Models, APIs and FHIR. A New World

Dave Iberson-Hurst, 23rd July 2020

Change History

- 11th July 2020 – Initial Draft
- 14th July 2020 – Second Draft
    - Updated with some comments from early reviewers
    - Initial references added
- 16th July 2020 – Third Draft
    - Fixed a typo
- 23rd July 2020 – Fourth Draft
    - Updated after further review comments received

First Off

Firstly, this discussion note is written by myself and no one else. There will be a lot of "I" in it. It is my personal opinion and I make no apology for that. Hopefully it can be a source of a discussion that may just result in something tangible to help us all.

Why

This week has been an interesting one. I sit here on a Saturday morning and my mind is full. It is overflowing with "stuff". Too many threads are running and the brain has reached capacity. It all needs to be written down such that order can be re-established.

It results from a week full of discussions. As always, many are about the day job: metadata, clinical studies, building them, automation, better ways of working. There have been discussions with customers, different viewpoints, solving today's needs and pain points; a dose of reality. I also took part in an interesting discussion with some people from Transcelerate. As always, such discussions bring you back to the reality of simple issues that we thought were long solved.

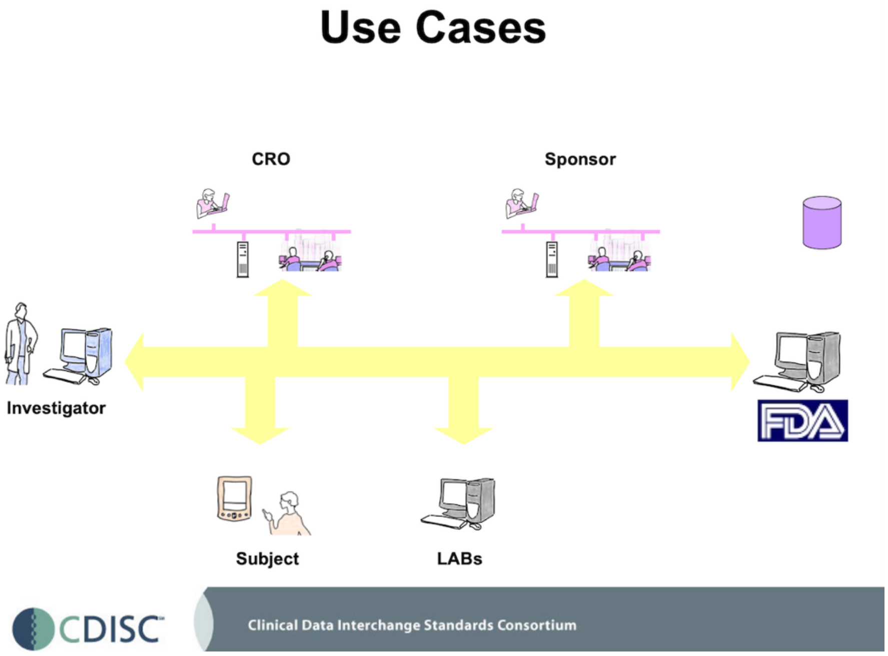
We still suffer, as an industry, from a lack of integrated systems, to readily exchange information amongst ourselves. We live in a batch world of file transfers, of sending SAS XPT files or large XML documents from one system to another, albeit electronically. We are unable to readily pool data without massive effort.

I have joked that our biggest enemy is the individual SAS programmer, one because they are cheap compared to the cost of a clinical study and two because they will implement some data transformation differently from the colleague five miles down the road working on the same problem but in a different sponsor company. We need consistency.

The Notion

During one of the conversations, I mentioned the notion of a data bus, something I am familiar with from the world of military systems in which I worked long ago. The procurer of the aircraft, the government, issues specifications that I, as a supplier, must meet. There is a physical hardware connection and a message specification that details the protocols and message content that will flow across that interface. My super new wizzo avionics box must do its job but it must also conform to the spec to allow it to communicate with the rest of the systems located within the airframe. The government can buy the component from me, or another supplier, as long as they both meet the spec. Standards are imposed from upon high.

While talking, I remembered a presentation from long ago that I subsequently dug out. I had suggested a similar approach while presenting at the FDA. The slide that I remembered is shown below. At the time it was all about ODM; CDISC's vision at the time was a world based on ODM. It had some technical validity but there was more than a pinch of politics in play. I tire of the politics.



Use Cases

CRO

Sponsor

Investigator

FDA

Subject

LABs

CDISC

Clinical Data Interchange Standards Consortium

Figure 1 - An Old Slide

I then looked at the date of the presentation. November 2004. 16 years have elapsed, and I feel, in some ways, we have not progressed. We have a lot of point solutions. SDTM has since been adopted by the FDA, define.xml, CDISC Library – big step forward there – HL7 FHIR but we still lack a coherent approach.

CDISC℠

Setting the
Global Standard
for Clinical Data

CLINICAL DATA INTERCHANGE
STANDARDS CONSORTIUM

FDA-CDISC Meeting
CDISC Operational Data Model (ODM)
4 November 2004

Dave Iberson-Hurst
Founder, Assero Limited

Figure 2 - A Long Time Ago

Since 2004 we have seen the emergence of new technologies – REST, JSON, Graph DBs - that can help along with some clarity of thought. Since 2004, we have learnt, we are more experienced, we have suffered the failures such that I feel that the time maybe right to have another go.

Some Detail

The idea, and it is not new, is that we evolve, as an industry what we build as standards. We view the clinical development data lifecycle as a data bus. You can start it at one end, as early as you like, but for the moment let's consider human clinical studies.

Every system has a need for data to perform its role. The system hopefully adds value and it creates outputs. Several vendors have similar systems but today each will use a different set of input and outputs. As an example, take EDC systems. We have Rave ALS files, we have ODM, we have tabular structures exported containing data. I know from experience that there are a variety of ways in which ODM is used to transport the same information with fields being overloaded in use; there are vendor specific extensions. ALS is a proprietary, if well understood format.

And with EDC we have protocol systems, we have study build systems, we have safety systems, we have regulatory systems. All share information or require subsets of that same information: a treatment, a study subject, an adverse event, the list goes on.

So, consider a simple example. Each system wants to inform or be informed about a Study. We have no method today to pass that information across all systems in a consistent manner. Today, many such transfers or interfaces will be custom-built point to point interactions. We need to move away from the siloed, single application view of the world.

So, employ the notion of a data bus. We use RESTful API technology to develop a consistent API for a Study to allow that information to be passed to and from systems that all vendors can implement.

To build such messages, we need to understand the content: what is a study and what are the relationships with other information of interest, e.g. a protocol? We need models. CDISC built, in collaboration, the BRIDG model. It is a good resource, but adoption has not been great due to, in my opinion, its size and the fact that it is difficult to understand and deploy; it is simply difficult to consume and users get overwhelmed by it. That said, it is underpinned by a significant amount of good thinking.

We need a reference model. I have long believed we need this so that we know how data relates. I saw the quote from Tony Seale [6] "The truth is meaningful data doesn't exist in isolation; everything is positioned within the context of everything else". In the past, I have illustrated this by asking people where they would place a Post-it that represents a lab test on a large blank whiteboard. Of course, there is no wrong answer. Then ask them to position a second Post-it that represents a vital sign observation in the correct place relative to the first lab test. Is it the same subject? Same Study? Same visit? You cannot do it. We need a framework to assist us to position these items. Much of the time this is a human reading a printed copy of the protocol and the annotated CRF.

And this is where I think we are entering Enterprise Knowledge Graph (EKG) territory [9]. This is where I feel we need to bite the bullet, we need to transform the knowledge in BRIDG into something simpler, readily consumable and able to meet our needs. I want to dig into this more, to learn about EKG and is it really the right way. What we need and the notion of an EKG are subtly different[1].

The Transcelerate Digital Data Flow (DDF) [7] project provided a glimpse of this with the Common Data Model. It provides one part of the model but, I believe, there are other parts that are necessary to provide what industry needs.

Within the model sits our data and where I see we need, and have advocated for the last decade, the Biomedical Concept [1, 5]. I don't care what you call then, FHIR uses resource, others use Clinical Models, LOINC see a pre-coordinated measurement represented by a code but they are essentially the same beast, a logical unit of knowledge composed of several variables that, if you remove one part, it loses meaning. I never wish to split it asunder. They are bound to terminology, thus providing value-level metadata (VLM) never needs to be derived, it is pre-specified. I hate to say it but VLM goes away as we always have access to it in the BC[2].

We need to rapidly build a BC library. I have prototyped a data mining application [3] that extract the definitions from define.xml files and that was successful. A similar approach can be used with actual SDTM datasets. The problem we face is that, not being a sponsor, we lack the number of defines or datasets to undertake a sensible mining operation. Given 3 to

[1] One early reviewer made the comment that we might want to refer to it as the Industry Knowledge Graph.

[2] A reviewer made the suggestion that we use the Unified Medical Language System (UMLS) to align terminology. I would agree as the terminology challenge is a significant one.

6 months and access to one large sponsor's datasets, I am confident I could produce draft BCs for 80% or more that the industry needs. These could then be shared for all to review, align and subsequently publish as a version 1. This alone would be a huge step and allow a new range of data quality checks to be employed by both sponsors and regulators[3].

These observations or BCs need to be placed at one point in the model, central to the whole landscape. We must place data at the centre [4], traditional application boxes take a secondary role.

From the model I can then derive my APIs to meet a set of use cases. What studies do you know about, tell me about study X. Give me the protocol for Study X. Has Study X enrolled any subjects. Give me the data for Subject Y, Study X, the list is endless. We need to decide which interfaces we want first.

But What About

But what about existing standards such as SDTM, ODM, Define.xml. Quick answer is they stay but we can use to build the APIs and improve.

As an example, consider ODM and Define.xml. Define is based on ODM but is designed for quite a different use case. Consider Figure 1 that sketches out the high-level construction of the two formats.

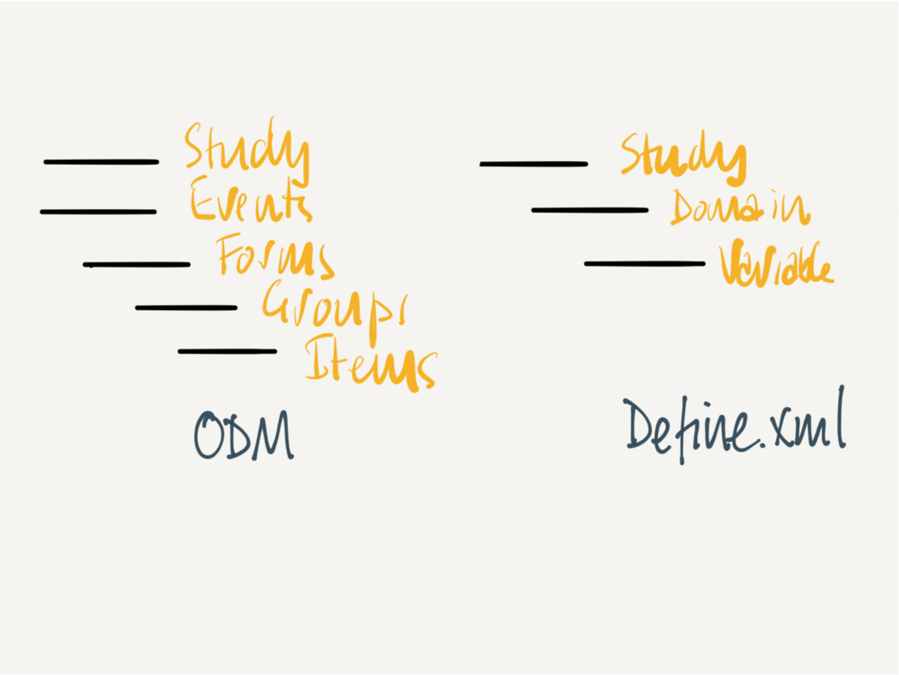


Figure 3 - ODM and Define.xml

[3] One suggestion is that this could be in the form of a dedicated hackathon.

They both have a study section with some very basic details about the study being referenced. We have other tools needing study information, as already noted. We can use the same API calls to get this information. Let's separate that piece out as one common API. Now every system can speak 'Study'.

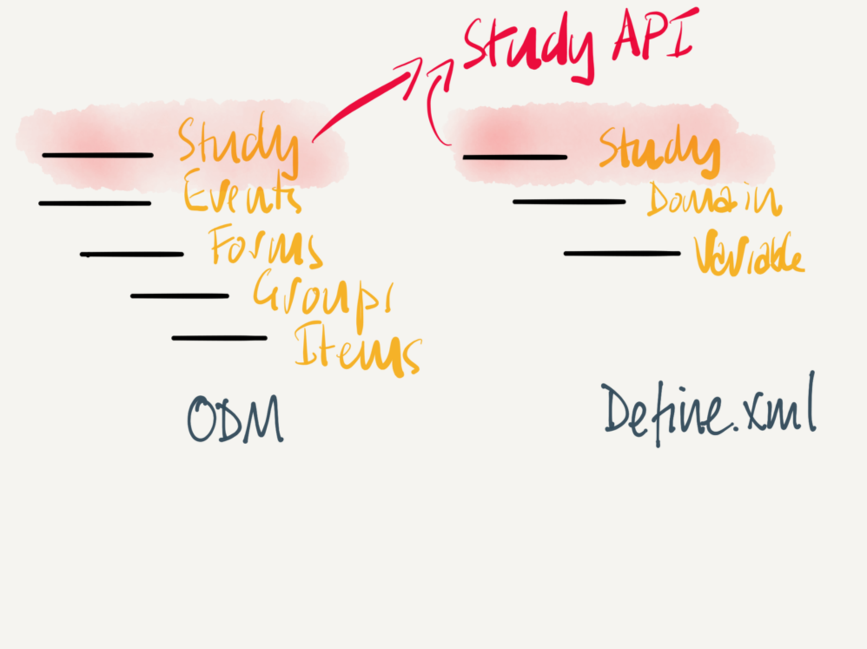


Figure 4 – Study

Looking at the other pieces, we can take the Study Events section of ODM and see similarities with the Schedule of Assessments that study build systems would require. Make that separate. This is DDF territory.

The Define.xml groups and items are holding metadata about domains and variables. Of course, this relates to SDTM tabular structures, and it would make sense to have an API for exchanging Domain metadata and data (also see Dataset XML which shares the define structure and holds the data). Long term we want to move away from these but they will need to remain in the near term; we cannot scare an industry to death and we need to keep the lights burning.

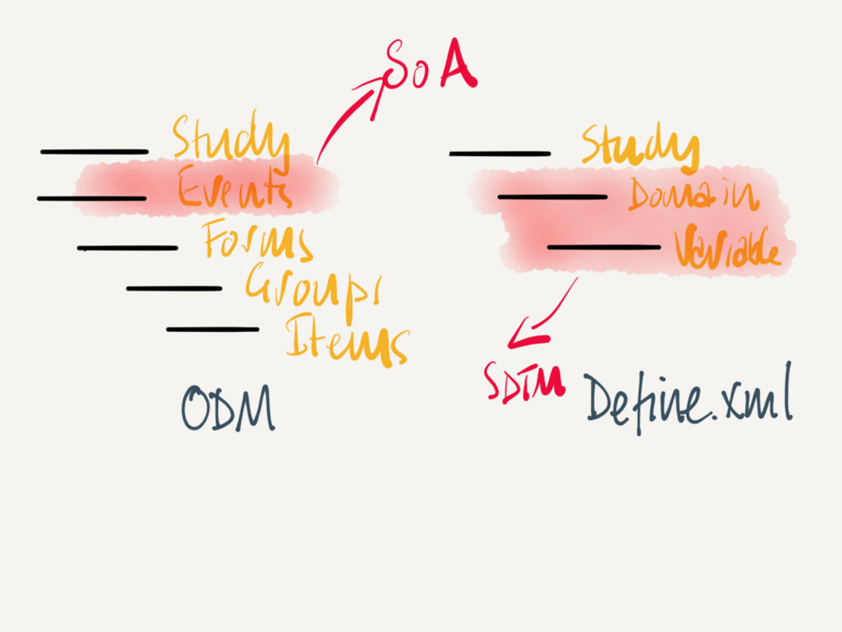


Figure 5 - SoA & SDTM

Forms should be an API section, while Groups and Items have a strong alignment with BCs. We need to consider as an industry how BCs can be used to improve data capture, not just in EDC systems but across all data feeds. Make a separate set of APIs for these items. But remember, the APIs cross-refer.

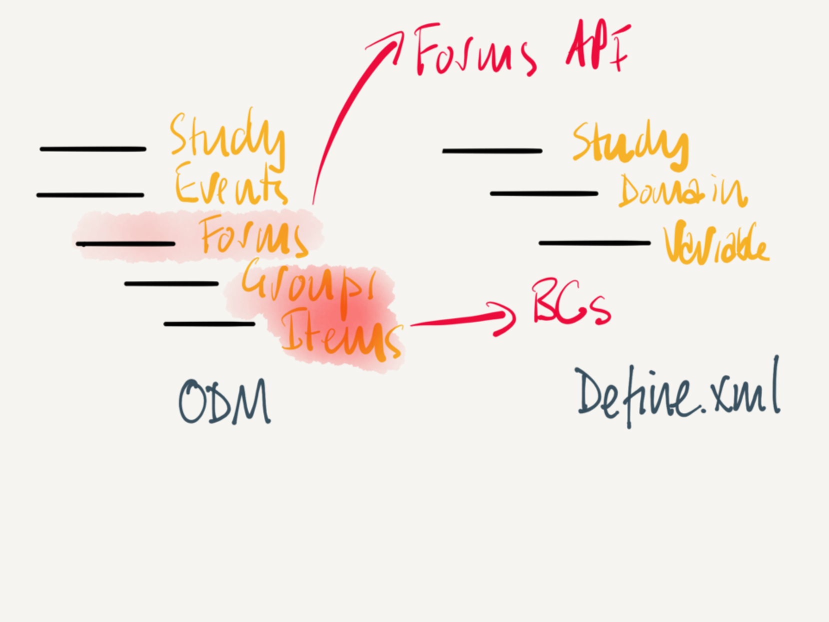


Figure 6 - Forms and Groups

As already noted, SDTM, as a set of metadata, obviously is the same as define.xml. Add data and we have datasets and alignment with Dataset XML. The API might allow me to request the definition, one domain of data, a set of domains.

But none of this should be a surprise, as all of our existing standards are but views and slices of our reality. Define.xml is but a set of columns, headers and associated information in an XML wrapper. I could equally format that as a JSON structure and send it using an API; in fact we could use XML structures in our API.

But the key is that we are doing two things. Firstly, we are using a consistent technology to interface and secondly, we are modelling to ensure we have consistent relationships that the community understands

The figure below shows the evolution from current interfaces, a mix of standards, files and proprietary methods to one based on API. But that landscape can be redrawn, not as a set of point to point interfaces but as a cooperating ecosystem, a community working together to ease everyone's burden. So, in the figure we see an uncontrolled set of interfaces evolve to a set of APIs that can then be organised as an ecosystem[4].

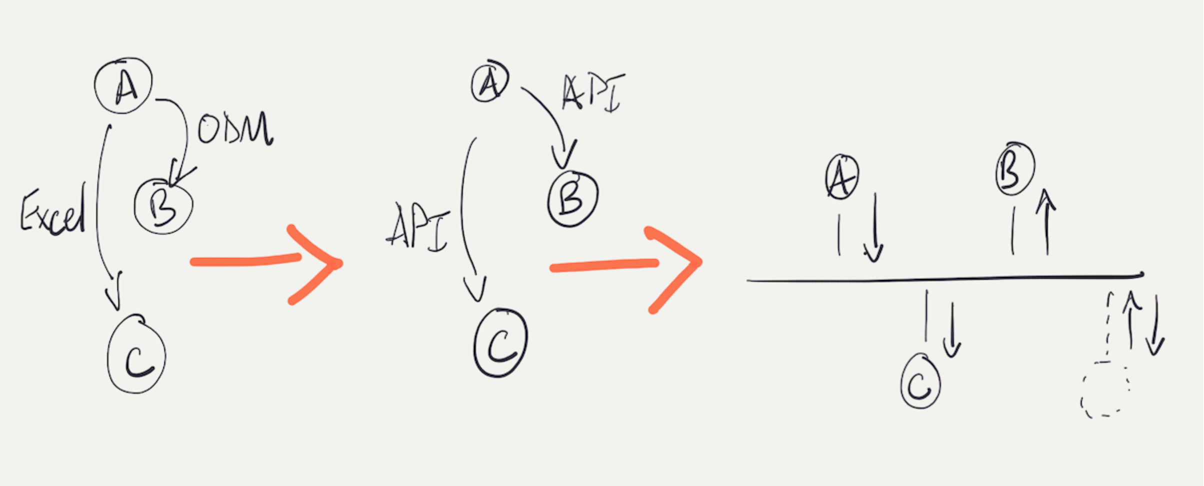


Figure 7 - Evolution

And now consider. If a sponsor system can request this range of information from its own systems, a submission could be a single message to the agency, Sponsor A submits X. That X contains enough information to allow the agency to request all the component pieces from sponsor systems, or at least the regulatory submission system used to assemble the submission. This might be going too far but there is no reason why the regulatory agency cannot employ the exact same APIs. They employ HL7 messages in the same manner. This is simply an extension of that notion.

[4] One reviewer noted the similarity of the right part of the figure with Estonia's X-Road infrastructure, see https://e-estonia.com/solutions/interoperability-services/x-road/

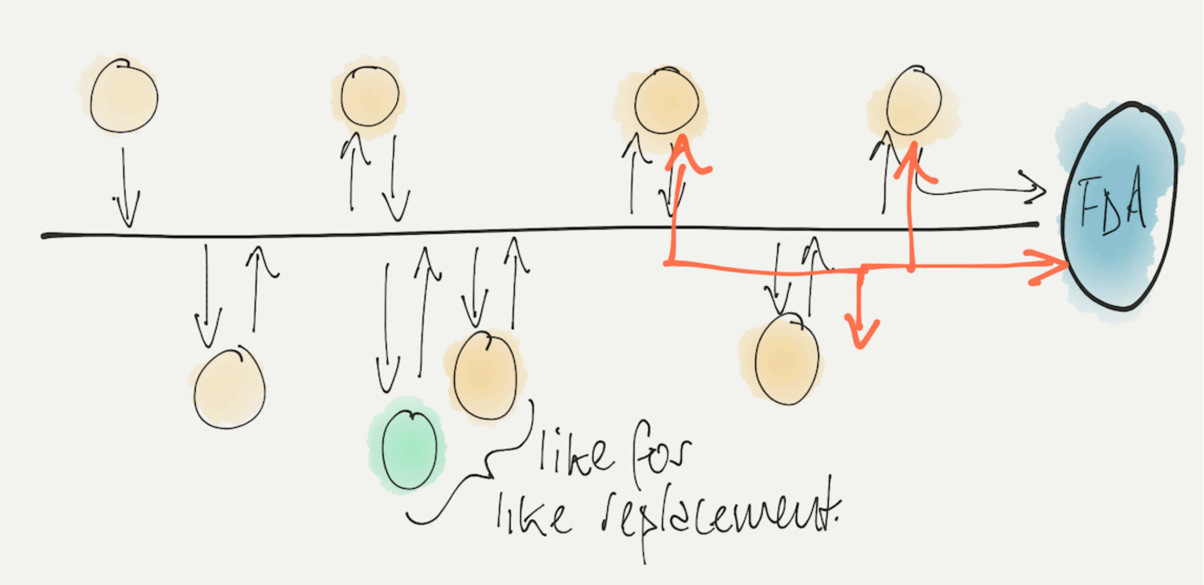


Figure 8 - FDA Submission

The API also allows for a swapping of conforming systems; the use of a different CRO and EDC systems for example on the next study rather than the ones used on the previous study. Flexibility becomes achievable. But we do need open APIs and open models, nothing must be hidden.

And one final thought to whet the appetite. A sponsor could, in theory, make a request to an EHR for data as part of some Real-World Evidence (RWE) exercise or study. Data can be combined with that from a study. I have built a simple prototype using FHIR messages to populate CRFs defined using BCs. We can do this [2]. Of course, we need to think about privacy and security. We need to be cognizant of such but HL7 have the same needs and much can be learned from them.

Organisational Issues

I have long felt, while being closely involved in standards organisations and lately while being less involved than I would like, that we need to recognize that we are not all aligned with where we should focus. We need to fix this. We need a single vision that the organisations buy into and a division of labour with no duplication of effort. Regulators also need to be on board. It needs collaboration, real collaboration, not one where the only connection to reality is a sentence in a press release.

As an aside it is worth reflecting on the standards we see in other industries and their method of production:

- Military Systems: we see standards are imposed by the governments because the customer owns the entire end product, they have total control
- Mobile technology and telecom standards: A mix of equipment manufacturers (handsets and network equipment) and the major buyers, the networks operators. It is not the ultimate user, the individual

- Consumer electronics: The vendors driving the standards, HDMI, USB and the like. Again, the end-user, the individual, is not involved.

But pharma is an odd space. The regulator is not buying the entire process, they approve the end product. Sponsor companies own most of the process but need to play nice with the regulator because of the submission. Vendors are involved but have a limited voice because the content aspects override the technical issues. We need to recognize these constraints and leverage the vendors more. Everyone needs to leave their agendas at the door on the way in.

Any resulting outputs need to be open and freely available; silos won't work.

Approach

So how to approach such an exercise? I would suggest:

1. Model, the "enterprise knowledge graph"
2. Use Cases
3. API specifications
4. BC Content
5. Test with Connecathons

I would sound one huge note of caution. Don't boil the ocean. I remember sitting at the back of a fairly crowded room at an ISO working group meeting. I found myself sitting next to Graham Grieve and we were discussing design by constraint and other ideas; to be fair Graham was educating me! I think he may have been at the start of his FHIR quest. He uttered those words and they have stuck with me for over a decade. "Don't boil the ocean"

I have immense respect for HL7 and what they have achieved with FHIR. They have applied the 80:20 rule, solved the 80% of cases. I contacted Wayne Kubick before embarking on writing this note about what were the factors that have made FHIR the success it is. I am still looking into this aspect, but it is important. We need to learn from their success. And we need to base any APIs on the FHIR standard.

> *Prior to writing this note Wayne mentioned a new HL7 Initiative called Vulcan which may address some of the issues discussed here.*
>
> *The Vulcan initiative has now been formally announced [8]. There is considerable overlap between what I have written here and the HL7 new accelerator programme*

We don't need to do it all at once, we need to learn and listen. How did EHR vendors start coping with FHIR, what approaches did they take. Such feedback will be useful to vendors embarking on upgrading their systems to conform with a new API.

Use Connectathons and other such events to test, get buy-in from vendors and show the community the progress. Learn from IHE, HL7 and others who have run such events for years.

Make use of the current CDISC Library API. It could provide a series of quick wins for vendors and there are ready baked specifications. Why re-invent the wheel? Move to FHIR for these existing interfaces when it makes sense.

As for the use cases we won't always get it right but do not be afraid of that failure or incompleteness. I always feel that the industry wants to get everything just perfect before releasing a specification. Stop it. We need progress not perfection. Make the 80% version, learn, add complexity and improvements. Don't go for perfection in round one.

Change Management is probably the biggest issue along with a general resistance to change. The usual why, what is in it for me? There are better placed people around to answer this one than me, but it is a significant issue

And Finally

This is but my view. I feel there is something there and I would like to hear people's views. But whatever we do, we need to do something because it is not working at the moment.

Issues

A series of issues yet to be detailed:

1. Single source of truth – industry has often spoken of this. Does this approach allow for such? Where should it be? Tied to this is repeated data. We have much today, how do we reduce it?
2. Access controls – let HL7 solve it for us. EHRs have this issue, use the solution. Again, don't reinvent the wheel.
3. Create marketing buzz around the whole thing, find a groovy name, learn from FHIR
4. Terminology, need a better approach

References

[1] CDISC Standards and the Semantic Web
https://www.lexjansen.com/phuse/2015/tt/TT09.pdf

[2] Into the Fire, Linking CDISC & FHIR
https://www.lexjansen.com/phuse/2018/si/SI12.pdf

[3] Towards a Biomedical Concept Library: Creating and Sharing Biomedical Concepts
https://www.lexjansen.com/phuse/2019/si/SI03.pdf

[4] Removing Silos: Placing Data at the Centre
https://www.lexjansen.com/phuse-us/2019/si/SI13.pdf

[5] Easing Your Pain with Biomedical Concepts
https://www.lexjansen.com/phuse-us/2018/si/SI19.pdf

[6] Tony Seale, LinkedIn

- https://www.linkedin.com/posts/tonyseale_ai-decentralised-knowledgegraph-ugcPost-6681501918516277248-XfDL
- https://www.linkedin.com/posts/tonyseale_future-data-graph-activity-6684107919652474880-gUqT
- https://www.linkedin.com/feed/update/urn:li:activity:6686697005844062208/

[7] Transcelerate Digital Data Flow (DDF)
https://transceleratebiopharmainc.com/initiatives/digital-data-flow/

[8] HL7 Vulcan
https://www.hl7.org/vulcan/

[9] Gomez-Perez J.M., Pan J.Z., Vetere G., Wu H. (2017) Enterprise Knowledge Graph: An Introduction.