

An Office Document Mashup for Document-Centric Business Processes

John M. Boyer
IBM Victoria Software Lab
4396 West Saanich Road
Victoria, BC, Canada V8Z 3E9
boyerj@ca.ibm.com

ABSTRACT

An office document mashup called ‘Dual Forms’ is presented to demonstrate the feasibility and advantages of imbuing an office document with intelligent interaction capabilities, access to web services of a service-oriented architecture (SOA), digital signatures for legally binding contractual agreements, and a self-submission capability that allows the document to flow through a collaborative network or business process.

Categories and Subject Descriptors

I.7.2 [Document and Text Processing]: Document Preparation—*Markup languages, Standards*; H.1.2 [Information Systems]: User/Machine Systems—*Human information processing*; K.4.4 [Computers and Society]: Electronic Commerce—*security*

General Terms

Standardization, Languages, Security, Legal Aspects

Keywords

Office Document, Business Process, ODF, web service, SOA, XForms, XML Signature, user interaction

1. INTRODUCTION

Business contracts, progress reports, customer presentations, sales results, ... Office documents clearly play a pivotal role in businesses and enterprises. Yet there are many obstacles to a seamless business process involving office documents. Documents can be complex to complete correctly, and end-users may need to learn a new content management or document management system in order to access the documents and collaborate on their completion. Then, there’s the challenge of auditable regulatory compliance in heavily regulated industries.

Imagine an office document with wizard-like interactivity for stepwise error correction and overall completion. Imagine the office document could become an auditable transaction record for a legally binding agreement. Imagine it could facilitate its own return to the server-side business process server or enterprise content management system. An office document mashup called ‘Dual Forms’ is presented in order to convert these imaginings to a reality of intelligent mobile office documents.

Copyright is held by the author/owner(s).
DocEng’08, September 16–19, 2008, São Paulo, Brazil.
ACM 978-1-60558-081-4/08/09.

2. A DUAL DOCUMENT FORMAT

At the document level, a Dual Form relies on three markup languages. First, ODF [5] is used to present the core office document, which may include free-flowing multipage text and rich content items like pie charts and bar graphs to serve as visual aids. The ODF presentation may also include form controls that capture key pieces of fill-in-the-blank data, which are consolidated in an XForms model [2]. XForms dramatically increases the simplicity and maintainability of interactive web applications for structured data collection [4]. In [3], the first author posits that this is due to better alignment with the Web 2.0 application model and fully explores the extension to semi-structured content collection on the web based on ODF and XForms 1.1. As a first prototype to demonstrate the feasibility and advantages of this approach, Dual Forms was created to fuse an XForms-based ODF document with an XForms-based XFDL [1] document.

An XFDL document is a precision electronic form capable of wizard-like behavioral interfaces, digital signature security, and web mobility through full document submission. It also has a document attachment capability, so the XFDL document acts as a container for the ODF document. It shares data from the XForms model with the ODF document. A digital signature on the XFDL document covers the ODF document as well, and the pair of documents inherits the web mobility of the XFDL document. The Dual Forms prototype enables the two rich document run-time processors to operate together in one application.

3. A DUAL COMPOSITE APPLICATION

Figure 1 depicts the client-side usage pattern for a Dual Form. First, the XFDL form containing the ODF attachment is obtained from a web resource, an email or the desktop. The end-user experiences the step-by-step wizard to help guide them through providing the more structured data required by the overall office document application.

Based on a condition built into the XFDL, the user can transition to the ODF document to interact with form controls bound to the structured data as well as the free-flowing text. This allows the user to add unstructured content, not just data, such as specialized terms and conditions of a contract. When the end-user finishes with the ODF, a button press returns the end-user to the XFDL form view, where the wizard view advances to the digital signature step, if applicable, or to the full document submission step.

Once the user submits the Dual Form document back to the server-side, it could enter the next stage of a workflow, such as an approval, or it could be saved to a repository and its data could be used to initiate a business transaction.

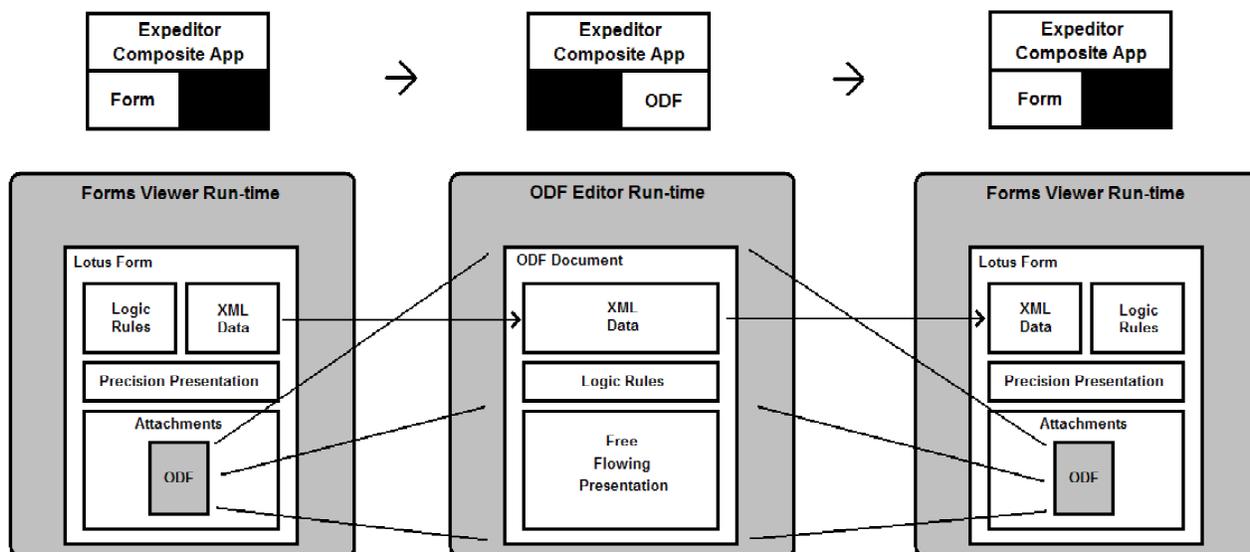


Figure 1: The client-side usage pattern for a Dual Form starts with the precision form experience. The user transitions to the ODF attachment for the office document fill experience, then back to the XFDL form to affix a digital signature and submit the document to the server.

At the run-time level, the above pattern was supported by using the Lotus Notes/Expeditor Composite Application Framework to create an office document mashup with two view components, one for the Lotus Forms XFDL viewer and the other for the Lotus Symphony ODF editors.

Figure 1 also depicts the mashup wiring between the view components. The XFDL to ODF transition request event bubbles up from the XFDL form viewer to the view component, which contains an event handler that uses the forms run-time API to obtain both the ODF attachment and the XML data from the XForms model. The wiring setter methods are called using first the ODF then the XML data.

The ODF view component listener on the receiving side of the mashup wiring first loads the ODF data, then it loads the XML data into the ODF document, then it brings ODF view to the front. Once the interaction with the ODF document is completed, the user presses a ‘Return to form’ button in the ODF view component toolbar, which causes the reverse wiring to be activated. The updated XML data and ODF content are obtained via the UNO API and fed to the setter methods of the ODF-to-XFDL wiring. The XFDL view component’s wiring handler receives the ODF and XML data, and it uses the XFDL forms run-time API to update the ODF attachment and the XML data in the XForms model of the XFDL form. This allows any subsequent digital signature or server submission to operate on the latest information from the dualized fill experience.

4. CONCLUSION

An office document mashup called ‘Dual Forms’ was presented to illustrate the feasibility and advantages of imbuing office documents with rich interactivity and web service access, digital signature security, and full document mobility within business processes. These advantages not only enhance the user experience and simplify application design and deployment, but they also mitigate vendor lock-in to specific document or content management server software.

5. ACKNOWLEDGMENTS

The authors are grateful for the Biztech Extreme Blue grant funding from IBM China Software Development Lab.

6. ADDITIONAL AUTHORS

Eric Dunn, IBM Victoria Software Lab; Maureen Kraft, IBM Lotus Development, Westford, MA; Jun Liu, IBM China Software Development Lab, Shanghai; Mihir Shah, IBM India Software Lab, Pune; He Feng Su, IBM China Software Development Lab, Beijing; Saurabh Tiwari, IBM India Software Lab, Gurgaon.

7. REFERENCES

- [1] John M. Boyer. Enterprise-level Web Form Applications with XForms and XFDL. *Proceedings of the XML 2005 Conference and Exposition*, November 14-18, 2005. Atlanta, GA, USA. Available at: http://www.idealliance.org/proceedings/xml05/ship/74/XFormsAndXFDL_Boyer.HTML
- [2] John M. Boyer. (ed.) *XForms 1.1*. W3C Candidate Recommendation, November 29, 2007. Available at: <http://www.w3.org/TR/2007/CR-xforms11-20071129/>
- [3] John M. Boyer. Interactive Office Documents: A New Face for Web 2.0 Applications. *Proceedings of the 2008 ACM Symposium on Document Engineering*, Sept. 16-19, 2008. San Paulo, Brazil.
- [4] Richard Cardone, Danny Soroker, and Alpana Tiwari. Using XForms to Simplify Web Programming. *Proceedings of the 14th World Wide Web Conference*, pp. 215-224, May 10-14, 2005. Chiba, Japan.
- [5] Patrick Durusau, Michael Brauer, and Lars Opperman (eds.). *Open Document Format for Office Applications (OpenDocument) v1.1*. Feb. 1, 2007. Available at: <http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.odt>