Chapter 6. Describing Relationships and Structures

Robert J. Glushko
Matthew Mayernik
Alberto Pepe
Murray Maloney

6.1. Introduction

A family is a collection of people affiliated by some connections, such as common ancestors or a common residence. The Simpson family includes a man named Homer and a woman named Marge, the married parents of three sibling children, a boy named Bart and two girls, Lisa and Maggie. This magical family speaks many languages, but most often uses the language of the local television station. In the English-speaking Simpson family, the boy describes his parents as his father and mother and his two siblings as his sisters. In the Spanish speaking Simpson family he refers to his parents as *su padre y su madre* and his sisters are *las hermanas*. In the Chinese Simpson family the sisters refer to each other according to their relative ages; Lisa, the elder, as *jiě jie* and, Maggie, the younger, as *mèi mei*.

Kinship relationships are ubiquitous and widely studied, and the names and significance of kinship relations like “is parent of” or “is sibling of” are familiar ones, making kinship a good starting point for understanding *relationships* in organizing systems. An organizing system can make use of existing relationships among resources, or it can create relationships by applying organizing principles to arrange the resources. Organizing systems for digital resources or digital description resources are the most likely to rely on explicit relationships to enable interactions with the resources.

In a classic book called *Data and Reality*, William Kent defines a *relationship* as an association among several things, with that association having a particular significance. “The things being associated,” the components of the relationship, are people in kinship relationships but more generally can be any type of resource (Chapter 4), when we relate one resource instance to another. When we describe a resource (Chapter 5), the components of the relationship are a primary resource and a description resource. If we specify sets of relationships that go together, we are using these common relationships to define resource types or classes, which more generally are called categories (Chapter 7). We can then use resource types as one or both the components of a relationship when we want to further describe the resource type or to assert how two resource types go together to facilitate our interactions with them.

We begin with a more complete definition of relationship and introduce five perspectives for analyzing them: semantic, lexical, structural, architectural, and implementation. We then discuss each perspective, introducing the issues that each emphasizes, and the specialized vocabulary needed to describe and analyze relationships from that point of view.

6.2. Describing Relationships: An Overview

The concept of a relationship is pervasive in human societies in both informal and formal senses. Humans are inescapably related to generations of ancestors, and in most cases, they also have social networks of friends, co-workers, and casual acquaintances to whom they are related in various ways. We often hear that our access to information, money, jobs, and political power is all about "who you know," so we strive to "network" with other people to build relationships that might help us expand our access. In information systems, relationships between resources embody the organization that enables finding, selection, retrieval, and other interactions.

Most organizing systems are based on many relationships to enable the system to satisfy some intentional purposes with individual resources or the collection as a whole. In the domain of information resources, common resources include web pages, journal articles, books, datasets, metadata records, and XML documents, among many others. Important relationships in the information domain that facilitate purposes like finding, identifying, and selecting resources include "is the author of," "is published by," "has publication date," "is derived from," "has subject keyword," "is related to," and many others.

When we talk about relationships, we specify both the resources that are associated along with a name or statement about the reason for the association. Just identifying the resources involved is not enough because several different relationships can exist among the same resources; the same person can be your brother, your employer, and your landlord. Furthermore, for many relationships the *directionality* or ordering of the participants in a relationship statement matters; the person who is your employer gives a paycheck to you, not vice versa.

In this chapter, we analyze relationships from several different perspectives:

Semantic perspective: The essential one; it characterizes the meaning of the association between resources. (§6.3)

Lexical perspective: How the conceptual description of a relationship is expressed using words in a specific language. (§6.4)

*Structural perspective: A*nalyzes the actual patterns of association, arrangement, proximity, or connection between resources. (§6.5)

Architectural perspective: Emphasizes the number and abstraction level of the components of a relationship, which together characterize its complexity. (§6.6)

Implementation perspective: How the relationship is implemented in a particular notation and syntax and the manner in which relationships are arranged and stored in some technology environment. (§6.7)

6.3. The Semantic Perspective

To describe relationships among resources, we need to understand what the relations mean. This *semantic perspective* is the essence of relationships and explains why the resources are related, relying on information that is not directly available from perceiving the resources. In our Simpson family example, we noted that Homer and Marge are related by marriage, and also by their relationship as parents of Bart, Lisa, and Maggie, and none of these relationships are directly perceivable. “Homer is married to Marge” is a semantic assertion, but “Homer is standing next to Marge” is not.

Semantic relationships are commonly expressed with a predicate with one or more arguments. A *predicate* is a verb phrase template for specifying properties of objects or a relationship among objects. In many relationships, the predicate is an action or association that involves multiple participants that must be of particular types, and the arguments define the different roles of the participants.

We can express the relationship between Homer and Marge Simpson using a *predicate(argument(s))* syntax as follows:

is-married-to (Homer Simpson, Marge Simpson)

The sequence, type, and role of the arguments are an essential part of the relationship expression. The sequence and role are explicitly distinguished when predicates that take two arguments are expressed using a *subject-predicate-object* syntax, often called a *triple* because of its three parts:

Homer Simpson → is-married-to → Marge Simpson

However, we have not yet specified what the “is-married-to” relationship means. People can demonstrate their understanding of “is-married-to” by realizing that alternative and semantically equivalent expressions of the relationship between Homer and Marge might be:

Homer Simpson → is-married-to → Marge Simpson
Homer Simpson → is-the-husband-of → Marge Simpson
Marge Simpson → is-married-to → Homer Simpson
Marge Simpson → is-the-wife-of → Homer Simpson

Going one step further, we could say that people understand the equivalence of these different expressions of the relationship because they have semantic and linguistic knowledge that relates some representation of “married,” “husband,” “wife,” and other words. None of that knowledge is visible in the expressions of the relationships so far, all of which specify concrete relationships about individuals and not abstract relationships between resource classes or concepts. We have simply pushed the problem of what it means to understand the expressions into the mind of the person doing the understanding.

We can be more rigorous and define the words used in these expressions so they are “in the world” rather than just “in the mind” of the person understanding them. We can write definitions about these resource classes:

- The conventional or traditional marriage relationship is a consensual lifetime association between a husband and a wife, which is sanctioned by law and often by religious ceremonies;
- A husband is a male lifetime partner considered in relation to his wife; and
- A wife is a female lifetime partner considered in relation to her husband.

(If these definitions upset you here, you will feel better in §6.6.1).

Definitions like these help a person learn and make some sense of the relationship expressions involving Homer and Marge. However, these definitions are not in a form that would enable someone to completely understand the Homer and Marge expressions; they rely on other undefined terms (consensual, law, lifetime, etc.), and they do not state the relationships among the concepts in the definitions. Furthermore, for a computer to understand the expressions, it needs a computer-processable representation of the relationships among words and meanings that makes every important semantic assumption and property precise and explicit. We will see what this takes starting in the next section.

6.3.1. Types of Semantic Relationships

In this discussion we will use *entity type, class, concept*, and *resource type* as synonyms. *Entity type* and *class* are conventional terms in data modeling and database design, *concept* is the conventional term in computational or cognitive modeling, and we use *resource type* when we discuss organizing systems. Similarly, we will use *entity occurrence, instance*, and *resource instance* when we refer to one thing rather than to a class or type of them.

There is no real consensus on how to categorize semantic relationships, but these three broad categories are reasonable for our purposes:

Inclusion Relationship

One entity type contains or is comprised of other entity types; often expressed using “is-a,” “is-a-type-of,” “is-part-of,” or “is-in” predicates.

Attribution Relationship

Asserting or assigning values to properties; the predicate depends on the property: “is-the-author-of,” “is-married-to,” “is-employed-by,” etc.

Possession Relationship

Asserting ownership or control of a resource; often expressed using a “has” predicate, such as “has-serial-number-plate.”

All of these are fundamental in organizing systems, both for describing and arranging resources themselves and for describing the relationships among resources and resource descriptions.

6.3.1.1. Inclusion

There are three different types of inclusion relationships: class inclusion, meronymic inclusion, and topological inclusion. All three are commonly used in organizing systems.

Class inclusion is the fundamental and familiar “**is-a,**” “**is-a-type-of**,” or “**subset**” relationship between two entity types or classes where one is contained in and thus more specific than the other more generic one.

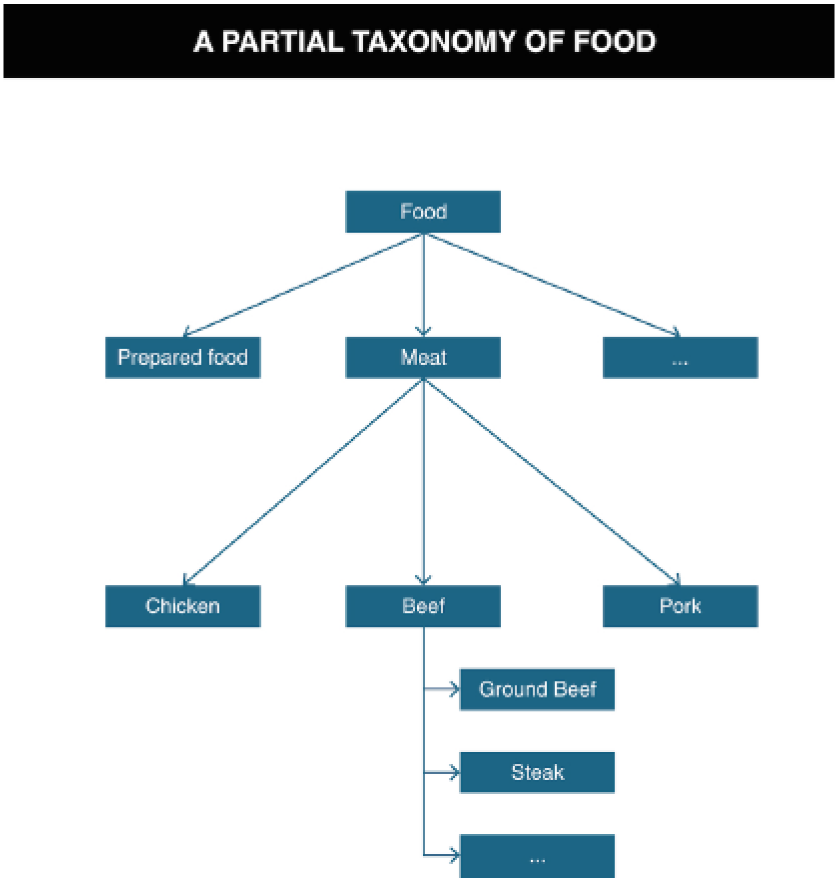
> **Meat → is-a → Food**

A set of interconnected class inclusion relationships creates a hierarchy, which is often called a *taxonomy*.

> **Meat → is-a → Food**
> **Dairy Product → is-a → Food**
> **Cereal → is-a → Food**
> **Vegetable → is-a → Food**
> **Beef → is-a → Meat**
> **Pork → is-a → Meat**
> **Chicken → is-a → Meat**
> **Ground Beef → is-a → Beef**
> **Steak → is-a → Beef**
> **...**

A visual depiction of the taxonomy makes the class hierarchy easier to perceive. See Figure 6.1, “A Partial Taxonomy of Food.”

Each level in a taxonomy subdivides the class above it into sub-classes, and each sub-class is further subdivided until the differences that remain among the members of each class no longer matter for the interactions the organizing system needs to support. We discuss the design of hierarchical organizing systems in §7.3, “Principles for Creating Categories.”



A partial taxonomy of food distinguishes the categories or prepared food from meat, distinguishes chicken, beef, and pork as subcategories of meat, and distinguishes ground beef and steak as subcategories of beef.

Figure 6.1. A Partial Taxonomy of Food.

All of the examples in the current section have expressed abstract relationships between classes, in contrast to the earlier concrete ones about Homer and Marge, which expressed relationships between specific people. Homer and Marge are instances of classes like "married people," "husbands," and "wives." When we assert that a particular instance is a member of class, we are *classifying* the instance. *Classification* is a class inclusion relationship between an instance and a class, rather than between two classes. (We discuss Classification in detail in Chapter 8.)

Homer Simpson → is-a → Husband

This is just the lowest level of the class hierarchy in which Homer is located at the very bottom; he is also a man, a human being, and a living organism (in cartoon land, at least).

instance→ is-member-of → class

Part-whole inclusion or *meronymic inclusion* is a second type of inclusion relationship. It is usually expressed using “is-part-of,” “is-partly,” or with other similar predicate expressions. Winston, Chaffin, and Herrmann identified six distinct types of part-whole relationships. Their meaning subtly differs depending on whether the part is separately identifiable and whether the part is essential to the whole.

- *Component-Object* is the relationship type when the part is a separate component that is arranged or assembled with other components to create a larger resource. In §4.1.1.1, “Resources with Parts,” we used as an example the component-object relationship between an engine and a car:

  The Engine → is-part-of → the Car

  The components of this type of part-whole relationship need not be physical objects; “Germany is part of the European Union” expresses a component-object relationship. What matters is that the component is identifiable on its own as an integral entity and that the components follow some patterned organization or structure when they form the whole. Together the parts form a composition, and the parts collectively form the whole. A car that lacks the engine part will not work.

- *Member-Collection* is the part-whole relationship type where “is-part-of” means “belongs-to,” a weaker kind of association than component-object because there is no assumption that the component has a specific role or function in the whole.

  The Book → is-part-of → the Library

  The members of the collection exist independently of the whole; if the whole ceases to exist the individual resources still exist.

- *Portion-Mass* is the relationship type when all the parts are similar to each other and to the whole, unlike either of the previous types where engines are not tires or cars, and books are not like record albums or libraries.

  The Slice → is-part-of → the Pie

- *Stuff-Object* relationships are most often expressed using “is-partly” or “is-made-of” and are distinguishable from component-object ones because the stuff cannot be separated from the object without altering its identity. The stuff is not a separate ingredient that is used to make the object; it is a constituent of it once it is made.

  Wine → is-partly → Alcohol

- *Place-Area* relationships exist between areas and specific places or locations within them. Like members of collections, places have no particular functional contribution to the whole.

The Everglades → are-part-of → Florida

- *Feature-Activity* is a relationship type in which the components are stages, phases, or sub-activities that take place over time. This relationship is similar to component-object in that the components in the whole are arranged according to a structure or pattern.

Overtime → is-part-of → a Football Game

Topological, *Locative* and *Temporal Inclusion* is a third type of inclusion relationship between a container, area, or temporal duration and what it surrounds or contains. It is most often expressed using “is-in” as the relationship. However, the entity that is contained or surrounded is not a part of the including one, so this is not a part-whole relationship.

The Vatican City → is-in → Italy
The meeting → is-in → the afternoon

6.3.1.2. Attribution

In contrast to inclusion expressions that state relationships between resources, *attribution relationships* assert or assign values to properties for a particular resource. In Chapter 5 we used “attribute” to mean “an indivisible part of a resource description” and treated it as a synonym of “property.” We now need to be more precise and carefully distinguish between the type of the *attribute* and the *value* that it has. For example, the color of an object is an *attribute* of the object, and the *value* of that attribute might be “green.”

Some frameworks for semantic modeling define “attribute” very narrowly, restricting it to expressions with predicates with only one argument to assert properties of a single resource, distinguishing them from relationships between resources or resource types that require two arguments:

Martin the Gecko → is-small
Martin the Gecko → is-green

However, it is always possible to express statements like these in ways that make them into relationships with two arguments:

Martin → has-size → small
Martin → has-skin-color → green

Another somewhat tricky aspect of attribution relationships is that from a semantic perspective, there are often many different ways of expressing equivalent attribute values.

Martin → has-size → 6 inches
Martin → has size → 152 mm

These two statements express the idea that Martin is small. However, many implementations of attribution relationships treat the attribute values literally. Unless we can process these two statements using another relationship that expresses the conversion of inches to mm, the two statements could be interpreted as saying different things about Martin's size.

6.3.1.3. Possession

A third distinct category of semantic relationships is that of possession. *Possession* relationships can seem superficially like part-whole ones:

Bob → has → a car
A car → has → wheels

However, in the second of these relationships "has" is an elliptical form of "has as a part," expressing a part-whole relationship rather that one of possession.

The concept of possession is especially important in institutional organizing systems, where questions of ownership, control, responsibility and transfers of ownership, control, and responsibility can be fundamental parts of the interactions they support. However, possession is a complex notion, inherently connected to societal norms and conventions about property and kinship, making it messier than institutional processes might like.

6.3.2. Properties of Semantic Relationships

Semantic relationships can have numerous special properties that help explain what they mean and especially how they relate to each other. In the following sections, we briefly explain those that are most important in systems for organizing resources and resource descriptions.

6.3.2.1. Symmetry

In most relationships, the order in which the subject and object arguments are expressed is central to the meaning of the relationship. If X has a relationship with Y, it is usually not the case that Y has the same relationship with X. For example, because "is-parent-of" is an *asymmetric* relationship, only the first of these relationships holds:

Homer Simpson → is-parent-of → Bart Simpson (TRUE)
Bart Simpson → is-parent-of → Homer Simpson (NOT TRUE)

In contrast, some relationships are *symmetric* or *bi-directional*, and reversing the order of the arguments of the relationship predicate does not change the meaning. As we noted earlier, these two statements are semantically equivalent because "is-married-to" is symmetric:

Homer Simpson → is-married-to → Marge Simpson
Marge Simpson → is-married-to → Homer Simpson

We can represent the *symmetric* and *bi-directional* nature of these relationships by using a double-headed arrow:

Homer Simpson ⇔ is-married-to ⇔ Marge Simpson

6.3.2.2. Transitivity

Transitivity is another property that can apply to semantic relationships. When a relationship is transitive, if X and Y have a relationship, and Y and Z have the same relationship, then X also has the relationship with Z. Any relationship based on ordering is transitive, which includes numerical, alphabetic, and chronological ones as well as those that imply qualitative or quantitative measurement. Because "is-taller-than" is transitive:

Homer Simpson → is-taller-than → Bart Simpson
Bart Simpson → is-taller-than → Maggie Simpson

implies that:

Homer Simpson → is-taller-than → Maggie Simpson

Inclusion relationships are inherently transitive, because just as "is-taller-than" is an assertion about relative physical size, "is-a-type of" and "is-part-of" are assertions about the relative sizes of abstract classes or categories. An example of transitivity in part-whole or meronymic relationships is: (1) the carburetor is part of the engine, (2) the engine is part of the car, (3) therefore, the carburetor is part of the car.

Transitive relationships enable inferences about class membership or properties and allow organizing systems to be more efficient in how they represent them. Transitivity enables implicit relationships to be made explicit only when they are needed.

6.3.2.3. Equivalence

Any relationship that is both symmetric and transitive is an *equivalence relationship*; "is-equal-to" is obviously an equivalence relationship because if A=B then B=A and if A=B and B=C, then A=C. Other relationships can be equivalent without meaning "exactly equal," as is the relationship of "is-congruent-to" for all triangles.

We often need to assert that a particular class or property has the same meaning as another class or property or that it is substitutable for it. We make this explicit with an equivalence relationship.

Sister (English) ⇔ is-equivalent-to ⇔ Hermana (Spanish)

6.3.2.4. Inverse

For asymmetric relationships, it is often useful to be explicit about the meaning of the relationship when the order of the arguments in the relationship is reversed. The resulting relationship is called the *inverse* or the converse of the first relationship. If an organizing system explicitly represents that:

Is-child-of → is-the-inverse-of → Is-parent-of

We can then conclude that:

Bart Simpson → is-child-of → Homer Simpson

6.3.3. Ontologies

We now have described types and properties of semantic relationships in enough detail to return to the challenge we posed earlier: what information is required to fully understand relationships? This question has been asked and debated for decades, and we will not pretend to answer it to any extent here. However, we can sketch out some of the basic parts of the solution.

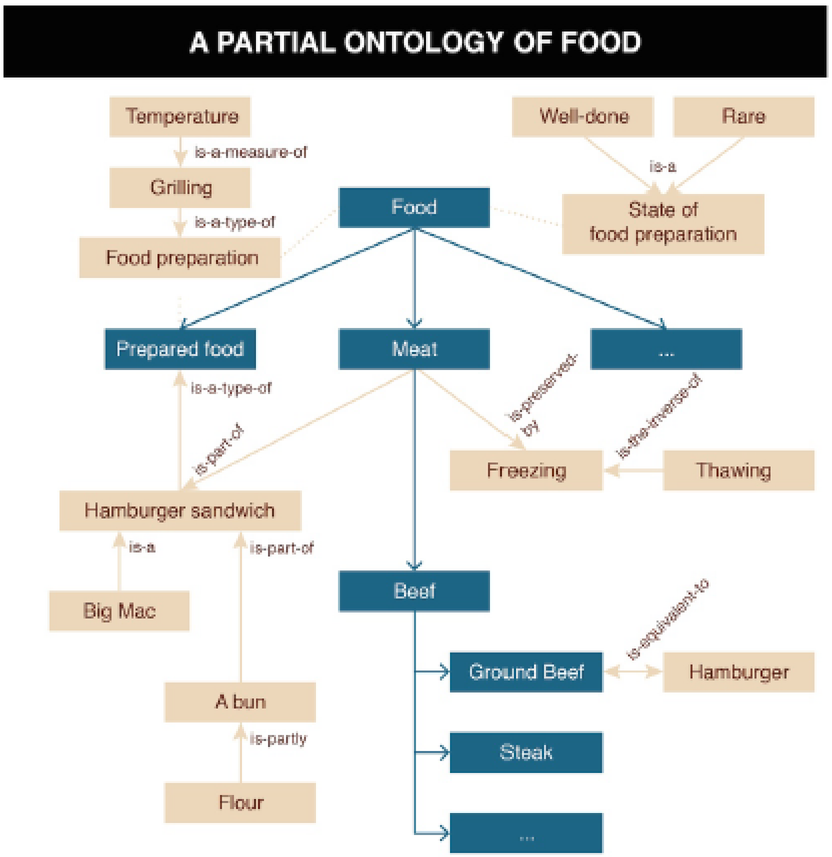
Let us begin by recalling that a *taxonomy* captures a system of class inclusion relationships in some domain. But as we have seen, there are a great many kinds of relationships that are not about class inclusion. All of these other types of relationships represent knowledge about the domain that is potentially needed to understand statements about it and to make sense when more than one domain of resources or activities comes together.

For example, in the food domain whose partial taxonomy appears in Figure 6.2, “A Partial Ontology of Food.”, we can assert relationships about properties of classes and instances, express equivalences about them, and otherwise enhance the representation of the food domain to create a complex network of relationships. In addition, the food domain intersects with food preparation, agriculture, commerce, and many other domains. We also need to express the relationships among these domains to fully understand any of them.

Grilling → is-a-type-of → Food Preparation
Temperature → is-a-measure-of → Grilling
Hamburger → is-equivalent-to → Ground Beef
Hamburger → is-prepared-by → Grilling
Hamburger Sandwich → is-a-type-of → Prepared Food
Rare → is-a → State of Food Preparation
Well-done → is-a → State of Food Preparation
Meat → is-preserved-by → Freezing
Thawing → is-the-inverse-of → Freezing
...

Figure 6.2. A Partial Ontology of Food.

In this simple example, we see that class inclusion relationships form a kind of backbone to which other kinds of relationships attach. We also see that there are many potentially relevant assertions that together represent the knowledge that most people have about food and related domains. A network of relationships like these creates a resource that is called an *ontology*. A visual depiction of the ontology illustrates this idea that it has a taxonomy as its conceptual scaffold. (See Figure 6.2, “A Partial Ontology of Food.”)



A partial ontology of food overlays the taxonomy of food with statements that make assertions about categories, instances, and relationships in the food domain. Example statements might be that “Grilling is a type of food preparation,” that “Meat is preserved by freezing,” and that “Hamburger is equivalent to ground beef.”

Ontologies are essential parts of some organizing systems, especially information-intensive ones where the scope and scale of the resources require an extensive and controlled description vocabulary. (See §5.3, “The Process of Describing Resources”.)

6.4. The Lexical Perspective

The semantic perspective for analyzing relationships is the fundamental one, but it is intrinsically tied to the lexical one because a relationship is always expressed using words in a specific language. For example, we understand the relationships among the concepts or classes of “food,” “meat,” and “beef” by using the words “food,” “meat,” and “beef” to identify progressively smaller classes of edible things in a class hierarchy.

The connection between concept and words is not so simple. In the Simpson family example with which we began this chapter, we noted with “father” and “padre” that languages differ in the words they use to describe particular kinship relationships. Furthermore, we pointed out that cultures differ in which kinship relationships are conceptually distinct, so that languages like Chinese make distinctions about the relative ages of siblings that are not made in English.

This is not to suggest that an English speaker cannot notice the difference between his older and younger sisters, only that this distinction is not lexicalized—captured in a single word—as it is in Chinese. This “missing word” in English from the perspective of Chinese is called a *lexical gap*. Exactly when a lexical gap exists is sometimes tricky, because it depends on how we define “word”—polar bear and sea horse are not lexicalized, but they are a single meaning-bearing unit because we do not decompose and reassemble meaning from the two separate words. These “lexical gaps” differ from language to language. We revisit this issue as “linguistic relativity” in Chapter 7.

Earlier in this book we discussed the naming of resources (§4.4.2, “The Problems of Naming”) and the design of a vocabulary for resource description (§5.3.1.3, “Scope, Scale, and Resource Description”), and we explained how increasing the scope and scale of an organizing system made it essential to be more systematic and precise in assigning names and descriptions. We need to be sure that the terms we use to organize resources capture the similarities and differences between them well enough to support our interactions with them.

For example, if we are organizing cars, buses, bicycles, and sleds, all of which are vehicles, there is an important distinction between vehicles that are motorized and those that are powered by human effort. It might also be useful to distinguish vehicles with wheels from those that lack them. Not making these distinctions leaves an unbalanced or uneven organizing system for describing the semantics of the vehicle domain. However, only the “motorized” concept is lexicalized in English, which is why we needed to invent the “wheeled vehicle” term in the second case.

Simply put, we need to use words effectively in organizing systems. To do that, we need to be careful about how we talk about the relationships among words and how words relate to concepts. There are two different contexts for those relationships.

- First, we need to discuss relationships among the meanings of words. (§6.4.1) and the most commonly used tool for describing them (§6.4.2).
- Second, we need to discuss relationships among the form of words. (§6.4.3, “Relationships among Word Forms”)

6.4.1. Relationships among Word Meanings

There are several different types of relationships of word meanings. Not surprisingly, in most cases they parallel the types of relationships among concepts that we described in §6.3, “The Semantic Perspective”.

6.4.1.1. Hyponymy and Hyperonymy

When words encode the semantic distinctions expressed by class inclusion, the word for the more specific class in this relationship is called the *hyponym,* while the word for the more general class to which it belongs is called the *hypernym*. George Miller suggested a formula for defining a hyponym as its hypernym preceded by adjectives or followed by relative clauses that distinguish it from its *co-hyponyms*, mutually exclusive subtypes of the same hypernym.

hyponym = {adjective+} hypernym {distinguishing clause+}

For example, robin is a hyponym of bird, and could be defined as “a migratory bird that has a clear melodious song and a reddish breast with gray or black upper plumage.” This definition does not mention every property of robins, just some that distinguish them from bluebirds or eagles.

6.4.1.2. Metonymy

Part-whole or meronymic semantic relationships have lexical analogs in *metonomy*, when an entity is described by something that is contained in or otherwise part of it. A country’s capital city or a building where its leader resides is often used as a metonym for the entire government: “The White House announced today...” Similarly, important concentrations of business activity are often metonyms for their entire industries: “Wall Street was bailed out again...”

6.4.1.3. Synonymy

Synonymy is the relationship between words that express the same semantic concept. The strictest definition is that *synonyms* “are words that can replace each other in some class of contexts with insignificant changes of the whole text’s meaning.” This is an extremely hard test to pass, except for acronyms or compound terms like “USA,” “United States,” and “United States of America” that are completely substitutable.

Most synonyms are not *absolute* synonyms and instead are considered *propositional* synonyms. *Propositional* synonyms are not identical in meaning, but they are equivalent enough that substituting one for the other will not change the truth value of the sentence. This weaker test lets us treat word as synonyms even though their meanings subtly differ. For example, if Lisa Simpson can play the violin, then because “violin” and “fiddle” are propositional synonyms, no one would disagree with an assertion that Lisa Simpson can play the fiddle.

6.4.1.4. Polysemy

We introduced the lexical relationship of *polysemy*, when a word has several different meanings or senses, in the context of problems with names (§4.4.2.2, "Homonymy, Polysemy, and False Cognates"). For example, the word "bank" can refer to a: river bank, money bank, bank shots in basketball and billiards, an aircraft maneuver, and other concepts.

6.4.1.5. Antonymy

Antonymy is the lexical relationship between two words that have opposite meanings. *Antonymy* is a very salient lexical relationship. Like synonymy, antonymy is sometimes exact and sometimes more graded.

Contrasting or *binary antonyms* are used in mutually exclusive contexts where one or the other word can be used, but never both. For example, "alive" and "dead" can never be used at the same time to describe the state of some entity because the meaning of one excludes or contradicts the meaning of the other.

Other antonymic relationships between word pairs are less semantically sharp because they can sometimes appear in the same context as a result of the broader semantic scope of one of the words. "Large" and "small," or "old" and "young" generally suggest particular regions on size or age continua, but "how large is it?" or "how old is it?" can be asked about resources that are objectively small or young.

6.4.2. Thesauri

The words that people naturally use when they describe resources reflect their unique experiences and perspectives, and this means that people often use different words for the same resource and the same words for different ones. Guiding people when they select description words from a *controlled vocabulary* is a partial solution to this vocabulary problem that becomes increasingly essential as the scope and scale of the organizing system grow. A *thesaurus* is a reference work that organizes words according to their semantic and lexical relationships. Thesauri are often used by professionals when they describe resources.

Thesauri have been created for many domains and subject areas. Some thesauri are very broad and contain words from many disciplines, like the Library of Congress Subject Headings (LOC-SH) used to classify any published content. Other commonly used thesauri are more focused, like the *Art and Architecture Thesaurus (AAT)* developed by the Getty Trust and the Legislative Indexing Vocabulary developed by the Library of Congress.

We can return to our simple food taxonomy to illustrate how a thesaurus annotates vocabulary terms with lexical and semantic relationships. The class inclusion relationships of hyperonymy and hyponymy are usually encoded using `BT` ("broader term") and `NT` ("narrower term"):

> **Food BT Meat**
> **Beef NT Meat**

The BT and NT relationships in a thesaurus create a hierarchical system of words, but a thesaurus is more than a lexical taxonomy for some domain because it also encodes additional lexical relationships for the most important words.

Because the purpose of a thesaurus is to reduce synonymy, it distinguishes among synonyms or near-synonyms by indicating one of them as a preferred term using UF ("used for"):

Food UF Sustenance, Nourishment

6.4.3. Relationships among Word Forms

The relationships among word meanings are critically important. Whenever we create, combine, or compare resource descriptions we also need to pay attention to relationships between word forms. These relationships begin with the idea that all natural languages create words and word forms from smaller units. The basic building blocks for words are called *morphemes* and can express semantic concepts (when they are called *root words*) or abstract concepts like "pastness" or "plural"). The analysis of the ways by which languages combine morphemes is *morphology*.

Simple examples illustrate this:

"dogs"="dog" (root) + "s" (plural)
"uncertain" = "certain" (root) + "un" (negation)
"denied" = "deny" (root) + "ed" (past tense)

Morphological analysis of a language is heavily used in text processing to create indexes for information retrieval. For example, *stemming* is morphological processing to remove prefixes and suffixes to leave the root form of words. Similarly, simple text processing applications like hyphenation and spelling correction solve word form problems using roots and rules because it is more scalable and robust than solving them using word lists. In addition, because natural languages are generative and create new words all the time, a word list can never be complete; for example, when "flickr" occurs in text, is it a misspelling of "flicker" or the correct spelling of the photo-sharing site?

6.4.3.1. Derivational Morphology

Derivational morphology deals with how words are created by combining morphemes. *Compounding*, putting two "free morphemes" together as in "batman" or "catwoman," is an extremely powerful mechanism. The meaning of some compounds is easy to understand when the first morpheme qualifies or restricts the meaning of the second, as in "birdcage" and "tollbooth." However, many compounds take on new meanings that are not as literally derived from the meaning of their constituents, like "seahorse" and "batman."

Other types of derivations using "bound" morphemes follow more precise rules for combining them with "base" morphemes. The most common types of bound morphemes are prefixes and suffixes, which usually create a word of a different part-of-speech category when they are added. Familiar English prefixes include "a-," "ab-," "anti-," "co-," "de-," "pre-," and "un-." Among the most common English suffixes are “-able,” “-ation,” “-ify,” “ing,” “-ity,” “-ize,” “-ment,” and “-ness.” Compounding and adding prefixes or suffixes are simple mechanisms, but very complex words like “unimaginability” can be formed by using them in combination.

6.4.3.2. Inflectional Morphology

Inflectional mechanisms change the form of a word to represent tense, aspect, agreement, or other grammatical information. Unlike derivation, inflection never changes the part-of-speech of the base morpheme. The *inflectional morphology* of English is relatively simple compared with other languages.

6.5. The Structural Perspective

The *structural perspective* analyzes the association, arrangement, proximity, or connection between resources without primary concern for their meaning or the origin of these relationships. We take a structural perspective when we define a family as “a collection of people” or when we say that a particular family like the Simpsons has five members. Sometimes all we know is that two resources are connected, as when we see a highlighted word or phrase that is pointing from the current web page to another. At other times we might know more about the reasons for the relationships within a set of resources, but we still focus on their structure, essentially merging or blurring all of the reasons for the associations into a single generic notion that the resources are connected.

Travers and Milgram conducted a now-famous study in the 1960s involving the delivery of written messages between people in the Midwestern and Eastern United States. If a person did not know the intended recipient, he was instructed to send the message to someone that he thought might know him. The study demonstrated what Travers and Milgram called the “small world problem,” in which any two arbitrarily selected people were separated by an average of fewer than six links.

It is now common to analyze the number of “degrees of separation” between any pair of resources. For example, a 2011 study using Facebook data computed the average “degree of separation” of any two people in the Facebook world to be 4.74.

Many types of resources have internal structure in addition to their structural relationships with other resources. Of course, we have to remember that we often face arbitrary choices about the abstraction and granularity with which we describe the parts that make up a resource and whether some combination of resource should also be identified as a resource. This is not easy when you are analyzing the structure of a car with its thousands of parts, and it is ever harder with information resources where there are many more ways to define parts and wholes. However, an advantage for information resources is that their internal structural descriptions are usually highly “computable.”

Management science is constantly reevaluating different structures for organizations. Many large businesses are organized similarly near the top, with a board of directors, a chief executive officer, and other executives who manage the vice presidents or directors of various business units. Within and across these business units, however, there are significant variations in how a business can organize its people.

Management strategies are built around the style of organization the business has chosen. These organizational choices reflect the CEO's management philosophy, the industry, regulatory requirements, operating scale, and other factors. Strict hierarchies are a traditional approach, with a tree structure leading from the lowest level worker directly up to the CEO. Other firms have a matrix structure in which an employee can be working on multiple projects, and reporting to a different manager for each one.

6.5.1. Intentional, Implicit, and Explicit Structure

In the discipline of organizing we emphasize "intentional structure" created by people or by computational processes rather than accidental or naturally-occurring structures created by physical and geological processes.

Some organizing principles impose very little structure. For a small collection of resources, co-locating them or arranging them near each other might be sufficient organization. We can impose two- or three-dimensional coordinate systems on this "implicit structure" and explicitly describe the location of a resource as precisely as we want, but we more naturally describe the structure of resource locations in relative terms. In English we have many ways to describe the structural relationship of one resource to another: "in," "on," "under," "behind," "above," "below," "near," "to the right of," "to the left of," "next to," and so on. Sometimes several resources are arranged or appear to be arranged in a sequence or order and we can use positional descriptions of structure: a late 1990s TV show described the planet Earth as the "third rock from the Sun."

We pay most attention to intentional structures that are explicitly represented within and between resources because they embody the design or authoring choices about how much implicit or latent structure will be made explicit. Structures that can be reliably extracted by algorithms become especially important for very large collections of resources whose scope and scale defy structural analysis by people.

6.5.2. Structural Relationships within a Resource

We almost always think of human and other animate resources as unitary entities. Likewise, many physical resources like paintings, sculptures, and manufactured goods have a material integrity that makes us usually consider them as indivisible. For an information resource, however, it is almost always the case that it has or might have had some internal structure or sub-division of its constituent data elements.

In fact, since all computer files are merely encodings of bits, bytes, characters and strings, all digital resources exhibit some internal structure, even if that structure is only discernible by software agents. Fortunately, the once inscrutable internal formats of word processing files are now much more interpretable after they were replaced by XML in the last decade.

When an author writes a document, he or she gives it some internal organization with its title, section headings, typographic conventions, page numbers, and other mechanisms that identify its parts and their significance or relationship to each other. The lowest level of this structural hierarchy, usually the paragraph, contains the text content of the document. Sometimes the author finds it useful to identify types of content like glossary terms or cross-references within the paragraph text. Document models that mix structural description with content “nuggets” in the text are said to contain *mixed content*.

In data-intensive or transactional domains, document instances tend to be homogeneous because they are produced by or for automated processes, and their information components will appear predictably in the same structural relationships with each other. These structures typically form a hierarchy expressed in an XML schema or word processing style template. XML documents describe their component parts using content-oriented elements like `<ITEM>`, `<NAME>`, and `<ADDRESS>`, that are themselves often aggregate structures or containers for more granular elements. The structures of resources maintained in databases are typically less hierarchical, but the structures are precisely captured in database schemas.

In more qualitative, less information-intensive and more experience-intensive domains, document instances become more heterogeneous because they are produced by and for people. The information conveyed in the documents is conceptual or thematic rather than transactional, and the structural relationships between document parts are much weaker. Instead of precise structure and content rules, there is usually just a shallow hierarchy marked up with Word processing or HTML tags like `<HEAD>`, `<H1>`, `<H2>`, and `<LIST>`.

The internal structural hierarchy in a document is often extracted and made into a separate and familiar description resource called the “table of contents” to support finding and navigation interactions with the primary resource. In a printed media context, any given content resource is likely only to be presented once, and its page number is provided in the table of contents to allow the reader to locate the chapter, section or appendix in question. In a hypertext media context, the same resource may be a chapter in one book while being an appendix in another.

6.5.3. Structural Relationships between Resources

Many types of resources have “structural relationships” that interconnect them. Web pages are almost always linked to other pages. Sometimes the links among a set of pages remain mostly within those pages, as they are in an e-commerce catalog site. More often, however, links connect to pages in other sites, creating a link network that cuts across and obscures the boundaries between sites.

The links between documents can be analyzed to infer connections between the authors of the documents. Using the pattern of links between documents to understand the structure of knowledge and of the intellectual community that creates it is not a new idea, but it has been energized as more of the information we exchange with other people is on the web or otherwise in digital formats. An important function in Google’s search engine is the *page rank* algorithm that calculates the relevance of a page in part using the number of links that point to it while giving greater weight to pages that are themselves linked to often.

Web-based social networks enable people to express their connections with other people directly, bypassing the need to infer the connections from links in documents or other communications.

6.5.3.1. Hypertext Links

The concept of read-only or follow-only structures that connect one document to another is usually attributed to Vannevar Bush in his seminal 1945 essay titled "As We May Think." Bush called it *associative indexing*, defined as "a provision whereby any item may be caused at will to select immediately and automatically another." The "item" connected in this way was for Bush most often a book or a scientific article. However, the anchor and destination of a hypertext link can be a resource of any granularity, ranging from a single point or character, a paragraph, a document, or any part of the resource to which the ends of link are connected. The anchor and destination of a web link are its structural specification, but we often need to consider links from other perspectives. (See the sidebar, Perspectives on Hypertext Links).

Theodor Holm Nelson, in a book intriguingly titled *Literary Machines,* renamed associative indexing as *hypertext* decades later, expanding the idea to make it a writing style as well as a reading style. Nelson urged writers to use hypertext to create non-sequential narratives that gave choices to readers, using a novel technique for which he coined the term *transclusion.*

Hypertext links are now familiar structural mechanisms in information applications because of the World Wide Web, proposed in 1989 by Tim Berners-Lee and Robert Cailliau. They invented the methods for encoding and following hypertext links using the now popular HyperText Markup Language (HTML). The resources connected by HTML's hypertext links are not limited to text or documents. Selecting a hypertext link can invoke a connected resource that might be a picture, video, or interactive application.

Perspectives on Hypertext Links

- A lexical perspective on hypertext links concerns the words that are used to signal the presence of a link or to encode its type. In web contexts, the words in which a structural link is embedded are called the *anchor text*. We can also analyze how different word signals in texts indicate relationships between texts or parts of them, like the subtle differences in polarity among "see," "see also," and "but see" as citation signals.
- Many hypertext links in web pages are purely structural because they lack explicit representation of the reason for the relationship. When it is evident, this semantic property of the link is called the *link type*.
- An architectural perspective on links considers whether links are *one-way* or *bi-directional*. When a bi-directional link is created between an anchor and a destination, it is as though a one-way link that can be followed in the opposite direction is automatically created. Two one-way links serve the same purpose, but the return link is not automatically established when the first one is created. A second architectural consideration is whether to employ *binary links*, connecting one anchor to one destination, or *n-ary links*, connecting one anchor to multiple types of destinations.
- A "front-end" or "surface" implementation perspective on hypertext links concerns how the presence of the link is indicated in a user interface; this is called the "link marker";

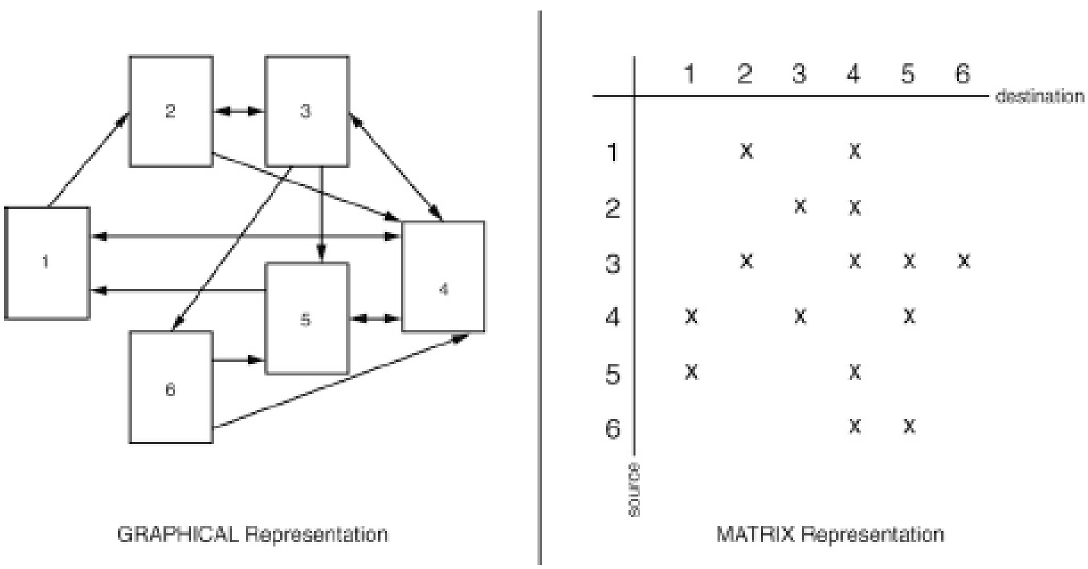
underlining or coloring of clickable text are conventional markers for web links. A "back-end" implementation issue is whether links are contained or embedded in the resources they link or whether they are stored separately in a *link base*.

6.5.3.2. Analyzing Link Structures

We can portray a set of links between resources graphically as a pattern of boxes and links. Because a link connection from one resource to another need not imply a link in the opposite direction, we distinguish one-way links from explicitly bi-directional ones.

A graphical representation of link structure is shown on the left panel of figure Figure 6.3, "Representing Link Structures.". For a small network of links, a diagram like this one makes it easy to see that some resources have more incoming or outgoing links than other resources. However, for most purposes we leave the analysis of link structures to computer programs, and there it is much better to represent the link structures more abstractly in matrix form. In this matrix the resource identifiers on the row and column heads represent the source and destination of the link. This is a full matrix because not all of the links are symmetric; a link from resource 1 to resource 2 does not imply one from 2 to 1.

Figure 6.3. Representing Link Structures.



A matrix representation of the same link structure is shown on the right panel of Figure 6.3, "Representing Link Structures.". This representation models the network as a directed graph in which the resources are the vertices and the relationships are the edges that connect them. We now can apply graph algorithms to determine many useful properties. A very important property is *reachability*, the "can you get there from here" property. Other useful properties include the average number of incoming or outgoing links, the average distance between any two resources, and the shortest path between them.

6.5.3.3. Bibliometrics, Shepardizing, Altmetrics, and Social Network Analysis

Information scientists began studying the structure of scientific citation, now called *bibliometrics*, nearly a century ago to identify influential scientists and publications. This analysis of the flow of ideas through publications can identify “invisible colleges” of scientists who rely on each other’s research, and recognize the emergence of new scientific disciplines or research areas. Universities use bibliometrics to evaluate professors for promotion and tenure, and libraries use it to select resources for their collections.

The expression of citation relationships between documents is especially nuanced in legal contexts, where the use of legal cases as precedents makes it essential to distinguish precisely where a new ruling lies on the relational continuum between “Following” and “Overruling” with respect to a case it cites. The analysis of legal citations to determine whether a cited case is still good law is called *Shepardizing* because lists of cases annotated in this way were first published in the late 1800s by Frank Shepard, a salesman for a legal publishing company.

The links pointing to a web page might be thought of as citations to it, so it is tempting to make the analogy to consider Shepardizing the web. But unlike legal rulings, web pages aren’t always persistent, and only courts have the authority to determine the value of cited cases as precedents, so Shepard-like metrics for web pages would be tricky to calculate and unreliable.

Nevertheless, the web’s importance as a publishing and communication medium is undeniable, and many scholars, especially younger ones, now contribute to their fields by blogging, Tweeting, leaving comments on online publications, writing Wikipedia articles, giving MOOC lectures, and uploading papers, code, and datasets to open access repositories. Because the traditional bibliometrics pay no attention to this body of work, alternative metrics or “altmetrics” have been proposed to count these new venues for scholarly influence.

Facebook’s valuation is based on its ability to exploit the structure of a person’s social network to personalize advertisements for people and “friends” to whom they are connected. Many computer science researchers are working to determine the important characteristics of people and relationships that best identify the people whose activities or messages influence others to spend money.

6.6. The Architectural Perspective

The architectural perspective emphasizes the number and abstraction level of the components of a relationship, which together characterize the complexity of the relationship. We will briefly consider three architectural issues: degree (or arity), cardinality, and directionality.

These architectural concepts come from data modeling and they enable relationships to be described precisely and abstractly, which is essential for maintaining an organizing system that implements relationships among resources. Organizing systems built without clear architectural foundations cannot easily scale up in size and scope to handle new requirements.

6.6.1. Degree

The *degree* or *arity* of a relationship is the number of entity types or categories of resources in the relationship. This is usually, though not always, the same as the number of arguments in the relationship expression.

Homer Simpson (husband) ⇔ is-married-to ⇔ Marge Simpson (wife)

is a relationship of degree 2, a **binary** relationship between two entity types, because the "is-married-to" relationship as we first defined it requires one of the arguments to be of entity type "husband" and one of them to be of type "wife."

Now suppose we change the definition of marriage to allow the two participants in a marriage to be any instance of the entity type "person." The relationship expression looks the same, but its degree is now *unary* because only one entity type is needed to instantiate the two arguments:

Homer Simpson (person) ⇔ is-married-to ⇔ Marge Simpson (person)

6.6.2. Cardinality

The *cardinality* of a relationship is the number of instances that can be associated with each entity type in a relationship. At first glance this might seem to be degree by another name, but it is not.

Cardinality is easiest to explain for binary relationships. If we return to Homer and Marge, the binary relationship that expresses that they are a married husband and wife is a *one-to-one* relationship because a husband can only have one wife and a wife can only have one husband (at a time, in monogamous societies like the one in which the Simpsons live).

In contrast, the "is-parent-of" relationship is one-to-many, because the meaning of being a parent makes it correct to say that:

Homer Simpson → is-parent-of → Bart AND Lisa AND Maggie

We can transform this more complex relationship architecture to a set of simpler ones by restricting expressions about being a parent to the one-to-one cardinality.

Homer Simpson → is-parent-of → Bart
Homer Simpson → is-parent-of → Lisa
Homer Simpson → is-parent-of → Maggie

The one-to-many expression brings all three of Homer's children together as arguments in the same relational expression, making it more obvious that they share the same relationship than in the set of separate and redundant one-to-one expressions.

6.6.3. Directionality

The *directionality* of a relationship defines the order in which the arguments of the relationship are connected. A *one-way* or *uni-directional* relationship can be followed in only one direction, whereas a *bi-directional* one can be followed in both directions.

All symmetric relationships are bi-directional, but not all bi-directional relationships are symmetric. (See §6.3.2.1, "Symmetry".) A relationship between a manager and an employee that he manages is "employs," a different meaning than the "is-employed-by" relationship in the opposite direction. As in this example, the relationship is often lexicalized in only one direction.

6.7 The Implementation Perspective

Finally, the implementation perspective on relationships considers how a relationship is realized or encoded in a technology context. The implementation perspective contrasts strongly with the conceptual, structural, and architectural perspectives, which emphasize the meaning and abstract structure of relationships. The implementation perspective is a superset of the lexical perspective, because the choice of the language in which to express a relationship is an implementation decision. However, most people think of implementation as all of the decisions about technological form rather than just about the choice of words.

In this book we focus on the fundamental issues and challenges that apply to all organizing systems, and not just on information-intensive ones that rely extensively on technology. Even with this reduced scope, there are some critical implementation concerns about the notation, syntax, and deployment of the relationships and other descriptions about resources. We briefly introduce some of these issues here and then discuss them in detail in Chapter 9, The Forms of Resource Descriptions.

6.7.1 Choice of Implementation

The choice of implementation determines how easy it is to understand and process a set of relationships. For example, the second sentence of this chapter is a natural language implementation of a set of relationships in the Simpson family:

> **The Simpson family includes a man named Homer and a woman named Marge, the married parents of three sibling children, a boy named Bart and two girls, Lisa and Maggie.**

A subject-predicate-object syntax makes the relationships more explicit:

Example 6.1. Subject-predicate syntax

Homer Simpson → is-married-to → Marge Simpson
Homer Simpson → is-parent-of → Bart
Homer Simpson → is-parent-of → Lisa
Homer Simpson → is-parent-of → Maggie
Marge Simpson → is-married-to → Homer Simpson
Marge Simpson → is-parent-of → Bart
Marge Simpson → is-parent-of → Lisa
Marge Simpson → is-parent-of → Maggie
Bart Simpson → is-a → Boy
Lisa Simpson → is-a → Girl
Maggie Simpson → is-a → Girl

In the following example of a potential XML implementation syntax, we emphasize class inclusion relationships by using elements as containers, and the relationships among the members of the family are expressed explicitly through references, using XML's ID and IDREF attribute types:

Example 6.2. An XML implementation syntax

```
<Family name="Simpson">
    <Parents children="Bart Lisa Maggie">
        <Father name="Homer" spouse="Marge" />
        <Mother name="Marge" spouse="Homer" />
    </Parents>
    <Children parents="Homer Marge" >
        <Boy name="Bart" siblings="Lisa Maggie" />
        <Girl name="Lisa" siblings="Bart Maggie" />
        <Girl name="Maggie" siblings="Bart Lisa" />
    </Children>
</Family>
```

None of the models we have presented so far in this chapter represents the complexities of modern families that involve multiple marriages and children from more than one marriage, but they are sufficient for our limited demonstration purposes.

6.7.2 Syntax and Grammar

The syntax and grammar of a language consists of the rules that determine which combinations of its words are allowed and are thus grammatical or well-formed. Natural languages have substantial similarities by having nouns, verbs, adjectives and other parts of speech, but they differ greatly in how they arrange them to create sentences. Conformance to the rules for arranging these parts makes a sentence syntactically compliant but does not mean that an expression is semantically comprehensible; the classic example is Chomsky's anomalous sentence:

Colorless green ideas sleep furiously

Any meaning this sentence has is odd, difficult to visualize, and outside of readily accessible experience, but anyone who knows the English language can recognize that it follows its syntactic rules, as opposed to this sentence, which breaks them and seems completely meaningless:

Ideas colorless sleep furiously green

6.7.3 Requirements for Implementation Syntax

The most basic requirement for implementation syntax is that it can represent all the expressions that it needs to express. For the examples in this chapter we have used an informal combination of English words and symbols (arrows and parentheses) that you could understand easily, but simple language is incapable of expressing most of what we readily say in English. But this benefit of natural language only accrues to people, and the more restrictive and formal syntax is easier to understand for computers.

A second consideration is that the implementation can be understood and used by its intended users. We can usually express a relationship in different languages while preserving its meaning, just as we can usually implement the same computing functionality in different programming languages. From a semantic perspective these three expressions are equivalent:

My name is Homer Simpson
Mon nom est Homer Simpson
Mein name ist Homer Simpson

However, whether these expressions are equivalent for someone reading them depends on which languages they understand.

An analogous situation occurs with the implementation of web pages. HTML was invented as a language for encoding how web pages look in a browser, and most of the tags in HTML represent the simple structure of an analogous print document. Representing paragraphs, list items and numbered headings with <P> and <LI> and <Hn> makes using HTML so easy that school children can create web pages. However, the "web for eyes" implemented using HTML is of less efficient or practical for computers that want to treat content as product catalogs, orders, invoices, payments, and other business transactions and information that can be analyzed and processed. This "web for computers" is best implemented using domain-specific vocabularies in XML.