

Why OOXML Is Not Ready for Prime Time

Jon Bosak, Corporate Standards, Sun Microsystems
July 2007

Introduction

Some observers were surprised recently by Sun's vote in favor of "Conditional approval (No with comments)" during the recent consideration of Draft International Standard (DIS) 29500 (aka OOXML, aka Ecma 376) by INCITS V1, the advisory committee responsible for recommending a U.S. national position on approval of OOXML as an International Standard, as well as taken aback by Sun's statement that "*we support DIS 29500 becoming an ISO Standard and are in complete agreement with its stated purposes of enabling interoperability among different implementations and providing interoperable access to the legacy of Microsoft Office documents.*"

Part of the surprise was due to the fact that this sentence was widely quoted by people who for some reason chose to omit the next sentence: "*Sun voted No on Approval because it is our expert finding, based on the analysis so far accomplished in V1, that DIS 29500 as presently written is technically incapable of achieving those goals, not because we disagree with the goals or are opposed to an ISO Standard that would enable them.*"

This paper explains the technical basis for Sun's vote in INCITS V1. The full INCITS V1 response to DIS 29500 is archived at <http://www.ibiblio.org/bosak/v1mail/200707/2007Jul17-133724.eml>.

JTC1 voting background

Many people have expressed strong opposition to OOXML for a variety of reasons relating to the real-world motives for its submission to JTC1 and the real-world effects its adoption would have on the attempt to create a truly open environment for office productivity data formats.

But standards experts don't get to take positions like this, at least not within the process leading to the approval of International Standards as specified in the Directives published by ISO/IEC JTC1. In that process, there are exactly three positions available when voting on a Draft International Standard (DIS):

9.8 Votes on Fast-track DISs

The period for fast-track DIS (or DAM) voting shall be six months, consisting of a 30-day JTC 1 National Body review period followed by a five-month ballot period. NBs [national bodies like INCITS, to which ANSI has delegated the responsibility] may reply in one of the following ways:

- Approval of the technical content of the DIS as presented (editorial or other comments may be appended);
- Disapproval of the DIS (or DAM) for technical reasons to be stated, with proposals for changes that would make the document acceptable (acceptance of these proposals shall be referred to the NB concerned for confirmation that the vote can be changed to approval);
- Abstention (see 9.1.2).

(Note: Conditional approval should be submitted as a disapproval vote.)

It can be seen that there is no position under the JTC1 Directives for simply rejecting a DIS outright, and in particular not a DIS that has already been through a standards process in some other forum, in this case, Ecma. While of considerable and obvious relevance in the real world, market-related factors are out of scope when considering the submission of a DIS. As standards people, we must follow the existing

process we are given through the Directives published by ISO/IEC JTC1, and we are thus forbidden from taking market-related factors into account in our voting, regardless of the many well-documented market-related problems with OOXML. The only allowable considerations in this part of the standards world are technical ones. As a company based on the concept of open standards, Sun does its best to honor this principle.

Reasons for our vote

So the simple answer to the question of why we didn't vote for an unconditional No is that it's not an option. And that's why Sun, along with every other V1 member that voted against Approval (including IBM, Oracle, and Red Hat), voted shortly thereafter for a U.S. position that puts us all on record as eventually supporting a version of DIS 29500 that meets its stated objectives — though Sun appears to be the only one taking the public position that in principle a Standard is better than a proprietary monopolistic application.

While all those voting against approval of DIS 29500 are on record, at least formally, as supporting the further progress of the spec, the fact is that the DIS in its current form is very far from something that would justify a vote to Approve as an International Standard. Purely on the basis of technical considerations, it is nowhere near ready for prime time, and Sun's vote No on whether to approve DIS 29500 (the first vote taken in V1) signifies our opinion on this as technical experts. As Sun's primary representative to V1, I'll explain the reasons why.

Why DIS 29500 is technically unacceptable

In its present form, DIS 29500 is technically unacceptable as an International Standard for three reasons: unacceptably low quality; failure to deliver on its promise of vendor-neutral interoperability; and failure to deliver on its promise of interoperable access to the legacy of binary Microsoft formats.

Reason 1: Low quality

Almost no one (including this author) has had the time to examine in detail the more than six thousand pages of DIS 29500; all that most of us can manage are core samples into the vast bulk of this submission. But no matter where reviewers begin, virtually every one of them comes up with technical defects. The errors found by Rob Weir of IBM are perhaps the best known, but there are many others that have also been widely published.

In addition to outright technical defects, the DIS contains a number of critical errors caused by failure to observe the rules for the construction of standards, in particular the difference between normative language and informative language. There are whole sections of the DIS in which normative language essential to implementation of the specification is put into notes, which are (as the DIS clearly states) purely informative. This is significant, because it means that a large amount of necessary detail relating to requirements is specifically labeled as irrelevant to conforming implementations.

After seeing a number of such errors, it becomes clear that DIS 29500 did not receive adequate quality assurance before it was submitted for consideration by the international community. James Mason, the representative of the U.S. Department of Energy who has served for more than 20 years as chair of JTC1 SC34 and its predecessor, JTC1 SC18 WG8, gave his expert assessment of the quality of the submission when he wrote his fellow V1 members that “the submitters obviously did not read — and edit — this submission into a consistent whole. If it were coming through the normal ISO process, I'd say it was in the state of a Working Draft and not yet ready for registration as a Committee Draft and assignment of a number.”

So the number one reason that OOXML is not ready for prime time is simple: it's just not up to the level of quality expected of an International Standard. Since Ecma is a legitimate industrial standards organization not ordinarily associated with shoddy work, it has to be assumed that the high defect level of DIS 29500 is a result of putting a specification about an order of magnitude larger than the average International Standard through a development and approval process run about three times as fast as the process used to create the average International Standard. But whatever the cause, a great deal of work needs to be done to fix these problems, and until those repairs are made, the DIS is simply not ready for approval.

Reason 2: Lack of interoperability

The JTC1 Directives devote an entire annex to the subject of interoperability. As a specification whose avowed purpose is to enable interoperability between different implementations, DIS 29500 must meet the minimum requirements set forth in that annex. In particular,

Standards designed to facilitate interoperability need to specify clearly and unambiguously the conformity requirements that are essential to achieve the interoperability. Complexity and the number of options should be kept to a minimum and the implementability of the standards should be demonstrable. Verification of conformity to those standards should then give a high degree of confidence in the interoperability of IT systems using those standards.

A Standard is not software

Perhaps surprisingly given its enormous size, DIS 29500 fails utterly to “specify clearly and unambiguously the conformity requirements that are essential to achieve the interoperability.” Some examples of this border on the laughable, as in the widely remarked “lineWrapLikeWord95” setting, but there are long stretches of the DIS that read as if its authors simply did not understand the difference between software documentation and standards writing. It's one thing to specify a set of parameters well enough to document a single vendor's APIs for internal use by engineers with access to the actual code; it's quite another to write an International Standard in detail sufficient to ensure interoperability among implementations developed from scratch by different individuals or companies that do not have access to that code. This pervasive underspecification makes interoperability among different implementations impossible.

Since so much of the DIS looks like recycled software documentation, I'm inclined to ascribe the underspecification problem to the same size and resource issues that are probably responsible for the high defect rate. But there's one critical point affecting the entire submission that carries this to an extreme we might call “radical underspecification”: Section 2.5 of Part 1 of the DIS, which states that “Application conformance is purely syntactic.”

Syntax and semantics

On the face of it, this astonishing provision would appear to indicate that the authors of the DIS did not understand the purpose of XML, which is to enable subject matter experts to associate a set of agreed-upon syntactical constructs with agreed-upon semantics. That's why we do this. In IT standards, interoperability lies precisely in an agreement about semantics. For something that purports to define an XML vocabulary to say, in a clause covering the entire specification, that it doesn't require applications to recognize semantics is to say that it fails in the primary purpose of an XML vocabulary. It would be like an International Standard for screw threads that defined what a screw thread is but didn't actually say how far apart the threads were supposed to be.

In reality, of course, the DIS does, in its insufficiently detailed way, attempt to define semantics — about six thousand pages of semantics. As stated in Section 2.3 of Part 1, “this Standard constrains both syntax and semantics.” It just doesn't require any application to recognize those semantics.

To use the proverbial phrase, this DIS wants to have its cake and eat it, too. It wants to define semantics but leave applications free to do what they please. So in the normative language of Section 2.5 it says that applications can ignore semantics, but then in the informative “Interoperability Guidelines” of Section 2.6, it tries to make sense of this by appealing to applications to conform to semantics after all. For example:

A presentation editor that interprets the preset shape geometry “rect” as an ellipse does not observe the first guideline because it implements “rect” but with incorrect semantics.

To take another example more or less at random, the statement that “application conformance is purely syntactic” directly contradicts the requirement in 11.3.1 that “A WordprocessingML consumer shall treat the contents of such legacy text files as if they were formatted using equivalent WordprocessingML, and if that consumer is also a WordprocessingML producer, it shall emit the legacy text in WordprocessingML format.” This clearly assumes a known semantic framework and requires a conformant application to respect that framework by translating the constructs of the legacy format into “equivalent” OOXML constructs, that is, constructs having the same semantics.

As V1 chair Patrick Durusau noted in email discussion of this point,

The net result of these clauses is that conformance to the semantics as defined by DIS 29500 can be changed for a document instance or by an application that claims conformance to DIS 29500 from those defined therein. In other words, users and application designers will be subject to arbitrary changes in the semantics allegedly defined by DIS 29500, resulting in a lack of interoperability and portability, which are goals for any standard issued under the JTC 1 Directives.

In practice, the effect of radical underspecification is to allow behavioral details to be determined on an ad hoc basis by the dominant software. This is contrary to both the JTC1 directives and the stated goal of the DIS. To be accepted as a satisfactory — or even meaningful — International Standard, DIS 29500 will need to be rewritten throughout to perform the most basic function of an XML IT standard by clearly identifying normative semantics in detail sufficient to enable interoperability among competing implementations.

Reason 3: Failure to deliver interoperable access to legacy documents

OOXML has been described from the beginning as an XML format that would do two things:

- it would create an open standards environment for competing office productivity applications, and
- it would bring the huge legacy of existing documents in proprietary Microsoft formats into this new environment.

As stated in the Introduction to Part 1 of the DIS:

The goal is to enable the implementation of the Office Open XML formats by the widest set of tools and platforms, fostering interoperability across office productivity applications and line-of-business systems, as well as to support and strengthen document archival [sic] and preservation, all in a way that is fully compatible with the large existing investments in Microsoft Office documents.

The first part of this promise (which, as already noted, is impossible to achieve in DIS 29500 as it currently stands) is intended to appeal to users wanting interoperability of competing applications *going forward*. The second part is intended to assure large users — governments, for example — that their *existing legacy* of documents created in the old proprietary environment will share in the benefits of the new competitive environment of interoperable applications. Unfortunately, the DIS as currently written is as incapable of delivering on the second part of the promise as it is incapable of delivering on the first.

The promise of interoperable access to the Microsoft legacy

The legacy aspect of the OOXML program is set forth in the Ecma document (Ecma/TC45/2006/374) submitted to JTC1 as Ecma's justification for asking JTC1 to approve the specification as an International Standard. The relevant portion of that document reads as follows:

2. PURPOSES FOR THE STANDARD

OpenXML was designed from the start to be capable of faithfully representing the pre-existing corpus of word-processing documents, presentations, and spreadsheets that are encoded in binary formats defined by Microsoft Corporation. The standardization process consisted of mirroring in XML the capabilities required to represent the existing corpus, extending them, providing detailed documentation, and enabling interoperability. At the time of writing, more than 400 million users generate documents in the binary formats, with estimates exceeding 40 billion documents and billions more being created each year.

The original binary formats for these files were created in an era when space was precious and parsing time severely impacted user experience. They were based on direct serialization of in-memory data structures used by Microsoft® Office® applications. Modern hardware, network, and standards infrastructure (especially XML) permit a new design that favors implementation by multiple vendors on multiple platforms and allows for evolution. [...]

Perhaps the most profound issue is one of long-term preservation. We have learned to create exponentially increasing amounts of information. Yet we have been encoding that information using digital representations that are so deeply coupled with the programs that created them that after a decade or two, they routinely become extremely difficult to read without significant loss. Preserving the financial and intellectual investment in those documents (both existing and new) has become a pressing priority.

The emergence of these four forces — extremely broad adoption of the binary formats, technological advances, market forces that demand diverse applications, and the increasing difficulty of long-term preservation — have created an imperative to define an open XML format and migrate the billions of documents to it with as little loss as possible. Further, standardizing that open XML format and maintaining it over time create an environment in which any organization can safely rely on the ongoing stability of the specification, confident that further evolution will enjoy the checks and balances afforded by a standards process.

Various document standards and specifications exist; these include HTML, XHTML, PDF and its subsets, ODF, DocBook, DITA, and RTF. Like the numerous standards that represent bitmapped images, including TIFF/IT, TIFF/EP, JPEG 2000, and PNG, each was created for a different set of purposes. OpenXML addresses the need for a standard that covers the features represented in the existing document corpus. To the best of our knowledge, it is the only XML document format that supports every feature in the binary formats.

In this passage from the Ecma submission to JTC1, the promise is very clear:

- there are billions of existing documents in Microsoft binary formats;
- OOXML supports every feature of those formats;
- OOXML creates a new interoperable environment; and
- a primary purpose of OOXML is to enable the migration of the old documents to the new interoperable environment.

This promise is hardly confined to the Ecma submission to JTC1. The 7 December 2006 press release announcing Ecma ratification of OOXML is subheaded “The new open standard safeguards the continued use of billions of existing documents and promotes document processing interoperability,” and a subhead further down in that release states the goal as “Opening Up Billions of Documents.” A Microsoft press release dated 16 May 2007 boasts that “a thriving community has emerged around the Ecma Open XML file formats (also known in standards circles as ECMA-376) because of the formats’ many capabilities,

including backwards compatibility with billions of existing documents...” An undated Microsoft statement targeted to health care professionals says that “Open Office XML is optimised to achieve backward compatibility with billions of existing documents, helping to preserve customers’ investments and meet their archival needs.” Another article from Microsoft’s web site says that “Users (including enterprises and government entities) have demanded that Open XML be designed so that it is backward compatible with the billions of existing documents created using earlier formats,” and it quotes a Microsoft executive as saying, “This is a very important concept because, if not implemented, billions of documents will be left out of the ‘eco-system’.”

Can this goal be achieved?

It seems to us beyond argument that “opening up billions of documents” formerly held captive to a single vendor’s product line and bringing them into the interoperable software ecosystem would be a tremendous benefit to global IT, and Sun has gone on record as supporting any contribution that would accomplish this. But DIS 29500 as currently written does not provide the piece that would make this promise a reality — a normative mapping from the old closed binary formats to the new open XML format.

Without such a mapping, the conversion of the existing nonstandard base to a standard form is impossible to accomplish in a consistent and therefore interoperable way (unless, of course, you happen to own the running code that did this in the first place). As V1 chair Patrick Durusau put it, “There is no doubt that inconsistent XML formats [i.e., inconsistent translations to and from OOXML] will lead to difficulties in searching, editing, automated processing and even inclusion of such documents in office systems based upon particular conversions of legacy documents.”

If OOXML were to become an International Standard, therefore, owners of Microsoft documents in binary format would be in exactly the same position that they are in today. An owner of legacy Office documents who wanted a reliable faithful rendering of those documents would still be required either to continue maintenance of the Microsoft products that created them or to purchase new Microsoft products that recognized the old formats using Microsoft’s proprietary knowledge of how those formats were intended to be rendered. No one other than Microsoft can have the necessary knowledge for how to do that, in spite of the promise of OOXML. On the other hand, an owner for whom a rough conversion of legacy documents to an XML format was sufficient could already accomplish this using existing free products like OpenOffice. In either event, OOXML by itself would add nothing to the ability of legacy document owners to access their documents, because OOXML does not put competing products on an equal footing with Microsoft products when it comes to correctly rendering the existing document base.

Strange arguments

When this point was brought up in V1, the conversation took a strange turn. Instead of attempting to refute this objection, the Microsoft representatives tried instead to disown the language quoted above from the Ecma submission to JTC1, calling the ability to represent legacy formats “marketing fluff” and denying that it was an objective of OOXML to provide interoperable access to the existing corpus of Microsoft binaries. One OOXML proponent went so far as to try to clarify the passage from Ecma quoted above by saying “the text says the ‘features represented in the existing document corpus’ ... not the actual document corpus” — as if the owners of those “billions of documents” were only interested in their features, not the documents themselves. Ecma/TC45/2006/374 says that the purpose of OOXML is “to define an open XML format and migrate the billions of documents to it with as little loss as possible.” It doesn’t say that the objective is to migrate the features; it says that the objective is to migrate the documents.

A two-day V1 meeting was held in Washington at the end of June in a valiant (though necessarily incomplete) attempt to resolve the hundreds of issues that had been identified by then. This issue was discussed at length, and I pointed out several consequences that would necessarily result from abandoning

the goal of open access to legacy documents.

What would happen to the DIS if this is not a goal

First, it would require some fairly significant changes to the DIS itself.

1. Given the near-universal belief fostered by OOXML supporters that OOXML does, in fact, provide open, interoperable access to legacy Microsoft Office documents, the Scope section of the DIS would have to be revised to clearly state that this is not an objective.

Example:

It is not an objective of this Standard to provide open access to the existing legacy of documents in Microsoft binary formats.

2. Certain provisions of DIS 29500 mandating conformant translation of legacy formats would have to be removed.

Example:

In Section 11.3.1 of Part 1, the content type of Alternative Format Import Part is defined as “any content, support for which is application-defined.” Needless to say, this covers a lot of possibilities. The requirement for conformant consuming software is equally broad: “A WordprocessingML consumer shall treat the contents of such legacy text files as if they were formatted using equivalent WordprocessingML, and if that consumer is also a WordprocessingML producer, it shall emit the legacy text in WordprocessingML format.”

The assumption in this passage that the Alternative Format can, in fact, be represented in OOXML is, of course, impossible to guarantee in the general case. XML alone (one of the possibilities informatively listed as an Alternative Format in Section 11.3.1) is capable of denoting an unlimited number of imaginable formatting semantics that are beyond the representational capabilities of this or any other particular version of OOXML.

But if we ignore this problem and assume that OOXML is capable of supporting every possible feature of every possible input word processing format, both now and in the future, the absence of a normative mapping would make it impossible for consuming software to properly and verifiably conform to the requirement that it “shall treat the contents of such legacy text files as if they were formatted using equivalent WordprocessingML.” Without standard mappings, there is no mechanism for establishing whether the output form is “equivalent” to the input.

In other words, conforming applications are required (“shall treat,” “shall emit”) to implement behavior whose conformance to the specification is impossible to establish. If no mappings are to be provided whereby conformance can be established, such passages will have to be identified and removed.

3. If access to legacy documents is no longer a goal, then a number of features included in DIS 29500 explicitly for the representation of Microsoft binary formats will have to be removed as no longer relevant.

The most obvious example is VML, the proprietary Vector Markup Language introduced as a W3C Note several years ago as competition to the open SVG format from W3C itself. The 600+ pages devoted to VML in DIS 29500 are introduced near the beginning of Part 4 Section 6.1 with what has to be one of the strangest passages in the history of International Standards:

The VML format is a legacy format originally introduced with Office 2000 and is included and fully defined in this Standard for backwards compatibility reasons. The DrawingML format [described separately in DIS 29500] is a newer and richer format created with the goal of eventually replacing any uses of VML in the Office Open XML formats. VML should be considered a deprecated format included in Office Open XML for legacy reasons only and new applications that need a file format for drawings are strongly encouraged to use preferentially DrawingML.

An existing standard (or a section or feature of an existing standard) is “deprecated” when, in the opinion of experts responsible for maintaining the standard, it has been superseded by a better approach. Deprecating a standard is always tricky and difficult because it means that users of what was previously considered the standard way of doing something have to be led gently, and usually over a long period of time, into making a transition to the newer method. It is never done lightly, and it always comes at a cost.

For this reason, you don’t launch the International Standardization of something that you already want people to stop using; that’s absurd. If you don’t think people should be using something, the last thing you want to do is make it a new standard. You don’t spend over 600 pages describing a format that you explicitly recommend not be used. And you certainly don’t standardize that format when a “newer and richer format” with better functionality is included in the same package.

The only conceivable justification for including VML in DIS 29500 is to serve as a target for the migration of legacy Microsoft documents. Take away that goal, and any reason for including VML — and uncounted other features of OOXML — just disappears.

As far as we can tell, all of the changes that would necessarily result from removing access to the legacy base as a goal can in theory be made through revision of DIS 29500. But beyond the specific points outlined above looms the larger question of whether JTC1 would then have any reason to consider OOXML for International Standardization.

If not interoperability with the binary formats, then what?

At the V1 meeting in Washington, an attempt was made to support the idea that we could just abandon the promise of open backward compatibility with the argument that the document that Ecma transmitted to JTC1 as an introduction to DIS 29500 was not actually part of the DIS and could therefore be ignored. But in terms of the transmittal to JTC1, this document, Ecma/TC45/2006/374, constituted the formal rationale for proposing that JTC1 consider the submission in the first place and put us through all this. If the motives we were formally given when accepting the fast track submission are now recognized as no longer operative, then JTC1 should halt the process and return the submission to Ecma so that we can start over.

To put somewhat the same point in less procedural terms: if the benefits of interoperable open access to the legacy are removed, then the only reason for the existence of OOXML is to provide a nice, open, full-featured XML format for the world to standardize on. Everyone agrees that a standard XML format for all the world’s office productivity applications would be a good thing, but we’ve already got one — ISO/IEC 26300, aka OpenDocument Format (ODF). So if all that’s intended here is a standard XML office format going forward, we don’t need to go to all the trouble of making another one. The sensible course in this case would be for *everyone* to join in SC34 and engage in the grand synthesis of ODF and UOF (the Chinese national document format standard) recently proposed by Sun Microsystems Chairman Scott McNealy.

While architected differently from OOXML, ODF and UOF are based on similar principles. If the objective is to create a standard markup that just does a good job of representing office documents in a vendor-neutral manner, then surely building on the work already invested in ODF and UOF is the best way to do that. In the absence of a set of objectives that would clearly distinguish OOXML from ODF and UOF, therefore, a genuine desire for interoperability would manifest itself in abandonment of OOXML in favor of joint work on this synthesis and not waste time and energy in the creation of a redundant and differently architected syntax for accomplishing the same thing.

Delivering on interoperability with the binary formats

As we’ve tried to make clear in V1, however, Sun does not support abandonment of the promises made for OOXML. We think that open, interoperable and full access to those billions of old Microsoft documents is a great idea. *If* that open access remains in the set of objectives actually realized by DIS

29500, and *if* all of its other defects can be repaired, then Sun would like to see the project continue and deliver on the promises made for it by providing a normative mapping between the legacy formats and OOXML. And this is what we said at the V1 meeting in Washington (as well as what we said in the passage quoted at the beginning of this paper, to the surprise of many who did not see it in its entirety).

Discussion at the Washington meeting made it evident that there is no technical impediment to providing such a mapping. Complete documentation of the Microsoft formats exists — indeed, one of the Microsoft representatives at the meeting noted that he had a copy on his hard drive — and has recently been made available to developers. And OOXML has been engineered from the beginning to be capable of completely and faithfully representing those formats. So with a complete description of the binaries on one end and a complete XML vocabulary designed to represent them on the other, all we need is a reference set of rules for getting from Point A to Point B so that everyone's software for providing access to legacy documents delivers the same results as everyone else's.

In the end, V1 agreed that keeping the legacy access goal and requiring the mapping was better than abandoning the only reason for creating a second International Standard for office document formats. The issue (as submitted by V1 chair Patrick Durusau) has been documented as follows:

Interoperability: DIS 29500 as written fails to establish interoperability between conforming applications that convert Microsoft Office 97 - 2003 documents to the format defined by this proposal. There is no defined mapping between Microsoft Office 97 - 2003 formats and the format defined by DIS 29500. In the absence of such a mapping, conversions to the defined format will be inconsistent [and] therefore lead to a lack of interoperability between the resulting conversions. Moreover, the lack of a defined mapping defeats interoperability between applications that use the defined format and the existing installations of applications that use Microsoft Office 97 - 2003 formats, resulting in a lack of interoperability with existing applications.

It was recognized that the standard mapping need not be physically included in the DIS itself, so the resolution of this issue agreed on by consensus of the members attending in Washington is that

DIS 29500 be amended to include a reference to a mapping from the Microsoft Office 97 - 2003 formats, to OOXML.

We endorse V1's resolution of this issue. Agreement to retain open access to the existing legacy of Microsoft documents as an explicit goal of DIS 29500 constitutes the primary reason that Sun still supports further work on the DIS, and our support for the eventual product will depend upon the existence of an approved and comprehensive mapping from the binary formats in addition to the repair of the many other defects present in DIS 29500 as it currently stands.

Summary

Our conclusion from the standpoint of technical standards definition is that a standard engineered for open interoperability of future documents and also open, interoperable access to legacy documents would have a defensible reason to exist and would not duplicate the function of the existing International Standard for generic office document formats, ISO/IEC 26300. But DIS 29500 as currently written is not that standard.

This document is made available under the Creative Commons 2.5 Attribution–Noncommercial–NoDerivs license.