



Published on Free Software Magazine (<http://www.freesoftwaremagazine.com>)

The day my father blew himself up

Open standards, free software, and old documents

By [Anthony Taylor](#)

The Commonwealth of Massachusetts is the visible front of the current standards battle royale: in this corner, at 220 pounds, Open Document Format (ODF)! In the other corner, the 800 pound gorilla, Microsoft Office 12 XML format! Hopefully, we won't get caught in the explosion.

The day my father blew himself up

I was four years old the day of the explosion. I'd like to say I remember it well, but I don't; I don't remember my mother's fear, or the excitement or terror. Sometimes I think I remember my father's absence, or staying with Grandpa while Dad was away, but I can't be sure of those memories. The older I get, the surer I am I invented a lot of my own history.

This is all true. Dad recalls the explosion with great clarity.

It starts with a drill, a drill designed to bite holes into solid rock.

The drills were big machines, driven by air. Each drill drove a shaft of steel with a ten-pound bit on the end. As the steel made its way into the rock, the driller added another ten-foot section of steel. After much noise and dust and physical labor, the driller had made a thirty-foot hole down into solid rock. Dozens of holes made up a pit, named for the large gap in the rock left behind after the explosion turned solid rock into gravel, and the gravel was turned into road.

After drilling this particular pit, the crew fed the explosives down the holes. In one hole, Dad fed a cardboard-wrapped tube of explosive, but it didn't slide to the bottom of the hole. He dropped another tube on top the first one, hoping to knock the first tube loose. It failed.

The two tubes of powder did their job as they exploded, breaking up solid rock and shooting brand-new jagged-edged gravel like shrapnel into the air, into my father's head, into part of his brain

On June 18th, 1971, it was standard practice to use a wooden pole to unblock a stuck hole. They kept such poles handy. Dad prodded the stuck explosives. They did not break free.

Dad tells me, "Some folks would use the drill to ream out a clogged hole. The bit would just eat right through the powder. You lost the powder, but saved the hole." As the holes were drilled in patterns designed to break the solid rock into gravel, each hole was important.

Sheer muscle was not going to open the hole. Dad attached the wooden pole to his drill, and used its might to push the powder down the hole.

The day my father blew himself up

The two tubes of powder did their job as they exploded, breaking up solid rock and shooting brand-new jagged-edged gravel like shrapnel into the air, into my father's head, into part of his brain.

"I was following a common procedure," Dad says to me. "These days the procedure's a bit different."

I can only imagine.

The importance of being standard

Standards exist for many reasons: to provide uniformity, to outline a best practice, to facilitate the sharing of information. In the computer world, the sharing of information is of paramount importance. Without acceptable standards outlining methods of information sharing, computing as we know it would not exist.

If processing power is the heart of computing, information is its soul. Without information, computers are merely the dry bones of a useless technology.

Information has economic and social worth only to the extent that it is both useful and available. A useful piece of information known to only a few can benefit only those few; worthless information known to everyone helps no-one. Beyond that, the worth of information is not universal. The fact that a wooden pole can be used to set off explosives does me very little good, but would have been quite valuable to my father.

If we are to make full use of computers, we must facilitate the sharing of information. Even the slightest inhibitor to sharing will reduce the effectiveness and eventual economic worth of our information.

Procedural standards such as the one with which Dad blew himself up are used to proscribe a safe, acceptable practice. These standards are inherently different from information-sharing standards; however, the lesson learned the day of my father's accident are still valid.



Drilling right-of-way in Southeast Alaska

Not everything that is labelled a standard is safe. Sometimes the cost of following a standard is higher than we might imagine.

Dad says to me, "DuPont claimed there was no way a wooden pole could set off that powder. They didn't believe me until they went out to the pit and saw the pole in the middle of the crater."

The first standard

Some standards seem almost axiomatic. ASCII is perhaps the best-known of all computer standards, one of the oldest, and one we take for granted.

The most fundamental aspect of information sharing is the representation of the building blocks of information. In the early 1960s, that meant choosing a method to encode individual characters. Until that time, there was no universal standard for sharing information between computers.

And now, a history lesson:

Alfred Vail invented Morse code before 1844. This was one of the first electronic character encoding schemes, and certainly the first to gain widespread acceptance. As telegraphy equipment evolved, so did the encoding schemes—from Baudot's code thirty years later, through Murray's code around the turn of the century, to ITA2 in the 1930s and the U.S. Army's FIELDATA in the late 1950s. Each of these codes contributed to the development of serial data communication by providing an encoding standard for the Roman alphabet, along with transmission control codes. [1]

By 1963, the American Standards Association produced a new character encoding format, the American Standard Code for Information Interchange (ASCII). The X3.4 committee, which produced the ASCII-1963 standard, was composed of representatives from the computing industry, including AT&T and IBM.

Virtually every computer today uses ASCII

ASCII is a 7-bit (not 8-bit, as is often assumed) encoding scheme, based on the serial communications encoding schemes that came before. The telegraphy coding solutions from the prior 100 years of work provided the solution to an emerging computer problem.

About the same time, in 1963 through 1964, IBM developed the Extended Binary Coded Decimal Interchange Code (EBCDIC), an 8-bit encoding scheme, based on the Binary Coded Decimal (BCD) scheme used in most IBM computers at the time. EBCDIC was designed to facilitate punch card codes, which was important at the time. Some versions of the brand-new IBM S/360 had no operating system, and loaded specialized programs at system startup, via punch cards.

Although an X3.4 committee participant, IBM eschewed the use of ASCII in favor of EBCDIC for many years. At the launch of the IBM S/360, IBM had few ASCII peripherals, and many EBCDIC peripherals; so, IBM chose EBCDIC for their peripheral communication standard. The huge success of the S/360 further slowed IBM's move to ASCII peripherals. Eventually, competitors also created and sold EBCDIC peripherals, entirely to support IBM's S/360 (and, later, S/370 and S/390) mainframe installations.

More recently, both Fujitsu-Siemens (with BS2000/OSD) and HP (with MPE) adopted EBCDIC.

Virtually every other computer today uses ASCII.

Character codes: the future

ASCII has proven robust and simple, but it is limited to the Roman character set, and so is unsuitable for today's international computing environment.

The day my father blew himself up

A new standard is emerging for data communications: Unicode. International in scope and unencumbered by restrictions on its use, Unicode is quickly becoming the accepted standard for internationalization and localization of computer software.

Blood-stained eye

Blown back in the explosion, Dad realized two things: first, he was the one who knew first aid; second, he wasn't breathing. So, he couldn't tell Kenny how to get him breathing again.

The standard operating procedure of the day required the pit boss to know basic first aid; no other member of the crew required first aid training. Valuable information was kept solely by one person, to be turned worthless in a single explosion.

"I couldn't think of any other way to tell Kenny what to do," Dad tells me, "so I started breathing again." Then he says, "I could feel rocks rolling down my throat."

He couldn't open his eyes, but he heard the voices around him, heard the fear. When they talked of calling in a helicopter to transport him out, Dad knew he wouldn't fit in the bubble. He imagined himself strapped to the skid, imagined the wind blowing down over the wound in his head. "No helicopters," he calmly told them. "Drive me into camp, and call a plane."

At the time, Thorne Bay was simply a logging camp, semi-permanent and small. There were no hospitals on Prince of Wales Island. The closest medical facility was in Ketchikan, 65 miles away, on a different island.

Dad tells me, "We got into camp. It was your grandfather's birthday; that's how I remember the date. I told Kessler not to touch me."

Kessler said, "But I have to clean you up, Gary."

"Leave him alone," Granddad said. "Just don't touch him."

They drove him down to the dock then, and put him on the float plane. Thirty minutes later he was loaded into the back of an ambulance.

"I was awake for everything," Dad says. "I remember they put me in the ambulance with my head facing the back." They gunned the engine, and raced for the hospital, up a steep hill. "They hit that hill, and I felt the blood rush up to my head."

There was a quick sharp pain in his right eye. "I felt the blood vessel pop right then. The hemoglobin stained the retina, and I haven't been able to see out of that eye since."

Networks of Babel

In 1969, computer networks were small, custom-built systems, used mostly to distribute terminals around campuses. That year saw the founding of ARPANet, a project intended to provide inter-computer communication across the United States. By the end of 1969, ARPANet boasts four nodes.

The ARPANet project eventually produced the protocols and standards that provide the bones and ligaments of the internet, but was considered simply an "academic" platform for the next two decades.

The day my father blew himself up

In 1983, Novell produced the first business-oriented, PC-based local area network (LAN). Based on the IPX protocol, Novell's Netware product became the dominant business networking software in the mid- to late-1980s. IPX itself was based on the Xerox Network Services (XNS) protocol.

Also in 1983, Sytec Inc. developed IBM's network protocol, NetBIOS and NetBEUI. NetBIOS handled high-level session-oriented tasks such as computer name resolution; NetBEUI was designed to exchange packets of information, similar to IPX.

Neither IPX nor NetBEUI were designed for wide-area networks such as the internet. In fact, NetBEUI was not designed to be routable at all.

Today, the internet has about 375,000,000 hosts, every single one of which runs TCP/IP

Microsoft adopted NetBIOS over NetBEUI as their networking standard in 1985, with the MS-Net and MS LAN Manager products.

For a decade, there were more computers running either IPX or NetBEUI than any other protocol. The fight for dominance between them heated up, with Microsoft determined to replace Novell as the dominant supplier of networking software.



A driller and his chuck tender aren't wearing hard hats

MS-Windows NT 3.1 was released in 1993, and directly challenged Novell's Netware server product. Both NT 3.1 and NT 3.5 installed IPX as the default transport protocol for NetBIOS.

Twenty years before, in the last month of 1974, Vint Cerf, Yogen Dalal, and Carl Sunshine released Request for Comment (RFC) 675, "Specification of Internet Transmission Control Program." Over the next six years, TCP developed into two separate protocols: IP and TCP. These standards, introduced in RFCs, were developed in an open fashion over the course of many years.

The most recent RFCs for TCP/IP are RFC-791 (IP) and RFC-793 (TCP).

At the beginning of 1991, the internet consisted of about 375,000 hosts. 1991 was the year everything changed. 1991 was the year of the World Wide Web.

Today, the internet has about 375,000,000 hosts, every single one of which runs TCP/IP.

The day my father blew himself up

How did that happen? In 1991, most businesses were using either IPX or NetBEUI for networking, both of which were designed and deployed for business. Today, IPX is almost dead, and NetBEUI is a faint memory even among us old-timers.

The answer is, like most answers to simple questions, very complex. Partly, the internet was itself a unique beast, a network-of-networks allowing cheap and instantaneous communications across the world. Partly, the World Wide Web gave muscle and skin to the skeleton of the internet, providing an easy and rich means of grabbing and providing information.

Mostly, though, TCP/IP was a well-designed standard that was open for implementation. No one company held the keys, no one company had an advantage over any other.

With the release of MS-Windows NT 3.51 in 1995, Microsoft switched the default NetBIOS transport protocol from IPX to TCP/IP. Novell switched from IPX to TCP/IP with the release of Netware 5.0, in 1998. Even among the business LAN giants, TCP/IP (and open standards) won out.

The battle that wasn't

The internet was growing tremendously before Sir Tim Berners-Lee gave us the World Wide Web, but nothing short of the internet itself is a greater success than the Web. In a time when private networks like AOL and CompuServe were controlling access to information, the Web was designed to release control. Berners-Lee developed the first HTTP server while at CERN. He also designed the hypertext markup language (HTML), and developed a simple text-only client. He released his code to the public, via the internet.

Tellingly, he used two open standards in his work: TCP/IP for the transport protocol, and ASCII for the character encoding. He could very well have made different choices, considering IPX was the most-used networking transport protocol (though not at CERN), and most document formats used binary codes for document markup and layout.

The commercial precursors to the modern internet used proprietary access methods, and gave the public very little ability to provide information. AOL, CompuServe, Delphi, Prodigy, Genie, and a slew of others all had their own access methods and on-line protocols. Each was an island of information, generally exchanging only email.

Delphi became the first major online service to allow full access to the internet. Over the next few years, every other surviving online provider followed suit. This shift from proprietary service to internet service provider was a result of the growing media coverage of the internet, especially the attention given to the World Wide Web.

Without open and freely-implemented communications standards, the Web would not exist as it does today

In 1993, the National Center for Supercomputing Applications (NCSA) released the Mosaic web browser. Mosaic provided support for in-line images and other multimedia, greatly influencing the HTML specification. It soon became obvious HTML needed standards oversight to curtail the use of non-standard markup tags.

The Internet Engineering Task Force (IETF) released the first HTML specification working draft in June 1993. No standard was released until November 1995, however, with the publication of IETF RFC 1866.

The day my father blew himself up

By this time, Berners-Lee had founded the World Wide Web Consortium (W3C). They released their first HTML standard in January of 1997, for HTML 3.2.

There was no real battle for the Web standards. A few skirmishes broke out in the early days, especially during the early days of Microsoft's Internet Explorer, but there was no real battle. Although different browsers implement the W3C standards with varying degrees of success, HTML and HTTP have proved extremely resilient to nonstandard modification, mostly due to the early open standardization efforts.

Without open and freely-implemented communications standards, the Web would not exist as it does today. The Web is arguably more responsible for the wide-spread adoption of home computers than any other factor. Many businesses exist solely as a result of the Web. Open standards increased one market, and created other markets wholecloth.

This is important.

The lobotomy effect

Speaking of his eye, Dad says, "I don't notice it anymore. I just can't see out of the one eye."

The medical facility in Ketchikan was inadequate. Tiny rocks had pierced his skull above his right eye, and the closest neurosurgeon was in Seattle. Dad was on the next plane South.

The surgery they performed was similar to the partial lobotomies they used to give violent criminals. A small part of Dad's skull above his right eye was cut away. They removed the damaged portion of the brain, and wired the square of skull back into place.

Simple as that.

I'm told my father is much mellower since the surgery, a lot less quick to anger. I don't remember much about pre-explosion Dad. I do remember him working rocks out from under the skin of his forehead for years. But in any case, I imagine being blown up changes a person.

A note on control

The free flow of information is just that: free. Free of cost, free of regulation, free of control.

There is very little money to be made from information if it is all freely available. A company that wishes to make money must either provide some other service, such as searching or organization, or create an information bottleneck, some sort of roadblock in the otherwise free flow.

The only way to introduce a roadblock is to control access to information. Before the internet, this was done by controlling the distribution channels of the physical media on which the information was imprinted. Since the advent of the internet, and the open standards on which it is based, the distribution channel is inherently uncontrollable. As a result, corporations wishing to extract money from the flow of information must use existing laws that control "ownership", or invent new laws that give them the control over the access to information.

This assault is currently underway. In the United States, this is accomplished through laws such as the DMCA, and in regulations such as the Broadcast Flag requirements set forth by the FCC (which has been

The day my father blew himself up

overturned; media companies are currently trying to enact broadcast restrictions as law, rather than as an FCC regulation). Other bills designed to restrict the flow of information are proposed every few months. Some will most likely become law.

Other legal avenues are also available. Patents are used to restrict access to specific file formats, allowing the patent holder to extract a fee before allowing access to information stored in that format.

One common technique is the “submarine patent”, a patent which is left unenforced on a format until it becomes a de-facto standard, and is then aggressively enforced. This technique was used quite successfully by Unisys and their LZW patent. GIF, introduced in 1987, was the primary graphic format of the early 1990s; it used LZW compression on images. Although the GIF specification was widely released and known, Unisys waited until December of 1994 to enforce their patent.

Forgent Networks is currently threatening the JPG image compression format in the same manner. JPG has been a widely-used format since the early 1990s, and replaced GIF as the primary image format on the Web.

Software patents provide one more means of controlling access to information. They are the tool of choice for the internet highwayman.

The new old battlefield

Standards are most important when information must be shared with others. Without an open, neutral, unencumbered specification, there are artificial roadblocks in the simple sharing of information. These roadblocks interfere with the free flow of information, decreasing both the economic and social benefits of that information.

Recently, the Information Technology Department (ITD) of the Commonwealth of Massachusetts released its Enterprise Technical Reference Model (ETRM). The ETRM covered only the executive branch of the Massachusetts government. Version 3.5 of the ETRM required all documents created by the executive branch to use the Open Document Format (ODF), an open and unencumbered specification that has been submitted to both the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) for ratification as an officially-recognized publicly-available standard (PAS).

As of November 25, 2005, the Commonwealth of Massachusetts is undergoing a political debate concerning the adoption of ODF, versus Microsoft’s Office XML file format. Many consider Microsoft’s format “open enough”. Others claim it is the de-facto standard, and as such deserves the honor of becoming Massachusetts’ primary document format.

What is “open enough”? Is it when other commercial vendors may license the format from Microsoft, but free software cannot? Or is openness achieved only when anyone may implement the standard in any fashion, at any time?

Eric Kriss, Secretary of Administration and Finance of the Commonwealth of Massachusetts, has stated the goal of using an open format was to ensure sovereignty. Sovereignty is the fundamental (and perhaps most important) issue. While the public does not directly control ODF, we are able to provide input and feedback in the future development of the standard.

As Microsoft controls future MS-Office formats, the public has no control whatsoever.

ODF allows free, unfettered implementation, by anyone, at any time, for any purpose.

The day my father blew himself up

Microsoft's XML format is potentially encumbered by Microsoft-owned patents, and is certainly encumbered by a license that shuts out free software developers.

Which format best meets the goal of sovereignty?

I believe there are also pragmatic reasons to adopt open standards. At the core, open communication standards ensure the ability to communicate seamlessly and effortlessly. Open standards promote competition, which contributes to both the economy and further advances in computing. Information encoded using an open standard today will still be available two decades from now.

This is not true of proprietary, closed formats. First, no real competition is possible, as one corporation controls the format. Secondly, as the format is not open, nor openly-published, future compatibility is not guaranteed. I have documents written in IBM Works for OS/2; I am able to access them only because I know how to extract readable ASCII characters from a binary document. Documents written in Microsoft Works in 1993 are similarly difficult to open.

Governments especially require document longevity. What if the Declaration of Independence had been written in Microsoft Works RE (Revolutionary Edition)?

The sound of inevitability

Practices changed after my father's explosion. Everyone on the crew was required to know first aid. They no longer prodded explosives with wooden poles. And so on.

Microsoft's resistance to the Open Document Format is understandable. If everyone is able to freely exchange information written in productivity software from any vendor, there is no MS-Office lock-in. There is no more loss of document layout when converting from one format to another. There is only a single, agreed-upon format, and the final layout is dependent solely upon the quality of the vendor's implementation.

I believe we can learn from history. I believe historically, free, unencumbered standards win out over proprietary specifications in almost all cases. I believe it is all but inevitable.

I mention a few such cases, some contentious, and some not. There are other cases, such as IBM's failed attempt to regain control of the PC market with their proprietary Micro Channel Architecture bus, or Microsoft's attempt to use ActiveX to take control of the web.

I won't even mention Macromedia Flash.

The explosion is happening around us, this very day. For many years proprietary document formats have interfered with the uninhibited exchange of data. That problem will only grow worse, and the solution is not found in a single product. Products come and go, and even Microsoft Office does not use a single document standard. Each version of MS-Office uses a new, incompatible document format.

When the shockwave has passed, and we pick the grit out of our scalps, I believe we will do what we have always done from the earliest days of computing: we will use an open, freely implementable standard. This is as inevitable as spam. Eventually, like ASCII, most of us will forget there was anything else.

Bibliography

[1] Tom Jennings, "[ASCII: American Standard Code for Information Infiltration](#)"

Biography

[Anthony Taylor](#) (/user/9" title="View user profile.): Tony Taylor was born, causing his mother great discomfort, and has lived his life ever since. He expects to die some day. Until that day, he hopes to continue writing, and living out his childhood dream of being a geek.

Copyright information

This article is made available under the "Attribution-Sharealike" Creative Commons License 3.0 available from <http://creativecommons.org/licenses/by-sa/3.0/>.

Source URL:

http://www.freesoftwaremagazine.com/articles/standards_battleground
