Ten good reasons why an HL7-XML message is not always the best solution as a format for a CDISC standard (and especially not for submission data)

Jozef Aerts, XML4Pharma
Version 0.5, November 2008

Introduction

CDISC standards have traditionally been based on XML technology, developed by teams of volunteers includung XML experts[1]. Examples include the CDISC Operational Data Model (ODM) standard, the Case Report Tabulation Data Definition Specification Standard (CRT-DDS), better known as define.xml, and the CDISC Lab standard, which has an ASCII implementation, a SAS implementation and an XML implementation.

Not very long ago, CDISC decided to develop the technical implementation of all new standards as extensions to the ODM standard. The most obvious reason for this is that the ODM standard is very succesfull in keeping as well metadata (such as study design) as clinical data[2]. As submission data (standardized according to the Submission Data Tabulation Model (SDTM) **is** clinical data, and a standard format has already been established (as an ODM extensions) to hold the submission metadata (define.xml standard), the most logical step was therefore to also develop an ODM-based format for holding the submission data itself. This format has been developed by a group of volunteers, consisting of as well submission specialists as highly skilled XML specialists, but has not been published by CDISC yet.

A current tendency however is being observed for wanting to format all new CDISC standards as HL7-XML messages. This tendency is driven by the desire for integration with the healthcare world, where HL7 is well-established, and by the FDA, who already uses a number of HL7-XML based standards, such as the annotated ECG standard (aECG). This idea is especially supported by people that have never been actively been involved in XML development.

To my personal opinion, and to the opinion of many other XML specialists, this is a tendency that is at least concerning. This article will list and discuss ten good reasons why HL7-XML messages should in many cases not be considered as being the best format for new CDISC standards, and especially cannot be a good format for holding CDISC submission (SDTM) data.

[1] Several CDISC volunteer teams have “XML-gurus” in their ranks, with considerable records of service in XML technologies

[2] But also reference data and clinical trial administrative data.

Ten good reasons

1. An HL7-XML message is not compatible with define.xml

The wonderful thing about the ODM standard is that it works as a “framework”. It is perfectly suited to hold as well study metadata as clinical data, in such a way that the clinical data can easily be validated against the metadata. Also define.xml, the well-established (and by the FDA embraced) standard for SDTM submission metadata is based on this concept. Developed as an ODM extension, it holds the metadata of the SDTM submission. The submission data themselves still need to be delivered in the legacy SAS Transport 5 format.
If the submission data itself however comes as an HL7-XML message (which has completely different concepts than define.xml), how will it be possible to validate the submission data agains the metadata? Of course one can argue that it is always possible to compare apples to oranges, but it will considerably more difficult. Writing software to validate ODM clinical data against their metadata is pretty easy, especially when using modern (but easy-to-learn) XML-languages such as XPath and Schematron. With an HL7-XML message however, these technologies will make little chance, and other, much more complicated technologies will need to be used.

2. HL7-XML messages take years to develop

CDISC volunteers (under which a few XML-veterans) have already developed an ODM-based extension to hold SDTM submission data that easily validate against define.xml metadata. This model could in principle be put into operation within a year, thus replacing the legacy SAS Transport 5 format. The latter still stems from the seventies, and is a binary format based on IBM mainframe technology, also completely outdated.
The latter is a binary format from the seventies, based on IBM mainframe, so completely outdated.
It is well known that HL7 messages take years to develop and to get approved. The FDA itself has foreseen that, in case an HL7-XML message is being developed for transporting SDTM data, the SAS Transport format will at least remain in place until 2015 or even later[3]. With all the problems the outdated SAS Transport format delivers, this is a timescale the industry cannot afford.

3. Many HL7-XML messages are overcomplicated

Those who have ever inspected an HL7 aECG-XML file in detail, may have been surprised by the enormous complexity of the XML. One of the reasons for this is that the XML structure (as defined in its XML-Schema**)** is not developed by XML specialists, but is derived from UML diagrams.

I have been teaching a lot of XML in the last ten years and have experienced that CDISC ODM can be learned in a one day course. No chance however to accomplish this with aECG. Therefore, the amount of people that really understand aECG-XML is very limited, this in contradiction to the amount of people that understand ODM-XML.

Similarly, though XML is usually defined as being “as well machine-readable as human-readable”, one may question whether the latter is applicable to some HL7-XML messages such as aECG: not only the complexity is overwhelming, but is also uses a lot of coded, ununderstandable for the human reader.

In 2006, Gartner issued a Note entitled "HL7 V3 Messages Need a Critical Midcourse Correction", stating that “*HL7 must act vigorously to make Version 3 messages easier to use and more compact*”

[3] My personal estimate is 2020.

[see e.g. http://2006.xmlconference.org/programme/presentations/141.html].

The direct consequence of this overcomplexity is that it is much harder (and thus much more expensive) to develop software to read and write HL7-XML than it is for ODM-XML. From my personal experience (30 years) in software development, I estimate that the cost of developing software for a complex HL7-XML message is at least the twentyfold than it is for ODM-XML.

Some critics of HL7 v.3 even started a blog-website, which highlights many of the problems with HL7 v.3. It can be found at: http://hl7-watch.blogspot.com.

Fortunately, not all HL7-XML messages are overcomplicated. A good example is the XML implementation of the CDISC Lab standard. Although it has some minor design errors, it is pretty easy to handle and to understand. So, if it is really necessary to develop an HL7-XML message for SDTM, it should not be more complicated than the CDISC Lab-XML standard.

4. XML is not UML

HL7-XML messages are developed in a somewhat curious way: first of all one or more UML diagrams are developed, and then the XML-Schema is derived from the UML. The UML is derived from the RIM (HL7 Reference Information Model) which currently has over 70 versions (!)[4].

Though the use of UML may be the perfect and well-established way for translating a software design to software classes, it is considered bad practice by XML specialists. Transformation of UML to XML-Schema in general leads to “spaghetti XML”, introducing unnecessary complexity. Of course it is an “easy” way: the world has much more UML specialists than it has XML-Schema specialists. Personally, I would consider transformation of UML to XML-Schema the “lazy man's way”. The result can however be catastrophical.

By the way, none of the most popular XML-based standards, such as MathML, VoiceXML, XHTML or XForms etc. have ever been developed using UML.

A discussion about XML-UML issues for HL7 messages can be found on the W3C website.

5. HL7-XML is nearly used in the industry

HL7 messages are pretty succesfull in the healthcare industry, especially in Northern America. However, when one looks in more detail, it is observed that over 99% of the implementations are HL7 version 2 implementations (not using XML), and that less than 1% are version 3 implementations (using XML). So the “market share” of HL7-XML within HL7 is less than 1%.

[4] The RIM itself is highly criticized by some IT specialists, see e.g.: http://ontology.buffalo.edu/smith/ppt/MIE/HL7_RIM_MIE06.ppt, http://ontology.buffalo.edu/HL7/doublestandards.pdf, and http://aurora.regenstrief.org/~schadow/Schadow-MIE06-r3.pdf

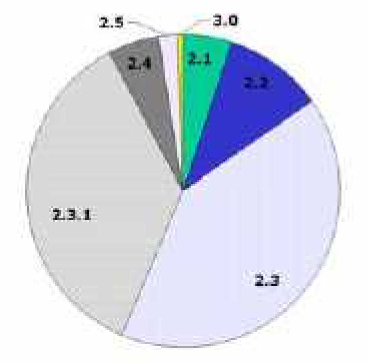


Figure 1: Approximate real-world usage of HL7 messaging standards. The vast majority of HL7 messaging is done using messages that approximate HL7 2.3 or HL7 2.3.1. Newer releases of HL7 (2.5, 3.0, and soon 2.6) represent a very small portion of real-world interfaces.

[REFERENCE: http://www.neotool.com/pdf/HL7-Version-3-with-HL7-Version-2-History.pdf].

Reasons mentioned for this relative non-success of HL7 version 3 are the enormous cost and lacking backward compatibility with version 2. Software developers and system architects that have succesfully implemented one the version 2 HL7 standards, have great difficulty with understanding and implementing version 3 HL7-XML (see e.g. http://albertderoos.nl/?p=106).

The “market share” of HL7-XML might further decrease, as many new HL7 messages are developed and further maintained as HL7 v.2 messages (non-XML)[5].

6. In many cases HL7-XML ignores existing XML standards

When developing the CDISC ODM standard and its extensions, the CDISC ODM team has always looked at other XML-based standards, and has always tried to prevent reinventing the wheel. An example is the implementation of the XML-Signature standard within the ODM, but also the fact that all ODM data types are based on native XML-Schema data types. For example, an ODM “date” is based on the native XML-Schema datatype “xs:date”, which is based on a subset of the ISO-8601 standard. Similar for time, datetime, incomplete-datetime, partial-datetime, and durations. In too many HL7-XML messages however (there are exceptions), all these are defined as being of data type “text”, making it very hard to validate instance data against the XML-Schema.

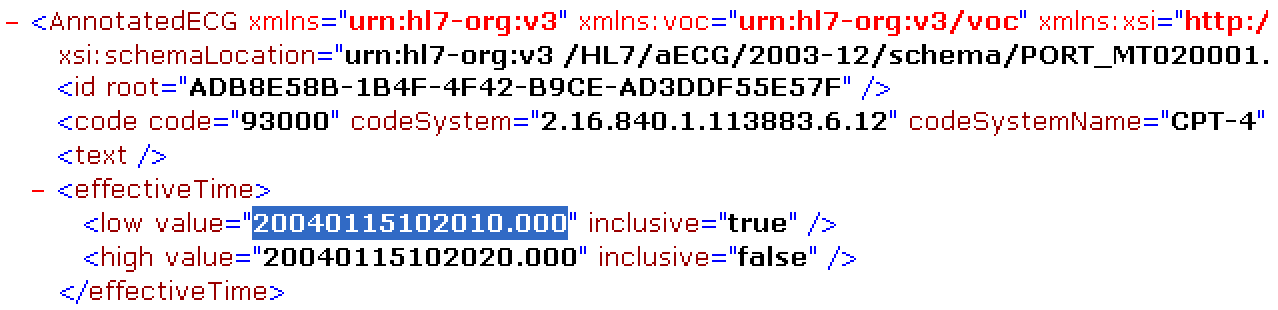
For example, accoring to the XML standard, dates should be formatted as YYYY-MM-DD. This has the great advantage that for example, a date “2008-02-30” will immediately fail when validating against the (ODM) XML-Schema (without writing any special software). A date “20080230” however (not standard-XML, but observed as date format in many HL7 XML-messages) is much more harder to detect as being an invalid date (special software is required).

ODM however has even defined some other ISO-8601 based XML data types, such as incompleteDataTime and partialDateTime, which can be of extreme importance for SDTM submissions in XML. In HL7-XML, these are based on string patterns, with the same limitations in validation.

In some HL7-XML messages we have even observed that two different ways of representing dates are being used in the same file – so the implementation is even not consequent.

Similarly for “date-time”: according to the XML standard, these should be formatted as YYYY-MM-DDThh:mm:ss. In many HL7-XML messages however, the XML standard is simply ignored, and date-times are declared as being of type “text” (instead of of type “datetime)”, making them unvalidatable without writing special software.

[5] Remark that HL7 v.3 is not downward compatible with v.2



```
- <AnnotatedECG xmlns="urn:hl7-org:v3" xmlns:voc="urn:hl7-org:v3/voc" xmlns:xsi="http:/
    xsi:schemaLocation="urn:hl7-org:v3 /HL7/aECG/2003-12/schema/PORT_MT020001.
    <id root="ADB8E58B-1B4F-4F42-B9CE-AD3DDF55E57F" />
    <code code="93000" codeSystem="2.16.840.1.113883.6.12" codeSystemName="CPT-4"
    <text />
  - <effectiveTime>
      <low value="20040115102010.000" inclusive="true" />
      <high value="20040115102020.000" inclusive="false" />
    </effectiveTime>
```

Incorrect formatting of date-time
extensive use of codes (without any codelist)
human-readable ?

This ignorance of existing XML standards may be a consequence of the fact that HL7-XML schemas are "derived" from UML, rather than developed from the ground up by XML-Schema specialists.

7. Why make it complex when it can be simple ?

There is no need for having SDTM submission data in a complex data structure. The SDTM standard describes all submission data as simple two-dimensional tables, not as complex objects. So a relative simple ODM-based data format (similar to define.xml and ODM-ClinicalData) is more than sufficient as a vehicle for porting SDTM data to the FDA. Arguments for an HL7-XML message for SDTM are based on the desire for integration with the healthcare world, where patient data are usually exchanged using HL7 messages. SDTM data are however much more simple, and can easily be transformed (if necessary at all) from complicated HL7 messages with patient data to the much more simple ODM format using XSLT stylesheets.

SDTM does not have the concept of "actors", "roles", ... SDTM data are simple two-dimensional tables with some table and column attributes (the metadata in the define.xml). So designing an HL7-XML message for transporting is a complete overkill.

8. Complex data structures can be transformed to simple datastructures by XSLT, the reverse is much more difficult

Operational clinical data are usually exchanged using the CDISC ODM, XML-based standard. This standard is now in its version 1.3, and is continuously improved by an enthusiastic group of volunteers, containing as well clinical data specialists, as XML specialists. This standard is embraced by the FDA, as it fully CFR Part 11 compliant, including audit trails and signatures. The ODM structure is however, compared to the HL7-XML structure, uncomplicated and not complex.

Several vendors[6] have already developed ODM to SDTM mapping tools, and have implemented an XML-based SDTM format, based on the ODM. As such, transforming operational clinical data (ODM) to such (ODM-based) SDTM-XML data is uncomplicated and straightforward. These vendor tools are even able to generate the XSLT transformation code to execute the transformation in an automated way!

XSLT transformation from a non-complex format to another non-complex format is easy. XSLT transformation from a complex format (such as HL7-XML) to an uncomplicated XML format is also

[6] Including XML4Pharma, XClinical and Formedix

pretty easy. Transformation from a non-complex format (such as the ODM) to a complex format (HL7-XML) however is very complicated. Automated generation of XSLT code to transform operational clinical data (ODM) to an complex HL7-XML message is probably a programmer's nightmare, meaning that mapping software will be very complex and very expensive.

9. CDISC standards are owned by CDISC, HL7 messages by HL7

The SDTM standard is a development by CDISC, not by the FDA nor by HL7. Though CDISC and HL7 have an “Associate Charter Agreement” (meaning a strong relationship), there may be ownership issues when an HL7 message is developed for carrying SDTM data. Who would be the owner of such a format? Would it be CDISC, who clearly is the owner of SDTM, or would it be HL7? Probably the latter. Or would it even be the FDA? This would lead to the strange (and dangerous) situation that the owner of the content standard and the owner of its transport format are not the same[7].

10. There is little or no XML knowledge at the FDA

The desire to have an HL7-XML message for submission (SDTM) data clearly comes from the FDA [http://www.fda.gov/CDER/REGULATORY/ersr/2003_06_17_XML/sld001.htm]. The amount of knowledge of XML technology (and especially XML-Schema technology) at the FDA is however very limited, if ever existent. Therefore we must suppose that the choice for an HL7 message is a political one rather than a technical or pragmatic one.

This means that the FDA will essentially obtain a “black box” from HL7, with the certainty that they will need to heavily rely on external parties for the development and deployment of software tools for working with the SDTM HL7-XML message[8]. For example, for viewing SDTM data in a tabular way, they will need extremely complicated stylesheets, or even very special software.

With ODM-based technology however, this danger is much smaller. For example, the CDISC define.xml team has developed several uncomplicated stylesheets to view the SDTM metadata in a standard browser. These stylesheets are being used by the FDA, but have also been refined by many sponsors. Similarly, several vendors have developed viewers for inspecting ODM clinical data, based on these simple stylesheets. If the SDTM data themselves come as an ODM-extension, the same technology can be used to visualize SDTM datasets, and the wheel does not have to be reinvented or redeveloped. Additionally, the metadata and data can be visualized together (define.xml and SDTM-ODM-XML) as they are based on the same base standard. Establishing the same for HL7-XML based SDTM datasets is probably an illusion.

An argument from the FDA for HL7-XML is that it is a common format for all information send to the FDA (from ICSR to aECG to Lab), for import in Janus, the FDAs warehouse. Whoever compared an ICSR file with an aECG file however (I encourage the reader to do so), will not find many things in common[9]. The only thing these have in common is that they are both totally different interpretations of the same high-level model.
So the argument of a common format is definitely a false one.

[7] This is not the case for the CDISC Lab standard which was developed by CDISC as an HL7 message. As it was developed by CDISC, the owner is CDISC.

[8] This was exactly the case with the aECG HL7-XML message. It was developed by B.D.Brown from Mortara, and F.Badilini from AMPS llc. The input of the FDA was limited to writing the user requirements.

[9] Ununderstandable for example is that ICSR does not use a namespace, whereas aECG does.

What is then the value of HL7-XML messages?

There is certainly a great value in HL7-XML messages. HL7 messages try, with a good amount of success, to capture the whole healthcare information and data streams. The healthcare world is a complex world, one of the reasons why HL7 messages are so complex. Very probably, electronic health records (EHRs) can never be described using a non-complex format like the one used in the ODM standard. For this, a system based on HL7-XML messages may be a good choice[10]. More and more countries are however deciding that the alternative OpenEHR standard is a better choice[11]. Assuming however that an HL7-XML message is therefore also necessary for operational or submission clinical data is an error. Operational (ODM) and submission (SDTM) data in XML format can better be transformed from EHRs using a complex-to-simple transformation (probably using XSLT), rather than using a complex-to-complex transformation[12].

Also HL7 is very important to CDISC for semantic interoperability. If information from EHRs need to be transformable into clinical data, it is very important the EHR and CDISC standards "speak the same language". This however does not mean that they need to use the same format.

Future articles

We are currently preparing a second article in this series. It will be named "**Ten things HL7 should do to make their HL7-v3-XML messages XML-compliant**". It will show, from the XML perspective, how HL7-XML messages can be made less verbose, more compact, better self-describing, and especially, compliant to the latest W3C XML standards.

[10] HL7-v3-XML was however never designed for EHR. This, and the XML-related problems with HL7-XML can be the reason for the fact that more and more countries are chosing for using OpenEHR instead of a system based on HL7-v3 for their EHR systems.

[11] Including Australia, Sweden, Denmark, ... Furthermore, a recent article on ClinPage.com states that the US is tremendously behind in EHR-implementation relative to Europe, where most countries are tending towards OpenEHR.

[12] I doubt whether EHR patient data will ever have to be directly transformed into SDTM data. Transformation to operational data (e.g.ODM) is straightforward, but transformation to SDTM data always involves an interpretation step.