

Leading Edge

Projecting InfoPath

A Form-Focused, Information-Gathering Tool for Smart Client Applications

By Peter O'Kelly

Let's say you're a leading-edge Microsoft customer who wants to build applications that offer:

- A rich information worker user experience, with familiar tools for spell-check, AutoCorrect, and other Office features.
- Deep support for industry standards and clean separation among structure (XML Schema), presentation (Cascading Style Sheets and XML Transformations), behavior (declarative and programmatic control), and content (XML Document Object Model).
- Strong synergy with other Microsoft products including Outlook, SharePoint, and BizTalk.
- Effective use of PC resources for logic and storage, reducing server round-trips and facilitating off-line application usage.
- A very powerful means for developers who seek to take full advantage of information resources and enterprise applications, and to make the most effective use of information captured and managed in Office applications.

If you read my previous *Smart Solutions* column on Office 2003's support for [smarter documents](#), you hopefully believe Office 2003 (especially Word and Excel 2003) is central to Microsoft's strategy for this type of application – and you're right. For forms-focused, information-gathering applications, however, and in some ways confusingly, Microsoft has also introduced InfoPath, a form-centric information-gathering application development tool and run-time environment.

InfoPath has been the source of a great deal of controversy since its introduction during October, 2002 (when it was code-named "XDocs"). Some have speculated about its potential as an Adobe Portable Document Format (PDF) killer, since InfoPath is focused on forms-centric application contexts, and since Microsoft would surely like to see PDF die (for many reasons). Others have described InfoPath's potential to displace IBM Lotus Notes/Domino for certain collaborative application types, since InfoPath has strong integration with Outlook, SharePoint, and BizTalk for messaging, collaboration, and workflow.

A few people, tracing InfoPath's lineage (starting as a project code-named "Netdocs" during the late 1990s), have observed that the product designers' initial goal was to build an entirely Internet standards-based Microsoft Office alternative, albeit indirectly, as a Microsoft response to then-expected Internet-based Office competitors, during the peak of the "Internet bubble". Unsurprisingly, when the born-on-the-Web would-be Office competitors such as Desktop.com failed, Microsoft stopped talking about Netdocs and its "universal canvas" potential, but key facets of the technology continued to evolve, and Netdocs was ultimately revised, refocused, repositioned, rechristened, and reintroduced as InfoPath.

In an attempt to dispel some of the market confusion about InfoPath, the rest of this column includes an overview of InfoPath's features, some enigmatic dimensions, and some projections about what Microsoft will likely do next with InfoPath.

What's Not to Like?

InfoPath is a forms-oriented product that exemplifies best practices in XML and XML Web services. It's actually two products in one; it includes a form designer tool for application developers and "power users" (people who are comfortable with Office macros, for instance) and an information worker-focused form-filler environment. See Figure 1 for an example of InfoPath's form designer (and <http://www.microsoft.com/office/preview/infopath/> for more screen shots and detailed product information).

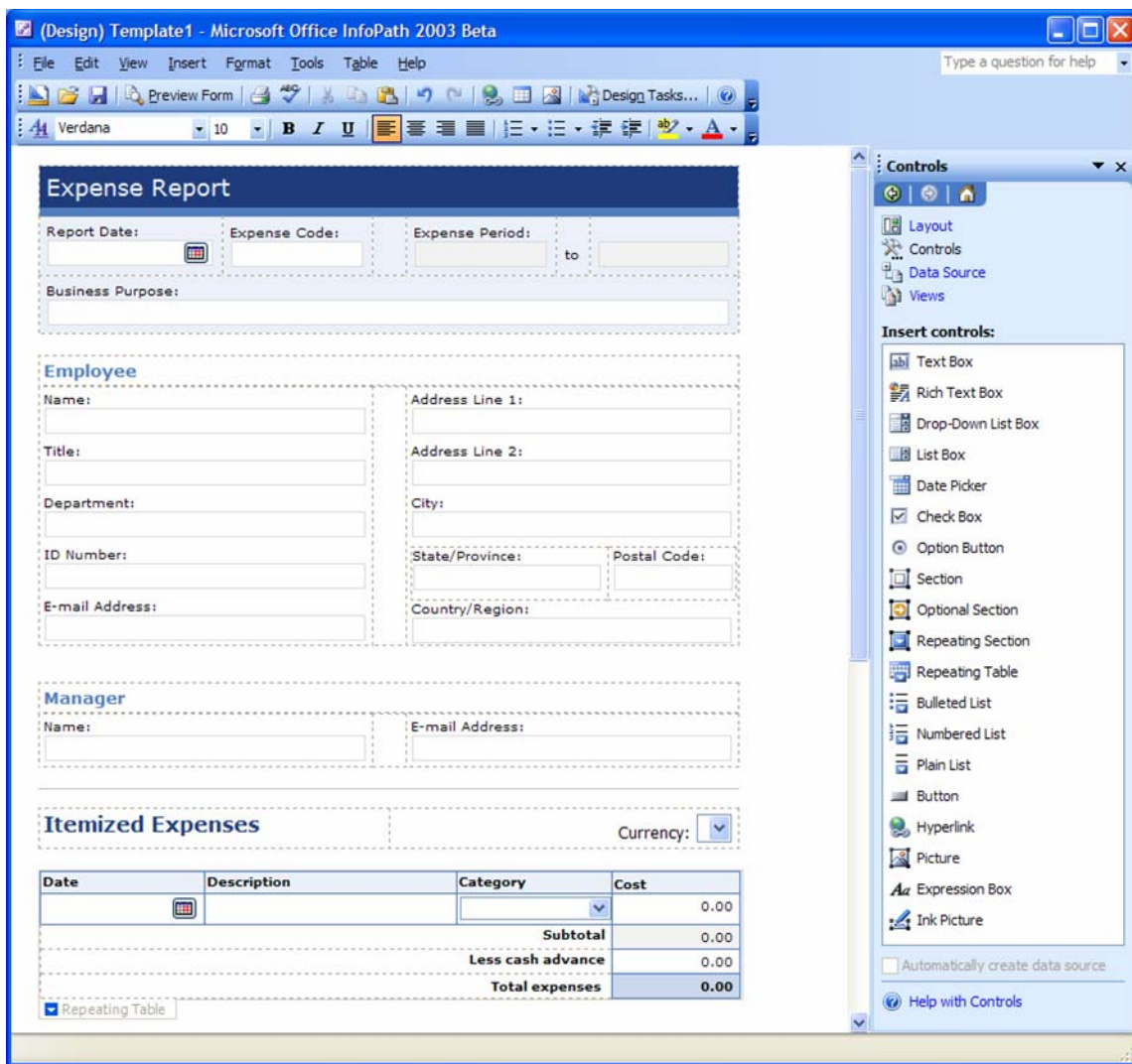


FIGURE 1: InfoPath's form designer editing a form based on the expense report sample included with the product.

More than twenty sample application templates are included with InfoPath, including expense reports, employee evaluations, and other business-focused form scenarios. InfoPath will be included with Microsoft Office 2003 Professional, Enterprise Edition and will also be available as a stand-alone product (stand-alone pricing wasn't yet announced when this was written). It requires Windows 2000 or Windows XP but does not require Office 2003. Several workflow and business process automation/management vendors are also planning to support InfoPath for client-side user interaction, since InfoPath's use of XML and Web services makes it easy to integrate with both Microsoft and non-Microsoft server and services-centric offerings (see [sidebar reference] for one Microsoft partner's perspective on InfoPath).

InfoPath has strong synergy with other elements of the Office System as well as the Windows Server System. Some examples:

- Many familiar Office features are used in InfoPath, including the Office spell-check service, AutoCorrect, and task panes.
- Outlook integration facilitates simple forms routing.
- Windows SharePoint Services can be used for optimal forms creation (launching new template-based documents from a SharePoint site) and processing (e.g., to consolidate and summarize multi-form workflow instances), and SharePoint lists may be programmatically used within InfoPath forms.
- BizTalk can be used for InfoPath application server-side workflow/business process automation needs.

Overall, InfoPath offers a compelling alternative to browser-based forms-centric applications and an attractive option for power users and developers who don't want to climb the Visual Studio.NET learning curve or to create Office 2003 smart documents (which can also be much more daunting than InfoPath applications).

XML Best Practices

XML aficionados have found plenty to like about InfoPath, since its foundation builds on:

- Extensible Markup Language (XML) 1.0 Second Edition
- Namespaces in XML
- XML Path Language (XPath) 1.0
- XML Schema (XSD) 1.0 Part 1: Structures, and Part 2: Datatypes
- Extensible Stylesheet Language Transformations (XSLT) 1.0
- Extensible Hypertext Markup Language (XHTML) 1.0
- Cascading Style Sheets (CSS)
- Document Object Model (DOM) 1.0
- XML Digital Signatures (XML DSig)
- Simple Object Access Protocol (SOAP) 1.1
- Web Services Description Language (WSDL) 1.1
- Universal Description, Discovery, and Integration (UDDI) 1.0

(Source: "Microsoft Office Access 2003 Beta 1 Preview:")

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc11/html/odc_acov.asp

If you haven't made tracking the myriad XML abbreviations and acronyms one of your primary hobbies, the key parts of this XML laundry list address Web services (SOAP, WSDL, and UDDI) and a variety of tools that are helpful in distinguishing among content, structure, presentation, and behavior dimensions of XML data management. While the boundaries are imprecise, in general XML, XHTML, and DOM are for content; XSD and namespaces address structure; CSS and XSL are for presentation; XPath, XSLT, and XML DSig address different aspects of behavior including queries, transformations, and digital signatures.

InfoPath is especially attractive in conjunction with BizTalk, for industry verticals that have embraced XSD, including healthcare, government, insurance, and many others. Microsoft offers "accelerator" solutions (see <http://www.microsoft.com/biztalk/evaluation/accelerators/default.asp>) for many verticals, and the integration between InfoPath and Jupiter (the next generation of BizTalk) will expand over time.

Again, there's a lot to like about InfoPath, both in terms of capabilities and in its deep support for XML standards. There are also some enigmatic dimensions, however, that have created quite a bit of confusion since InfoPath was introduced.

Enigmatic Dimensions

The initial release of InfoPath is in some ways inconsistent with other related initiatives, both from Microsoft and in the industry standards domain. Some inconsistencies when compared to Office 2003 and Visual Studio.NET 2003:

- InfoPath does not include popular Office features/services such as grammar checking and smart tag technology. InfoPath's programmable custom task pane model differs from the smart document Document Actions task pane used in Word and Excel 2003. These features and omissions reflect Microsoft's focus and priorities for InfoPath; it will gain additional Office features and services over time. InfoPath also has very strong synergy with other Office tools; for example, information captured in InfoPath can be incorporated into more rich document-oriented Word documents, and Excel can be used to manage and present analytical information about InfoPath applications (e.g., charts depicting current workflow status).
- InfoPath and its new Office sibling OneNote are the only Office applications that don't adhere to the traditional File/Save... model; InfoPath users generally *save* partially-completed forms for later work and *submit* forms when they are complete. Users will probably find this more intuitive than the traditional Office model, but it may take some getting used to.
- Microsoft Access and FrontPage are also strong candidates for some of the types of applications InfoPath addresses. Access and FrontPage developers are probably somewhat perplexed about InfoPath's positioning today, but Microsoft will offer guidance to help developers determine which Office tool is most appropriate for their application requirements.
- InfoPath programmability is based on the Microsoft Script Editor (VBScript and JScript, either-or on a perform basis). VBA, however, is still the most popular programming model option for Word, Excel, and Outlook, and the Visual Studio Tools for Office transform Office into a set of programmable components that can be used within Visual Studio.NET. FrontPage is the only other Microsoft product that is focused on the Microsoft Script Editor for programmability. It's reasonable to expect that InfoPath's programming model will expand to incorporate more .NET support over time.

- InfoPath's dual modality (form designer/filler) includes some unusual integrated development environment (IDE) dimensions, e.g., it does not offer IDE IntelliSense and does unusual things with the Office task pane model, such as using it (instead of a floating toolbar) as a control type pallet.
- While InfoPath offers powerful wizard-based options for incorporating database (SQL Server or Access), XSD, and Web service resources into form designs, it uses ADO instead of the newer, .NET-based ADO.NET. ADO may be more familiar to Office-oriented developers, but Microsoft has done a very effective job during the last few years at explaining why ADO.NET is preferable to ADO for multi-tier applications.

InfoPath also differs from traditional forms-oriented products in several respects:

- InfoPath offers neither a stand-alone form-filler option (a version that doesn't include the designer mode) nor a browser-based client option (it does offer installation and application-level options for disabling the designer mode, however). Browsers and other XML clients can be used to read InfoPath forms, since their native file format is XML, but only InfoPath can be used to create or modify InfoPath forms.
- Significantly, since InfoPath is only available for PC versions of Windows, this also means InfoPath forms can't be used (except for read-only purposes) on non-PC device types such as PDAs and smart phones. This is another instance of priorities for InfoPath's initial release; Microsoft intends to address non-PC clients for InfoPath over time.
- InfoPath defers to Windows for permissions, i.e., it has no built-in support for encryption or password-protected forms. This and the lack of a freely distributable form-filler client largely limit InfoPath's applicability to Windows intranet-based contexts for which the client configuration can be controlled.
- InfoPath does not offer pixel-level precision, a feature often required for applications that replace paper-based forms. InfoPath's designers focused on flexible, dynamic form features rather than screen or printing precision, at least for its initial release.

On the standards front, while InfoPath is a market-leading XML-based product, Microsoft apparently has no plans to either support the W3C XForms initiative or to propose an alternative to XForms, instead insisting that XSLT meets its InfoPath needs. Given the potential of XForms (or its successor; it's not yet clear if XForms will be a successful standards initiative) to extend browser clients into a wider range of "smart client" application contexts, with potentially very negative implications for Office's application client role, it's not surprising that Microsoft isn't an enthusiastic XForms supporter.

A Systemic and Systematic Competitor

In terms of feature/function capabilities and target applications, InfoPath has some competitive overlap with:

- Traditional forms-oriented products such as Adobe Acrobat (which in turn incorporates features from the eforms market-leading Accelio, originally known as Jetform and acquired by Adobe during 2002), Cardiff, and PureEdge. Microsoft doesn't intend to directly compete with these vendors, however, and instead positions InfoPath as the best choice for information-gathering-oriented, dynamic forms applications.
- XML content creation and management tools such as those from ArborText and Corel (which acquired SoftQuad and its XMetaL product in 2001). InfoPath doesn't address the same scope/complexity for structured XML documents, however.
- Web-oriented forms tools such as Macromedia Dreamweaver (for developers) and Contribute (for end users). InfoPath is more narrowly focused and does not offer a browser-based client option.
- IBM Lotus Notes/Domino and other form-based collaboration products. InfoPath can address some of the same target collaborative application types, but its lack of built-in encryption and security, and its reliance on Outlook, SharePoint, and BizTalk for forms processing, mean InfoPath is only part of the Microsoft solution for form-based collaborative applications.

While product evaluators and pundits focused on these market segments may find InfoPath lacking in many respects, they'd also be missing Microsoft's bigger-picture goal: Microsoft wants to compete with the Office System and the Windows Server System, not with a laundry list of individual tools, and in that context InfoPath makes perfect sense as a focused offering.

InfoPath is Microsoft's XML forms-based tool for information-gathering applications. It complements many other Microsoft products and, despite some overlap with Office 2003 smart documents and some short-term enigmatic dimensions, InfoPath is an attractive option for Microsoft customers seeking to leverage XML for forms-based applications. It's also an outstanding vehicle for Microsoft to gain insights into market requirements and trends for XML-based applications. In many ways InfoPath is a can't-lose proposition for Microsoft customers, as its deep

support for XML and Web services means applications and data created in InfoPath will be readily usable with other Microsoft and non-Microsoft tool and server/service offerings.

Peter O'Kelly (email: pbokelly@acm.org; blog: <http://pbokelly.blogspot.com/>) is an independent software industry analyst, consultant, and author with more than 20 years of experience in software product management and strategy, application development, and consulting.

Sidebar by Hank Barnes, Ultimus Vice President Marketing: <http://www.ultimus.com>; hbarnes@ultimus.com
Ultimus provides a complete business process management solution that automates both human workflow processes and process that involve enterprise applications (using XML, Web Services, and integration with Microsoft BizTalk Server). A key component of our solution is supporting the use of electronic forms to allow people to enter and view information related to their processes. Our approach in this area has been to support widely used form client technology, supplementing it with capabilities that we provide directly. To that end, we currently support the use of Adobe PDF forms as well as Web forms (generic HTML and component-enhanced), which can be developed independently or using a robust forms design capability provided in our product. We will also support InfoPath when it's released.

We decided to support InfoPath for three reasons. First, we believe that Microsoft's model of including InfoPath with Office 2003 for enterprise customers, as well as supporting it as a standalone module for earlier operating systems, provides it with the distribution muscle required to have it become widely used very quickly. Second, we believe that XML will become more important for forms technology over time, and by supporting InfoPath we get early exposure to a widely used XML product, experience that will enable us to more easily achieve our strategy of supporting any XML based form technology in the future. Finally, we think InfoPath has some great features that will be familiar and appealing to end user participants in automated business processes.

A key aspect of our solution is that we allow different technologies to be used at different points in a business process. This will enable you to mix and match interfaces based on your users' preferences, and to expand the user population to include Web clients (which are not addressed by InfoPath). For example, you may choose to automate a customer care process for your business. The process starts with a customer request. Since you can't require your customer to use InfoPath, you could use an Ultimus Web form for the initial submission. That information would then be routed to internal users, who might see the task in their SharePoint portal and then view the information and respond using InfoPath. This might occur for one or more steps, with the business process participants all using their preferred form client environments. Finally, once the customer request is completely addressed, the information could be fed into an Adobe template, producing a PDF document for your online knowledge base.

We're excited about the possibilities that InfoPath presents for Ultimus, our customers, and the market in general, and we look forward to using it in our continuing mission to help our customers automate all of their business processes.