

Watch this: forms move centre stage

[Terrence A. Brooks](#)

The Information School
University of Washington
Seattle, WA 98195

From simple containers to fundamental structures

From static presentation: Since the original purpose of the HyperText Markup Language (HTML) was presentation, user interaction was limited to a small number of elements, such as buttons and text fields. The `<FORM/>` element was merely a container for various *controls* (e.g., checkboxes, radio buttons, etc.). Users modified the controls and then submitted the form for processing.

To foundations of interactivity: Increasing the sophistication of Web presentation usually means increasing its interactivity. Responsive Web content is generally considered as more interesting, or at least "stickier" (i.e., readers linger longer, thereby providing a strong economic incentive for increasing Web-site interactivity). Potentially, Web presentation can exhibit "intelligence" as Web servers analyse user input and choices, and then respond with customized content. These are the dynamics that shift focus to the `<FORM/>` element that hosts the Web widgets of interactivity.

By Autumn 2003, forms emerge as the dominant structural element for Web presentations, as well as the doorway of Extensible Markup Language (XML) information into organizations:

- Microsoft's .NET initiative evolves the Web page to the "Web form." Legacy Active Server Pages (ASP) script was mixed with HTML presentation code, and might include one or more forms. .NET Active Server Pages (ASPX) separates scripting code from HTML presentation. Script resides in "code-behind" files while the fronting HTML file subsumes everything to the `<FORM/>` tag.

Translation: Each Web page is structured as a single form. The `<FORM/>` tag becomes the dominant structural element of the Web.

- "Smart" forms check the validity of user input on the client machine, thus reducing load on server machines. Smart forms can reach out to resources on the Web to populate fields with appropriate data, and run scripts that modify subsequent information based on earlier input. As XML emerges as the fundamental technology of information exchange among applications, efficiency suggests that users submit information with forms that immediately structure information as XML.

Implication: Watch for the role played by Web services (see [Watch This: Web services](#)) in populating fields in smart forms.

Why is the evolution of forms significant?

Information can be structured and validated at the periphery of an organization. Incoming information is thereby already groomed and ready to be shredded (i.e., individual fields of the incoming XML document are broken out) and submitted to the appropriate applications (read: databases, spreadsheets, etc). Reward goes to practice that emphasizes re-use of information and reduces the inefficiencies of keyboarding information more than once.

Show me an example:

The following InfoPath form features rich controls such as a date picker, radio boxes and a text field for anecdotal input.



Terry's InfoPath Form

Favorite Ice Cream ☒ Vanilla ☐ Chocolate

Notes

I ate some delicious vanilla ice cream this afternoon.

Today's Date:



This form produces native XML ready for immediate processing (This example has been edited to reduce markup clutter).

```
<?xml version="1.0" encoding="UTF-8"?>
<my:field1>true</my:field1>
<my:field2>false</my:field2>
<my:field4>I ate some delicious vanilla ice cream this afternoon.</my:field4>
<my:field6>2003-11-10</my:field6>
</my:myFields>
```

What to do next: Process this file by using an [XPath](#) query to focus on the contents of "field4" and send it to the memo field of a database; use another XPath query to send "field6" to the date field of a database, etc.

Emerging Technologies and Applications

- [XForms 1.0](#), W3C Recommendation 14 October 2003

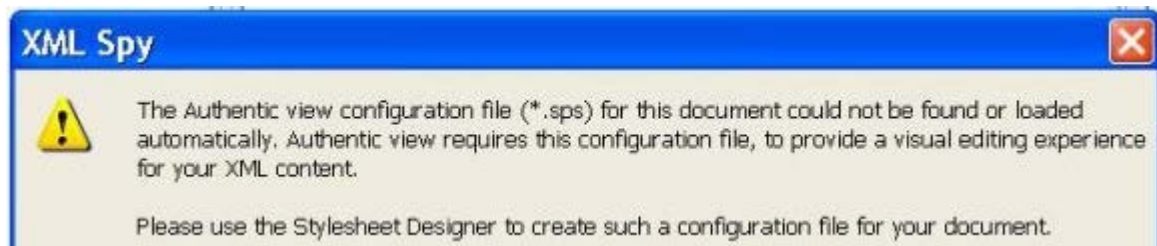
Provides the general description, motivation and strategy of using forms that split instance data from user interface markup. As described, XForms is not a separate technology, but meant to be incorporated into XML technologies such as XHTML (Extensible Hypertext Markup Language) or SVG (Scalable Vector Graphics) pages.

- Microsoft's [InfoPath](#)

InfoPath is Microsoft's version of the XForm recommendation expressed as a combination of word processor and forms processor. Peter Kelly in [Microsoft InfoPath 2003 Decision Tree: Comparing InfoPath Forms to Other Microsoft Solutions](#) situates InfoPath forms in relation to smart documents in Word 2003 and Excel 2003. In short, these standard applications now support XML schema mappings. This means that XML becomes the fundamental format for information exchange in the Office suite of applications. It also means that the smart document phenomenon will spread among Office applications.

- [Altova](#)

authentic 2004 is a free XML document editor. Its look and feel is closer to Altova's flagship product XML Spy rather than to non-technical office product. It requires the construction of a ".sps" file with the Stylesheet Designer that is a required purchase, as the following screenshot indicates.



Other Commercial (and more costly) Applications

- [Ektron control Pro + XML](#)

Advertises the ease for non-technical users to work with XML in a word processor-like interface. Cost around \$500 (US)

- [Corel XMetal](#)

Advertises a "tags-off" XML editing interface for users who prefer working in a simplified word processing environment. Cost around \$900 (US)

Last Word and A Caution:

The essence of XML is the separation of content from the contingencies of a particular presentation. Consider the following example created with the InfoPath form above. Any further processing of this XML would have to remove the and tags which were created in the word processor-like environment and embedded in the XML. (This fragment has not been edited to remove the clutter of markup.)

```
<my:field4>Grant and Brenda
<font xmlns="http://www.w3.org/1999/xhtml" size="4">
<strong>love</strong></font>
<font xmlns="http://www.w3.org/1999/xhtml" size="3">vanilla ice cream.</font></my:field4>
```

Note for the folks who don't appreciate the significance of this example: If you're passing XML back and forth between applications, you don't want one person's particular style choices to ride around with the content. *Oops!*

Date: December 2003

For further information:

- **XForms Essentials** by Micah Dubinko, O'Reilly, 2003.
Introduces XForms; final chapter considers InfoPath.
- **Putting XML to Work in the Library: Tools for Improving Access and Management** by Dick R. Miller

and Kevin S. Clarke, American Library Association, 2004.

"Once a final standard is reached, all Web browsers will probably support XML-based Web forms. If this happens, XML will simplify the task of processing patron data submitted from a Web interface." (p. 176)

- <http://www.w3.org/MarkUp/Forms/>
- <http://webservices.xml.com/pub/a/ws/2003/01/29/cocoon-xforms.html>
- http://www.idealliance.org/papers/xml02/dx_xml02/papers/04-04-02/04-04-02.html
- <http://www.xml.com/pub/a/2001/09/05/xforms.html>

How to cite this paper:

Brooks, T.A. (2003) "Watch this: forms move centre stage" *Information Research*, 9(2) TB0401

[Available at <http://InformationR.net/ir/9-2/TB0401.html>]

© the author, 2003 Updated: 12th December 2003

[Contents](#)

23784

[Web Counter](#)

[Home](#)