# And Your Future BPMS Is? Microsoft Office

## *Reflections on Announcements from Microsoft for a Windows Workflow*

**Howard Smith**

If you are a *business* process owner, have you ever considered how hard it must be for your company's Chief Information Officer (CIO) to select, agree upon, and consistently apply, technical standards for the development of the information technology (IT) systems—the systems you rely upon for the efficient execution of the processes for which you are responsible? Java, J2EE, J2SE, NetBeans, Dot NET, ASP, C#, Web Services, XML, Application Servers, Eclipse, Visual Studio, Portals, Struts, NetBeans, Apache, Beehive, Cocoon, Geronimo, Jakarta, Perl, Tomcat … the IT acronyms just go on and on. A wrong turn can be very expensive indeed. Making sense of tech trends is a life long career. No wonder that many CIOs end up relying upon "off the shelf" business solutions from companies such as SAP and Oracle.

Packaged software solutions are always a compromise between the unique processes you need and the processes that the IT organization is practically able to deliver and maintain. While not the state-of-the-art in software development, commercial off-the-shelf (COTS) packages nevertheless provide a quick and easy starter pack of pre-packaged tools and methods, honed over years at many end-user sites. Using these packages, the CIO and the IT-shop is able to grow and manage a suite of enterprise applications that serve you, the process owner. They provide a stable environment in which software developers are able to build up their software development skills, reusable coding assets, and working practices. Yet even these popular products and the deeply entrenched and successful companies that supply them have to move with the times.

Technology is forever changing. Software development techniques that worked on older computers cannot take advantage of modern platforms. No matter how successful a software product line, it will inevitably need to be updated and moved to new technical platforms and computer operating systems. Eventually it will have to be replaced. Managing technology transition is both difficult, time consuming and resource intensive. No one wants it, but everyone ends up with it.

Who doubts the influence on IT development practices emerging from the open source, Web, and Java communities? Even the most successful of enterprise software vendors, SAP, is now making the switch to open standards. There's a buzz about SAP Netweaver[1]. Why? Netweaver is the *brand* under which SAP is conducting an extensive program of new development, providing its customers the tools and technologies to move SAP applications and SAP development teams to more modern techniques. Surprise: SAP's new focus is upon Java, XML, Web services and executable process modeling languages.

For SAP and similar firms, whose value in the market rests primarily upon the business processes and business logic supplied as software modules, migration to a new programming standard must feel like an immense distraction. End user organizations buy SAP software for business purposes. They seek solutions for finance management, human resources management, customer relationship management and supply chain management. They don't buy SAP for 'J2EE' and 'XML' any more than they buy for it SAP heritage programming language, 'ABAP'. A shift of platform provides no new business value per se. CIOs feel the same way. As they make similar transitions they are battered by technology trends. CIOs are no more interested in the syntax of new computer languages than other members of the CxO board. All that the CIO wants to do is to delight users at the desktop, enable business growth and demonstrate the positive value of IT, not just run a commodity. Yet CIOs and other tech-savvy executives know that if they *can* adopt new technologies life downstream will be improved.

[1] http://www.sap.com/solutions/netweaver

More capable technology, running on commonly available platforms, lowers IT procurement costs and opens the door to a wider community of developers and sourcing options. This all adds up to a reduced *process design to deployment time and resources cost.* A major plank of Microsoft's success has been based on this principle. The company replicates previously expensive "enterprise" technology and scales it so that it can be supported on commodity hardware (Intel) around which they then develop a community of developers. This approach is of great interest to CIOs, but it is not the only reason CIOs pursue consumerized and commoditized IT. CIOs also seek to avoid being caught in proprietary traps, traps that could later be exploited by a powerful incumbent software supplier. Smart CIOs therefore push down on the technology stack to create commodity service, while at the same time building the foundation for the next layer of business value to be built on top.

## Towards A Process-Oriented Development Approach

You are a *business* person. No doubt your eyes glass-over whenever someone mentions XML, Web Services, .NET, or Java. If so, think again, for it is precisely these modern IT platforms that will soon deliver, at commodity prices, what you have come to expect from enterprise-class business applications. Microsoft knows this, and that is why the firm has recently moved into areas previously only occupied by vendors such as SAP and Oracle. With the acquisition of application components to support finance, customer relationship, and supply chain, and working through a network of Microsoft solution partners, Microsoft is now, in part, in the same market currently occupied by SAP.

Microsoft built its early business success by supplying commodity operating systems, everyday software development tools, and personal-productivity apps. So successful have they been in this market that Microsoft has established de facto standards for commodity programming and applications development. Microsoft technologies such as .NET Framework, Visual Basic, Visual C++, and Visual C# are pervasive, with many times more developers than SAP has covering ABAP. Yet even Bill Gates does not control all of computing. Microsoft's programming languages compete with open source, Java and its communities, which are larger still. So even though companies like SAP and Oracle are immensely powerful in the collaborative enterprise applications space, they are required to interoperate with and support both of these mainstream platforms, .NET and J2EE. SAP does not have the luxury of betting on one platform of another; they must support both.[2] So what's the 'standard' for the CIO? Is it Microsoft's .NET, open J2EE, or proprietary SAP! To those with knowledge of the specific technical standards and tools the question looks naïve. In fact, it's a very real dilemma.

Companies originate, operate, and optimize business *processes.* Those processes depend critically on IT automation. Process owners cannot avoid IT. If you are building business processes, choices have to be made. Should the process be written in C#? Should it be written in Java? Should it just "come with the package?" The fact is that none of these approaches are ideal for *processes.* Software development is too costly, packages too standard and difficult to customize. And few computer languages were *designed* for processes. The most successful to date have been workflow systems. Now that idea has been partly re-invented, and partly evolved, creating the category of Business Process Management Systems (BPMS). The idea was a category of software built around a programming language designed for *processes.* In this language, sometimes called a Business Process Modeling Language (BPML), new processes could be developed and, unlike existing technology, also used to re-describe old processes (in existing systems) so as to expose and isolate business functionality from the intricacies of the underlying technical platforms, and reused to create new functionality.

What are companies like SAP and Microsoft doing to help process owners to get closer to the business process without the distraction of technology? While you may not have thought about it this way before, that is just what "BPM" is for IT vendors.

[2] http://www.sap.com/solutions/netweaver/technicaldetails/microsoftnet.epx & http://www.sap.com/solutions/netweaver/technicaldetails/java.epx

## What BPM Is

It was vendors of electronic forms, document management, and workflow who first realized that business people sorely needed a software application that *directly* reflected the unique design of the specific processes they wished to put in place using information technology. With a workflow engine and a workflow design tool, any flow of work could be brought to life in an IT system. Workflow systems enabled repetitive tasks to scale while remaining efficient, enabling business growth and maintaining consistency for process execution.

Software applications based on workflow consist of electronic forms, task routing and operations on key business documents. The idea had merit, but suffered from two deficiencies. Firstly, workflow vendors could not agree upon how to *represent* workflows in computer systems. Each workflow package could execute flows, but the tool could not read the flows and change them in response to changing circumstances. There was no standard computer-language for workflow. Secondly, each workflow vendor added proprietary scripting languages to extend workflow technology into areas occupied by traditional programming languages and data processing. Workflow was a partial solution, needing extensions before it could be used to create a business solution. As a result, workflow systems never crossed the chasm to become a mainstream programming paradigm. Workflow remained an application "feature," differing among competing workflow vendors.

It was the BPM community, specifically BPMI.org, who subsequently proposed and developed a standard language for all processes, encompassing both workflow and data processing. This innovation, the Business Process Modeling Language (BPML), provided a natural form of *process-oriented programming.* While its developers were clear that it would never be used for all programming tasks, the language was complete from a computational perspective and needed no extensions. The idea was that, using a BPMS that supported BPML, business people could specify all aspects of commonly needed processes, right down to the nitty-gritty details, without the need to dip into the technical plumbing. The approach decoupled the process from the IT solution. Technical matters were hidden in the BPMS, freeing business people to concentrate upon the unambiguous specification of the executable process.

BPML was innovative. Previous computer languages such as Visual Basic, C, Java and many older languages could only describe *single threads* of processing. Programmers using such languages who wished to create *processes* therefore had to 'call out' to operating system features such as "messaging," "threads," "semaphores," and "rendezvous" in any case where two or more programs needed to communicate. BPML, by contrast, had *concurrency* built into the language from the get-go, unifying computation and communications. BPML is therefore an ideal, and possibly unique, language in which to develop processes.

The BPM *technical* community's focus on BPML as a *programming* language was misunderstood by some BPM practitioners on the *business* side of the house, especially those who had never seen a BPMS in practice. They dismissed it as "just another programming language." Nevertheless, BPML took hold. BPEL is a similar idea. These languages will, as they evolve, be successful, because business buyers want to do better than re-design business processes and throw requirements over the wall to IT for their implementation. Without a language like BPML, IT systems cannot be easily and cost effectively kept in step with changing business processes.

## Business Process, The New It Application Logic

BPML was developed by a group of innovative technology start-ups and IT service industry giants, including Intalio and Computer Sciences Corporation. While new computer languages created by start-ups and IT service giants are interesting, it is more interesting still when those innovations are adopted by the giants of technology R&D, thereby signalling their significance. BPML was replicated in part by IBM and Microsoft who sought an XML document-syntax to describe the new distributed processing paradigm of Web Services. A "copycat" language, BPEL, was created, sometimes called BPEL for Web Services

(BPEL4WS) or WSBPEL. But not everyone who jumped onto the BPEL bandwagon had the same vision for BPEL as those who developed BPML. As a result, cracks began to appear in the BPEL development strategy. While the name BPEL is clearly the adopted market term, it is important to recognize the steps and stumbles it has taken, and the state of its development.

- First, key members of the BPEL support group noted that the language needed to be extended with programming features. Odd. They proposed BPEL-J[3], an extension to BPEL that, in the words of the authors, "allows each language to do that it does best". For example, the BPEL-J authors state that "calculating a value to be inserted into a document" or "calculating a value that will be used to affect the flow of control," should best be programmed in Java, and not BPEL. Not so. BPML was a complete language. Why can't that be the case for BPEL? It's to do with the standardization process. Some vendors held back from *allowing* BPEL to be complete because they worried about the cost of implementing a BPEL engine to match a more complete specification. BPML engines already existed.
- Second, some vendors supporting BPEL began privately adding new XML "tags" (programming features) to the language, in one case over fifty new tags. They did this in order to make up for shortfalls in BPEL and to differentiate their BPEL implementations, thus admitting that BPEL was, in quite a few ways, 'incomplete.' The implication was that BPEL could not easily, or efficiently, do what is required of a new computing language system. This created uncertainty about the base definition of BPEL, relegating it in the eyes of some analysts to a programming "feature" within distributed programming systems as opposed to a general purpose computer language. For many vendors, BPEL was indeed 'tacked into' existing solutions – for example, EAI (Enterprise Application Integration) solutions. As a result, BPEL "compliance" now means very little. Unlike a language such as Java, for which compliance is very clear, there are many potential use cases and scenarios for how BPEL can be used. While some vendors are making serious efforts to create BPEL engines, others claim compliance just by exposing existing functions that were there before BPEL existed. For those vendors, BPEL is little more than syntax.
- Third, BPEL did not provide a way to extend BPEL. Thus, some developers of BPEL engines felt the need to create a sub-language beneath BPEL just to allow their products to be more easily adjusted to any "official" BPEL variant that gained market acceptance. This was so that they could still continue to claim compliance with "the standard." After all, start-up companies don't have the resources to re-develop code in response to pre-emptive strikes by more powerful players. In short, some vendors ended up implementing a better, and more extensive, and flexible language, *inside* their products than the BPEL specification required.
- Fourth, extensions were proposed to BPEL that, in effect, would "hard-code" into BPEL features to support higher-level processes. One example is a workflow task allocation step. In one proposal, the authors state that "human user interactions are currently not covered by the Web Services Business Process Execution Language (WS-BPEL), which is primarily designed to support automated business processes based on Web services. In practice, however, many business process scenarios require user interaction."[4] They went on to state, "BPEL4People [a new specification] is defined in a way that it is layered on top of the BPEL language so that its features can be composed with the BPEL core features whenever needed. We envisage that additional BPEL extensions might be introduced that can use the BPEL4People extension introduced here." Sounds nice, but what does it mean for BPEL engines? If BPEL is forever to be extended in this way, how can one procure a known engine capable of supporting future extensions? Shouldn't higher-level processes be capable of being *programmed* (in BPEL) from lower-level primitives provided by the standard (BPEL) language and supported by a standard (BPEL) engine? BPEL4People and similar proposals effectively admit that BPEL is unable to express higher-level processes, with the implication that BPEL engines may be unable, in and of

[3] http://ftpna2.bea.com/pub/downloads/ws-bpelj.pdf
[4] http://www-128.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/

themselves, to execute higher-level processes without the addition of code to replicate features of, for example, a standard workflow product.

- Fifth, it appears as if BPEL does not allow for the programming of reusable sub-processes. Two supporters of BPEL, one originally supporting BPML, found that when they tried to create complex processes with BPEL it lacked support for this modularization and reuse. So they proposed another BPEL extension, BPEL Extension for Sub-Processes, or BPEL-SPE.[5] IBM, an author of BPEL-SPE, stated that the BPEL language currently "does not support the explicit definition of business process "fragments" that can be invoked from another (or the same) business process. The only way to approximate similar behavior today is by defining a complete business process as an independent service and invoking it using an <invoke> activity. That the invoked activity is really implemented as another process is completely hidden from the parent process; in other words, there is no chance to establish any coupling of process instance lifecycles." But wasn't that the whole point of a *process*? We can be sure that a proposed language (BPEL) was not developed to be a real computer language (e.g. Java) if something as simple as a sub-routine was forgotten in the original draft.

Developing a computer language by holding committee meetings over years with no clear end-goal is unlikely to yield a usable and implementable language. If BPEL is not a complete programming language—requiring extensions to do things normal languages already do and extensions of its run time system to cover commonly needed cases—how can BPEL possibly be positioned as a standard for BPM solutions and future IT procurement? It's all quite confusing to end users and doesn't bode well for the status of BPEL as a computer *language* in the grand sweep of the history of computing. Time will tell. Perhaps these deficiencies can be fixed in a cleaner, and hopefully simpler, BPEL 2.0.

With that in mind and with no prospect of a standard *executable process modeling* language in the near term (other than BPML) what is the future for process-oriented programming? The original idea behind BPMS has been somewhat lost in the BPEL standards process. While BPML exists in commercially available software, if you want BPEL and write that down in your software Request For Proposal (RFP), look again, for it might say BPEL on the box, but what the heck is BPEL! No one can supply for BPEL that which comes with any standard computer language—a manual. And the news just seems to get worse.

## BPEL 2.0 Is Delayed

OASIS, the committee and facilitation group shepherding the development of BPEL, has stated that BPEL version 2 is being delayed until the first half of 2006, while vendors continue sorting through approximately three dozen issues, whittled down from a list of outstanding BPEL issues that had totalled around two hundred and thirty[6].

While vendors press ahead and have implemented various *versions* of BPEL, a formal standard appears to be many months away, and *de facto implementations,* not carefully agreed standards, are likely to drive its development. One vendor stated, "There's blocking and tackling pretty much going on right now." More confusing, the OASIS group wish to add a human interaction component to BPEL, allowing, for example, a clerk to approve a loan in a banking application. Not only will such features *not* be in BPEL 2 (according to observers), but also it is not clear why these need be added in the first place. As mentioned above, if BPEL cannot be used to implement a simple human step, what kind of programming language is it?

It's easy to imagine the design of a task-step process, containing process *participants,* such as "task," "stack," "work-item," "business role," etc., providing the *logic* of how the task participant can be allocated from one worker role to another. This *pattern* can then be reused in any other process. This *logic* could be written in any language, even Java, and definitely including BPML. Indeed, BPML is ideal for this, since it deals with the parallel constructs and

[5] http://xml.coverpages.org/ni2005-10-13-a.html
[6] http://www.infoworld.com/article/05/10/28/HNbpel_1.html

interactions perfectly, providing a *process logic.* So why not using BPEL? Why add such new built-in tags for that which can be programmed? The fact that BPEL architects feel the need to add something to BPEL to make such a thing work indicates either that something very important is missing in BPEL or that the BPEL developers have a very different vision for BPMS than those that developed BPML.

Analyst comment about BPEL 2.0 plans is also confusing. One analyst stated that the lack of human workflow in BPEL 1.0 and predicted in BPEL 2.0 would be a "serious, fatal flaw." Not true. The *flaw* is an *inability to program those things, build a library, and enable reuse.*

## A Long And Winding Journey

As the story of BPEL illustrates, the journey towards a standard *process* language has been somewhat tortuous. Beginning in 2000 with the drafting of BPML 0.1, BPMI.org members voted positively for BPML 1.0 in 2003, only for this to be overturned by announcements around BPEL leading to the current elongated standards agreement process. This is a shame, because the theory of how BPML or BPEL can work, the pi calculus, is well accepted among theorists – little to argue over there. Yet the process for agreeing a *definition* of a process language, driven as it has been by various vendor interest groups and the resulting ad-hoc 'feature-creep' additions, has not yet delivered a simple, clean, complete, open and formalized specification. There is a saying that things should be as complex as they need to be, and no more complex. In the case of BPEL, it's been a case of "as simple as it can be to claim compliance with whatever we already can do, and we'll extend later."

What does all this mean? For the time being, "innovation" in the field of executable process modelling languages is likely to continue. We are not out of the woods yet and we cannot see clearly over the next hill. Thus, it was with great interest to many when Microsoft, who often pre-empts markets with de-facto standards, announced and released Microsoft Windows Workflow Foundation (WWF).[7]

## What's Microsoft Up To?

Microsoft spotted the significance of the pi calculus around the same time as others working in the open source communities, around 1999. They used it, as BPMI.org later did with BPML, as the inspiration for a new computer language, a language Microsoft dubbed XLANG (Xml LANGuage).

XLANG was built into Microsoft's BizTalk integration solution. Later, Microsoft "abandoned" XLANG to create BPEL with IBM, adding features from IBM's Web Services Flow Language (WSFL). WSFL was inspired by workflow. So if Microsoft now supports BPEL and has dropped XLANG, what's with the new Windows Workflow Foundation? Don't let the word "workflow" fool you.

Microsoft's Workflow Foundation is an extensible programming model (process) and runtime component (engine) for building solutions on the Windows platform. In other words, it's a new programming language. While the descriptions of WWF focus on workflow-enabled applications, WWF is more. It includes support for both system workflow and human workflow across a wide range of scenarios including: workflow within a line of business application, user interface page-flow, document-centric workflow, human workflow, composite workflow for service oriented applications, business rule driven workflow, and workflow for systems management. More significantly, Microsoft is building these workflow characteristics directly inside familiar MS Office desktop applications, bringing together document management, records management and Web content management for unified, integrated, scalable enterprise content management.

With WWF, Microsoft states that their goal is "to provide every information worker in an organization with a simple way to create, share, manage, archive, and find a diverse

[7] http://www.windowsworkflow.net/

spectrum of content, and to do that in a way that is extensible for developers." What Microsoft is describing is a *business process.* And they go on to state that with Windows Workflow "developers can build new kinds of solutions on top of Microsoft Office" and that "Office will provide out-of-the-box support for common workflow solutions such as document creation and management, content archival, records management, policy-based retention, and Web content management."

Microsoft has also stated in press releases that future releases of BizTalk "will use Windows Workflow Foundation to enhance support for integrating applications, trading partners and business process management." So the language inside WWF will also be the new language of BizTalk. A new book entitled *Presenting Windows Workflow Foundation*[8] is co-authored by Scott Woodgate, lead product manager for BizTalk Server at Microsoft corporate campus in Redmond. BizTalk 2004 already includes a technology called Human Workflow Services (HWS), which supports automated business processes that involve the routing of documents. It looks like HWS will be a part of BizTalk 2006 but eventually will be phased out in favor of WWF in BizTalk 2008, or whatever the successor of 2006 is dubbed.

In a Microsoft keynote address, executives discussed how Workflow Foundation would create "new opportunities for Office developers, including workflow solutions and beyond … to make it easy for developers to create and extend enterprise information [knowledge] worker solutions." The company went on to state that with Windows Workflow "developers today can use Extensible Markup Language (XML) and Web service interfaces to integrate Office programs with corporate applications and business processes."

It's fair to say that with Workflow Foundation, Microsoft has just become serious about workflow. Indeed, David Chappell, independent consultant and a writer for Microsoft, states, "Without the right foundation, writing workflows is hard."[9] That's an understatement. Without a programming language for processes, and an engine, it's darn nigh impossible. That's why Microsoft developers have struggled to include those features in Microsoft applications so far. Chappell correctly observes, "Meeting the needs of a long-running business process, supporting the dynamic behavior that people require, and handling the other challenges that workflows present all require more time and effort than most developers can invest." Spot on. That's why any company wishing to implement workflow doesn't try to "develop" it, but rather uses a commercial full-featured off-the-shelf workflow or BPM solution. After all, they don't want to develop the engine to drive workflows, they just want to create and deploy the flows of work. And that's what Microsoft has recognized: BPM. They are beginning with a programming model to mirror the capabilities of a BPMS. They will later supply a BPMS. End users want a BPMS, not another programming model.

## Usurping Third Parties

Chappell goes on to predict that, "After years of being relegated to specialized uses and applications, workflow is about to go mainstream" – by which he and Microsoft could mean something completely different. Isn't this more likely that they really mean this: After years of being supported by non-Microsoft products, Microsoft has recognized the value of workflow *approach* and so workflow will be built into Windows, given away for free, thereby eroding the business-case for best-of-breed workflow *solutions.*

It is always interesting to listen to the detail of announcements from Microsoft. As if to rub salt into the wounds for smaller Microsoft-based workflow vendors, Eric Rudder, senior vice president of Servers and Tools at Microsoft said, "Today, workflow solutions are ad-hoc, complex, and limited in use because developers must implement their own workflow engines … Windows Workflow Foundation makes workflow an integral Windows platform capability, promising to dramatically increase the availability and simplicity of workflow-enabled

[8] http://www.samspublishing.com/bookstore/product.asp?isbn=0672328488&rl=1
[9] Introducing Microsoft Windows Workflow Foundation: An Early Look, http://msdn.microsoft.com/windowsvista/building/workflow/

applications."[10] Isn't that the wrong way around? Isn't it rather that many applications embody ad-hoc, complete and limited workflows *not* because developers implement their own engines, but, rather, because they *don't include* (in the application) a best of breed workflow engine and *instead, try to hard-code* the processes in code! While Microsoft's intention to provide a workflow engine in Windows is entirely worthy and welcome, and will improve significantly the quality of Windows-based applications, let's not invert the logic.

## The Third Wave

During his talk, Rudder also announced a new set of tools designed to allow software makers to open up their programs to be customized by others. The software, known as Visual Studio Tools for Applications, is based on the company's existing Visual Studio suite. Rudder and other Microsoft executives pitched the tools as a way to enable software to be customized without placing the entire burden on the application writers.[11] This is a key BPM theme. As Rudder says, "We're overburdened, as developers, adding features to applications." Microsoft is now admitting that which many of us in the BPM community already knew so well, that to continue to rely on programmers to create processes is both counter-productive and unnecessary.

In our 2002 book, *Business Process Management: The Third Wave*, co-authored with Peter Fingar, we used an analogy from Walt Disney's *Snow White* to explain. This can be found on Page 1:

"In The Max Strategy, Dale Dauten told some interesting stories about Walt Disney, or "Uncle Walt," as he liked to be called. Now there was a man fizzing with intelligence. Someone once asked him his "secret" and he said this: "Do something so well that people will pay to see you do it again."

There's a scene in Snow White where Snow is standing beside a well. And she tells a flock of doves that it's a wishing well. She demonstrates, saying something like "I wish my prince would come." Then, we see her from the bottom of the well, right through the water. We watch her face, shimmering in the surface of the water, as drops of water fall into the well and create ripples moving out. Now imagine drawing a shimmering face reflected in water that's rippling out in circles. Imagine how hard that would be, especially since this was long before computer animation.

As workers and as consumers, both online and offline, each of us is enveloped by a myriad of business processes—the intricate, dynamic, ever-changing manifestations of the economic activity of companies. Whether we are disinterested, or actively engaged, in these processes, in large part determines the wealth of those who weave them. Companies are looking for secrets, skills and tools that will enable them to create and mesh together business processes that are so outstanding that customers will "pay to see them" time and time again.

Like Walt Disney, companies are not lacking in imagination, but unlike *The Walt Disney Company* in 1937 that could afford to employ a thousand animators, companies today cannot afford to be distracted by the labor-intensive animation process [i.e. software development and integration]. To create the compelling business processes they so desperately seek, companies are now looking for the business-process equivalent of Pixar's computer-assisted animation methods—the ones Disney now uses."

That's BPM.

## Commoditizing Workflow

Following Microsoft's announcements, analysts were quick to point out that there are "A mob of companies currently offering software for business process management and integration."[12] They were referring, of course, to members of IT industry associations such as the

[10] http://www.crmdirectory.com/content/anmviewer.asp?a=2237&z=1
[11] http://news.zdnet.co.uk/software/windows/0,39020396,39218352,00.htm
[12] http://www.pcworld.com/news/article/0,aid,122555,00.asp

Workflow Management Coalition (WfMC.org). The analysts went on to state, "While enterprise customers used to buy the software as a stand-alone product, now they increasingly are purchasing it as part of a suite of Web development infrastructure software offered by companies such as IBM, BEA Systems, and Oracle." Hurts doesn't it.

Taking its usual competitive tack, Microsoft is including WWF across the major applications of its Windows platform, so developers building applications on Windows, at some point in the future, will no longer need to use third-party software to provide business process integration and management. Following Rudder's speech, Microsoft Office senior vice-president Steven Sinofsky outlined new business process and content management features in Office 12, an update that is scheduled to ship in the second half of next year.

Microsoft is targeting workflow towards the environment that all people who *work in, and work with,* business processes already have: Office. And that's not a bad thing. And yes, you guessed it, according to Microsoft, "Support for Windows Workflow Foundation will be available for **free** as an add-on to Visual Studio 2005."

"We're making it possible for application developers and ISVs to put workflow within their apps," said Scott Woodgate, group product manager in the connected systems unit. "Certain high-end apps have been heavily invested with a workflow engine, but smaller [vendors] couldn't afford it. We're commoditizing workflow so any ISV can put workflow in their application. This isn't meant to compete with [other workflow-enabled apps]; it lets people build [workflow-driven] apps much more readily."[13] Yup, Microsoft are trying to relegate workflow and BPMS to platform status, and moving the action into the area of process development. That's very much in line with BPMI.org's ideas. We have often compared BPMS with databases, upon which applications like ERP are built.

## Tracking Documents Across Applications

Microsoft understands the potential of BPM. They also understand how challenging it is to achieve technically. Kirk Koenigsbauer, a general manager in Microsoft's Information [knowledge] Worker group, has said, "Microsoft's vision for enterprise content management means tracking a document, no matter what its file format, through every aspect of its life cycle in an enterprise." He acknowledged this will be a "huge task that Microsoft is still figuring out exactly how to tackle," as the company has, in the past, merely "dipped its toe" into this market. "For enterprise content management we're looking at the whole life cycle of that document," Koenigsbauer said. "We are building technology to facilitate that entire process—the entire life cycle of a random piece of content."

Other articles began to pick up on what Microsoft had announced. TheServerSide.net stated that "much in the same way that Windows Communication Foundation (Indigo) is the plumbing for communicating across applications, Windows Workflow Foundation is the infrastructure for creating workflows. Also like Indigo, it [WWF] executes in the process space of the application to allow flexibility and scalability. Workflows are defined either graphically using the new Workflow Designer, or in code by using the *System.Workflow* namespace. The graphical designer can also output an XML representation of the workflow which is then constituted into *System.Workflow* objects at runtime."[14] The article went on to explain, "Workflow components, called Activities, can then be dragged into the workflow from the tool palette. Each Activity is actually just an instance of a .NET class in the *System.Workflow* namespace. Activities include 'If Else' for comparisons and 'Parallel' to launch an asynchronous branch of a workflow. … Several Activities come by default with Windows Workflow Foundation which use their own representation, in this case XAML, but Microsoft is planning to release another set that will generate BPEL compliant activities. Developers and ISVs can build their own activities using Visual Studio 2005 and the custom Activity Wizard which generated .NET source code, either VB.NET or C#." Once again echoing BPM themes,

[13] http://www.sdtimes.com/article/story-20051015-03.html
[14] http://www.theserverside.net/news/thread.tss?thread_id=36547

Microsoft stated that an important feature of Workflow Foundation was its "ability to modify a running workflow at runtime. This allows workflows to change dynamically as new tasks are required." One analyst commented, "In some ways, Microsoft is changing the traditional definition of workflow. So what Microsoft describes as workflow isn't necessarily the same as companies now offering workflow products."[15] True. Microsoft is not really talking about workflow at all. The name Windows *Workflow* Foundation has been chosen to appeal to everyday users. It is not so much about the traditional functions of e-forms, document management and task lists. It is really a new lifecycle for content. It's BPM.

## Towards Executable Flowcharts

In presentations, I have often used the analogy of flowcharts coming to life as executable processes. Microsoft is adopting a similar analogy. In the Microsoft authored book on Windows Workflow Foundation, it says, "Inside classrooms around the planet students often receive their first exposure to computer program design through the concept of a flowchart. The *flowchart* provides a graphical model that is used to formalize the design of program structure—with squares, diamonds, and triangles representing the various steps in the flow such as activities and decisions. Although the flowchart model is an excellent learning tool, the flowchart is not directly represented in running software. The expense of maintaining a flowchart model in software from a CPU-cycles perspective means that the flowchart is nothing more than documentation. After the concepts of flowchart program design are mastered by students, the flowchart model is forgotten and programs are written directly in code."[16] The authors go on to state, "Although writing programs directly in code has been the main development paradigm for more than 25 years, and many millions of programs have been created, a pure coding approach is not without issues." Spot on again. If they can be made to execute, flowcharts as the metaphor for process logic would open up process development to a much larger number of developers who do not have the skills to learn code syntax and semantics. This was the inspiration for the development of BPMI.org's Business Process Modeling Notation (BPMN).

During early BPMI.org meetings, some members struggled with the idea of using flowcharts. The process modeling community felt uncomfortable with the need for formal semantics. At that time, process modeling was really process *drawing,* and few process diagrams could ever come to life as a system since the diagrams had no inner logic. BPM is changing that. A quick scan of Google Images for "flowchart" reveals the depth to which people still use flowcharts, all over the planet and in every walk of life. Microsoft knows this, and, hence, following the acquisition of Visio, a leading diagramming tool, Microsoft has the interface necessary to enrich Workflow Foundation to offer numerous diagrammatic idioms that will come to life as executable processes in future versions of Office.

## Workflow, Tomorrow's Application Logic

To explain what is happening here, SDTimes wrote to its audience, "Remember back in 1993, when people were discovering this "Internet" thing? To use it, you needed an add-on protocol called TCP/IP, which didn't come natively in Windows 3.1. Today, IP access is native to cell phones. It's just a part of our daily lives. Microsoft aims to do the same with workflow support planned for Windows Vista … which will provide a single technology for building workflow support in business process automation for both system and human workflow tasks. This could range from the movement of a document through an organization to the business rules for system management."[17] SDTimes also went on to point out the potential impact on workflow vendors. "A number of companies have built their own workflow systems, including Adobe Systems, Computer Associates, IBM, Novell and Oracle. But these are tied to the product. Microsoft wants to make workflow a part of its major product lines so anyone can use it, just as it took the one-time add-on TCP/IP protocol and made it a native part of Windows."

[15] http://www.microsoftmonitor.com/archives/010495.html
[16] http://www.informit.com/articles/article.asp?p=418010&seqNum=4&rl=1
[17] http://www.sdtimes.com/article/story-20051015-03.html)

## The Third Wave of Microsoft Office

In *Business Process Management: The Third Wave*, Peter Fingar and I wrote that "Business process management products are available from many vendors, in versions ranging from departmental workgroup solutions to enterprise-scale infrastructure—a range of solutions to meet all needs. It is possible that personal BPM tools, akin to the commodity databases [e.g. Access] that form part of commonly used office productivity suites, will emerge. Imagine a 'Process Office' suite, providing an integrated, process-centric approach to collaboration, computation, work management, process modeling and simulation."[18] When these words were written in 2002, we had little idea an answer would come as early as 2005, although we always suspected it would come from Microsoft.

BPM features have been available in enterprise-class products for some time. With the announcement of Workflow Foundation, Microsoft is moving those features to the desktop. If the tech-giant is serious in its intent, workflow will become an ubiquitous characteristic of the everyday-computing landscape as has occurred with word processing, spreadsheets, and databases. A *process tool* will have been added to Office and knowledge workers will be enriched once again.

From announcements, it looks as if Microsoft has "Got Process" and is not planning to "tack-on" workflow as an add-on to specific applications. This is good news. Rather, they appear to be referring to BPM as a *platform,* an integral component of the way all future content, intelligent documents, and applications will work. For *processes* are all of those things working together. If this can be achieved, and with BPML there is evidence it can, it will give independent workflow, BPM, and ERP vendors much to think about. The market for process solutions should expand for all. If successful, Microsoft will be able to place business process *control* directly into the hands of those who are best placed to use it – those who perform *work* and those tasked with *improving the work of others.*

The first BPM vendors showed how to give business users control of the processes around them. Microsoft will now do this in everyday office productivity packages. As a result, productivity will take on new meaning for Office users. Information and knowledge workers are not just interested in their personal productivity, but the collaborative productivity of the teams around them. Their work is not just *in* processes—executing tasks handed to them by a workflow engine—but *with* processes.

## When Office Is A BPMS

To what extent should independent workflow and BPM vendors be concerned about Microsoft Windows Workflow?

In *Business Process Management: The Third Wave,* we grew an analogy between the emergence of the BPMS and the development of the RDBMS. Only time will tell whether Microsoft's BPMS, MS Office, will be the BPMS-equivalent of Microsoft Access (a personal and workgroup tool), SQL Server (a departmental and in some cases enterprise tool), or Oracle Database 10g, or IBM DB2 (high end databases)?

If Microsoft can move *beyond* workflow as a programming tool, and offer an integrated BPMS, smaller players at least should be concerned. Yet it will take time for Microsoft to integrate this programming model into its products and build tools to manage the end-to-end lifecycle of content and process. In that time, expect further progress from BPM vendors. In parallel with those developments, Microsoft's new workflow engine will empower vendors old and new to build new solutions (processes), ever more easily. Some are already adjusting their product development plans and aim to take advantage of Microsoft's new workflow platform. Such vendors will, for a time, differentiate themselves on the basis of their upper-level toolsets. But with ever-greater integration of workflow into Visual Studio and other Microsoft toolsets, how

[18] *Business Process Management: The Third Wave,* 2002, MKPress, Appendix B, Page 234.

long will that last? Surely Microsoft will build on their new workflow technology, to provide tools for process discovery, design, deployment, operations, measurement, and optimization?

Just as application vendors gave up on the idea of writing their own database once good enough databases existed, won't the same vendors give up workflow in the same way? And let's not forget the impact on users, since that's very much Microsoft's focus. Just as numerical models mushroomed in the era of spreadsheet programs, so a myriad processes—the "intricate, dynamic, ever-changing manifestations of the economic activity of companies"—will mushroom in the era of Office BPMS.

Does Microsoft care about so-called BPM language standards, like BPEL? Did it care about the proprietary format of Word documents until the US government began to dictate document interchange requirements? On the other hand, does it care about the science of distributed processes? You bet.

In the *Epilog* to *Business Process Management: The Third Wave,* we pointed to current problems in IT practice, namely, "Over the last five years, delivering business applications has become much more complex, with layer upon layer of new infrastructure requirements and new features. While this has been good for IT industry players that sell new products for new layers, it is not necessarily so good for companies that use them as business tools. When complexity mounts and eventually becomes unmanageable, it's time for action." It now looks like Microsoft is one more company taking that action.

Microsoft has clearly adopted process centric thinking and is building the foundations for a BPMS. Its future name is likely to be Office. When will business users finally "Grok process?"[19] When, using Office, they create a new document and realize that its collaborative development is just a process.

---

**Howard Smith** is the Chief Technology Officer (Europe) of Computer Sciences Corporation (CSC) and co-author of *Business Process Management: The Third Wave* and *IT Doesn't Matter -- Business Processes Do.* He can be reached at hsmith23@csc.com

[19] Do You Grok Process? Howard Smith and Peter Fingar, BPTrends.com, April 2004

 