

Delivered electronically

Juha Vartiainen
IT-standardisation
Finnish Standards Association SFS
PB 116
FI-00241 HELSINKI
FINLAND

Opinion regarding the “Standard format for document exchange and archiving” discussion paper

First of all I want to thank for an opportunity to give some constructive feedback for the discussion paper.

Document formats have raised some strong feelings after/during ODF and OOXML standardisation processes.

However, there is still need to continue standardisation related to document formats and the discussion paper ¹ prepared by SFS is a good starting point.

Thinking from policy execution, also called administration, there is still room for debates related to document formats. This discussion paper is a good starting point also to political and/or administrative debate, even though it is probably not the main objective.

Annex 1 holds information of copyright, licence and disclaimer.

Best Regards,

Jukka Rannila
citizen of Finland

signed electronically

¹ SFS, Standard format for document exchange and archiving, Discussion paper for comments, Draft: 2009-06-02, prepared by Tommi Karttaavi, Martti Poutanen and Juha Vartiainen (electronic document)

Information technology – short curriculum

As an idea information technology is quite simple. I have used following points to describe information technology systems:

- document, database or combination of document and database
- add data
- retrieve data
- change data
- remove data
- communications protocols of sending data to remote place
- communications protocols of retrieving data from remote place
- persons using system is classified to different classes
- administrator of the systems(s).

There is tendency to hide this simplicity of information technology when there is discussion and quarrel about programming languages, communications protocols, data format protocols, ownership of programs, licences, etc. etc.

The result of this discussion and quarrel is that information technology field is divided to many competing collections of persons and legal entities, i.e. companies, joint ventures, foundations and associations, etc.

General problem of information technology

The general problem for continuity of any [human] entity is clear. How to keep operations going on, when information technology is constantly changing?

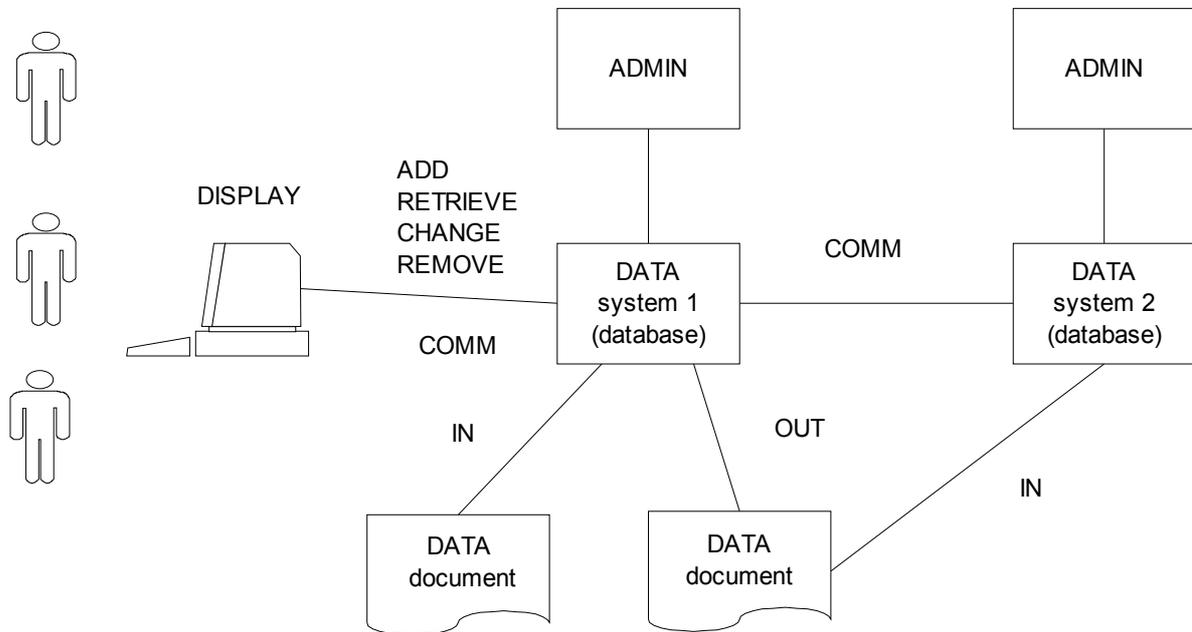
Successful commercial entities have sometimes an obligations that can last for decades. A good example might be a commercial nuclear facility, which can run literally decades, and last maintenance persons are not yet born when the nuclear facility starts its functions. When thinking for a while, there are other similar examples.

The problem for public sector is clear. Public sector units have an obligation to sustain certain activities as long as there is legislative foundation to have this activities. This means, that a certain public sector unit might be using certain information technology system long after the originating commercial company for that information technology system might be disappeared.

In both cases, public and private, there is need for using documents for several decades.

Basic ideas in information technology – a picture

Sometimes it is said that a figure can explain something better than words. I try that in the following figure.



Assessing the situation, based on the discussion paper

In the discussion paper is the following assessment:

Current document format standards (OOXML, ODF) do not take these considerations into account in a meaningful way:

- they are focused on the presentation of the document on the expense of the content.
- the XML format of the document must also support the full functionality of the editor
- which can make the XML structure very complex and prone to changes when the application evolves.
- the complexity may cause information loss over decades of storage.

(copied from discussion paper)

When thinking more carefully, it can be said, that ODF and OOXML are Data Display Standards. Like the picture indicates following actions are possible with Data Document:

- adding data
- retrieving data
- changing data
- removing data.

Then these Data Documents can be transmitted between systems, IN-OUT-IN.

ODF and OOXML

At this point I have to make an assumption that the reader has at least some understanding what are ODF² and OOXML³ standards. If not so, then the reader is advised to read some basic information about these standards.

Admin(istrator)-to-Admin(istrator) communications

When looking the picture there is two ways for information system administrators to communicate:

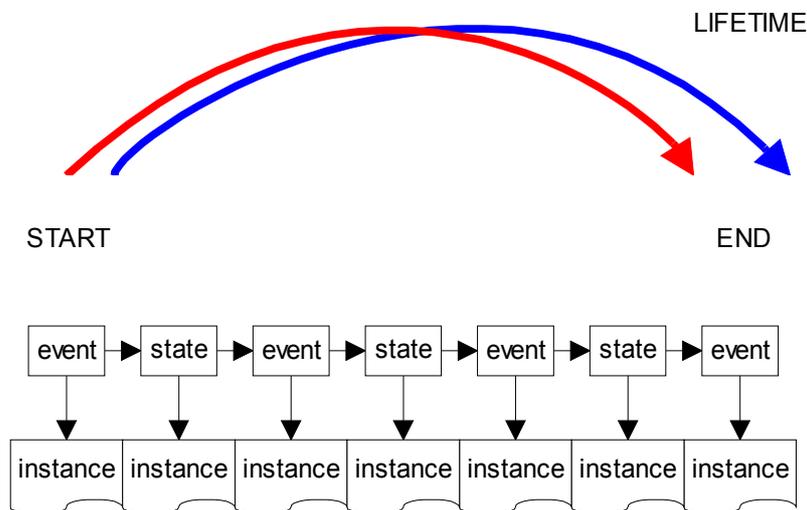
- system-to-system communications
- system-to-document-to-system communications.

I conclude that in this case we are talking to system-to-document-to-system communications, also IN->document->OUT->document->IN.

Events, states and lifetime

Now we can add following concepts to our picture:

- events
- states
- lifetime.



In practical reality there is events and states changing all the time in large data(base) systems. It is totally possible to have a document (instance) from every change in states and events.

² ISO/IEC 26300:2006 Open Document Format for Office Applications (OpenDocument) v1.0.
e.g. <http://en.wikipedia.org/wiki/ODF>

³ ISO/IEC 29500:2008, Office Open XML.
e.g. <http://en.wikipedia.org/wiki/OOXML>

When thinking juridical agreements this division is quite clear:

- every agreement has a starting point (start) and an ending point
- during an agreement there can a wide variety of events
- it is possible to change the agreement in every event
- it is possible to have both an electronic and paper document (instance) from every event and state.

What is actually the problem?

In the discussion paper there is following assessment:

Problem [in the discussion paper]: Organisations and individuals should be able to exchange documents regardless of the software the documents have been created with. On the other hand, certain documents need to be archived for long periods time, even hundred years and over. There is also a need for adding semantic information to the documents in a machine readable format. For example, there could be a classified section in an otherwise public document. The confidential part of the document needs to be marked, so that it will not be presented in the published version. In this paper the ability to exchange documents is focused on data exchange between information systems, not office applications.
(copied from discussion paper)

As can be seen this is clearly ADMIN-to-ADMIN communications through documents.

I see that the actual problem is in **nature of electronic document instances.**

In paper documents every instance can have its history, states and events, marked physically to the paper document itself. In electronic document instances there is no limit to distribute instances, since copying electronic documents is relatively easy. When thinking forward, from one juridical agreement there might several electronic document instances, and they might be in several computers.

Therefore, when opening an electronic document, the person using an electronic document (instance), she/he must know state and/or events related to the document.

In other words, **an electronic document instance should “know” its events and states.**

Division between data and programs

The division between data and program has been the main issue for decades.

In ODF and OOXML standardisation the main issue has been that can we:

- have ODF and OOXML documents,
- have several programs, and
- several programs could use both ODF and OOXML documents?

After some considerations I have started to think about a combination of document and program, so called **document-program**.

Document-Programs

I think that I am not the first person think about combination of document and program. But when there is an invitation to discuss with a discussion paper, why not then think aloud something.

What this kind of Document-Program should contain? Supposedly following:

- intelligence part (program and program code)
- communicator part (COMM)
- state and event data part (META-DATA)
- data part (DATA)

Relating document-programs to the proposed solution

In the discussion paper there is following proposal:

Proposed solution [in the discussion paper]: There should be a XML-based document standard that keeps the document in the simplest possible format without layout information. The document semantics are captured to proper metadata model that stores the document type, author, dates etc. Document type could be used to re-create the semantics (and the styling) of the generic content elements (e.g. sect1/title in a board meeting memo).
(copied from discussion paper)

Lets no relate my proposal of document-programs to the proposed XML-base document standard(s).

Intelligence / Programs

When thinking intelligence of an XML document, there is none.

XML documents can be very sophisticated, but there is no intelligence in XML documents, and therefore there is always the need for a program to create something intelligent from XML documents.

Since XML documents as such are dumb, they don't know their previous states and events. Event and state information must be added by programs.

Communications

When thinking XML document for communications, it is quite useful tool. Generally speaking XML documents are quite easily transmitted between different systems.

But when thinking intelligent communications, there is need for human intervention in several points.

State and event data (also called meta-data)

When there was no electronic documents, it was possible to add state and event data physically to the paper document.

It is possible to add state and event data to an electronic document, but the problem is with instances, since the same state and event data is not transmitted to all instances. An document instance should “know” when there is new meta-data to be added.

Data

This is quite obvious, since a document is carrying data, being it paper or electronic document.

When thinking an XML document, it is relatively easy to change data in the document. But the problem is once again, that there should be information, which denotes the need for updating only one instance or all instances.

How would a Document-Program work?

Now we have defined a Document-Program, which would have four parts.

Data	Meta-Data
Communications	Program Program code

Document-Program

According to my proposal it would be something like this.

1. A data document is created, e.g. ODF or OOXML.
2. A Document-Program is initiated.
3. Data from the data document (ODF/OOXML) is added to the Document-Program.
4. Meta-Data is initiated in the Document-Program.

Then the key issue should be decided.

5. Is this Document-Program created unique?

The need for the broker?

You can not hide it, you can not run away from it. Its the broker or middleman, which is needed in many information technology solutions?

Since Document-Program is not a new idea, there are commercial solutions where there is a broker somewhere, and then Document-Programs can be used efficiently.

Now we must suppose that we are thinking also solutions that are not purely commercial.

Need for universal broker / uniqueness ?

Without broker there is no way to determine that some electronic document is unique. We have already mentioned that copying electronic documents is relatively easy.

Since we have to make the key decision, we decide that this Document-Program is unique. This will lead us to the following phase.

6. Document-Program gets unique identifier from the universal broker.

We can try all kind of transactions between XML documents and programs, but they are always separated and there is no way to have certainty of the uniqueness of certain electronic document.

Not-unique Document-Programs

What would not-unique Document-Programs then be? With these there can be communications with different rules:

7. One Document-Program is unique: all data and/or meta-data can be communicated between instances.
8. There is no unique Document-Program: data and/or meta-data is communicated between instances.

Rules

Then there is obvious part to unique and non-unique Document-Programs.

9. There are different rules (Meta-Data) to handle data.

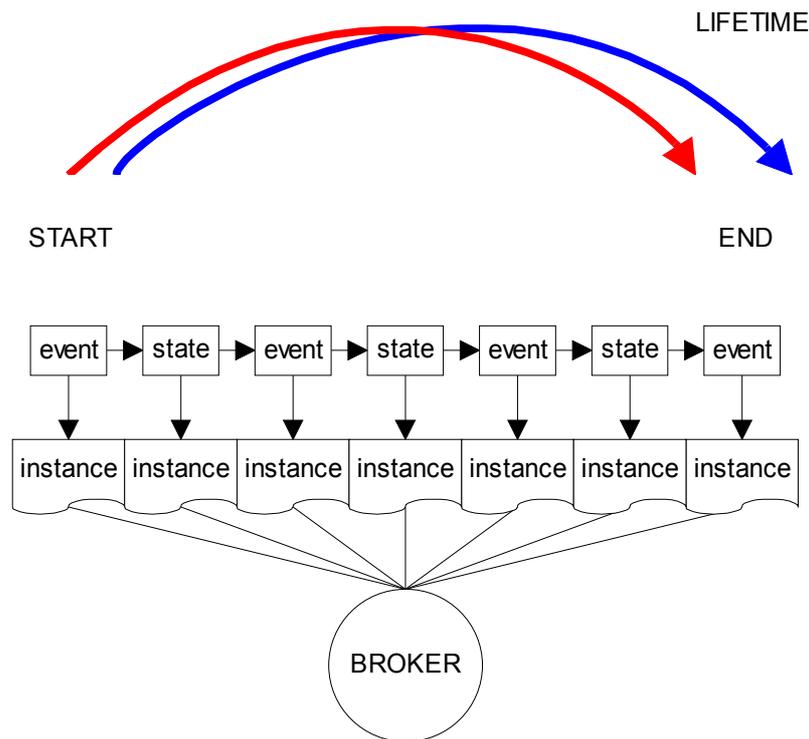
Is it more of creating the universal broker?

The problem with creating these brokers is, that creating one universal broker is sheer impossibility, since it should reach through all countries in the world. Knowing the complexity of the world, it is

quite feasible to think that one universal broker is close to impossibility.

Therefore we have to conclude that there will be several brokers, i.e. federated, and their co-operation is a possibility or an option.

In simple form there would be only one broker and one kind of Document-Programs.



Back to the practical reality?

When thinking the practical reality and the proposal of Document-Programs, there are really major problems.

- I) Joining documents and programs would mean more security risks, since there would be constant communications with the broker or brokers.
- II) Joining documents and programs would mean more risks, i.e. these combinations become obsolete easier when programs and documents are separate.
- III) Brokers have their own lifetime, and when a broker ceases to exist, this leads to several problems with Document-Programs.
- IV) Computers develop and then Document-Programs are more easily locked to certain technological combination than pure Documents.
- V) Neutrality of a broker can be a matter for really complicated disputes.

Proposal / A way forward

In the discussion paper there are following conclusions.

Conclusions [in the discussion paper]: an international project to define the base XML-schema for document exchange and archiving should be started. The possibility to use DocBook as the starting point should be explored.
(copied from discussion paper)

When relating these conclusion to Document-Programs proposal, we can make following proposals:

1. DocBook is a good start to have data in Document-Programs.
2. For meta-data there should be separate standard, or a major extension of DocBook.
3. For rules handling data and meta-data there should separate standard, or a major extension of DocBook
4. Program part of Document-Program have to be interchangeable, i.e. the program part of a Document-Program removed and replaced.
5. Communications part of Document-Program have to be interchangeable, i.e. the Document-Program can use different communications methods.

What this would mean in practise?

1. Data part could have public and confidential parts, confidential parts encrypted.
2. In meta-data there is information of document history, states and events.
3. Meta-data part could have public and confidential parts, confidential parts encrypted.
4. Program could be changed, e.g. when Document-Program is moved from a (traditional) computer to a hand-held device.
5. Communication method could also be changed, e.g. (traditional) computer might have different communication standards than hand-held device.
6. If there is a broker or brokers needed, in data and meta-data there is information about communication rules with the broker(s).

What about standardisation of Document-Programs? According to my understanding, complete standardisation would take several years. There reasons could be following:

1. Data part of Document-Program could be defined in a shorter timetable.
2. Meta-data part of Document-Program could take more time, since there is a wide variety for rules handling state and event data.
3. Encryption methods need some time more, since this means defining conformance of different programs with these encryption methods.
4. Creating Document-Program rules where a Program could be changed in a Document-Program means more standardisation
5. Corollary creating communications part in the Document-Program means more standardisation.

Conclusions

Based on these thoughts my conclusions are following:

- a) An international project to define the base XML-schema for document exchange and archiving should be started.
- b) An international project to define the meta-data rules could be started, or it is extensions to the previously mentioned base XML-schema project.
- c) Based on these two international projects, there should also be project to create conformance methods for computer programs.
- d) There could be a series of seminars to debate on Document-Programs.
- e) If there is a concrete need to have Document-Programs, then the Document-Programs standardisation project can be started after serious deliberation.

When thinking previously mentioned problems (I-V) with Document-Programs, a series of seminars (d-e) should concentrate on solutions for these problems.

Good luck!

Standardisation project does not mean, that the end result, i.e. standard, will be used everywhere.

Proposal to have an international project to define the base XML-schema for document exchange and archiving is a good idea. Unfortunately we do not know beforehand, if there will be a “standard” or another standard, which will implement the idea regardless of the well-meaning standardisation project.

But we must try standardisation, still knowing the risks ahead. After all, information technology is almost always about standards.

With kind regards,

Jukka Rannila
citizen of Finland

signed electronically

ANNEX 1

DISCLAIMER

Legal disclaimer:

All opinions in this opinion paper are personal opinions and they do not represent opinions of any legal entity I am member either by law or voluntarily. This opinion paper is only intended to trigger thinking and it is not legal advice. This opinion paper does not apply to any past, current or future legal entity. This opinion paper will not cover any of the future changes in this fast-developing area. Any actions made based on this opinion is solely responsibility of respective actor making those actions.

Political disclaimer:

These opinions do not represent opinions of any political party. These opinions are not advices to certain policy and they are only intended to trigger thinking. Any law proposal based on these opinions are sole responsibility of that legal entity making law proposals.

These opinions are not meant to be extreme-right, moderate-right, extreme-centre⁴, moderate-centre, extreme-left or moderate-left. They are only opinions of an individual whose overall thinking might or might not contain elements of different sources. These opinions do not reflect past, current or future political situation in the Finnish, European or worldwide politics.

These opinions are not meant to rally for a candidacy in any public election in any level.

Content of web pages:

This text may or may not refer to web pages. The content of those web pages is not responsibility of author of this document. They are referenced on the date of this document. If referenced web pages are not found after the date when this document is dated that situation is not responsibility of the author. All changes done in the web pages this document refers are sole responsibility of those organisations and individuals maintaining those web pages. All illegal content found on the web pages referenced is not on the responsibility of the author of this document and producing that kind content is not endorsed by the author of this document.

COPYRIGHT

This opinion paper is distributed under Creative Commons licence, to be specific the licence is “Creative Commons Attribution-NoDerivs-NonCommercial 1.0 Finland”. The text of the licence can be obtained from the following web page:

<http://creativecommons.org/licenses/by-nd-nc/1.0/fi/legalcode>

The English explanation is in the following web page:

<http://creativecommons.org/licenses/by-nd-nc/1.0/fi/deed.en>

⁴ Based on the Finnish three-party system there is phenomenon called extreme-centre in Finland.