Chapter 7: Describing Relationships

Chapter authors: Matthew Mayernik, Alberto Pepe, and Robert Glushko
(mattmayernik@ucla.edu, apepe@ucla.edu, glushko@berkeley.edu)

Table of Contents

7.1 Describing Relationships: An Overview 1
   7.1.1 Defining "Relationship" 3
   7.1.2 The Components of Relationships 3
   7.1.3 The Semantics of Relationships 5

7.2 Describing Relationships Among Concepts 6

7.3 Describing Relationships Among Things 7

7.4 Describing Relationships Among Words 8

7.5 Implementing Relationships in Organizing Systems 8

7.1 Describing Relationships: An Overview

A small boy in a medium-sized American family might describe that family as containing himself, his sister, his two brothers, and his mother and father. But he might realize that his mother’s husband is not his biological father, and as he gets older, he might describe that person as his stepfather or as his mother’s second husband. Maybe his sister is technically his half-sister, and his brother is his stepfather’s son born before his mother and father split up, but there isn’t a single word in his language that describes that relationship. The boy was a little confused and uncomfortable the last time he visited his father’s house because he didn’t know what to call his father’s new wife or her parents, who insisted on being called grandma and grandpa, giving him three sets of people who he calls by those names.

The parents have an easier time because their point of view in the family leads them to say “the kids” or “the children” and take a more general perspective on family structure. They can replace the pair-wise descriptions (“he is my brother”) with descriptions that are one-to-many—“I am the parent of these children”—by combining them, ignoring the properties that distinguish them in doing so.

In contrast, if this little boy were growing up in a China, he might not find these family relationships so confusing, because the Chinese kinship system makes very precise distinctions about generations, lineage, gender, and relative age (Baker, 1979). Languages and cultures differ greatly in how they distinguish and describe kinship, so our small boy might find the system of “family organization” easier to master in some countries and cultures than in others.

Kinship relationships and terms are widely studied, and their role in culture and daily living make them a good starting point for understanding relations, which are central to organizing systems for every kind of thing or type of information. Relationships are crucial to organizing systems for they describe the ways
by which entities are associated to one another. In the realm of information systems, relationships have been described as “the stuff of which information is made” (Kent, 1979): they use as their components the pieces of information (Chapter 2) that we use when we describe individual things (Chapter 3), and when we define classes or categories or types of things (Chapter 5), we are specifying sets of relationships that go together.

In this chapter, we use examples from the kinship domain because the participants in kinship relations are people, a type of entity familiar to us. Likewise, the names and meanings of kinship relations like, “is parent of” or “is sibling of” are familiar. Entity and relation types differ across domains, of course; in the domain of information resources, common entities include web pages, journal articles, books, data sets, metadata records, and XML documents. Relations in this domain include relationship statements such as “is the author of,” “is published by,” “has publication date,” “is derived from,” “has subject keyword,” “is located at,” and so on.

With these preliminary notions in mind, it is important to introduce a more abstract or domain-independent vocabulary for talking about relationships between entities, about the structural and semantic dimensions on which relationships differ, and about the relationships between words because words are needed to express relationships.

Why do we care about describing relationships? We care about relationships for two main reasons: 1) everything is related to something, and 2) relationships are powerful means of navigating and finding resources.

Everything is related to something; nothing in our world exists in complete isolation. Humans are inescapably related to generations of progenitors, and in most cases we also have social networks of friends, co-workers, and casual acquaintances to whom we are related in various ways. Similarly, a hammer has a manufacturer, an owner, and relations to product lines and other forms of hammers. Information resources also abound with relationship. Books have creators, publishers, and owners, and they are often related to other books (such as translations, compiled works, or screenplays of novels, as we learned in Chapter 2’s discussion of the abstraction hierarchy). Scientific data sets are related to journal publications, technical reports, and news articles. Business documents are related to companies, projects, accounts, or reports. Web sites relate to other web pages through hyperlinks.

As these examples indicate, relationships are powerful means of navigating and finding resources. The power of relationships is well known in human societies. We often hear that our access to money, jobs, and political power is all about “who you know.” Access to information is similarly highly social. “Networking,” the act of meeting and getting to know other people in specific social situations, is all about building relationships with people who might have access to important, valuable, or useful information. In information systems, relationships are also a central means of information organization and retrieval. Hyperlinks, for example, provide direct connections between related web sites, but many other kinds of relationships between information resources can be leveraged to help users of information systems as well. Library catalogs allow users to find all the works by a single author or about a particular subject. Similarly, article databases allow users to search for all articles published in a particular journal, indexed under a particular term, or that contain a user-specified search phrase or set of terms.

We discuss many of these relationships in more depth later in this chapter, but first, let’s step back and talk about what exactly we mean by “relationships”.

7.1.1 Defining “Relationship”

William Kent (1979) defined a relationship as “an association among several things, with that association having a particular significance” (p. 73). Kent points out that the name of a relationship is typically a statement of the reason for the association. In fact, as he states, “we sometimes invent names which are whole phrases, such as ‘is-employed-by,’” if our language does not contain a term appropriate to defining a particular relationship. Kent continues, “Note that the reason is an important part of the relationship. Just identifying the pair of objects involved is not enough; several different relationships can exist among the same objects, and the order of the objects in the relationship usually matters” (p. 73). So, in discussing relationships, we need to know more than just which entities are related, we need to know how they are related.

7.1.2 The Components of Relationships

Relationships do not exist by themselves; they only exist as associations between two or more “things” in the world. Thus, in order to understand and express relationships, we must understand what kinds of “things” populate the domains we are describing. Entity-relationship modeling is a common method used to build this understanding. Entity-relationship modeling was developed in the mid-1970s as a technique for building database structures (Chen, 1976), but its underlying scheme is found in many contemporary data representation formats. In a nutshell, an entity is a “thing” that can be distinctly identified, while a relationship is an association between entities. In information organizing systems, a common conceptualization of the entity-relationship model is the subject-predicate-object information structure. For example, the relationship between two spouses (Barack and Michelle Obama) could be expressed as follows:

\[
\text{Barack Obama} \rightarrow \text{married to} \rightarrow \text{Michelle Obama}
\]

This simple information structure, called a triple, consists of a subject (Barack Obama), an object (Michelle Obama), and a predicate, expressing the relationship existing between subject and object (→ is married to →). Although several variants of this conceptual scheme exist, by and large, relationships between entities can be expressed using a subject-predicate-object model. As discussed above, some relationships naturally involve more than two entities. For example, Barack and Michelle Obama have two children, Sasha and Malia. Yet, these relationships can always be collapsed to triples. For example:

\[
\begin{align*}
\text{Barack Obama} \rightarrow \text{is parent of} \rightarrow \text{Sasha Obama} \\
\text{Barack Obama} \rightarrow \text{is parent of} \rightarrow \text{Malia Obama} \\
\text{Michelle Obama} \rightarrow \text{is parent of} \rightarrow \text{Sasha Obama} \\
\text{Michelle Obama} \rightarrow \text{is parent of} \rightarrow \text{Malia Obama}
\end{align*}
\]

Thus, the notion that Barack and Michelle Obama are the biological parents of Sasha and Malia Obama cannot be expressed in a single relationship. It is expressed as a number of separate triples. Yet, this notion can be reconstructed by using reasoning to interpret the relationships expressed in our collection of statements. In this context, reasoning is the capability of inferring a set of conclusions from a set of statements. For the example above, one could create an inference rule by which, if two subjects are found to be parents (“is parent of”) of an object, then they are its biological parents—i.e., Barack and Michelle Obama are biological parents of both Sasha and Malia. By extension, one could also create a rule by which, if two objects have the exact same set of parents, then they are full siblings—i.e., Sasha and Malia are biological sisters.
As this example illustrates, relationships can vary in dimension. The number of participants in a relationship is called the **arity** or **cardinality** of the relationship. There might be N things in a relationship, but N-ary relationships can always be modeled as a set of binary ones, which are usually easier to understand and implement in an organizing system—but not always. For example, the “child-of” relationship, as illustrated by “Sasha Obama -> child-of -> Michelle Obama,” is N-ary because a mother can have more than one child. Treating the Obama family’s “child-of” relationships as multiple binary relationships, “Sasha Obama -> child-of -> Michelle Obama” and “Malia Obama -> child-of -> Michelle Obama,” introduces redundancy in the representations and loses the explicit representation that the children are siblings. This can be inferred through an algorithmic computation, but such processing always has a cost in time and efficiency.

So there are at least two things in a relationship, where a “thing” consists of one of the “entities” discussed in Chapter 3. It is also worth repeating from Chapters 5 and 6 that we can describe the same things as part of simple relationships or aggregations or classes. This means we can express relationships between instances and between the classes or categories or types of things that the instance is. The former can be considered as concrete or specific relationships and the latter as abstract ones. In the kinship domain, a concrete relationship is an expression involving specific people: Barack Obama > is married to > Michelle Obama. Or to relate an entity to a class we could say “Barack Obama — belongs to—> Presidents of the United States,” indicating that is the member of the class “Presidents of the United States.” An analogous abstract relationship involving entity types rather than entities is Person > is married to > Person. This abstract relationship can be considered to be a collection of all the concrete relationships with the same relation.

From this description, it becomes clear that a number of relationships about a given subject, when taken together, constitute a network. We can model a network such that the entities are the vertices and the relationships are the edges connecting the vertices. For example, a simple network composed of the above statements is depicted in Figure 7.1.

**Figure 7.1. Entity-relationship representation of the Obama family**
This subject-predicate-object information structure is used to describe resources on the Semantic Web, which is discussed in more depth below.

7.1.3 The Semantics of Relationships
In order to describe relationships among information resources, we need to understand the semantics of the relationships -- what the relations mean, or the reasons why the resources are related. Some relationships exist because physical, biological, or structural constraints cause certain types of things to go together. For example, a child has two—and only two—parents by biological necessity. Other relationships are established by explicit design to satisfy some human or computational requirements. For example, the manufacturers of digital cameras designed the EXIF specification to define the information that is associated with a digital image when it is captured by the camera. If some futuristic digital camera used biometric security techniques like retinal scanning so that only its owner could use it, it would be an explicit design choice to extend EXIF to associate the photographer’s name with a picture.

Information systems often face a trade-off as to whether relationships will be explicitly or implicitly designed and used. Explicit relationship structures, such as subject headings and uniform titles in catalog records, provide formal structures for retrieving related resources. On the other hand, web search engines leverage implicit relationships (the co-occurrences of query terms) to retrieve web pages.

The **Semantic Web** is a set of technologies designed to make the meaning of web resource relationships more explicit (See sidebar).

*Sidebar on Semantic Web goes here*

The vision of a “Semantic Web” was proposed by Tim Berners-Lee, the inventor of the Web, James Hendler, and Ora Lassila in a classic 2001 paper (Berners-Lee at al, 2001). The Web’s astonishingly rapid adoption as a publishing medium was largely due to the conceptual and technical simplicity of HTML for using tags to mark up pieces of text according to how they should appear. Berners-Lee consciously made the tradeoff for simplicity rather than expressive precision, and the idea embodied in HTML that “how you tag is what you see” is so easy to understand that school kids and computer-wary grandmothers can make web pages.

But designing the Web “for eyes” means that it couldn’t be understood “by machines” and Berners-Lee and others came to realize that the Web could reach its full potential only it could represent what text meant and not just how it should appear. For this goal HTML was fundamentally inappropriate, because its fixed tag set cannot express anything about the meaning of the content between its tags or assert relationships between pieces of content or other web resources.

So we have to invent XML and other technologies... (more to come in this sidebar; cite some xml and semantic web technologies references)

*End sidebar*

Semantic web technologies include XML, a data structure standard for encoding information, and RDF, a standard model for data interchange. RDF provides a computer-readable syntax for creating the subject-predicate-object information structures described above. RDF allows the nature of relationships to be
precisely defined in a way that normal web hyperlinks do not. If hyperlinks declare that a relationship exists between two resources, RDF triples declare that a relationship exists and describe exactly the relationship is, as in the Barack Obama $\rightarrow$ is married to $\rightarrow$ Michelle Obama example.

The “semantic” nature of the Semantic Web should not be understood in the literal sense of creating a web of meanings. As Karen Coyle (2010a) describes it, “The Semantic Web uses the term as it is defined in an area of mathematics known as formal languages. For example, computer programming languages use formal semantics that define the set of possible computations that the language can perform, like addition, subtraction, and division” (p. 19). According to this understanding of “semantic,” RDF allows relationships to be precisely defined in syntax that enables computer understanding and processing.

Using the Barack Obama $\rightarrow$ is married to $\rightarrow$ Michelle Obama example, defining these relationships in a RDF syntax would enable us to state that there are two entities, Barack Obama and Michelle Obama, and that they are related by the “is married to” relation.

These statements, however, are not enough for a machine to understand what marriage means or even to know that this relationship is bi-directional. Yet, we can declare that the relationship “is married to” is bi-directional. This would allow a machine to use formal logic to infer that Barack Obama $\rightarrow$ is married to $\rightarrow$ Michelle Obama is equivalent, by definition, to Michelle Obama $\rightarrow$ is married to $\rightarrow$ Barack Obama. We will discuss in section 7.5 how describing relationships using Semantic Web technologies is becoming more widespread in many information-intensive institutional settings.

### 7.2 Describing Relationships Among Concepts

Information professionals have put much effort into describing relationships between concepts. Classification schemes, taxonomies, and ontologies are the most common means to formally structure and describe conceptual relationships. All three of these tools are very similar in concept, with taxonomies and ontologies typically referring to specialized forms of classification.

A classification, discussed in chapter 6, is used to organize a collection of information resources. In classifications, classes are ordered using a predetermined set of principles. A taxonomy, according to Garshon (2004), is a “subject-based classification that arranges the terms in the controlled vocabulary into a hierarchy without doing anything further” (p. #?) Biological taxonomies are the most well known taxonomic systems. For example, the Linnean biological taxonomy uses Kingdom, Phylum, Class, Order, Family, Genus, and Species ranks to classify living organisms.

Ontologies are another form of classification. “Ontology,” in the philosophical sense, refers to the study of what exists in reality and the general features and relations of whatever that might be (Hofweber, 2009). The term ontology, however, has been adopted by computer science to refer to a computer-usable engineering artifact that maps terms, relationships, and meanings in a specific knowledge domain. An ontology is “constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words” (Guarino, 1998, p. 4, italics in original). An ontology is a kind of classification in that it relates concepts to terms and definitions, provides maps of relationships among terms, and supports information retrieval, among other functions (Soergel, 1999).

Classifications, taxonomies, and ontologies are important information organization and retrieval tools for a number of reasons. First, they all enable a common understanding of the structure of information in a particular domain. They also help to make the assumptions about knowledge and information in a
particular domain explicit by declaring term definitions and relationships between terms. Finally, they
are useful in separating the domain specific terms from more general terms, and in this way provide a
tool to analyze structures of domain knowledge.

7.3 Describing Relationships Among Things
As we discussed briefly in Chapter 2, things, or entities, in our world are often parts of larger things or
might themselves be composed of multiple parts. For example, we can name many components of a car,
but we can also still refer to a “car” as a singular object. These kinds of whole/part relationships are
called meronomies. A meronomy usually takes the form of a “has-a” or “is-part-of” relationship, as
when we say “a car has a steering wheel” or “a steering wheel is part of a car.” Meronomies are a type
of hierarchical relationship.

Hypernyms and hyponyms are another type of hierarchical relationship. Instead of indicating a
part/whole relationship, hypernyms and hyponyms indicate a general/specific relationship. For example,
“bird” is a hypernym of “pigeon,” “penguin,” etc., and correspondingly, “pigeon” and “penguin” are
hyponyms of “bird.” These are examples of “is-a” relationships, such as “a penguin is a bird,” where the
hypernym subsumes the hyponym.

Transitivity is another common type of relationship that can be observed between things in the world.
Transitivity refers to the way that relationships can be implicit in two other relationships. The most
common way to describe transitivity is to say, “if A is B, and B is C, therefore A is also C”. For example,
one of the most used relationships expressed on the Semantic Web is the “owl:sameAs” statement,
defined by the Web Ontology Language (OWL). The owl:sameAs statement simply indicates that two URI
references on the Semantic Web actually refer to the same thing. Thus, if a URI is the object of a
owl:sameAs statement and the subject of another, a transitivity relation applies. For example, the
statements President Obama - owl:sameAs -> Barack Obama, and Barack Obama - owl:sameAs ->
Barack Hussein Obama imply that President Obama - owl:sameAs -> Barack Hussein Obama.

In some relationships, however, the order in which the components are expressed doesn’t affect the
meaning of the relationship. These are referred to as “symmetric” relationships. Because “is a brother
of” is a symmetric relationship, “A is a brother of B” means the same thing as “B is a brother of A.” But
more often, the order of the components matters, as in transitive relationships. Hypernyms and
hyponyms, for example, are transitive relationships, but not symmetric. All pigeons are birds, but not all
birds are pigeons.

7.4 Describing Relationships Among Words
Most information systems are based on words. Relationships between words are therefore central to
information organization and retrieval systems. Individual words may be related in many different ways.
Here, we focus on word-to-word relationships that are based on word forms and meanings.

Word form relationships are based on the idea of root words. A root word is simply the base word from
which other words are derived through inflectional or derivational processes. For example, “cloud” is
the root word of “cloudy,” “clouds,” and “cloudier,” while “hammer” is the root word of “hammered,”
hammering,” and “hammers.” Words can be related through pluralization, through different tense
forms (e.g., gerund, past participle), and through prefix and suffix modifiers like “re-,” “de-,” “-ment,”
and “-ly.” Suffixes and prefixes modify the meaning of words in significant ways while retaining
important relations to the root words.
These relationships are important in organizing systems because which ones that are explicitly recognized determine what word forms are treated as equivalent. Similarly, these relationships are important to recognize in search engines and natural language processing, where inflections and derivations are often “undone” to identify the root form of words. Stemming refers to the process of removing prefixes and suffixes from words in order to determine the root word of interest. For example, if a user enters “clouds” into a search box of an information system that uses stemming, the system will return items that use the word “cloud” along with those that use the word “clouds.” Similarly, a search for “hammering” will return items that use the word “hammer” when stemming is occurring (this is a simplification; Chapter 11 presents a more detailed and nuanced view of stemming because organizing systems and search engines don’t all use the same stemming algorithms, and some do not use stemming at all).

Relationships based on word meaning also occur in a variety of ways. A single word may exhibit polysemy, that is, it may have multiple meanings. For example, the word “pitch” can refer to any number of objects and activities, from roofing material, to baseball, to boating, to music. Different words may be synonyms, and have identical or closely related meanings, such as “teach” and “instruct.” In contrast, words may be antonyms and have directly opposite meanings, such as “strength” and “weakness.” Classifications, taxonomies, and ontologies use “see” and “see also” references, which point from one word to another, to indicate a meaning-based word-to-word relationship.

7.5 Implementing Relationships in Organizing Systems

We use relationships when we create or follow links between web pages, but the reason for or the type of the link isn’t often explicitly or consistently encoded. Conventions for expressing the relationships between documents are better defined and followed in scholarly writing; an author might say “See Glushko & Borgman, 2011” when referring to it as consistent with his point of view, and use “but” as a contrasting citation signal and say “But see Glushko & Borgman 2011” to express the relationship that the cited document disagrees with him. The pattern of citations for individual authors can be used to characterize their research interests and point of view; the pattern of citations between authors can be used to infer an “invisible college” or detect an emerging discipline (Price, 1963; Lievrouw, 1989; Zuccala, 2006).

The expression of relationships between documents is especially nuanced in legal contexts, where the use of legal cases as precedents makes it essential to distinguish precisely the point that a new ruling lies on the relational continuum between “Following” and “Overruling” with respect to a case it cites. Furthermore, because legal opinions not only reference direct statues but also interpretations and rulings from other opinions, a single document can be related to hundreds of other documents of varying types.

Much of our thinking about relationships in organizing systems for information comes from the domain of bibliographic cataloguing of library resources and the related areas of classification systems and descriptive thesauri. Ad hoc techniques devised for describing the items in ancient libraries progressively became more robust and systematic over time. By the late 19th century, numerous sets of cataloguing rules had been proposed that standardized author and title names and their uses in bibliographic descriptions. More importantly, the scope of descriptions was expanded to include cross-references to related items. Cross references dramatically increase the value of the library catalog, because it is no longer just an aid to finding a specific item like a particular edition of Shakespeare’s
Macbeth. Instead, the network or web of relationships between items conveys the scope and depth of the library’s contents by bringing together everything that relates to the same abstract work (remember the abstraction hierarchy from Chapter 2); in the Macbeth example, this set includes scores of editions and translations as well as numerous expressions in film, opera, ballet, and other forms. In addition, cross-references in subject classifications assigned to bibliographic entities express the conceptual structure of the domains of knowledge represented in the collection.

When printed card catalogs gave way to online catalogs in the second half of the 20th century, it became both possible and necessary to formalize models for bibliographic and referential relationships to make online catalogs and digital libraries operate effectively. Bibliographic relationships provide an important means to build structure into library catalogs. Barbara Tillett (1991), in a study of 19th and 20th century catalog rules, found that many different catalog rules have existed over time to describe bibliographic relationships. Out of this study, Tillett developed a taxonomy of bibliographic relationships: equivalence, derivative, descriptive, whole-part, accompanying, sequential or chronological, and shared characteristics. Smiraglia (1994) created a taxonomy that expanded on Tillett’s derivative relationship, and included seven categories: simultaneous derivations, successive derivations, translations, amplifications, extractions, adaptations, and performances.

These studies of bibliographic relationships influenced the Functional Requirements for Bibliographic Records report (FRBR, 1998). Section five of the FRBR report focuses on the relationships between entities and their context within the model. As described in FRBR, the role of relationships is to “serve as the vehicle for depicting the link between one entity and another, and thus as the means of assisting the user to navigate the universe that is represented in a bibliography, catalogue, or bibliographic database” (FRBR 1998, pg. 56). The four Group 1 entities can be related to each other in a variety of ways: work-work, expression-expression, work-expression, expression-manifestation, and so on. This again is the abstraction hierarchy of the entities that participate in relations and prescribes the structure and semantics of those relations introduced in Chapter 2.

Bibliographic relationships are common among library resources. Smiraglia and Leazer (1999) found that approximately 30% of the works in the Online Computer Library Center’s (OCLC) WorldCat union catalog have associated derivative works. Relationships among items within these bibliographic families differ, but the average family size for those works with derivative works was found to be 3.54 items. Additionally, “canonical” works that have strong cultural meaning and influence, such as Shakespeare’s plays and the Bible, have very large and complex bibliographic families. These findings, consistent with other studies (Smiraglia 1994, Tillett 1992), make a strong case for the development of library information systems that explicate relationships between items. The Resource Description and Access (RDA) next-generation cataloging rules are attempting to make the description of these bibliographic relationships more important in cataloging practice.

RDA utilizes RDF to declare and store relationships among bibliographic materials. The move in RDA to encode bibliographic data in RDF stems from the desire to make library catalog data more web accessible (Coyle, 2010b). MARC records can be formatted and displayed on the web—online library catalogs do this—but the data within the records are not available for re-use, re-purposing, or re-arranging by researchers, patrons, or librarians. As web-based data mash-ups, application programming interfaces (APIs), and web searching are becoming ubiquitous and expected, library data are becoming increasingly isolated. The developers of RDA see RDF as the means for making library data more widely available online.
In addition to simply making library data more web friendly, RDA seeks to leverage the distributed nature of the Semantic Web. Once rules for describing resources, and the relationships between them, are declared in RDF syntax and made publicly available, the rules themselves can be mixed and mashed up. Creators of information systems that use RDF can choose elements from any RDF schema. For example, we can use the Dublin Core metadata schema (which has been aligned with the RDF model) and the Friend of a Friend schema (a schema to describe people and the relationships between them) to create a set of metadata elements about a journal article that goes beyond the standard bibliographic information. The Dublin Core allows us to state the title of the article, the publication information, and the author of the article, while the Friend of a Friend schema allows us to state supplementary biographical and social information about the author (Harper, 2010).

This use of individual elements from multiple metadata schemas stems from the recognition that any single metadata schema may not be sufficient to describe all of the particulars of a single resource or collection, an issue we mentioned in Chapter 4. For example, many cataloging researchers have recognized that online catalogs do not do a very good job of illustrating bibliographic relationships among items, both due to catalog display design and to the limitations of AACR2/MARC data within catalog records (Tillett, 2005). Author name authority databases, for example, provide information for variant author names, which can be very important in finding all of the works by a single author, but this information is not held within a catalog record. RDA is being designed to leverage multiple metadata schemas in order to provide more complete and interconnected data about works, authors, publications, publishers, and subjects.

As of late 2010, RDA is still in the development and testing phase, but the progress in moving to RDF is well underway. The FRBR entities, RDA data elements, and RDA value vocabularies have been defined in alignment with RDF using the Simple Knowledge Organization System (SKOS, 2010). The SKOS is a “RDF-compliant language specifically designed for term lists and thesauri” (Coyle, 2010c). The SKOS web site provides lists of registered RDF metadata schemas and vocabularies. From these, information system designers can create application profiles for their resources, selecting elements from multiple schemas, including the FRBR and RDA vocabularies.

Related formalization has occurred with specifications for controlled vocabularies and thesauri used in labeling and describing the components of relational expressions and the relations themselves. Hierarchical relationships are commonly used in information organization and retrieval systems. In addition to the FRBR work-expression-manifestation-item hierarchical concept model, a common usage of hierarchical relationships is the use of “broader term” and “narrower term” designations in classifications and thesauri, such as the Library of Congress Subject Headings. The benefit of using hierarchical relationships, such as meronomies, is that properties of higher-level entities can be inherited down the hierarchy. For example, the owner of a car is also the owner of the steering wheel. Not all properties are inherited down a hierarchy, however, as a blue car may or may not have a blue steering wheel. Word clustering algorithms in information retrieval systems also take advantage of hierarchical relationships. For example, hypernyms and hyponyms can be used to automatically cluster terms together for metadata schemes, retrieval algorithms, and web search interfaces (Hearst, 1992; Hearst, et al., 2002).

A somewhat analogous evolution has taken place in thinking about relationships in the domains of information system design and information architecture. Ad hoc programming practices that did not cleanly separate the representation of information structures from the processes that operate on them gradually gave way to more structured techniques for software architectures and information
management in which the information components and objects used by applications were explicitly represented. This evolution was initially driven in the 1970s and 1980s by the need to improve software productivity and increase software reuse, and since then by the need to integrate information systems and achieve interoperability in a global information economy.

Comparing and combining information between different systems involves identifying corresponding components and relationships in each system and possibly transforming them to some degree of equivalent meaning, as we’ll see in Chapter 11 when we look at interoperability. For example, an Internet shopping site might present customers with a product catalog whose items come from a variety of manufacturers who describe the same products in different ways. Likewise, the end-to-end process from customer ordering to delivery requires that customer, product and payment information pass through the information systems of different firms. Creating the necessary information mappings and transformations is tedious or even impossible if the components and relationships among them aren’t formally specified for each system. In contrast, when these models exist as data or document schemas or as classes in programming languages, identifying and exploiting the relationships between the information in different systems to achieve interoperability or to merge different classification systems can often be completely automated. Because of the substantial economic benefits to governments, businesses, and their customers of more efficient information integration and exchange, efforts to standardize these information models are important in numerous industries.

References:


http://dx.doi.org/10.3115/992133.992154


