeRose S, XML Path Language (XPath) Version 1.0, W3C Recommendation: REC-xpath-19991116 [EB/OL] 1999, <http://www.w3.org/TR/xpath>
- 20 <http://www.w3.org/DOM/>.
- 21 Muench S, Scardina M, Fernandez M, XPath Requirements Version 2.0, W3C Working Draft: WD-xpath20req-20010214 [EB/OL] 2001, <http://www.w3.org/TR/xpath20req>
- 22 Muench S, Scardina M, XSLT Requirements Version 2.0, W3C Working Draft: WD-xslt20req-20010214, 2001 [EB/OL] <http://www.w3.org/TR/xslt20req>
- 23 <http://www.w3.org/2000/xp/> [EB/OL] 2001. 2
- 24 X-Smiles: an open xml-browser for exotic devices, Telecommunications software and multimedia laboratory [EB/OL] Helsinki University of Tehcnology, <http://www.x-smiles.org/>.
- 25 Mosquito Technologies AG [EB/OL] <http://www.mosquito.org/>.
- 26 MSXML 4.0, 2001. 7, (c) 2001 Microsoft corporation [EB/OL] <http://msdn.microsoft.com/downloads/default.asp?url=/downloads/sample.asp?url=/MSDN-FILES/027/001/677/msdncompositedoc.xml&frame=true>
- 27 <http://www.jdom.org/>.
- 28 <http://www.theminelectric.com/>.
- 29 <http://java.sun.com/xml/download.html>
- 30 <http://alphaworks.ibm.com/tech/xml4j>
- 31 Layman A, Jung E, Maler E, Thompson H S et al XML-Data, W3C Note: NOTE-XML-data-0105 [EB/OL] 1998, <http://www.w3.org/TR/1998/NOTE-XML-data/>.
- 32 Bray T, Frankston C, Malhotra A, Document Content Description for XML, Submission to W3C: NOTE-dcd-19980731 [EB/OL] 1998, <http://www.w3.org/TR/1998/NOTE-dcd-19980731>.