Simplicity in Data Models

by Karen Coyle

Linked Data and the Charm of Weak Semantics

EDITOR'S SUMMARY

Evolving from database models using punch cards, strict linear relational databases and predefined object-oriented data structures, the triple statements underlying Semantic Web technologies bypass many design constraints to offer endless flexibility. Overcoming structure is challenging, especially the relatively recent structure formalized in the Functional Requirements for Bibliographic Records (FRBR). Though geared to easier access and interoperability and recognizing a multilevel bibliographic model, FRBR remains tied to translating entity-relation diagrams to data structures. Resource Description Framework (RDF) provides a more flexible way to express concepts, in which bibliographic models may be thought of as graphs of properties and relationships. But even RDF-based models can undermine that flexibility by mixing concept classes and data structures. The advantage of RDF classes is to provide semantics that enable a user to focus on similarities, not bound by contextual constraints.and success metrics.

KEYWORDS

data structures

Functional Requirements for Bibliographic Records

RDF

bibliographic records

linked data

graphs

Karen Coyle is a librarian with over 30 years of experience with library technology. She consults in a variety of areas relating to digital libraries, has published dozens of articles and reports (mostly available on her website, kcoyle.net) and has served on many standards committees. She works primarily on metadata development and technology planning and is currently investigating the possibilities offered by the Semantic Web and linked data technology. She can be reached at kcoyle<at>kcoyle.net

The graph pattern of Semantic Web data is a significant departure from the data models that preceded it. The first data processing methods on punched cards were essentially an automation of the ordered, linear card file. A punched card stack, however, had to be processed from the first entry to the last in order to extract data, which clearly had some disadvantages. Early database management systems used a hierarchical model that could query particular paths in order to arrive at results. Like the classified library shelving system, these hierarchies forced designers to provide one and only one place for each information unit, which naturally cut off some possible data combinations at the same time that it facilitated others.

The 1980s saw a great improvement in design in the form of the relational database management system. A relational database was much more efficient for searching on combinations of stored data elements. However, before you could store your data in such a database, you had to go through a tedious process of deciding on one, just one, view of your data that would form the physical database design. The resulting data structure was fixed – fixed before any data was entered into it – and changes were difficult and painful; adding a new kind of data meant changing the underlying structure of the database. This view had to accomplish a number of things simultaneously:

- It had to reduce duplication of data, striving to arrive at a design in which each data instance, like an inventory number, existed once and only once in your database.
- It had to establish identifiers for each separate unit of data and relationships between those units of data.
- It had to allow for efficient update and querying of the database.

The late 20th century movement toward object-oriented design took advantage, in programming and database design, of a human capability for organizing things based on commonalities. In object-oriented (O-O) design, those commonalities are determined by attributes and processing functions. All entities with similar attributes could be gathered into a single class, with the class defining the attributes that they have in common and with subclasses for more specific or variant concepts. In the O-O data view, classes are containers and controllers for data and functions. O-O designs were more extensible than relational ones, but still required the pre-definition of a basic data structure.

Because only pre-defined elements could be stored in these database management systems, we developed gatekeeper programs that only allowed in certain data and that ruled over the input forms used by anyone keying data directly into the database. We created macros that limited what searches could be done because some searches could bring the system to a grinding halt.

Semantic Web technologies, based on three-part statements (triples) and a graph structure, make many of these design constraints obsolete. Yet one of the hardest things about moving into today's data landscape is learning to give up old ideas of the fixed structure of data. It's like telling a technology developer that one day his pet can be a cat, the next day it can be a fish. This is unsettling. Yet there is much that is positive about this state of affairs, not the least being that we can give up our role as the tyrants of data and allow change to happen naturally.

What would natural change look like? I'll take my examples from the data environment that I am most familiar with – library resource metadata. Recent efforts to create a new model for library and archive resource metadata show how difficult it is for us to move past our grounding in structured data and to embrace the ever-changing, conceptually infinite graph. This new model centers on a conceptual model that we know as Functional Requirements for Bibliographic Records (FRBR).

FRBR, 1998

This story begins more than a half a century ago, and reaches its apex before the turn of the new millennium. It is rooted in a 1946 document, *Studies of Descriptive Cataloging: A Report to the Librarian of Congress*, an analysis of the library cataloging rules by a newly arrived Library of Congress cataloger, Seymour Lubetzky. Lubetzky's analysis noted, among other cogent observations, that the library cataloging rules of the time provided no explanation of the functional motivations for the decisions that catalogers were required to make. Unlike a proper domain analysis that was the expected technology focus of the latter decades of the 20th century, the cataloging rules contained no connection between data and services.

During that same period, the International Federation of Library Associations (IFLA) was working to formulate cataloging standards that would allow libraries across the globe to exchange information about their resources. At an IFLA meeting on international cataloging rules in Stockholm in 1990, the participants decided that a new vision of cataloging was needed. The new rules should adhere to Lubetzky's suggestion that each rule be grounded in the actual desired service to users of the catalog. The rules should also eliminate any data elements that were not absolutely necessary, promote sharing of data and reduce the costs of cataloging.

The task of developing this new vision was assigned to a small group of library representatives and of experts in the area of cataloging. The study group began its work in 1992, and in 1998 released its final report (www.ifla.org/en/publications/functional-requirements-for-bibliographic-records). The report is heavily influenced by a particular data design technique called entity-relation (E-R) analysis. Like the analyses that were performed by database designers in the 1980s and 1990s, FRBR outlined a structure that was designed, albeit imperfectly, around E-R principles. The FRBR conceptual model defined three groups of entities: bibliographic entities (work, expression, manifestation, item), agent entities (person, corporate body) and subject entities (concept, object, place, event). The latter two groups are compatible with the library practice of name and subject authorities, but the first group introduces, for the first time, a multilevel view of the bibliographic entity. Although the report does address user needs and also defines a core set of bibliographic elements, it is the E-R design that dominates the final report.

The design, which is often referred to in FRBR-related documentation as an “entity-relation/object oriented” view of bibliographic data, is, however, true to neither technology. Instead, it is a purely conceptual view of bibliographic data, as is affirmed in the final report from the study group. A conceptual E-R diagram, like FRBR, is a talking-points view that exhibits primarily a human-understandable model of the data realm under analysis. However, the difference between conceptual design (often considered the first step in developing an actual database model) and data design is not made clear in the practice of FRBR.

The IFLA group that continues to manage FRBR followed up the final report with a direct translation of the FRBR conceptual model into the W3C's Web Ontology Language (OWL), part of the suite of Semantic Web standards. This undertaking is compatible with a common misconception that the FRBR E-R diagrams translate directly to data structures. Other implementations of FRBR have also often treated the conceptual elements of the FRBR model as physical records. This treatment reflects another common misconception, which is that classes in the Semantic Web are structural in the way that classes are structured in object-oriented design.

Classes and Graphs

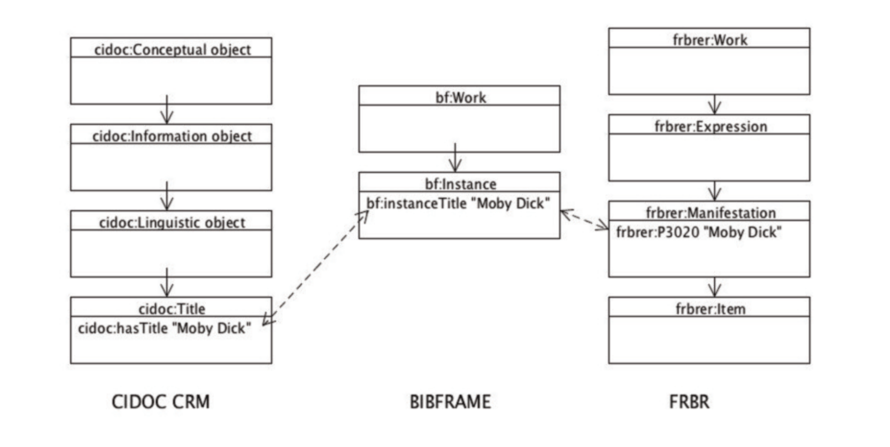
The Semantic Web protocol Resource Description Framework (RDF) comes out of work of the artificial intelligence (AI) community, which has an entirely different approach from that of previous information technology. Although machine intelligence and human intelligence are significantly different, AI attempts to model human thinking rather than the business processes that were the primary motivator for developments in previous information technology design. The study of human cognition has many facets, one of which revolves around our use of common concepts to understand the world. We walk down the street and say “Hey, look at that cat!” If instead we said “Look at that four-legged mammal” we would be considered either strange or facetious. Many conceptual categories are not exclusive. Said cat cannot be both a black cat and a white cat, but it can be a cat, a pet, the thing that howls in the night and also something owned by your next-door neighbor. We recognize apples even though some are red, some green and some yellow; regardless of their color, they all have “appleness.” If our categories were strict, if only red apples were allowed to be apples, we would be faced with an overwhelming number of different things to keep track of in our daily life. Communication would be tedious because there would be none of the generalizations that allow us to agree on a meaning with only a small hint at the nature of the thing being described.

Coming from AI work, RDF has a very different view of classes than that of hierarchies and taxonomies or even object-oriented data processing. In RDF, classes are not containers; they are non-exclusive categories that can help us use conceptual shortcuts in our interaction with the world. Like humans, your artificial intelligence must be able to understand that Bob can be a father, an employee, a baseball coach and a knitter. Where a hierarchical system would have forced you to decide on one and only one category for Bob, RDF allows you to describe Bob with as many categories as are useful, and none of them determines the sole nature of Bob. It is also the case that none of them determines a single data structure for Bob.

Multilevel Bibliographic Models

Prior to the FRBR study group, bibliographic records were monolithic, with each record representing one or more identical physical items in the library's collection. FRBR broke that monolith by creating a multilevel bibliographic model. While the four entities of FRBR's bibliographic model (work, expression, manifestation and item) are presented as universal, in fact there is significant debate, especially among catalogers of non-textual items, about the suitability of those four entities to items that they must catalog. Some communities, such as the museum community (www.cidoc-crm.org/), have found a need for more than one work or expression level in their cataloging. Others, like the art community, are describing a unitary creation where there is difficulty separating the work from the item. As communities of practice have developed their own models of bibliographic description, questions arise about the creation of bibliographic silos that will resist aggregation because of the difference in their data structures. The bibliographic framework work at the Library of Congress (www.loc.gov/bibframe) has been heavily criticized for proposing a new approach that uses a multilevel model that is not the same as FRBR.

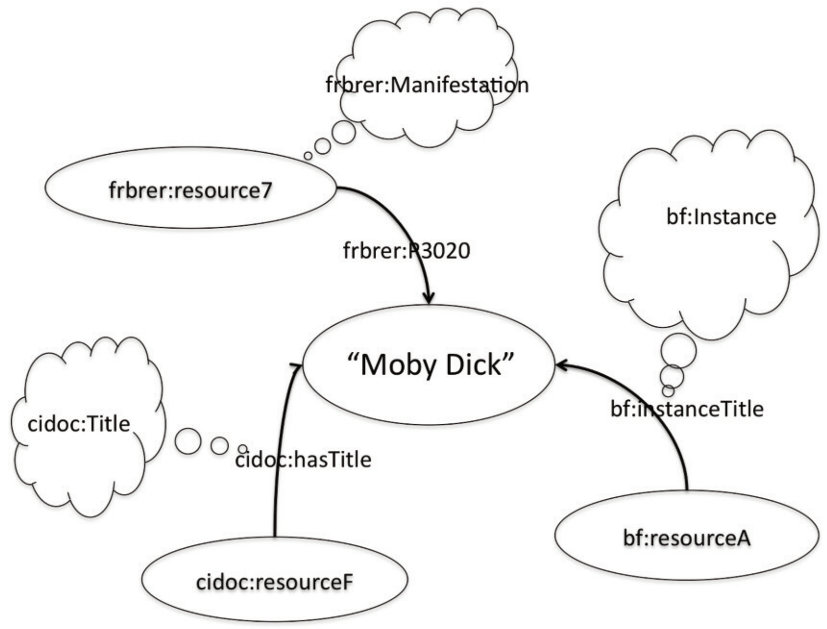
FIGURE 1. Structured view of bibliographic models



Each of these models makes use of RDF technology, but they also each confound classes, in the RDF sense, and data structures. They pre-define the bibliographic entities as entirely separate entities, each with only one set of exclusive attributes. In spite of the flexibility afforded us by RDF, these models imitate the pre-defined, fixed models of E-R and O-O.

It is the atomic nature of properties in graphs that gives the linked data cloud its potential to allow widespread sharing of data among different communities and data sources. By viewing the above bibliographic models as graphs of properties and relationships rather than structured records, it becomes clear that they have more in common than is evidenced in their respective high-level diagrams. The role of RDF classes is to provide additional semantics to properties for the purposes of inferencing, not to bind properties to a set structure, and the properties themselves can interoperate independently with other data in the linked data cloud. With this difference in mind it becomes easier to see that the data described in RDF graphs is recombinant precisely because properties are not bound by the context of their high-level models. Defining our data in RDF means that we free it from the pre-determination that was required by past data structures so that we can take advantage of similarities, not be thwarted by differences. ■

FIGURE 2. The recombinance of properties


frbrer:Manifestation
frbrer:resource7
frbrer:P3020
bf:Instance
"Moby Dick"
cidoc:Title
bf:instanceTitle
cidoc:hasTitle
bf:resourceA
cidoc:resourceF