PhUSE 2016

Paper PP23

Constructing Interoperable Study Documents From A Semantic Technology-based Repository[1]

Colin de Klerk, UCB BioSciences GmbH, Monheim, Germany

ABSTRACT

Significant maturity has been achieved in organising and exchanging clinical data using standard models (e.g. SDTM, ADaM). The procedures underlying the data are, however, usually expressed in highly variable text buried in poorly-structured (even when template-driven) documents. It is almost impossible to process or reuse these descriptions automatically.

Meanwhile, many organisations are engaged in extracting, generalising and organising into standard models (e.g. BRIDG) and ontologies the myriad terms and concepts encountered in clinical and analytical procedures. Published models are made available to authorised end users.

If we embed these curated, standardised concepts into clinical documents (e.g. study protocol) we gain automated interoperability, traceability, reusability and more. Semantic technology metadata offer a mechanism to do that.

The prototype shows how new, structured and semantic-aware clinical documents can be generated from a repository connected to external standards and containing enterprise-level concepts and extensions.

INTRODUCTION

Standards reflect industry best practises and ideally support the efficient re-use of elements common to individual studies. While standards are inherently conservative, they also need to be flexible enough to be used in novel situations as well as to be adaptable in an environment with many moving parts including shifting regulatory guidance, scientific and technological advances and the varying needs of different functional teams.

Document standards, specifically, are often implemented in the form of predefined templates. While document templates can facilitate the task of standardisation, it is clear that the user must apply a good deal of cognitive input to use a template in a particular situation -- by reading, applying and finally deleting instructional text in the document. Some of this task can be automated and certain fields can be pre-populated, perhaps from a database.

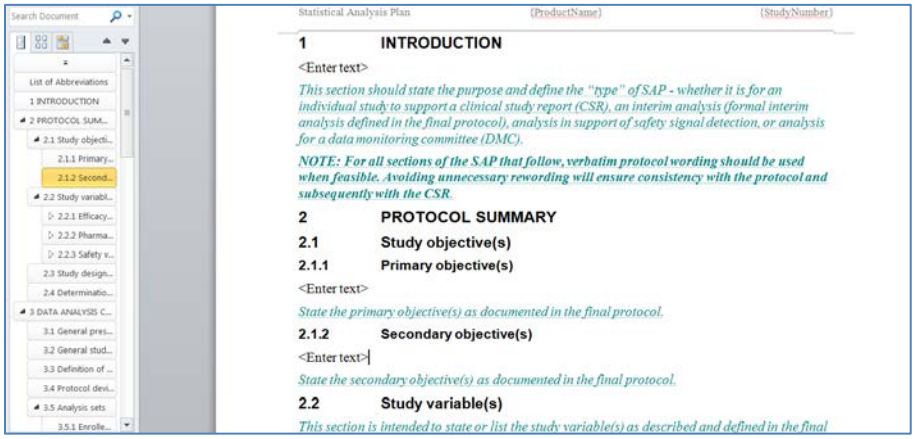

Figure 1: Typical template for a statistical analysis plan

The design of the study protocol template, while superficially straightforward, has been continually revisited by different sponsors and biotech companies over the years, and hence suffers from a severe lack of consistency. This presents challenges when stakeholders from different companies and regulatory authorities need to exchange information about studies via the study protocol. Over time, industry-wide initiatives such as the *Common Protocol Template* project maintained by TransCelerate at http://www.transceleratebiopharmainc.com/initiatives/common-protocol-template/ have emerged to promote standardisation of study protocol templates.

Document templates are often organised within a document management system (CMS) that manages storage and versioning, provides search facilities, and controls access to content. Efficient use of a CMS relies on how well the content is organised inside the CMS and this, in turn, depends on the addition of expressive metadata – keywords, descriptions and identifiers - to allow users to find and use items stored in the repository. Failure in this respect makes subsequent search and retrieval challenging at best.

Figure 2: Functions of a Content Management System[2]

At the same time, busy people working on tight project schedules do not have time to craft elaborate metadata. Add ever more users and ever more poorly-labelled documents to the mix and over time, the CMS becomes unwieldy. Ultimately it can degenerate into a graveyard of lost content rather than living up to its promised advantages.

Hence adding value to the metadata process, while challenging, appears to offer great rewards. How can we help users to determine relevant, intuitive metadata to associate with the content documents of the CMS to improve the accuracy of, and reduce the complexity of retrieval of content from the CMS? How efficiently can we attach metadata to the content to which it refers? How do we ensure that the metadata remain relevant in a dynamic environment of changing versions and shifting requirements?

WHERE ARE WE NOW?

Intuitively, it seems we could solve these problems if we could somehow teach a system to extract and identify metadata from the content documents. Many systems already support functionality to extract keywords from document text using techniques ranging from simple frequency counts at one extreme right up to intricate heuristics derived from statistics, text mining and natural language processing (NLP) at the other. Note that this only applies to analysis of text-based documents. A significant proportion of content, however, is not simple text and must be pre-processed, e.g. by optical character recognition or image analysis, to convert non-text content into text that can then be analysed as described.

While metadata allow us to gain an idea of what a manuscript contains, the longer and more complex a text, and the more themes it contains (e.g. study protocols that cover every aspect of a clinical study in one document) the more difficult it becomes to characterize the document completely with just a few keywords. This complexity only grows as we add a dynamic aspect of tracking changes to the document over time.

While this is an issue if individual documents are treated monolithically -- as whole documents -- there may be some promise in breaking long documents up into smaller chunks that contain fewer, related themes and can thus be better described by targeted metadata. As an additional advantage, these smaller fragments would fit better into the area of expertise of a particular functional group and hence streamline the review process. For example, the study statistician only needs to review relevant aspects of the study design such as the methods to determine sample size. While it would still be necessary to review the document as a whole to ensure consistency, it seems the overall review process could be simplified.

Indeed, a repository of document fragments that are well-described by metadata could even suggest turning the whole approach on its head: rather than extracting metadata from a document only to regurgitate the whole document later, perhaps we could *generate* the required document from self-describing chunks that combine with study data for a novel instance (e.g. a study protocol for a brand new project) and thus pre-populate much of the template. This achieves a double advantage of assisting adherence to standards and also automating some of the work.

So we have a lofty goal. Is it practical, or even feasible? Perhaps semantic technologies could help. We will digress briefly to discuss what semantic technologies are and how they might help us realize the goal of a value-added CMS.

SEMANTIC WEB, RDF AND LINKED DATA

The problem of effective categorisation and labelling of unstructured content is not unique to applications in clinical research and countless bright people have proposed myriad solutions over time. A particularly intriguing approach began to emerge around the turn of the millennium. Known by various names such as web 3.0 or linked data, but most often called *semantic web technologies*, it involves adding a layer of *meaning* to the otherwise chaotic and unstructured content found on the Internet – of course this is an oversimplification.

A fundamental design principle of the semantic web is to enable the automatic processing of machine-readable metadata including themes or keywords, and especially the relationships within and between documents. Apart from cataloguing content with a view to improving information retrieval, this processing extends to inference: using techniques derived from first-order logic to find rules and relationships beyond those that content providers stipulate explicitly.

Linked data as an expression of the semantic web comprises statements of the form: "*subject→predicate→object*" where *subject* and *object* are *resources* identified by *international resource identifiers* (IRI) which are a superset of *universal resource identifiers* (URI), which are again a superset of the *universal resource locators* (URLs) we are accustomed to using on the Internet. IRIs extend URIs, using Unicode to represent resources using any alphabet (e.g. http://ヒキワリ.ナットウ.ニホン), not just ASCII (e.g. http://www.phuse.eu). *Predicates* can also be regarded as resources and can have their own URIs. *Subject→predicate→object* statements are expressions of the *resource description framework (RDF)*. RDF resources can be represented by nodes, and RDF predicates by edges connecting two nodes in a mathematical graph. While RDF and related standards (e.g. SPARQL, SPIN, OWL) as maintained by the World Wide Web Consortium[3] (W3C) are best known, there are plenty of other semantic technologies.

Predicates in RDF statements can be thought of as verbs in a sentence. The statement: "Jemma is 43 years old" can be restated as "Jemma has-the-age 43". So "Jemma" is the subject, "43" is the object and "has-the-age" is the verb or predicate. Imagine that Jemma has an account at example.org where she has the user ID 123. The URL http://example.org/123 can then be used to link to Jemma as a resource. In this example the object of the statement requires no identifier because it is a literal value of 43 with an integer data type. Assume that "ex:age" represents the predicate "has-the-age" and we have the situation depicted in Figure 3.

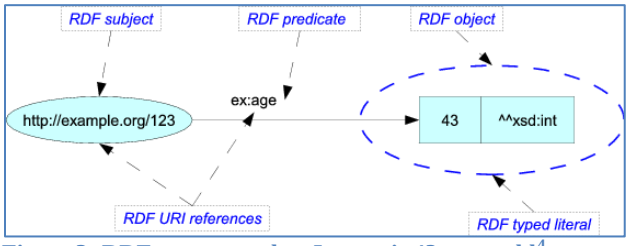

Figure 3: RDF statement that Jemma is 43 years old[4]

Based on this simple framework, we can make further statements about Jemma and so build up a knowledge base that models aspects of Jemma that interest us in some domain. We can even combine "our" set of facts about her with additional sets of facts about Jemma that have been published by other entities, perhaps in other languages. Provided we can equate our URI "http://example.org/123" with theirs (ideally we could all agree on the same URI that points to Jemma), we can be sure we are all talking about the same Jemma. This distributed or *federated knowledge base,* even if it is maintained by separate entities, could then be cleaned to eliminate duplicate or redundant statements and to resolve conflicts. At the end we are left with a comprehensive, internally consistent asset that can be used in useful applications.

On a massively greater scale, the Linking Open Data Cloud pictured in Figure 4 as of 2014 represents countless RDF statements and thus linked knowledge pooled across a huge number of domains maintained by thousands of different entities. DBPedia[5], the largest bubble in the centre of the diagram refers to the RDF knowledge base that underlies aspects of Wikipedia. Domains relating to life sciences in the outlined segment represent a significant proportion of the LOD cloud.

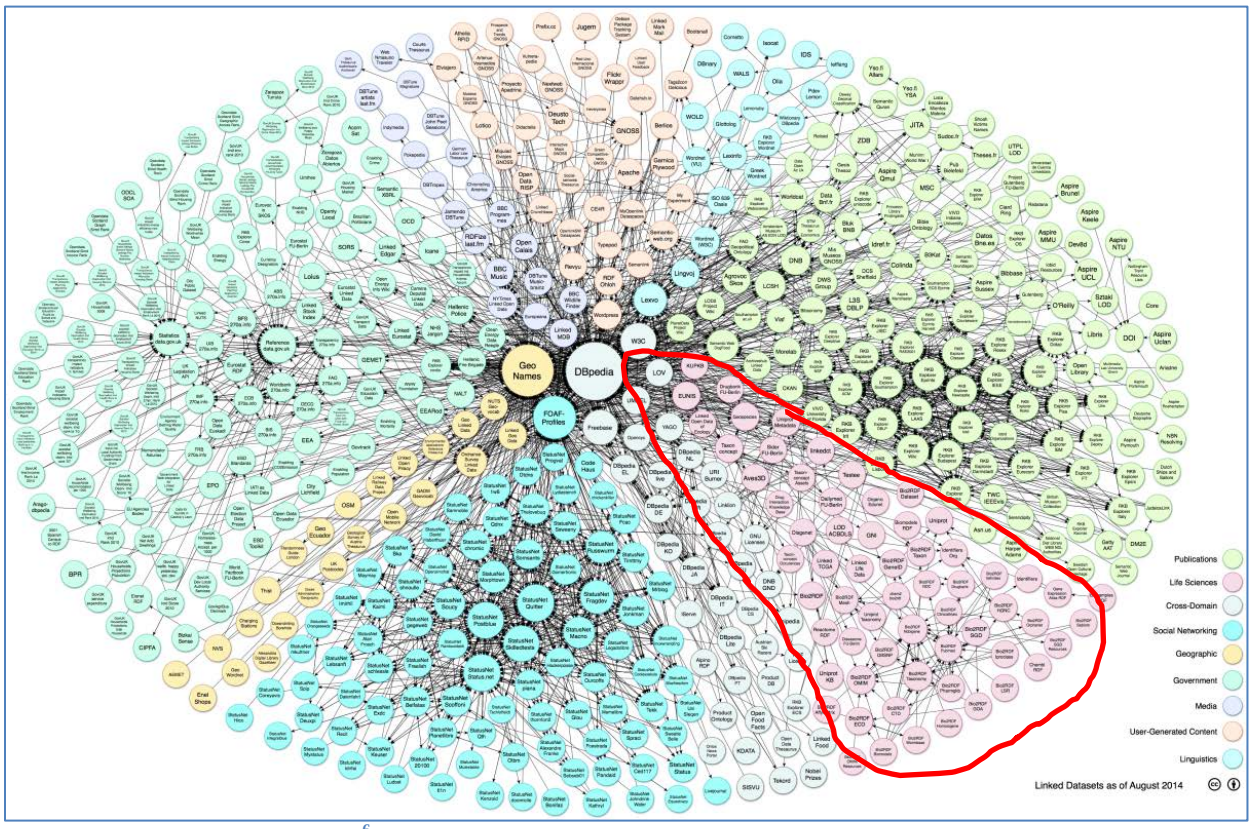

Figure 4: Linked Open Data Cloud[6]

AUTOMATIC INFERENCE – EXPANDING WHAT WE KNOW

RDF statements can be stored and processed in a number of ways. They can be queried using a structured query language called *SPARQL* [7]. RDF statements can also be analysed to infer new statements that follow logically from others. Thus "Zoe has-mother Jemma" and "Kyle has-mother Jemma" logically entail the symmetrical relationships "Kyle has-sibling Zoe" and "Zoe has-sibling Kyle". These logically entailed facts can be automatically derived by a suitably capable *inference engine, a* capability supported by many tools (see tools and software), and which are generally compliant with another W3C standard: *SPARQL Inference Notation* (SPIN[8]).

ORGANISING WHAT WE KNOW – THESAURUSES AND ONTOLOGIES

Conceptually, there is nothing to stop us from just gathering any number of RDF statements in a database, using SPIN to refine the collection by adding logically entailed constraints and statements and then finally running SPARQL queries to extract (hopefully useful) information from the collection. However, given limitations in data storage capacity and in the shear processing power required to run such inferences and queries with currently available tools, it is vital to help the process along.

In the interest of efficiency, the resources referenced in RDF statements are often collected in curated (which implies that some recognized authority maintains canonical definitions of the resources: each entity shown in Figure 3 would represent such an authority) *thesauruses* of controlled vocabulary or terminology. Examples include medical dictionaries like MedDRA or WHO Drug, or MeSH, the Medical Subject Headings used to index articles on PubMed. Similarly, RDF knowledge bases are often organised into commonly-accepted *ontologies* which are logical structures that formalise concepts, agents, processes and the relationships between them in some domain. Ontologies range from simple to highly complex as we shall presently see in the BRIDG model.

Ontologies can be expressed in many ways. Here we consider *web ontology language* (OWL) which is itself based on RDF and *RDF schema* (RDFS) and has three variants with increasing levels of expressivity and thus of complexity. Domain ontologies concentrate on a particular field of activity, e.g. the "*Bio2RDF project aims to transform silos of life science data into a globally distributed network of linked data for biomedical knowledge translation and discovery*."[9]

TOOLS AND SOFTWARE

An increasingly large number of software platforms, some of them open-source, provide capabilities to store and process RDF, run SPARQL queries, perform SPIN inference and much more. A comprehensive list is available at http://semanticweb.org/wiki/Tools.html.

APPLYING SEMANTIC NOTIONS TO A CMS – THE CONCEPT

Now that we have a better understanding of what semantic technologies are and an inkling of how they could assist us, we return to our goal: a value-added CMS that can *generate* study documents on demand.

Creating such a "smart" semantic technology-based content management system is an ambitious project. We want a system that not only performs the classical functions of a CMS: storage and versioning, search via metadata and access control, but also one that at some level *understands* the content we entrust to it. A semantic CMS (semCMS) would offer many advantages over contemporary CMSs that treat their content as black boxes wrapped up in keyword metadata. These advantages include more sophisticated search and retrieval functionality and also the ability to *generate* new documents on demand that look like the template document in Figure 1, except with much of the content already pre-filled.

While the goal of an all-purpose, general semCMS lies well beyond our grasp at present, for the purposes of this paper and to illustrate how a more general implementation might work, we will concentrate on a few core documents of clinical studies starting with the study protocol. Note that in the ensuing discussion, we use the terms *trial* and *study* indiscriminately.

A *study protocol* is a document that describes, in detail, the plan for conducting a clinical trial. The study protocol explains the purpose and function of the study as well as how to carry it out. As such, it fulfils multiple purposes for diverse stakeholders at different times. While the study protocol *document* merely describes the many *activities* that various people or stakeholders (*agents)* perform in their *roles* in the study at different times, the orderly *conduct* of the clinical trial itself, including the regulatory need to document it, is what really drives the process of compiling a study protocol document.

If we wish to add support for study protocol documents to our semantic CMS, we need to supply it with a model of the study protocol. Analogous to CDISC models such as SDTM that deal with the *data* arising from clinical studies in a well-described format that is understood and accepted by stakeholders such as drug companies, CROs and regulatory authorities, we need a comprehensive model that contains all agents or roles, activities, workflows, interactions and exchanges of both physical items (e.g. contracts, blood samples) as well as abstract items (e.g. data files) that contribute to clinical study and are described in a study protocol.

But how would we go about creating such a model? Fortunately there is no need to reinvent the wheel. As can be deduced from Figure 4, hundreds if not thousands of organisations have spent years analysing and modelling processes in life sciences domains and many of them have published or licensed their models, some even in the form of linked data. Even if a relatively small proportion of these organisations specialize in analysing clinical studies, a huge amount of useful material is still accessible for applications such as our intended semantic technology-based content management system.

THE BIOMEDICAL RESEARCH INTEGRATED DOMAIN GROUP (BRIDG) MODEL

Consider the *BRIDG Model* (BM), which is described as a *domain information model* (DIM), essentially a reference or conceptual rather than a physically implemented model. It arose from the confluence of invaluable independent work performed by different stakeholders with different initial goals and approaches in domains including clinical trials, biomedical applications, and translational medicine. The current version, BRIDG version 4.1.1, was released in July 2016. Compressed, the release package weighs in at just under 41MB.

> *The Biomedical Research Integrated Domain Group (BRIDG) Model is a collaborative effort engaging stakeholders from the Clinical Data Interchange Standards Consortium (CDISC), the HL7 BRIDG Work Group, the International Organization for Standardization (ISO), the US National Cancer Institute (NCI), and the US Food and Drug Administration (FDA). The goal of the BRIDG Model is to produce a shared view of the dynamic and static semantics for the domain of basic, pre-clinical, clinical, and translational research and its associated regulatory artifacts.*[10]

Given its diverse origins the BM is comprehensive to say the least! Figure 5 shows a *unified modelling language* (UML) *class diagram* of the BRIDG model as a whole. It is frankly illegible at this scale due to its size and complexity. Indeed the BM has grown over more than a decade to incorporate many more sub-domains than needed for our immediate goal of modelling a study protocol. The BM has long since transcended its original scope covering "just" clinical trials. Machine-readable versions of the BM and related models can be obtained by licensed members from the CDISC SHARE *Metadata Repository*[11]. Version 4.1.1 of the BRIDG model is not yet available in RDF/OWL format and the UML representation is still considered the canonical form.

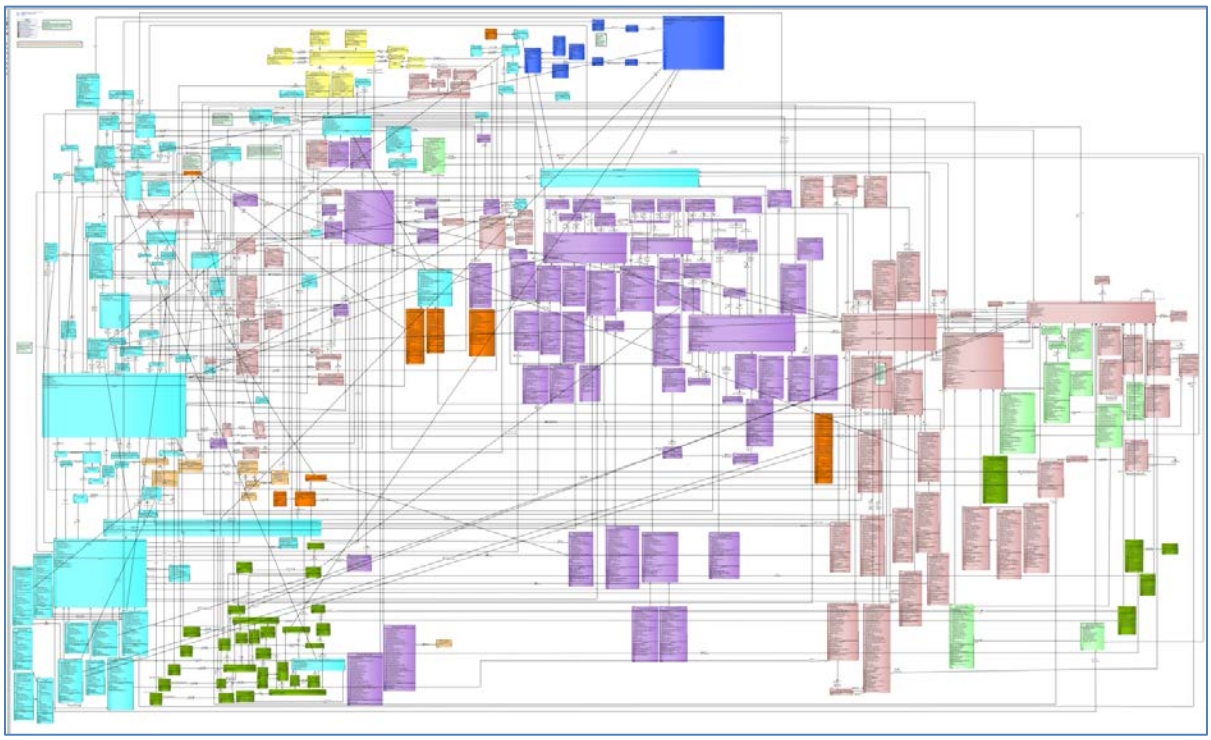

Figure 5: Overview of version 4.1.1 of the BRIDG model[12]

In earlier versions, the creators (the BRIDG Working Group, WG) applied a colour scheme to identify to which package or sub-domain a particular UML class belonged (e.g. the purple items in the middle of the diagram pertain to the study protocol, yellow at the top to statistical analysis, green for adverse events). As the model continues to grow over time, this will no longer be practical, but for now we can still gain an impression about the relative sizes of the sub-domains.

So in our example, we will concentrate on the aspects of the BM that deal with the study protocol and are hence the kind of information we would expect to find in a given study protocol document. Fortuitously, this domain was also a priority for the BRIDGE WG and is hence very well described. We will also need to look at characteristics of the general sub-domain and the study conduct sub-domain.

To use the BM effectively, it is important to appreciate the *caveats* mentioned in the documentation. They include discussions about the harmonization of terminology used in the disparate precursor models (for example, The HL7 *Reference Information Model* (RIM) is a highly abstract comprehensive information model for the healthcare domain and tends to use more generic classes than other components of the BM), considerations about when to expand the "standard" model, when to leave out aspects that do not fit the model, and when to create so-called "localizations". More technical aspects are also discussed, such as the *HL7 data types*, the principles of *computable semantic interoperability* (CSI), and the introduction of various rules or *constraints* in the model (e.g. (unique) identifiers, qualifiers and exclusive-or relationships). Unsurprisingly, these reflect many of the same principles as semantic technologies generally.

BRIDG is a dynamic model and attempts to handle dynamic processes and/or aspects of time. In abstract terms, an activity performed by someone at a particular time and in a particular place has a number of different phases. An activity can be *defined* generally, *planned* by drawing it into a specific study context, *scheduled* to be performed for a particular study subject in the future, and finally *performed* – which means the activity *was* performed for the subject (past tense). If the activity was performed without having been scheduled first, it is understandably an *unscheduled* event.

Splitting the activity across these "pillars" means that the only attributes of the activity class relevant to the specific pillar need to be shown for each one, thus simplifying the model and avoiding redundancy. Figure 6 shows the four sub-classes (DefinedActivity, PlannedActivity, ScheduledActivity and PerformedActivity) of the basic Activity class where the attributes are distributed across them as appropriate. As before, legibility may be an issue at this scale, so for clarity we will highlight a few differences between the four. Thus for *DefinedActivity*, these include *nameCode*, *description* and *repeatFrequency*, A *PlannedActivity* links via a *StudyActivity* to a particular study and includes *studyDayRange* and *blindedDescription* among its attributes. Both the *ScheduledActivity* and *PerformedActivity*

classes link indirectly via the *StudySubjectExperienceDocumentVersion* class (not shown) to a subject. Both also include the attributes *dateRange* and *idealDateRange*. *PerformedActivity* contains additional attributes to contain information about the duration of the performed activity.

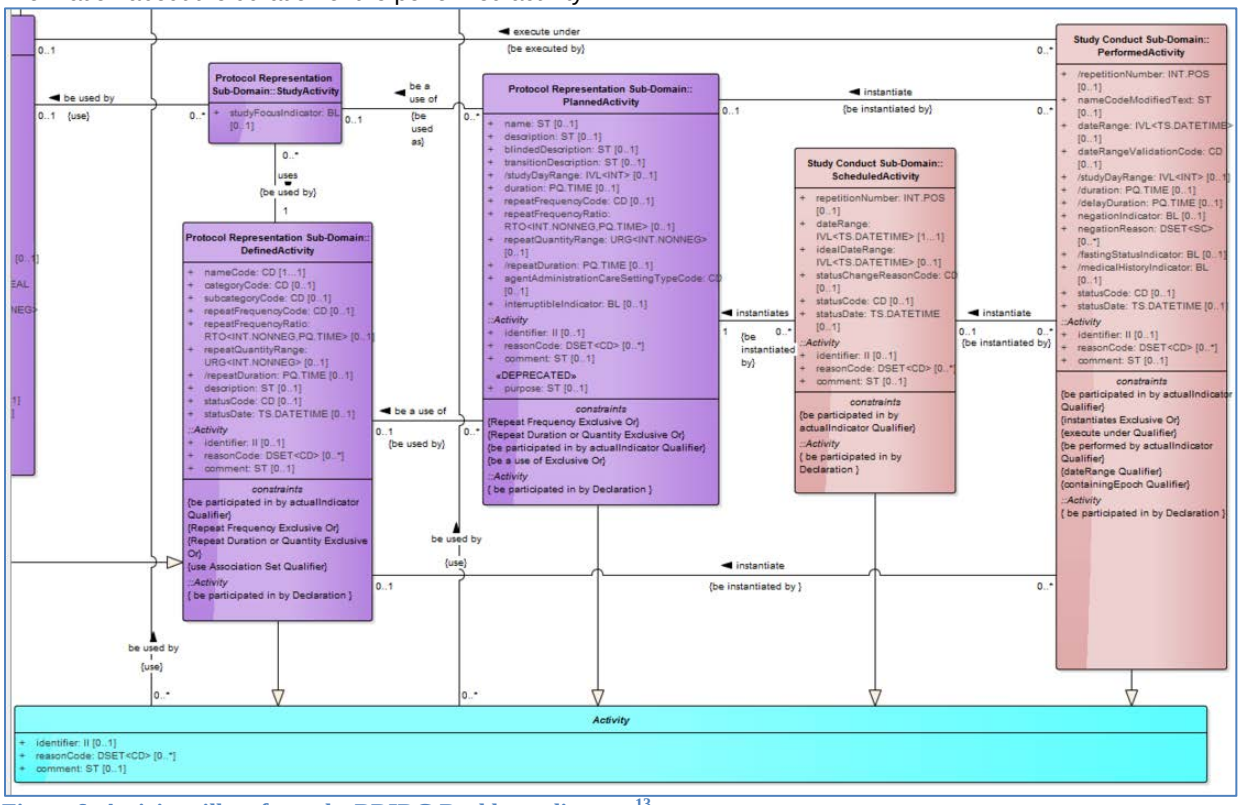

Figure 6: Activity pillars from the BRIDG Backbone diagram[13]

APPLYING SEMANTIC CONCEPTS AND BRIDG TO A CMS

Now we have introduced semantic technologies and the BRIDG model, we return to our discussion of a semantic CMS that can handle study protocol documents in a value-added fashion.

Since the v4.1.1 model is not yet available in RDF format and we did not have access to EA (the BRIDG project uses Sparx Systems' Enterprise Architect modelling tool: a free viewer is available for download[14]), it was necessary to convert the UML version of BRIDG from the XML Metadata Interchange format in which it was published, to RDF. Though the hardware used was nothing special: an Intel i5 laptop with 4GB of RAM running Windows 7 Enterprise SP1, the run-time performance was very good. Python 3.4 was used in PyCharms Community Edition version 2016.1.4 to manipulate the XML using the XML DOM library and thus generate RDF text. This text was subsequently imported into TopBraid Composer FE version 5.01.

CONVERTING FROM UML TO RDF

So as an example, consider the fragment of the BRIDG Backbone UML class diagram from the Common sub-domain shown in Figure 7. The XMI representations of the UML classes ***StudyProtocol*** *(purple)*, ***Study*** *(cyan)* and ***ResearchProject*** *(cyan)* are:

```
<packagedElement xmi:type="uml:Class" visibility="public" xmi:id="…47AAFCF4453E"
      name="StudyProtocol" />
</packagedElement>
<packagedElement xmi:type="uml:Class" visibility="public" xmi:id="…611C206337A8"
     name="Study" >
     <generalization xmi:type="uml:Generalization" xmi:id="…173CB3439D4E"
          general="…FAB88CA21A0D"/>
</packagedElement>
<packagedElement xmi:type="uml:Class" visibility="public" xmi:id="…FAB88CA21A0D"
     name="ResearchProject" >
     <generalization xmi:type="uml:Generalization" xmi:id="…B8A7FFB31D93"
          general="…C6AD05E990B6"/>
</packagedElement>
```

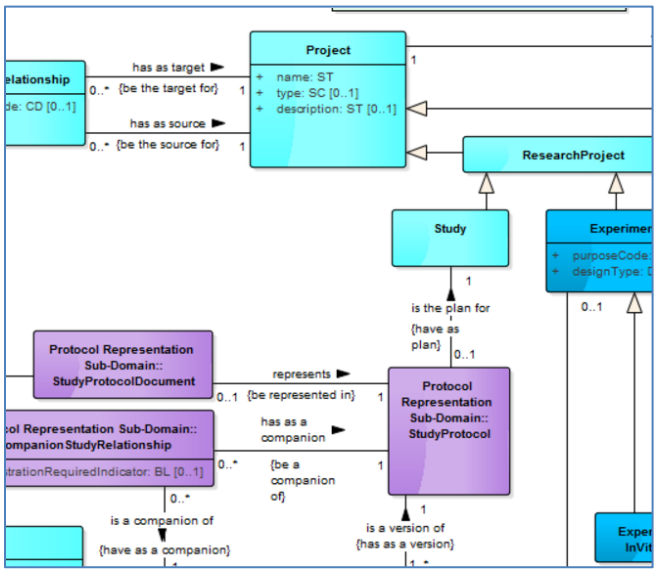

Figure 7: Fragment of the BRIDG Backbone diagram

As can be seen in the figure (the open triangle arrow represents a generalisation association in UML; the arrow points to the more general class) *Study* generalises to *ResearchProject*, which in turn generalises to *Project*. Both XMI and RDF can represent these relationships and also the UML association "*is the plan for*" or "*(have as plan)*" that binds *StudyProtocol* to *Study*. The 42-characater values of the xmi:id attributes that uniquely identify elements have been abbreviated. After conversion, we obtain these RDF statements:

```
model:…47AAFCF4453E
       rdf:type uml:Class ;
       rdfs:label "StudyProtocol"^^xsd:string
.
model:…611C206337A8
       rdf:type uml:Class ;
       rdfs:label "Study"^^xsd:string ;
       rdfs:subClassOf model:…FAB88CA21A0D
.
model:…FAB88CA21A0D
       rdf:type uml:Class ;
       rdfs:label "ResearchProject"^^xsd:string ;
       rdfs:subClassOf model:…C6AD05E990B6
.
```

TopBraid proved superbly capable of importing the prepared RDF statements and we ended up with 332 classes and 447 associations spread across 14 packages, just as it was in UML (see Figure 5). With the model in place, it was now possible to run SPARQL queries like "what are all the attributes of class StudyProtocolVersion?" All 29 were returned, reassuringly.

Clearly we are now in a much better position to ask, and have a hope of *answering* the kinds of questions we would need to ask, in order to set up a study protocol document in a semantic CMS. But it is worth noting at this point that despite its intricacy, we still have only the *generic* domain information model describing the classes and associations typically encountered in clinical trials.

INSTANTIATING THE MODEL: ADDING REAL STUDY DATA

In order to pose questions such as "how many StudyProtocolVersions are associated with StudyProtocol X" – which may tell us how many protocol amendments there have been for a particular study – we need to *instantiate* the model with data from real studies.

If the semantic CMS were to be set up within a single pharmaceutical company that has run or outsourced clinical trials in the past, there would already be a ready supply of real study documentation to add to the model. It should additionally be possible to add (or link to) data such as that publicly available at Clinical Trials (https://clinicaltrials.gov/) or the equivalent for the European Union: EU Clinical Trials Register

(https://www.clinicaltrialsregister.eu/). Either way, there is no shortage of data available to run against the generic model, once the necessary due diligence has been performed on copyright and intellectual property issues.

As an example, consider the results returned by searches for the study NCT01550003 with UCB as sponsor. https://clinicaltrials.gov/ct2/show/record/NCT01550003?term=UCB+juvenile+idiopathic+arthritis&rank=1 returned several pages of (human-readable) information organised into tabs about the *Pediatric Arthritis Study of Certolizumab Pegol (PASCAL) study*. Much of the information shown, and even the order it in which it was presented, resembles the template for a study protocol document as shown in Figure 1. This includes primary and secondary objectives, descriptive information such as the phase, the study design including treatment arms, and administrative information such as sponsor contact details and even the current recruitment status. A similar search at http://bio2rdf.org/describe/?uri=http://bio2rdf.org/clinicaltrials:NCT01550003 returned comparable human-readable information with the option to obtain machine-readable versions of the results in various formats (e.g. N3-Turtle – one of many *serializations* or formats in which RDF data can be saved). A section of this is shown in Figure 8[15].

```
…<<omitted>>…
ns1:NCT01550003     rdfs:label    "Pediatric Arthritis Study of Certolizumab Pegol
[clinicaltrials:NCT01550003]"@en .
@prefix dcterms:    <http://purl.org/dc/terms/> .
ns1:NCT01550003     dcterms:title "Pediatric Arthritis Study of Certolizumab
Pegol"@en .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
ns1:NCT01550003     dcterms:identifier  "clinicaltrials:NCT01550003"^^xsd:string .
@prefix void: <http://rdfs.org/ns/void#> .
ns1:NCT01550003     void:inDataset
       <http://bio2rdf.org/clinicaltrials_resource:bio2rdf.dataset.clinicaltrials.
R3> .
@prefix ns7: <http://bio2rdf.org/bio2rdf_vocabulary:> .
ns1:NCT01550003     ns7:identifier      "NCT01550003"^^xsd:string ;
       ns7:namespace "clinicaltrials"^^xsd:string ;
       ns7:uri       "http://bio2rdf.org/clinicaltrials:NCT01550003"^^xsd:string .
@prefix ns8: <http://identifiers.org/clinicaltrials/> .
ns1:NCT01550003     <http://bio2rdf.org/bio2rdf_vocabulary:x-identifiers.org>
       ns8:NCT01550003 ;
       ns2:acronym   "PASCAL"^^xsd:string ;
       ns2:actual-enrollment      163 ;
       ns2:arm-group <http://bio2rdf.org/clinicaltrials_resource:NCT01550003/arm-
group/8c950ce4b9f63efb5dde3295163b7432> .
…<<omitted>>…
```

So to combine the generic aspects of the BRIDG model expressed in RDF (Figure 9[16]) with aspects of the real study data expressed in RDF (Figure 8), we need "only" match the different representations of the same information – e.g. studyProtocolVersion has the attribute "acronym" and we find this in the bio2rdf data articulated in the RDF statement:

```
ns1:NCT01550003 ns2:acronym "PASCAL"^^xsd:string
```

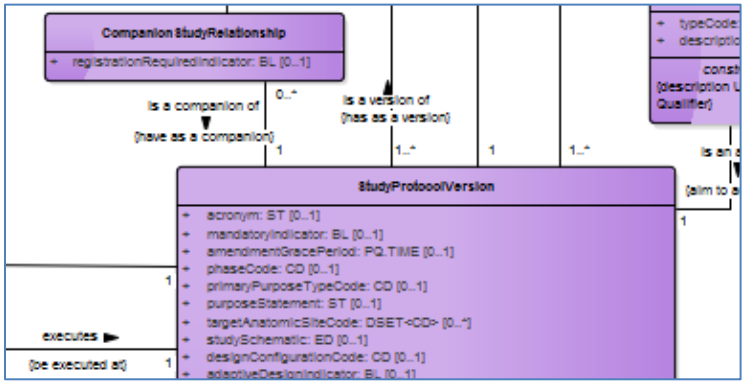

Figure 9: Fragment of BRIDG View: Protocol Representation

Additionally, without going into more detail, we would create new resources as instances of all the classes for which we find a match, instantiate their attributes and so on, until all the information we can extract from the Internet sources and the study protocol document has been matched against the model. While there should be a good fit, there will no doubt be elements of the real study that cannot be accommodated in the model and similarly, not every aspect of the model will find expression in the real study. These mismatches could presumably drive the development of the model so that it can progressively handle more aspect of real studies, or alternatively could contribute to creating more standardised future studies. Once all instantiations are in place, we can query them and/or run SPIN inferences on them, potentially finding inconsistencies in the study design and perhaps avoid repeating the issues in future studies.

Realistically, this step of matching the model and real trials may well be the most difficult part of the entire undertaking; extracting the metadata of the real study from its study documentation presents a challenge, to say the least!

Over and above the RDF data we could obtain from bio2rdf, summary information downloaded from a source such as clinicialtrials.gov is likely to be only semi-structured (in the sense of being able to derive data fields filled with data, e.g. sponsor = “UCB”, if necessary by manipulating the HTML page source) at best. Even assuming we had access to a corpus of study protocol documents, it would still be a major task to convert their contents to the kind of structured data needed. Hence it might be necessary to delve into text-mining, machine-learning and natural language processing techniques, to automatically extract the sense of the study protocol from real documents. In our initial investigations, we examined libraries in R and the python *Natural Language Toolkit* (NLTK), but regrettably with limited success, mainly due to time constraints. If the number of studies to be entered is manageably small, it may well be worth processing their study protocol documents manually.

PUTTING IT ALL TOGETHER – SEMANTIC CMS

In addition to posing SPARQL queries on the generic model and on study data stored against the model, we know that it is also possible to infer new facts from existing statements using SPIN, for example. Indeed when converting the UML/XMI version of BRIDG to RDF, we relied upon this capability to construct aspects of the associations between classes by using RDF *reification* – constructing RDF statements out of other statements – and then using SPIN to fully realise these new statements in the model. Such derived statements are then available to the system in the same way as explicit facts.

While this is interesting in itself, the ability to infer information from the model clearly comes into its own when we use it to compare different studies (e.g. how many studies require subjects to terminate early based on a particular scenario), or even to construct a hypothetical study based on certain criteria. This, now, seems to be going in exactly the right direction to assist us in our stated purpose of using the semantic content management system not just to store content, but also to *generate* it.

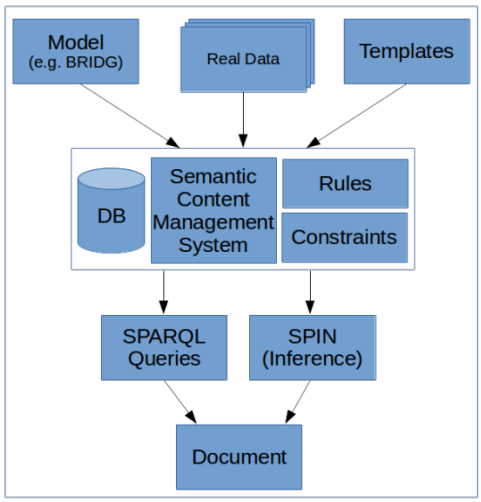

Figure 10: Overview of semantic CMS

Hence we could conceivably ask a system with the components shown in Figure 10 to construct a first version of a study protocol for a prospective study in the same indication as other studies already in the system, and if we supplied the broad study design (e.g. active study drug versus comparator X and possibly placebo), the system could deduce many of the investigations we would need to conduct the proposed study protocol, and even suggest interfaces and data sources.

REALITY CHECK

Note that we are not proposing to replace the kind of data-driven enquiry currently well handled by established technologies in clinical trials. In the (possibly specious) example question above, we wish to query the *metadata* associated with studies to analyse something about a particular early termination scenario – which is essentially an enquiry into the study *design*. Traditional statistical analysis of the clinical data continues to be imperative in discovering whether the *objectives* of the study are being met.

The main use case of the paper continues to be the generation of new study documents from a knowledge base of prior documents.

We might also decry the lamentable lack of semantically-aware authoring tools. Just as we propose in this paper to use the semantic CMS to *generate* documents based on stored content (from which the meaning has been laboriously scraped post facto, possibly by using NLP and text mining methods) and models, so too we might ask ourselves whether it would not be better to build semantic awareness into the original content documents themselves.

There is some cause for optimism here with initiatives such as Dokieli[17] which would help to create documents with some awareness of what they contain and whom they concern:

> *[D]okieli is a decentralized article authoring, annotation, and social notification tool which works from a Web browser. While it is a general purpose tooling to write articles, it is fully compliant with the Linked Research initiative and principles, and provides features and interactions for scholarly communication.*

While approaches like this are still very new and the world may not be quite ready for them, there are already many examples of their adoption[18].

CONCLUSION

This paper was motivated by a desire to improve document standardisation and hence to allow different stakeholders to use and exchange information in more efficiently. Thus, as discussed, the BRIDG model that underlies our initial proposal of a semantic CMS is the product of intensive collaboration by respected industry participants. As the industry moves from traditional clinical trials into translational research, and as the focus grows on increasingly well-informed study subjects who demand more information about themselves and the trials they participate in, the demand to produce accurate, standard documents for particular audiences can only increase.

While we have mainly discussed study protocols in this paper, it should be clear that the system could theoretically support any kind of document provided we could locate its contents somewhere in the models (as comprehensive as it is, we need not restrict the system to the BRIDG model) and for which we have examples of real data in the form of documentation to match against the models. Documents such as the *statistical analysis plan* (SAP) may represent a natural progression from study protocols since they handle much the same material, albeit with different emphasis. For example, the study protocol document might treat a topic such as sample size superficially; we would expect a more thorough discussion in the SAP.

A semantic CMS may help to manage some of this complexity as it learns to handle additional document types, thus assisting to promote compliance with document standards and at the same time enabling the required flexibility.

REFERENCES

These have been provided in the text and in end notes.

ACKNOWLEDGMENTS

Thanks to Tim Williams for inspiring me get started with semantic technologies and for his continued great suggestions.

RECOMMENDED READING

- The Semantic Web, article in Scientific American by Tim Berners-Lee et al: http://bit.ly/2bCLEC7
- RDF resources - https://www.w3.org/RDF/
- RDF 1.1 Primer - https://www.w3.org/TR/rdf11-primer/
- XML Metadata Interchange - https://en.wikipedia.org/wiki/XML_Metadata_Interchange

CONTACT INFORMATION

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

Colin de Klerk
UCB BioSciences GmbH
Alfred-Nobel-Straße 10
40789 Monheim, Germany
Tel: +49.2173.48.1549
Email: colin.deklerk@ucb.com
Web: www.ucb.com

END NOTES

[1] Ideas never occur in a vacuum, and the reader is advised to refer to the PhUSE initiative "Representing CDISC Protocol Representation Model in RDF" at http://bit.ly/2bI6U7K (regrettably on hold at present).
[2] Adapted from http://quickenwebsites.com/services/content-management-system
[3] The World Wide Web Consortium (w3) http://www.w3.org
[4] From http://dublincore.org/documents/dc-rdf/
[5] DBPedia: http://wiki.dbpedia.org/
[6] The Linked Open Data Cloud http://lod-cloud.net
[7] SPARQL: https://www.w3.org/2009/sparql/wiki/Main_Page
[8] SPIN: https://www.w3.org/Submission/spin-sparql/
[9] Bio2RDF: https://datahub.io/organization/about/bio2rdf
[10] About BRIDG: http://bridgmodel.nci.nih.gov/
[11] CDISC SHARE *Metadata Repository* http://www.cdisc.org/standards/share
[12] BRIDG Model – Overview UML class diagram (Figure 5):
http://bridgmodel.nci.nih.gov/files/BRIDG_Model_4.1.1_html/EARoot/EA3.htm
[13] BRIDG Backbone diagram: http://bridgmodel.nci.nih.gov/files/BRIDG_Model_4.1.1_html/EARoot/EA3/EA60.htm
[14] Free Enterprise Architect viewer: http://www.sparxsystems.com/products/ea/downloads.html
[15] Bio2Rdf query for the PASCAL study: http://bio2rdf.org/sparql?query=define%20sql%3Adescribe-mode%20%22CBD%22%20%20DESCRIBE%20%3Chttp%3A%2F%2Fbio2rdf.org%2Fclinicaltrials%3ANCT01550003%3E&output=text%2Fturtle
[16] BRIDG Protocol Representation View:
http://bridgmodel.nci.nih.gov/files/BRIDG_Model_4.1.1_html/EARoot/EA6/EA210.htm
[17] Dokieli – official website: https://dokie.li/
[18] Dokieli usage examples: https://github.com/linkeddata/dokieli/wiki#examples-in-the-wild