

DITA — The Industry that Almost Wasn't

The untold story behind the story

Michael Iantosca



—

Over the past two-plus decades, an entire industry has grown and evolved around a single XML standard called the **Darwin Information Typing Architecture (DITA)**.

But exactly *how* did DITA come to exist? A search or a prompt would tell you that DITA originated at IBM, but it won't say how it originated. Such is the lore of internal corporate politics and ambition.

The entire industry built on DITA—the technology, the systems and tool providers, the consultants, the countless webinars, conferences, and more, would not exist without one fateful conversation—and a nasty fight that ensued.

DITA Would Not Exist Without a Fight

Without several pioneers' gritty leadership, persistence, and vision, DITA would never have existed, at least not in its current form. Instead of a widely adopted, open XML standard, IBM may have ended up with yet another SGML-based proprietary system, which would have limited its scalability and global influence.

This story illustrates and underscores how internal corporate battles over technology can massively impact the future of industry standards and an entire industry. Thanks to grit and determination, DITA became the global standard for structured content management, influencing how worldwide enterprises manage technical documentation today. Official accounts omit and obscure this fascinating and crucial part of DITA's origin story.

Getting Into This Predicament

My undergraduate degree was in communications and journalism from Marist College, after which I began working at the American Broadcasting Companies (ABC-TV) at their headquarters and broadcasting studios in Manhattan. However, my career took a different turn when I pursued graduate studies in computer science at Marist while seeking a technical career path. I was permanently hired by IBM in 1983 as an Information Developer (aka, Technical Writer) at the company's mainframe operating systems software development lab. In the company's grand wisdom, all operating system technical writers were required to undergo the same grueling software development training as software engineers. During that time, I gained advanced programming and software architecture chops.

My first writing assignment was to convert several books into a new electronic format called Generalized Markup Language (GML), which had been invented at IBM. Little did I know that this initial exposure to GML would eventually lead me down the path of structured content and DITA.

I also had a penchant for developing publishing software and tools.

Crossing Paths with SGML Pioneers

I became a highly productive technical writer, quickly outpacing many senior peers by developing and leveraging new automation tools and technology. Along the way, I had the fortune of meeting Dr. Charles Goldfarb, the father of structured document markup (GML) and the inventor of **Standardized General Markup Language** (SGML) that followed. Charles encouraged me to pursue a patent on my early work that began around the dawn of the IBM PC, developing electronic books and integrating multimedia with hypertext. Unfortunately, with their mainframe-only mindset at the time, the legal

department decided to file the invention as an “*invention publish*” in the US Patent Office and the Library of Congress, which made embedded media players based on markup forever public domain. This was a year or two before the birth of HTML and the World Wide Web. Ponder that for a moment - I kid you not: <https://priorart.ip.com/IPCOM/000109088>.

My work on multimedia and early hypertext development caught the attention of my colleagues who were deeply involved with SGML, including Elliot Kimber. Charles and Elliot convinced me to rebase my multimedia work from GML to SGML, marking my transition to structured content official. Elliot’s subsequent departure led to my corporate assignment to design and develop IBM’s first SGML-based content production system, the **Information Development Framework**, which later evolved into the **ID Workbench**, which was developed at IBM’s Rochester, Minnesota, development lab. I later became the IDWB product development team (PDT) lead for many years, leading the development of multiple generations of technical documentation publishing systems.

XML 1.0 – A New Horizon

I attended the SGML '97 conference in Boston, where the **Extensible Markup Language** (XML) 1.0 draft specification was unveiled with great fanfare. XML emerged from the SGML community before it became a W3C standard. There had been several previous attempts to create a simplified version of SGML, which was euphemistically referred to as **monastic SGML**. Excited by its potential, I evangelized XML across the company, publishing a widely circulated internal white paper and presenting it to then-CEO Louis Gerstner in person. Although the famed IBM Watson Research Lab quickly produced the industry’s first XML parser, there remained significant resistance, with some executives suspicious of the new standard and preferring a new web-oriented SGML-based DTD called **WebDoc**.

The Genesis of DITA: A Battle for the Future of Documentation

In the late 1990s, IBM stood at a crossroads. The company had long relied on a proprietary SGML (Standard Generalized Markup Language) system, IBMIDDoc, to manage its extensive technical documentation. As web technology advanced and digital transformation accelerated, IBM recognized

that its documentation system needed an overhaul to be less book-centric and more web-centric. The leadership at IBM leaned toward developing **WebDoc**, another **SGML-based successor**, which seemed like a logical extension of the existing infrastructure.

However, I stood in the way of this decision. As a staunch advocate for structured content in IBM's technical publications department, I believed that WebDoc was not the solution for the future. Instead, I saw the potential of a new, more flexible, scalable approach built on XML rather than SGML. What followed was an internal battle between me and a senior manager on the corporate Information Development team that would determine the future of the company's documentation strategy and the foundation of an entirely new standard that would shape an entire industry.

A Major Turning Point: The SGML vs. XML Battle Begins

At the time, SGML was still widely used in enterprise content management, and the company had heavily invested in SGML-based technologies. Senior managers believed a modernized SGML solution would be the next transition, preserving IBM's existing tools, infrastructure, and workflows. WebDoc, their proposed replacement for the IBMIDDoc SGML markup language, was intended to be an improved SGML system that would maintain continuity with the web and past investments.

Having deep experience with SGML, I saw the flaws in this approach. SGML was cumbersome, rigid, and difficult to adapt to new digital formats. XML, by contrast, was emerging as a more straightforward, adaptable standard that could better support web-based content delivery.

While leadership pushed for WebDoc, I fought back, advocating that the company **pivot away from SGML entirely and embrace XML**. This was not a small request—such a move would require a massive shift in documentation strategy, rethinking technology, and process.

Building the Case for XML and Topic-Based Authoring

Realizing that executive buy-in would not come without the proverbial knock-down, drag-out fight with City Hall (corporate), I acted. The most fateful moment was when I initiated an emotional meeting with my manager, Elmer

Hill, in his office one afternoon. I explained that “***There’s no “X” in XML!***” meaning that while XML had e**X**tensible in its title, it was not that much more extensible than SGML. Having previously led the enterprise development of the company’s SGML platform and the mind-boggling conversion of millions of pages of content, I convinced him to sponsor a workgroup to address what I viewed as a significant deficiency and omission in the XML standard. I convinced him to sponsor a dedicated workgroup of about ten specialists, including Don Day and others. We aimed to build new content architecture leveraging XML’s strengths while addressing IBM’s documentation challenges. Michael Priestley would join Don, myself, and the others several months hence. Priestley, in particular, brought key notions of DITA to the team. If that one pivotal conversation in my manager’s office had never happened, DITA—and the entire industry and communities that have grown around it—might not exist today.

If that one pivotal conversation in my manager’s office had never happened, DITA—and the entire industry and communities that have grown around it—might not exist today.

The team’s key innovation was a topic-based, modular approach to documentation. Instead of writing and managing massive, monolithic documents, writers could create self-contained, reusable topics and components that could be combined differently for various products and audiences. This approach would significantly reduce redundant content and improve efficiency, making updating and managing technical information easier.

We knew we were fighting an uphill battle. IBM corporate resisted change, and shifting to XML meant abandoning significant SGML investments and introducing risk. Still, the workgroup pressed forward, developing proof-of-concept demonstrations and pilot projects to highlight how an XML-based system could outperform WebDoc.

Winning the Internal Battle

Despite resistance from IBM leadership, we refused to back down. We demonstrated how XML and topic-based authoring could:

- **Enhance Content Reuse:** Writers could reuse existing content across multiple documents instead of rewriting similar information repeatedly.
- **Enable Multi-Channel Publishing:** Documentation could be seamlessly outputted into HTML, PDFs, and other formats without requiring separate authoring processes.
- **Streamline Content Updates:** Changes could be made to a single source and reflected everywhere that content was used.
- **Support Specialization:** Writers could create customized content variations without duplicating or breaking the structure.

Gradually, the evidence became undeniable. The efficiency gains, cost savings, and flexibility offered by XML and DITA were too compelling to ignore. After months of internal debates, discussions, and demonstrations, we won the battle—IBM leadership officially abandoned WebDoc and SGML in favor of the XML-based approach championed by our team.

With this victory, the foundational elements of DITA were set in motion.

The Birth of DITA and Its Impact on the Industry

Once IBM committed to XML and modular, topic-based authoring, the next step was formalizing the new framework. The workgroup refined the model, introducing key concepts such as:

- **Specialization:** Allowing users to extend and tailor the architecture without breaking the base structure.
- **Topic Types:** Defining standard content categories, concepts, tasks, and references.
- **Structured Authoring:** Encouraging a consistent, standardized approach to writing documentation.

By 2001, the XML-based system had proven its worth within IBM, and the company recognized that it had broader potential beyond its internal use. The team extracted its core, built the **DITA Open Toolkit**, and made it freely available on Source Forge, laying the foundation for an industry-defining standard. In 2005, IBM donated DITA to OASIS (Organization for the Advancement of Structured Information Standards), making it an open standard available to the world.

Today, **DITA is the leading structured authoring standard** prevalent in software, manufacturing, healthcare, finance, and many other industries. Its adoption continues to grow. DITA's flexibility, scalability, and efficiency have made it the backbone of many enterprise content management systems, proving that our vision was right for the entire world of technical documentation. With the advent of generative AI, structured content is emerging to become more critical than ever, breathing new life into DITA.

Patent-ly Unfair

The core team that developed DITA involved about ten or so people. As a working member, I was responsible for implementing the DITA platform, operationalizing it, and leading the enterprise-wide conversion from SGML to DITA XML.

Unfortunately, a middle manager from headquarters replaced the former corporate lead about halfway through the workgroup's development of DITA. Inexplicably, the new manager submitted a patent application for DITA that included himself and excluded half of the workgroup members who had toiled to develop DITA for many months. Those omitted included Rick Dennis, Elaine Petrone, Brian Gillan, Robert Berry, and yours truly. The patent was granted and enshrined in the history books (most notably, Wikipedia); **the rest of us got T-shirts with the DITA logo.**

Regarding the actual intellectual property, I credit Michael Priestly and Don Day as the principal inventors and thought leaders behind DITA.

A Legacy of Innovation and Determination

DITA might never have existed had I not fought leadership's push for another SGML-based solution. The insistence on embracing XML and a new way of

thinking about documentation transformed the company's approach to content management and shaped the entire technical writing industry.

This story serves as a testament to the power of innovation, persistence, and challenging the status quo. Our team's efforts ensured that DITA became an open, global standard, empowering organizations worldwide to manage complex documentation efficiently.

As structured content and content reuse become even more critical in an increasingly digital and AI-driven world, DITA remains as relevant as ever—all because we dared to say, **"We can do better."**