The Emergence of Intelligent Content

The evolution of open content standards
and their significance

"The future is already here – it's just not very evenly distributed."
William Gibson

Abstract

This paper traces the history of open content standards and related technologies in an effort to understand the nature and significance of intelligent content. What is illustrated is that a common thread runs through SGML, HTML, XML, Web 2.0, the Semantic Web, DITA, OOXML, EPUB and HTML5 and that the evolution of open content standards has enabled the emergence of intelligent content and with it new forms of organizational and technological agility.

This whitepaper has been prepared as a corollary to the presentation "Content Fusion: There's a Piece of Data Lodged in my Document" delivered at Intelligent Content 2009, Palm Springs California, January 30, 2009.

www.intelligentcontentconference.com

Joe Gollner, M.Phil.
Director
Gnostyx Research Inc. (www.gnostyx.com)
6 January 2009
Updated 26 September 2010 & 18 February 2013

INTRODUCTION

After decades of evolution, content technologies have arrived at a critical threshold. All of the pieces are now in place and it has become possible to genuinely talk about "intelligent content" – by which we mean content that expresses its *salient meaning* in a way that is openly accessible both to applications and to people. This is important because with intelligent content comes an entirely new class of solutions the emergence of which could not come at a better time. To fully appreciate the potential of intelligent content, we should first understand the evolution of open content standards that has led to its emergence. This is the focus of this paper.

KEY TERMINOLOGY

"If you wish to converse with me," Voltaire once said, "define your terms." Accordingly, we should start by defining, with some help from the Oxford English Dictionary (OED), how we will use some key terminology:

- **Content.** Related to the word "*contain*" (and therefore *container*), content is derived from the Medieval Latin term *contentum* (thing contained) or *contenta* (things contained). By definition, content refers to the *meaning* held within, and transported by, a *container*. Historically, the term *content* has been used to differentiate between the *contents* of something and its format, style, rendition, packaging, or delivery method. Seen from a different angle, content can be described as the persistent expression, in physical form, of our intended meaning that we exchange when we seek to inform others. In being persistent and physical, content can become subject to management and to being shared across a wide variety of spatial, media and temporal boundaries.

- **Context.** The OED defines context as the "circumstances that form the setting for an event, statement or idea and in terms of which it can be fully understood". It follows that the context of content is integral, even central, to the meaning and value of the content. The term context is based on the Latin *contextus*, *con*-"together" plus *texere*-"to weave" and this reminds us that context is very much determined by the relationships amongst various content artifacts. It further follows that contextual metadata is itself content, albeit with a special role, and that this content must also be openly amenable to management, interchange and processing.

- **Publish.** Although often used more broadly, “publish” specifically means to “prepare and issue material (such as a book or a piece of music) so as to make it *generally known*” and often “for public sale” (OED). Essentially meaning “to make public”, publishing refers to the process of producing the final *renditions* of the content, formatted predominantly but not exclusively for use *by people,* and of delivering these renditions to the intended audiences.

- **Document.** A document is an exchange artifact that provides information to someone, supplies evidence about an event or serves as an official record. Interestingly, the term document is derived ultimately from the Latin *docere*, to teach (OED). A document is the rendered and delivered form of communication that contains the *content* and, in its production and exchange, transforms that content from an *intended meaning* into an *authoritative transaction*. Documents become an integral part of business events and their persistence becomes a matter of process accountability. However, and this is important, once documents are deployed they become records and therefore often fall under formal constraints on how they can be revised if at all. It is at the layer of the underlying content that change occurs and managing these changes and how these changes are formally released as revised documents is very much the function of a content management system. It is also important to emphasize that documents can take many forms and this includes electronic renditions that are accessed in many different ways.

- **Intelligent Content.** Intelligence refers to the ability to acquire and apply knowledge (normally a quality attributed to people but not exclusively), or to a collection of information of value in a particular context (OED). Content can be considered intelligent when it expresses, in an open way, the *salient meaning* underlying a communication such that the data, information and knowledge being expressed can be easily accessed and effectively leveraged by both people and the software applications that support them. In practical terms, intelligent content is a persistent expression of meaning that has been encoded using an open standard and that can be efficiently processed by automated systems to facilitate its management, validation, discovery, and publication *given the full range of uses to which that content may potentially be put*. Intelligent content realizes is full potential value when it is published as an *authoritative document* and deployed effectively as part of a *business transaction*. It is important to highlight that today *smart documents* leverage the intelligence of the source content to enable application behaviour that is combined with published information to create *interactive work instruments* that facilitate streamlined business events.

DOCUMENTS AND COMPUTERS

In late 2001, at an XML World conference I was chairing in San Francisco, the closing keynote speaker was Dr. Charles Goldfarb, widely acknowledged as the inventor of the Standard Generalized Markup Language (SGML). With the subsequent rise of the Extensible Markup Language (XML), Dr. Goldfarb has been accorded less and less attention although this is unfortunate because many of his observations and insights remain just as relevant for XML as they had been for SGML. At this gathering, Dr. Goldfarb proved this point with a talk that highlighted the fact that, until the advent of computers, all business transactions, with which we are familiar, were performed using printed documents. Due to the limitations of early computing technology, applications were developed and deployed that would automate how selected documents were maintained: inventories, ledgers and personnel records. It could be said that software developers only tackled the simplest documents, those that exhibited the most consistent structural patterns, and that they built software that aggressively accelerated how these selected documents could be updated, exchanged and processed.

What this really meant was that the “data versus documents” dichotomy, which is frequently raised as a barrier that separates traditional information technology from the world of content management and publishing, is an entirely false one. Instructively, this fictional dichotomy has only appeared due to the limitations in traditional computing technology and specifically its inability to effectively store, manage and process the more complex patterns that constitute the vast majority of document content.

THE EMERGENCE OF SGML

The one area where computing was applied to documents in the early days was the automation of publishing processes. Typesetting systems were commonplace and each came with its own proprietary procedural markup language that effectively told a specific output device how to place and format the associated text and graphics. As early as the late 1960s, questions began to be asked about whether this was the right way to apply automation to documents. By the 1970s, efforts were well underway to produce, refine and deploy a generalized markup strategy that would, as its founding principle, seek to declare what content *was* instead of repeatedly defining how it was to be formatted. When this notion of descriptive markup was combined with formal rules governing how this markup should be declared and applied, and with validation routines that ensured document conformance with these rules, we arrived at what was established as an ISO standard in 1986 as the Standard Generalized Markup Language (SGML).

Although someone new to the world of XML could be forgiven for not knowing it, the real strengths of XML, such as they are, derive entirely from the innovations associated with SGML. In truth, XML amounts to a series of constraints applied to SGML in order to accommodate the limitations of early web technologies. Again, the limitations of technology have determined how well, or how completely, document content can be handled by automated applications.

Also easily forgotten, or perhaps intentionally hidden, is the inherent radicalism of SGML. One of the notable things about the SGML community was how diverse a range of backgrounds was reflected in its initial cadre of advocates and contributors. The majority of the most energetic champions hailed from fields far removed from computer science and it is possible, even irresistible, to see this group as the voice of the document creators, owners and users declaring, loudly, that their content was not to be locked into, constrained by, or diminished in its potential value because of, the format into which it was encoded. SGML can therefore be seen as a determined attempt to force computers to deal with content the way people actually create and use it. This attempt led, inexorably, to an elegant grammar for creating and applying markup languages that, despite its elegance or perhaps because of it, proved insurmountably challenging for application developers to work with.

Despite its complexities, SGML could be used to create simple markup languages that would, it turns out, achieve great success.

THE WEB REVOLUTION

It is impossible to overstate the importance of what we would rightly call the *web revolution*. The reasons for its success have been the subject of numerous inquiries but one of the contributing factors that is most frequently underrated was the tactile simplicity of the Hypertext Markup Language (HTML). While some at the time declared that everything was important about the web except HTML, which was dismissed as technically uninteresting, the real success of the web lay in the fact that not only was web content easily accessible anywhere in the world but absolutely anyone could create that content. The simplicity of HTML was completely central to the success of the web. And HTML was a simple markup language created using SGML.

For those who had eagerly thrown themselves into the deep end of SGML complexity, including yours truly, the lesson of the web revolution rang out loud and clear - *simplicity works*! It is a lesson that bears repeating at the start of each content management project meeting.

The use of SGML for the creation of HTML, and with it the web, seems to have been one of the most productive accidents in the history of technology. The CERN laboratory, where Sir Tim Berners-Lee was creating the initial prototype of the web, was also by chance one of the precious few computing environments with an operating SGML-based publishing system. Legend has it that the use of SGML to create the content markup language for this prototype was recommended partly as a political expedient. Using SGML would ensure that the publishing group would support the project as it moved ahead. And so it was that members of the CERN publishing group, themselves SGML practitioners, played a key role in framing the initial HTML Document Type Definition (DTD). The myriad of benefits that accrued to the fledgling web with this accidental deployment of SGML are fundamentally important even if they are also largely unacknowledged.

Despite the success of HTML, it was not long before there were questions arising about why web content needed to be so entirely bereft of semantic meaning and why different formatting needs could not be better accommodated. If a more intelligent markup format could be introduced into the web then new types of content could be supported and applications could be deployed to perform more useful services. But in the early days of the web, these rumblings were very much on the periphery because the main strength of HTML, that of simplicity, had allowed the world's content to flood online. The fact that HTML was a purely rendition-oriented language that did not care what it was delivering in fact proved liberating and this continues, to this day, to allow all manner of content to enter into global circulation.

THE RISE OF XML

By early 1996, the rumblings were growing louder about the need for a more intelligent content format to become part of the rapidly evolving web. It was understood that what was needed was an extensible markup language, one that would allow communities of application developers to design and apply markup languages that suited their individual needs. Whereas HTML provided one tag set, the future of the web lay in a universe of tag sets all implemented according to a common set of general rules. This led, once again, back to the door of SGML. The inevitability that the web would need to move beyond simple HTML was obvious within the SGML community where efforts had already been underway to provide an "SGML Lite" or an "SGML for the Web". It was this community that provided the substantive contributions that ultimately produced the Extensible Markup Language (XML), a simplified subset of SGML designed specifically for web applications.

The SGML community's James Clark was even responsible for the "acronym upgrade", from the four letter SGML (sometimes translated as *Sounds Good Maybe Later*) to the three letter XML.

Arguably the most important advocate for bringing content intelligence to the web was the great Yuri Rubinsky who should rightly be remembered as *the spiritual father of XML*. Within the SGML community, it was Yuri who responded most quickly, passionately, coherently and concretely to the appearance of the web and it was his voice, and actions, that pointed the way forward to a future where intelligent content would be universally available as the lingua franca of the web. However another accident of fate, this time an unpleasant one, saw Yuri pass away early in 1996, before he could see his vision of intelligent content on the web come to fruition – the first step towards which occurred later that same year with the release of the initial draft of the XML recommendation.

Among the key goals the community finalizing XML set for itself, when creating a trimmed-down version of SGML, was called the 'desperate PERL hacker test'. A programmer, it was declared, working in an unheated basement and without the benefit of doctoral studies in computer science, should be able to develop, in less than two weeks, a conforming XML parser and perhaps even a useful application. The target, then, was once again *simplicity*.

And XML was successful in achieving this objective, at least initially. But as had happened with the web, simplicity begot complexity and suddenly organizations found themselves able to share content with partners whose systems and processes, like their own, had been designed and developed with blissful disregard for any such possible integration. Many things, it became clear, would need to change in this new and more intelligent web. In reaction to this eventuality, the XML recommendation quickly became the family of XML recommendations as new capabilities were added feverishly to keep up with the demands being placed on XML. In fact, before long the family of XML recommendations began to look more like the large and boisterous XML family reunion of recommendations where the more seasoned standards mixed freely with the new and unproven.

One fruitful way of looking at XML is as the offspring of two quite different parents. On one side is the web, where simplicity above all else ruled as the governing spirit. On the other side is SGML, bringing as it does a formidable level of sophistication. I have been known to declare that XML still exhibits an unresolved tension between simplicity and sophistication and, to continue along this line of reasoning, that this is not a bad thing at all. The future of the web ultimately rests on finding a new balance between these two poles: one that encourages rapid development and another that offers a genuinely effective answer for those who need to implement more sophisticated content services.

THE REAL FOCUS OF XML

While it was somewhat peripheral to the interests of many who were contributing to its initial finalization, the real focus of XML was not actually directed towards *document content* at all. It was focused, instead, on resolving a number of persistent and interrelated problems obstructing the progress of software technology development both generally and on the web specifically. These problems basically boiled down to facilitating data interchange and integrating heterogeneous applications. This could be seen, from one perspective, as an odd turn of events because so many within the mainstream technology community were openly hostile to SGML, or any of its offspring, and did not consider any possible contribution from XML to be a serious candidate for solving these problems. But from another perspective, and one that was abundantly clear to those who saw a bright future for "web applications", XML could solve the raft of stubborn problems thwarting the integration of historically proprietary, incompatible, inflexible and monolithic software products. The openness inherent in SGML and now largely passed to XML could be again leveraged to allow data to be harvested from software silos and exchanged across platform boundaries that were previously insurmountable barriers to integration. Being able to pass data in an open, validating and processable format would give birth to a new generation of software tools, ones that could be deployed rapidly, changed quickly and even adapted on-the-fly to the individual needs of a given user.

In this way, XML became the darling of the technology community and the early naysayers quickly changed their stripes and joined the stampede. The adoption of XML in fact traced a trajectory that has effectively set the standard for hype-curve ascent. A number of factors contributed to this success. One was the fact that the global economy of the mid-1990s had arrived at a point where the historical approach to Electronic Data Interchange (EDI) had been proven to be unable to scale up to the demand being introduced as re-engineered supply chains looked to establish increasingly diverse and adaptable networks. Anyone who had battled through EDI implementations in the past knew full well that this dog was not going to hunt in the new web environment. The arrival of XML immediately provided an open adaptable format for encoding data that could be exchanged using a wide range of mechanisms and imported into a variety of applications.

This same process occurred around application integration with XML immediately being recognized as an extensible mechanism for encoding and exchanging application messages, regardless of what those messages might contain. Software itself has been irreversibly changed with this recognition and with it we entered the era of web services and service oriented architectures. The impact of XML on the technology market cannot be overstated even if the full potential is rarely seen in practice mostly because true openness was initially seen as a threat to many software business models. The emergence of new models, such as software as a service (SaaS), illustrates that XML can indeed foster significant business change and that the vendors who had found the openness of XML initially threatening now had more tangible reasons to lie awake at night.

It should be said that the major industry player who probably "got" XML and its significance most deeply and completely was none other than Microsoft. By around 1998, it was clear that the team from Redmond had made their second web revelation – the battle was no longer for the desktop, it was for the space between desktops, teams and corporations that mattered. Integration was now the goal with the internet providing the platform and XML providing the interchange medium. It was also clear that with XML would come a wave of technology innovation that many of Microsoft's aspiring competitors, such as Netscape, simply could not survive. So Microsoft had good reasons to back XML energetically and events indeed proved them right. Machiavelli would have been proud.

THE WEB 2.0 PHENOMENON

Web 2.0, as a moniker, began life as a nebulous umbrella concept under which a variety of loosely related phenomena could be grouped. While not really aspiring to anything resembling a definition, Web 2.0 has become a widely used term by which people refer to the emergence of the next generation of web applications, ones that exist exclusively within the web as their native environment and that enable the type of dynamic, ubiquitous social networking that has met with feverish popularity and universal adoption. The Web 2.0 phenomenon combines two levels of innovation, one social and another technical. On the social level, the mantra is *collaboration* whether in the personal, professional or project modes. On the technical level, the mantra is *simplicity* and the treatment of software applications less as canonical products than as continuously evolving services that respond dynamically to changing needs as determined through communal feedback. For now, it is the technical level that attracts our attention.

When we lift the hood to see what underlies the Web 2.0 phenomenon we find, to the surprise of some, a collection of very simple building blocks. These building blocks, we discover, are applications of XML or automation methods that leverage XML. The interoperability that has been the result of the core focus of XML innovation, with web services and such protocols as Really Simple Syndication (RSS), essentially supplies the infrastructure upon which the social aspects of the

Web 2.0 phenomenon were able to emerge. Once again, the combination of openness and simplicity enabled a surge of popular adoption and allowed this adoption to evolve rapidly in many directions.

Other Web 2.0 innovations are relevant to our inquiry as well. A new emphasis on dynamic collaboration, inclusive community building, and experimental content repurposing (mash-ups) are welcome additions to the repertoire of the author, illustrator, editor and content manager. These are behavioural patterns that content management industry desperately needed. The underlying technical simplicity of the Web 2.0 environment also manifests itself in a new generation of simplified user interface models that allow people to work intuitively and interactively on shared content. When it is considered that many of the previous content management tools confronted users with diabolical interfaces (as one of my customers memorably termed it), the possibility of using flexible, engaging web clients instead is welcome news indeed. And within these emergent Web 2.0 environments, we see a new fusion of document content and interactive application behaviour. The nature of the content emerging as a result of this fusion is quite novel with some content being clearly intended for people to consume, some content being intended to facilitate its discovery and navigation, and still other content included for guiding application components that enable dynamic downstream interactions. When manoeuvring around a content-rich social networking site, as the poet W.B. Yeats once asked of how one could *tell the dancer from the dance*, it can be difficult to distinguish the content from the network environment and application components through which it evolves.

THE SEMANTIC WEB

A subject of growing interest is the possibility of adding to the web a level of semantic precision and logical rigor that will enable applications to operate across the web with new levels of autonomy. While its adoption has been more gradual, the impact of the semantic web is already being felt in specific areas of web functionality. Interestingly, one of the areas of immediate utility is in harvesting and organizing the tsunami of content being spawned by the Web 2.0 phenomenon.

What is sometimes referred to as the *semantic wave* is somewhat different than the revolutions surveyed thus far in that simplicity is not a defining feature of the semantic web. On the contrary, the semantic web is currently distinguished by a sobering level of conceptual and technological sophistication. What can be foreseen however is the fact that the semantic web is pursuing a valid, even inevitable, goal and that it will just take one element of simplicity to set off a semantic web revolution. Simplicity, as it has been in the past, will act as the catalyst.

At this time however, the semantic web is advancing through two different mechanisms. Within highly specialized domains, leading-edge projects have been refining techniques for creating, managing, integrating and leveraging sophisticated conceptual models using open standards. Within the more mainstream marketplace, web portal projects that are focused on improving content discovery and navigation have been importing specific tools and techniques from the more specialized initiatives in order to implement practical mechanisms for specifying, managing, applying and leveraging metadata for content resources. In both of these cases, the semantic web amounts to the introduction of a descriptive layer of particularly ornate content the traversal of which facilitates the discovery, interpretation and use of the content resources that people (or applications) access and use. However specialized it may be, the semantic web remains a type of content around which there are emerging new applications that interact with this type of content to deliver innovative services.

It is again noteworthy that all of the developments surrounding the semantic web have been firmly grounded in standards and tools that are themselves based on XML. Sometimes this has been seen by the advocates of the semantic web as a necessary evil because the types of functionality that would ideally be enabled might be greatly accelerated if efforts could proceed within a neatly-controlled, proprietary environment. However, the value of rich semantic resources is effectively nil if they cannot be openly exchanged with others and this fact means that all investments in realizing the semantic web must find their way back into an XML-based expression if they are to be worthwhile.

The semantic web represents the future more so than the present but this does not diminish its importance. As many luminaries in the web community believe, it will be the realization of the semantic web that lifts the web itself to its full potential. For organizations with substantial investments in information technology, there is a practical interest in the evolution of the semantic web. It will be the arrival of mature semantic technologies that will make the vision of Service Oriented Architectures (SOA) a reality whereas the current implementations travelling under that title are, at best, pale facsimiles of the envisioned goal. The full SOA vision would see system functionality being assembled and tailored to suit new or changing circumstances in much the same way we now talk about assembling and tailoring content for publication. The dynamic responsiveness associated with these visions of just-in-time content and just-in-time functionality will only be fully realized when the environment itself can "understand" the situational context within which a request is made. This type of contextual awareness for automated components will then permit a dynamic matching of the situation with the right content and application resources and thereby deliver just-in-time services that exactly fit the user's need and that of the governing business process.

Only with this SOA vision being fully realized will the information technology infrastructures of organizations become a strategic advantage not just for executing business transactions but for continuously evolving new capabilities and realizing new efficiencies. If we are honest with ourselves, our current technology investments, including and perhaps especially those pertaining to content management, are rarely if ever a strategic advantage and more often than not these investments, once made, become encumbrances that suffocate, more than facilitate, change.

XML IN THE WILDERNESS

In the last couple of years, a number of opportunities have presented themselves for us to reflect back on XML's first ten years. One of the interesting themes to appear in several of these retrospectives has been the accentuation of the differences between XML and its predecessor SGML and the importance of the break between them. From the perspective of stakeholders interested in facilitating data interchange and application integration, the appearance of XML stands as an historic milestone. Similarly, the wider community of stakeholders interested in evolving web applications was well served by the simplification of SGML to form XML and this is graphically illustrated with the emergence, and success, of the Web 2.0 phenomenon. The question, however, can be legitimately asked as to whether or not the introduction of XML has been an unqualified benefit to those who are primarily interested in how we design, create, manage, and publish *complex content resources*.

The answer to this question is more mixed than many might expect. Certainly the broad adoption of XML to facilitate data interchange and application integration has provided several noteworthy benefits to those we might call the *content stakeholders*. Documents that directly depended upon the integration of traditional document content with selections from dynamic data sources, such as part inventories or customer profiles, enjoyed immediate improvements. The available infrastructure had evolved to make these types of integrations far simpler to implement and to maintain. The XML revolution was also a boon in that it meant that once document content was ready for publication the XML capabilities of the technology infrastructure could be leveraged to disseminate the published content to all of its possible targets. As XML support proliferated, it became progressively more feasible to channel content assets not only into personalized renditions, such as would be accessible on a portable device, but also into application environments through which people would perform their work. It should be emphasized that these changes represent significant gains for all content stakeholders. However, it should also be noted that the XML revolution also shifted the focus of attention away from some of the core concerns surrounding the design, creation, management and publishing of complex content and even introduced some new challenges.

One of the areas of greatest expectation for XML, at least amongst the SGML community, was that it would expand and enhance the variety of tools for use in designing, creating, managing and publishing content. Aside from the more general examples cited above, this expectation has gone largely unsatisfied within the ten years that followed the finalization of the XML recommendation in 1998. This is largely explained by the fact that the main focus of attention for the XML community fell elsewhere, as has already been addressed. Another contributing factor was that many of the more document-centric technology vendors that had arisen before XML were not successful in capitalizing on the new market and, by pursuing many ill-considered schemes, they frequently spent themselves into oblivion. Content stakeholders, as a result, have found themselves with fewer tools, not more.

It really needs to be said that some of the activities surrounding the XML recommendation, and specifically those that were primarily sponsored by the *technology stakeholders*, spawned innovations that, however meritorious for data interchange or application integration, often created obstacles to the design, creation, management and publishing of high quality documentation content. In effect, complexities have been progressively added to the palette of mainstream XML tools and these changes did not help to address core content challenges but rather they introduced markup overhead and processing complexities that complicated the life of authors and implementers alike. As perhaps the most tangible example, the general migration towards using *XML Schema* for the specification of document content rules has, despite all the associated good intentions, typically produced content models that are noticeably poorer than those that had been historically produced using DTDs. When it is considered that most DTDs, and associated applications for working with them, left a lot to be desired, this is a somewhat depressing observation. That the introduction of the additional capabilities and controls that came with XML Schema has prompted this decline is not too surprising. The primary failing amongst SGML DTDs and applications was self-defeating complexity that was more often than not unnecessary given the main business requirements that needed to be addressed. With XML Schema, solution designers seem to have indulged this proclivity towards over-engineering even more energetically than before with the consequence that the gap between the investments made and the returns realized has typically been widened and at times has reached comical extremes.

This story does end on a positive note. The period between 1996 and 2008 for XML has been unashamedly focused on addressing problems in the technology infrastructure. Like an ancient tale of a hero finding revelation in the desert, XML, as a markup language and as an aggregate of interrelated standards and technologies, has definitely benefitted from the years of use by technology stakeholders as a tool of interchange, integration and interaction. The newly tempered capabilities of XML are now available to other stakeholders and can be used to enhance the ways in which complex content is managed and delivered. This adventure in integration is a big part of the XML story and intelligent content, along the lines of how we have defined it, could only have become a reality now because XML has undergone this evolution and because XML has changed the fundamental nature of the technology infrastructure within which intelligent content has been able to evolve.

XML AND DOCUMENT CONTENT

The last few years have seen a renewed, and welcome, interest being directed towards improving how we design, create, manage and publish document content. When this renewed focus is added to the rapidly improving capabilities of the technology infrastructure, something new becomes visible on the horizon.

DARWIN INFORMATION TYPING ARCHITECTURE (DITA)

The Darwin Information Typing Architecture (DITA) has been the subject of significant interest recently and deservedly so. After decades as a vanguard implementer of markup technologies, with this extending back to before the ratification of SGML as an international standard, IBM launched an internal initiative to assemble and formalize the best practices that had proven to be repeatedly useful. The result of this effort was DITA which IBM then made available as a public standard managed under the auspices of OASIS (The Organization for the Advancement of Structured Information Standards, formerly SGML Open). It is only partly in jest that DITA can be defined as a collection of "SGML dirty tricks" which can be used to implement content creation, management and publishing environments that actually work and that are cost-effective to maintain. That DITA hails from the hard-fought experiences of IBM working with SGML is a very good thing and for a number of reasons. For one, it means that the DITA standard, unlike all too many "standards", is actually a distillation of practical experience and better still experience that has been seasoned over a significant period of time. For another, it means that at its heart DITA is built on some of the very good ideas that grew up within and around SGML and that were explicitly directed towards solving the real problems of designing, creating, managing and publishing complex content resources.

The central innovation of DITA, although one seen in various forms in other initiatives that were evolving in parallel and addressing similar requirements, is that of introducing an *extensibility framework*. Under this framework, content models and application logic can be specified as *specializations* with reference to a base model and default application behaviour. This was important because experience had shown, repeatedly, that content management and publishing solutions that do not offer this type of extensibility mechanism rapidly become unsustainable as even minor changes in requirements spawn complex development, testing and management tasks. When technology solutions must be deployed across multiple workgroups, or across multiple organizations, then an extensibility mechanism changes from being a prudent capability to being a life-saving necessity.

The success of DITA, at least insofar as measured by popularity amongst implementing organizations and technology vendors, cannot really be traced to the extensibility framework, important as it is. Few organizations in fact make any use of specialization at all and those that do tend to do so only to the slightest extent. The success of DITA is in fact rooted in the simplicity of its base models and the immediacy with which those base models can be used to address widespread documentation needs. The prospect of a significant community of implementing organizations coordinating their content management investments with reference to a single managed content model was greeted like a liberating hero by product vendors who had struggled valiantly over the years to support the wild variety of implementations that XML, or worst still SGML, made possible. The formation of a community around a common model also permitted a body of knowledge and practice to establish itself around the actual authoring, management and publishing tasks that need to be addressed in conjunction with any technology deployment. Momentum behind DITA, for all these reasons, continues to build.

OFFICE OPEN XML (OOXML)

Another area of significant change that bears directly upon document content is the decision by Microsoft to base its ubiquitous office automation suite on an XML-specified format and then to release the specification for this format to the public as an official standard (ISO/IEC 29500:2008). While the design of the format, commonly referred to as OOXML (Office Open XML), is obviously geared to supporting the Microsoft Office product line, it is nonetheless complete and exhaustively documented. For those that need to harvest content from mainstream office document sources or to channel content into them, the OOXML standard offers an undeniably valuable resource. With OOXML it has become possible to truly say that XML is everywhere. XML can now be found in the deepest enclaves of even the most traditional office. All this makes it possible to deploy XML-enabled solutions that deliver concrete benefits to a broad range of users and customers by injecting intelligent content into the normal flow of business documents. OOXML by itself does not achieve this vision. But OOXML in combination with the capabilities of DITA and collaborative workspaces fundamentally change the way organizations will handle their information. Most importantly of all, these changes are bringing the benefits of XML-enabled intelligent content to a new, and much broader, community of *business stakeholders* and this will energize the economics underlying the content management market.

ELECTRONIC PUBLICATION (EPUB)

Although various preceding efforts had been undertaken around standardizing how electronic books (eBooks) could be encoded and packaged, it is only in the last couple of years that the marketplace has shown itself ready to embrace eBooks on a major scale and to broadly adopt a predominant electronic publication (EPUB) standard. The recent uptake of eBooks, and the associated mobile reading devices (eReaders), has been nothing short of explosive. While there remains divergence within the marketplace, the underlying similarity among the competing standards, and the growing momentum around the EPUB standard itself, points towards an approaching convergence. Ideally, the publishing industry would take a page from history and chart a path towards the openness, simplicity and standardization that has been shown to work so well in the past.

Although it is often used to refer in general to the emergence of electronic publications, EPUB in fact refers to one specific standard managed by the International Digital Publishing Forum (IDPF), an industry association. The standard sets out relatively simple guidelines for publishing and packaging content that will be accessed using eReader devices. In this, the EPUB standard builds upon already existing standards such as the XML compliant representation of HTML (XHTML) and World Wide Web Consortium (W3C) Cascading Style Sheets (CSS). The EPUB standard even provides mechanisms for introducing “XML islands” that open the door to channelling the intelligence of content sources through to interesting behaviour within the eReader environment.

While it is still too early to declare a winner in the battle for dominance amongst electronic publication standards, the EPUB standard from the IDPF seems to provide the range of capabilities, simplicity and extensibility to become that dominant standard. The resulting convergence of eBook content and eReader innovations would be a boon to the publishing industry as a whole. Those publishers who manage to adapt to the new electronic publishing marketplace will discover new opportunities for growth by enriching content experiences in ways as yet unimagined.

HTML5

A brief mention should be made of the confluence of efforts that have been going into HTML5. It has been observed that the finalization of HTML5 has been taking a long time. Given the scope of what is being attempted, and the benefits that will come with its realization as a broadly supported standard, the time being taken is completely understandable. The threads that have been feeding into HTML5 are worthy of their own study so we will need to content ourselves with a cursory treatment here. Essentially, HTML5 is an effort to bring together a number of techniques and subordinate standards into a coherent whole – one that will put into the hands of web content professionals a formidable array of capabilities. Whether we are considering the introduction of mathematics, metadata and vector art work into web pages, or facilitating smart interactions between text and multimedia resources, HTML5 is emerging as the key way to reach and engage online users. In summary, HTML5 can be seen as pulling together and popularizing best practices that have been hammered out during the first 20 years of the web. HTML5 is a fundamentally worthwhile undertaking, one that will add a critical piece to the intelligent content puzzle and therefore one that is worth waiting for.

THE EMERGENCE OF INTELLIGENT CONTENT

Throughout the story we have been tracing, a number of themes have surfaced. One is the observation that technology innovation, in order to make a significant and lasting impact on successive innovations, must achieve a measure of popular adoption. Another observation is the fact that the single most important criteria in determining what innovations meet with popular adoption, as opposed to becoming enlightening but irrelevant footnotes, is *technical simplicity*. These observations do require one qualification in that the technical simplicity that leads to popular adoption is invariably a simplification, or simplified application, of much more sophisticated concepts that encapsulate important knowledge of how people understand their world, interact with their peers and establish collaborative frameworks for effective action. The notion of technical simplicity is also a relative one in that a common source of this simplicity results from the alignment of innovative solutions with the inherent capabilities of the technologies that are in broad use at a given point in time. And as the capabilities of the mainstream infrastructure evolve, it becomes increasingly more possible to introduce heightened sophistication into solutions and it is this dialectic process that is now making intelligent content a practical reality.

It is among the underlying propositions within this paper that the original SGML standard stands as the font of innovation from which HTML, XML and Web 2.0 have repeatedly drawn. This remains true even though it is rarely acknowledged and at times even actively denied. As we have seen, SGML sought to establish an elegant, but complete, way to model the structural patterns of human communication while at the same time addressing the physical challenges of exchanging the resulting representations across a wide range of boundaries. This does not take anything away from the importance of the contributions made with HTML, XML or Web 2.0 but it does help us to better understand the nature of, and reasons for, their success.

At the present time, the story of the evolution of open content technologies has arrived at an important juncture. Markup standards and strategies are reaching a level of seasoned maturity and implementers now have access to 20 years of experience to help guide investments towards practices that work well and away from others that offer questionable returns. DITA in many ways can be seen as an attempt to codify and reduce into practice many of these lessons. The web application infrastructure has, with Web 2.0 and HTML5, arrived at a level of availability and interactivity that allows implementers of content-oriented solutions, for the first time, to deploy collaborative content development and management environments *that people will actually use*. It is within these collaborative and interactive environments that people can design, develop and deploy intelligent content resources that can be disseminated into large communities themselves interacting digitally. The introduction of semantic technologies, which leverage layers of intelligent content to facilitate the discovery, management and interchange of content assets, allows enhanced automation to help organizations capitalize on their bursting stores of digital content. And finally, it is with the proliferation of mainstream support for XML, as the universal data interchange and application integration mechanism, that the value of intelligent content resources really becomes noticeable. The network effect applies and as the number of people and applications leveraging a given content asset expands, and as the variety of uses to which that content can be put increases, the returns being realized on the investment in content intelligence grows at an exponential rate.

When a single topic of content can find itself simultaneously guiding application behaviour within an enterprise system and informing a field service technician equipped with a hand-held device, then it is clear that intelligent content has arrived. When this same content has been authorized by a duly designated executive who reviewed and approved that content in the form of formal document, then we get the sense that something quite new is happening. And when we note that the approval of this document occurred within a collaborative online environment wherein key business processes are accessed and traversed using an interactive visualization that selectively exposes the underlying content details and metrics, we catch a glimpse of the magnitude of the changes occurring under our feet. The domain of people and business transactions, through the auspices of intelligent content, is beginning to directly inform, guide and control the domain of technology and this opens the door to new levels of organizational agility. As Dorothy noted to her little dog Toto, we get the feeling that we're not in Kansas anymore.

It is now possible, for the first time really, to create content that genuinely expresses its full meaning in an open way and for organizations to deploy mainstream technology resources to create, modify, publish and exploit that content as a normal part of doing business. In being able to reduce the costs of creating and exploiting these content assets, organizations are finding that the return-on-investment calculations are starting to yield quite different results. Content technologies have crossed an invisible line and have become increasingly compelling investments. And when it is considered that the content associated with any product or service is a critical part of its value, and that many products and services exist at least to some extent online, these content technologies start to be recognized as much as strategic necessities as prudent investments.

The evolution of open content standards, and the associated technologies, has brought us to the point where intelligent content is a reality. While this may only be recognized by a small community today, the future will belong to those organizations that can assemble the various tools and techniques that have emerged into combinations that allow their people and partners to innovate and adapt in the face of what can only be accelerating change. For the content management technology market, the emergence of intelligent content represents a tremendous opportunity provided the vendors can make their offerings relevant and sustainable in a landscape that is fundamentally more open and dynamic.

We are entering *the age of intelligent content* and the only hope we have of foreseeing at least some of what might happen next will be in understanding the evolutionary trajectory that has led to the emergence of this phenomenon. And if this history teaches us anything, it is that openness and simplicity, however challenging they may be to achieve and sustain, are the best principles by which to guide the evolution of solutions, organizations and the *intelligent content* that propels them.

ABOUT THE AUTHOR

Joe Gollner is the Director of Gnostyx Research Inc. (www.gnostyx.com), an initiative focused on advancing open content standards and on helping organizations leverage intelligent content technologies to realize concrete business benefits. Previously he was the Vice President and Chief Solution Architect at Stilo International where he led a world-class team of specialists in the design, development and delivery of state-of-the-art content processing solutions that leveraged the proven OmniMark platform. Prior to that, Joe had been the founder and president of XIA Systems Corporations, a firm that he had built up into a leading XML solution integrator and that he sold, in 2004, to Stilo. He has worked in the content management market, with an emphasis on open standards and large-scale systems, for 25 years and he has implemented dozens of systems across a variety of industry sectors. He was educated in a *wide* range of subjects at Queens University (Bachelors of Arts) and the University of Oxford (Masters of Philosophy) and has completed graduate programs in project management, business analysis and knowledge management. His blog, together with a selection of his whitepapers, can be found at www.gollner.ca.