Paper SI03

Towards a Biomedical Concept Library: Creating and Sharing Biomedical Concepts

Dave Iberson-Hurst, S-cubed, Copenhagen, Denmark

ABSTRACT

There has been much discussion over the last few years about Biomedical Concepts (BCs), end-to-end and automation. CDISC itself has recently initiated the 360 Project to investigate the use of BCs and how they can meet these goals. However, to reach our goal we need the actual BCs for both the safety domains and therapeutic areas.

This paper reports on work investigating whether it is possible to create BCs from existing content like define.xml. We will demonstrate how a semi-automated process using tools can create BCs, show how those familiar with the CDISC standards can use the process to create new BCs and how the results can be shared with the community. The impact of BCs on the CDISC TAUGs and how BCs can improve the definitions of TAUGs is also discussed.

THE PROBLEM

In April of 2019 CDISC launched the 360 Project. As the CDISC web site [1] states:

> *CDISC 360 is an ambitious new project geared toward innovating clinical data standards to ensure they remain valuable and relevant into the future. CDISC 360 aims to support standards-based, metadata-driven automation across the end-to-end clinical research data lifecycle and represents a significant next step toward realizing an increased return on investment in standards implementation that our stakeholders expect – substantially improved efficiency, consistency, and re-usability.*

One of the key pieces to all of this is Biomedical Concepts (BCs) [2]. [3] & [4] and how they contribute to the CDISC 360 vision. The author has written about, and worked with, BCs for some time. When discussing BCs, one issue that often arises is the number of BCs that we expect to create. The answer is in the 10,000 area. The next reaction from people is that this is too big a task and that we will never accomplish it. However, the good news is that there is a way forward and this paper is intended to show that we can do this, and the target is eminently achievable. It should be remembered that a BC is essentially structuring an already known set of information, adding relationships and creating code list subsets for such items as qualifiers. This information is created repeatedly every time a study is built, and a data item is being defined. A BC is a standardized template for this and will make this process more effective and consistent. The author strongly believes we have the necessary base information, we just need to release it, format it, fill in the gaps and version control the resulting outputs.

SOLUTION

As stated above, we have a way forward in that we have the knowledge to create the necessary set of BCs at our fingertips albeit in a format(s) that is not that accessible. Most of the information we need is known to us; we use it every day in our work, there are a few new observations we make as part of a clinical study, maybe some novel laboratory test but the bulk of the definitions are well known. What BCs allow us to do is package this knowledge as reusable definitions that we can share to reduce, and even eliminate, the variation across the industry. We do not need to create a mass of new content for each new study, the amount is limited and can be accommodated in a structured manner. One other advantage is that these reusable packages become mini standards in their own right and are version managed, they are not final, they have a life cycle of their own and can be updated. Once created in a modern structured package the BCs can assist in today's studies, they can be exported in tabular structures for use within today's processes.

METHOD

So, while it is believed that BCs can be found within our existing content, it is necessary to prove it. Therefore, the author and a few colleagues undertook to develop a prototype web application such that we could test the theory. As with all prototypes and research we wanted to keep the scope tightly constrained so we restricted the use case to SDTM findings and the associated test codes, but the principles apply to the other areas and can be readily applied.

The method taken is to take a well-known and available – to sponsors and the FDA – data source, namely the define.xml file. A define file contains the metadata for a single study. Take N define.xml files and, in theory, we should have the metadata for N studies. If we look at a single finding, we should have N variants of that finding with the necessary qualifiers, code list subsets etc. In reality we end up with less than N definitions so making N as large as possible is certainly beneficial; simply put, the more define files the better.

The work is based on one major assumption, namely that the define.xml files have been built correctly and they have used the value-level metadata structure of define correctly. For this prototype only define version 2 files were used.

It is expected that there will be variation. This variation may present itself in many ways. It may be in code lists used, the list of units for example between studies, possibly one study used imperial while another used metric. Qualifiers may or may not be present, one study collected a certain qualifier while another did not. Looking across multiple studies allows us to build the superset from the N definitions. This superset then forms the basis of the BC that can then be reviewed and curated by humans.

The overall process is shown in Figure 1 and described in the following text.

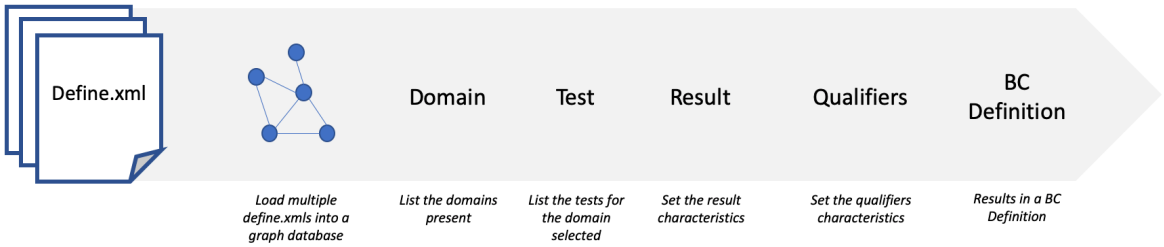

Figure 1: Overall Process

To test this, an application was created that uses a linked data approach to import the N define.xml files and load the content into a single graph. The application then queries the database to obtain the set of domains that have been described within the set of define files loaded.

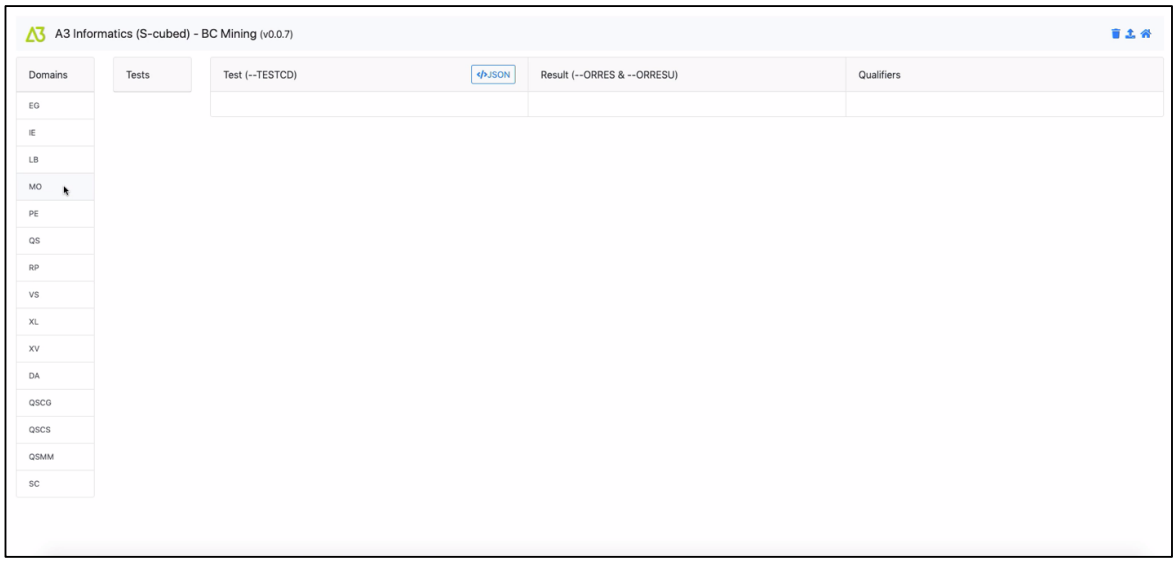

Figure 2: Select Domain

The user can then select a domain and see the test codes present within the set of define files

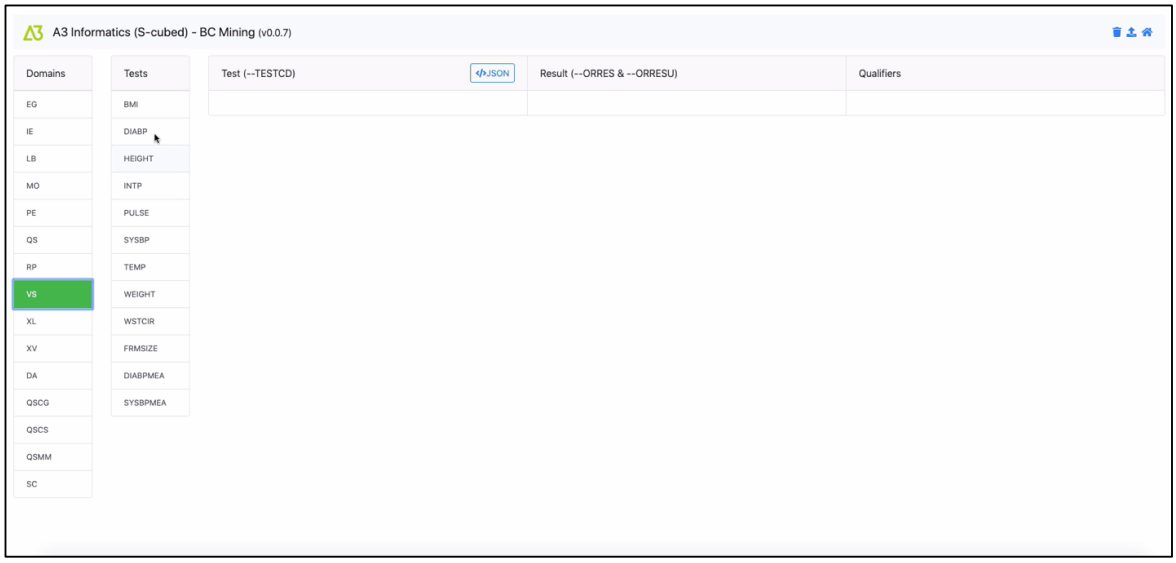

Figure 3: Select Test

On selecting a test code, the database can then be mined to extract the various metadata from the set of define files. In Figure 4, we see a view where we can examine the raw result of the database mining. This example is taken from some test define files. We see that two studies used the unit “inches” for the Height test while another two used “centimeters” and one did not contain any metadata.

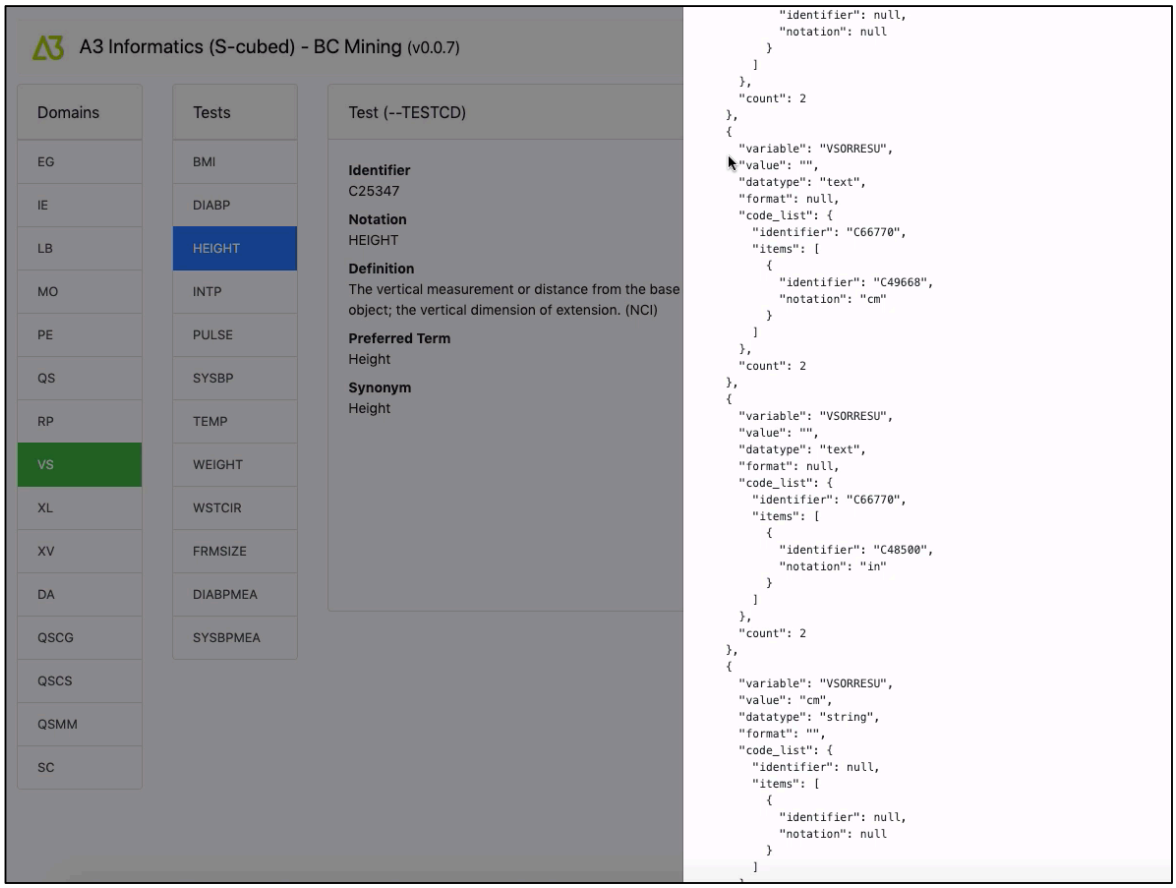

Figure 4: Variation

This is where we begin to see the variation. Variation can occur in a number of places. An observation may have a numeric result. That result might be integer in one study and a float in another. The units may differ from one study to the next due to a variety of factors such as geographic location of the study. If the result is coded, a subset of potential result codes may have been used in one study and a different subset in a another. By looking across the studies we can build the superset.

Figure 5 shows a screen for the Height test. We see four studies having a float data result but some variation in the formats reported. We see the inches and centimeters units already selected. This is all based on the query of the set of define files. The application then allows the various choices to be made and further terminology to be added to the units code list, in the screen shot meters gets added. The application also allows for a coded response to be added.

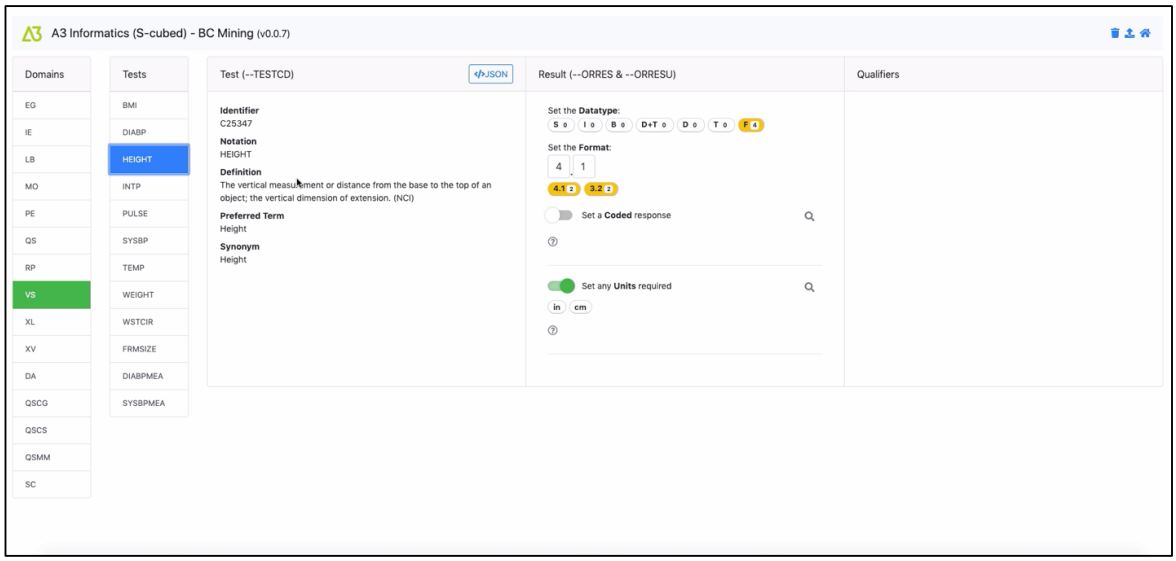

Figure 5: Test and Result

The same issue of variation occurs with the qualifiers. For example, one study does not capture a body position, a second does but fixes it at standing. A third captures one of several responses. By looking at all of the studies, we build a more complete picture for qualifiers and build the correct subset for the qualifier for the test.

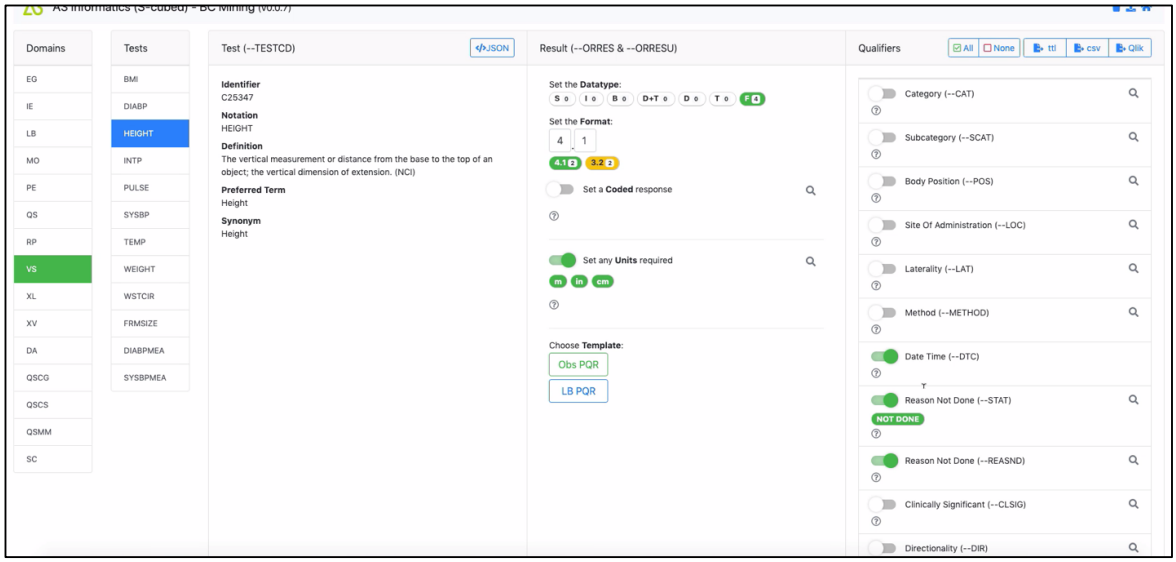

Figure 6: Qualifiers

One issue that arises, when looking at the metadata, is what to actually look for. The result, be it coded or a numeric value with units, is obvious. But with the qualifiers, we need guidance. This is where the BCs help. They are formed from a small set of templates that define the type of information required that includes a list of qualifiers. As an aside the template content is based on the SDTM and BRIDG models. This allows us to know what to search for when

querying the database, what goes where in the BC and what information might be required and thus present to the user. In the application the selection of a template is based upon the test result type. The system looks at the result type and presents the user with the appropriate subset of templates. If the datatype changes the template selection will change.

When addressing the qualifiers, it is not always the case that the defines contain all of the terminology required. Hence, we need the human and their knowledge to allow additional terminology items to be added, where they do not exist within the define metadata. We hope that the greater the number of define files reduces this need. This is achieved by searching the existing terminology (from a metadata repository) and adding the required terms.

Qualifiers can also be enabled and disabled. A test may require a method, a lab test would probably require a specimen type whereas a vital sign would not. However, it is important then when they are included (enabled) that they are fully defined with the appropriate terminology subset.

One aspect we did not include in the prototype application but would be sensible to include in a production environment is the ability to preset qualifiers. One that comes to mind is status – has the test been performed – and the reason for not preforming the test. The NOT_DONE value is common across tests and could thus be in a library of preset values for qualifiers.

Having gone through the process of setting up the result and qualifiers, we have a draft BC that could then enter review and curation. The BC can be exported in various formats. One export setup in the prototype is a CSV format. This allows the information to be exported into existing processes where they could be used for data quality checks, better EDC setup and other such processes.

The demonstration user interface has been designed around a manual process for the benefit of demonstrations. A production environment would automate much of what is seen here, so as to speed the processes.

THEARAPUETIC AREA USER GUIDES

One benefit of creating a set of BCs would be in the definition of CDISC's Therapeutic Area User Guides. Having definitive data specifications for each of the observations would vastly improve the quality of the specifications. These definitions could be curated from a set of define.xmls from the relevant therapeutic area.

SECONDARY USE CASE

The work, as stated above, was originally intended to mine BC definitions. However, several other use cases presented themselves when we started working on the project. In particular the same approach could be used to check the quality of a set of define.xmls by looking at the variation on the metadata. By looking at each test and looking at the variability an assessment as to the quality of the data across a set of studies could be made.

The use cases for the tool should not be confused with the use cases for BCs themselves as described in [3]. Once we have good BC definitions, they bring great power to the overall end-to-end study process. The use case(s) here are about creating BCs not deploying them across the life-cycle.

CONCLUSION

BCs offer a tangible and concrete way forward for the industry, but the industry needs the definitions. This work clearly demonstrates we can, to a great extent, automate the production of these definitions.

REFERENCES

[1] CDISC 360 project.
[accessed 2019 Oct 16]
https://www.cdisc.org/cdisc-360

[2] Biomedical Concepts and SHARE.
[accessed 2019 Oct 16]
https://www.cdisc.org/standards/share/biomedical-concepts

[3] Langendorf. PhUSE US Connect 2018, SI19. Easing Your Pain with Biomedical Concepts
[accessed 2019 Oct 16]
https://phusewiki.org/docs/2018_US%20Connect18/SI%20STREAM%202/si19%20final%20.pdf

[4] Iberson-Hurst. Phuse US Connect 2019, SI13. Removing Silos: Placing Data at the Centre
[accessed 2019 Oct 16]
https://www.s-cubed-global.com/wp-content/uploads/Removing-Silos-Placing-Data-at-the-Centre-Paper-.pdf

CONTACT INFORMATION (HEADER 1)

Your comments and questions are valued and encouraged. Contact the author at:

Dave Iberson-Hurst
S-cubed Aps
Lille Strandstræde 20C 5.
DK-1254 Copenhagen K
dih@s-cubed.dk:
https://www.s-cubed-global.com/