against a blue sky might already strike “bird” and “sky” from the set of acceptable words, leaving users to suggest words such as “flying” and “cloudless.” Van Ahn also invented the reCAPTCHA technique that presents images of text from old books being digitized, which improves the accuracy of the digitization while verifying that the user of a web site is a person and not a robot program.[4]

However, if short text descriptions or low-level image properties are the only features available to train an image, otherwise irrelevant variations in the position, orientation, or illumination of objects in images will make it very difficult to distinguish objects that look similar, like a white wolf and the wolf-like white dog called a Samoyed. This problem can be addressed by using deep neural networks, which exploit the idea that low-level image features can be combined into many layers of higher-level ones; edges combine to form motifs or patterns, patterns combine to form parts of familiar objects, and parts combine to form complete objects. This hierarchical composition enables the highest-level representations to become insensitive to the lower-level variations that plague the other approaches.

In 2012, when deep learning techniques were applied to a dataset of about a million images that contained a thousand different object categories, they reduced the error rate by half. This spectacular breakthrough, and the fact that the deep learning techniques that derive layers of features from the input data are completely general, rapidly caused deep learning to be applied to many other domains with high-dimensional data. Facebook uses deep learning to identify people in photos, Google uses it for speech recognition and language translation, and rapid captioning for images and video are on the horizon. Wearable computers might use it to layer useful

4. (von Ahn and Dabbish 2008).

information onto people's views of the world, creating real-time augmented reality.[5]

Describing Music

Describing music presents challenges quite different from those involved in describing texts or images. Poems and paintings are tangible things that we can look at and contemplate, while the aural nature of music means that it is a fleeting phenomenon that can only be experienced in the performative moment. Even musical scores and recordings, while as much tangible things as paintings and poems, are merely containers that hold the potential for musical experience and not the music itself. Most contemporary popular music is in the form of songs, in which texts are set to a melody and supported by instrumental harmonies. If we want to categorize or describe such music by its lyrical content, we can still

5. The key idea that made deep learning possible is the use of "backpropagation" to adjust the weights on features by working backwards from the output (the object classification produced by the network) all the way back to the input. Mathematically-sophisticated readers can find a concise explanation and history of deep learning in (LeCun, Bengio, and Hinton 2015). LeCun and Hinton were part of research teams that independently invented backpropagation in the mid 1980s. Today, LeCun heads Facebook's research group on artificial intelligence, and Hinton has a similar role at Google.

rely on methods for describing texts. But if we want to describe the music itself, we need to take a somewhat different approach.

Several people and companies working in multimedia have explored different processes for how songs are described. On the heavily technological side, software applications such as Shazam and Midomi can create a content-based *audio fingerprint* from a snippet of music. *Audio fingerprinting* renders a digital description of a piece of music, which a computer can then interpret and compare to other digital descriptions in a library.[6]

On the face of it, contemporary music streaming services represent the apex of music classification and description. Pandora, for example, employs trained musicologists to listen to the music and then categorize the genres and musical materials according to a highly controlled musical vocabulary. The resulting algorithm, the "Music Genome," can essentially learn to define a listener's musical tastes by means of this musical tagging, and can then use that information to suggest other music with similar characteristics.[7]

But musicians have been thinking about how to describe music for centuries, and while the Music Genome certainly brims with complexity, it pales in comparison to the sophistication of the much older "pen-and-paper" methods from which it derives. Ethnomusicology (loosely defined as the study of global musical practices in their social contexts) has arguably made greater strides towards comprehensive descriptions of musical resources than any other field of musicological study. Since the late 19th century, ethnomusicologists have created complex methods of notation and

6. (Cano et al. 2005).

7. (Walker 2009).

stylistic taxonomies to capture and categorize the music of both Western and non-Western cultures.

Hungarian composer and scholar Béla Bartók collected and transcribed thousands of Eastern European folk songs to which he applied a complex classification system to group them into "families" derived from melodic archetypes. More recently, American ethnomusicologist Alan Lomax's Cantometrics project classified songs collected from around the word according to 37 style factors in an effort to create a common controlled vocabulary that would facilitate cross-cultural comparison and analysis.[8]

8. Bartók's method for transcribing and categorizing each tune into families was as follows:

    1. All tunes end on the note "g" for ease of comparison;
    2. Tunes are divided and categorized according to the number of lines;
    3. Tunes are classified according to the placement of the final note of various tune lines with the final note indicated by figures;
    4. Sub-groups are categorized according to the number of syllables to each tune line;
    5. Tunes are categorized according to their melodic compass with the lowest and highest note of each tune labeled.

It is not difficult to see the parallels of this method with the Pandora algorithm, as well as the greater level of descriptive detail afforded by Bartók's method. See (Bartók 1981).

Every folk song collection contains several examples of the same song performed at various times by the same singer or by many different singers. Often these different songs (or "variants") are so drastically different that we begin to ask the question: "At what point does a variant become a completely new piece of music?" Up until the beginning of the twentieth-century, many scholars believed that variants were simply poor performances by folk singers who were attempting to recreate a pristine, archetypical version of the song.

It wasn't until Australian collector Percy Grainger suggested that variants represented a vital and dynamic performance practice among folk singers that the idea of variants as flawed archetypes gave way to one in which all performances were unique entities unto themselves that possess what Wittgenstein would eventually refer to as "family resemblances" with one another. (Grainger 1908)

More recently, Newsweek magazine compiled a list of 60 different versions of Leonard Cohen's "Hallelujah," many of which differ so drastically from Cohen's original as to seem to be completely different songs with only the

A DJ Describes and Organizes Music

Casual music fans might describe their music using the names of the songs or performers and might organize it according to genres like "Pop," "Rock," or "Classical." A professional DJ, however, emphasizes different properties, especially the beats per minute of the music.

This annotated photo shows a portion of the music

most rudimentary family resemblances. Does "Hallelujah" as a "work" even exist anymore? Or is it simply an idea, a potential for music that only exists during each varied performance? (http://www.newsweek.com/60-versions-leonard-cohens-hallelujah-ranked-303580).

collection of noted DJ "Kid Kameleon" (http://kidkameleon.com/).

(Photo and annotation by Matt Earp. Used with permission.)

On a more granular level, musicians are endlessly innovative in finding ways to categorize, describe, and analyze not simply large-scale musical genres, but the notes themselves. In the accompanying photo showing the record collection of professional DJ "Kid Kameleon," we see that the records are arranged not simply by genre, but also by beats-per-minute (BPM). For Kid Kameleon, these records represent the resources of his musical creative process, and arranging them by BPM allows him to pull exactly the correct musical material he needs to keep the music flowing during a performance. His classification system is therefore a taxonomy that moves from the broad strokes of genre down to the fine grains of specific arrangements of notes and rhythms. This photo is not simply a picture of a record collection: it is a visual representation of an artist's creative process.[9]

9. This method of organizing musical resources for ready access (physically and cognitively) is one that has both an illustrious past and a fascinating future. Musicologist Robert Gjerdingen has studied the way in which composers in 18th century Naples learned their art by studying an organized system of musical schemata that could be expanded, strung together, and varied to create an endless series of pleasing compositions in the galant

Describing Video

Video is yet another resource domain where work to create resource descriptions to make search more effective is ongoing. Video analytics techniques can segment a video into shorter clips described according to their color, direction of motion, size of objects, and other characteristics. Identifying anomalous events and faces of people in video has obvious applications in security and surveillance.[10] Identifying specific content details about a video

style of the period (Gjerdingen 2007). A current approach to this same idea can be found in the work of composer David Cope, whose Experiments in Musical Intelligence software ("Emmy" and the next-generation "Emily Howell") can analyze existing music, break its musical resources down into identifiable schema, and then recombine those schema to create a musical output in the style of the original musical input (Cope 2001). Emmy can recombine these elements in millions of different ways to produce compelling, convincing, and somewhat unnerving works in the style of any composer whose music has been fed to her. Different though they may all seem, 18th century Neapolitans, Kid Kameleon, and Emmy all represent a creative process dependent on the input, description, organization, recombination, and output of musical resources.

10. (Regazzoni et al. 2010) introduce a special issue in IEEE Signal Processing on visual analytics.

currently takes a significant amount of human intervention, though it is possible that image signature-matching algorithms will take over in the future because they would enable automated ad placement in videos and television.[11]

11. One organization that sees a future in assembling better descriptions of video content is the United States' National Football League (NFL), whose vast library of clips can not only be used to gather plays for highlight reels and specials but can also be monetized by pointing out when key advertisers' products appear on film. Currently, labeling the video requires a person to watch the scenes and tag elements of each frame, but once those tags have been created and sequenced along with the video, they can be more easily searched in computerized, automated ways (Buhrmester 2007).

35. Key Points in Chapter Five

- How are resource descriptions used in information retrieval?

  Information retrieval is characterized as comparing a description of a user's needs with descriptions of the resources that might satisfy them. Different property descriptions determine the comparison algorithms and the way in which relevance or similarity of descriptions is determined.

  (See "Naming {and, or, vs.} Describing")

- What are some of the different names given to the terms used in resource descriptions?

  In different contexts, the terms in resource descriptions are called keywords, index terms, attributes, attribute values, elements, data elements, data values, or "the vocabulary," labels, or tags.

  (See ""Description" as an Inclusive Term")

- What is a descriptor?

  In the library science context of *bibliographic description*, a *descriptor* is one of the terms in a carefully designed language that can be assigned to a resource to designate its properties, characteristics, or meaning, or its relationships with other resources.

  (See ""Description" as an Inclusive Term")

- What is a bibliographic record?

  A bibliographic description of an information resource is most

commonly realized as a structured record in a standard format that describes a specific resource.

(See "Bibliographic Descriptions")

- What is the relationship between metadata and bibliographic description?

  Metadata is structured description for information resources of any kind, which makes it a superset of bibliographic description.

  (See "Metadata")

- What is a database schema?

  A relational database schema is designed to restrict resource descriptions to be simple and completely regular sets of attribute-value pairs.

  (See "Metadata")

- What is RDF?

  The Resource Description Framework(RDF) is a language for making computer-processable statements about web resources that is the foundation for the vision of the Semantic Web.

  (See "Resource Description Framework (RDF)")

- What is an aggregated information object?

  An aggregation is a set of information objects that, when considered together, compose another named information object.

  (See "Resource Description Framework (RDF)")

- What is the dominant historical framework for resource description? What is an alternative?

The dominant historical view treats resource descriptions as a package of statements, an alternate framework focuses on each individual description or assertion about a single resource.

(See "Frameworks for Resource Description")

- What are the key principles guiding design of a description vocabulary?

  Design of the description vocabulary should focus on the user of the descriptions. Svenonius proposes five principles for a description vocabulary: user convenience, representation, sufficiency and necessity, standardization, and integration.

  (See "The Process of Describing Resources")

- What is the disciplined process for describing resources?

  The process of describing resources involves several interdependent and iterative steps, including determining scope, focus and purposes, identifying resource properties, designing the description vocabulary, designing the description form and implementation, and creating and evaluating the descriptions.

  (See "The Process of Describing Resources" and Figure: The Process of Describing Resources.)

- Why are schemas or models important in resource description?

  A collection of resource descriptions is vastly more useful when every resource is described using common description elements or terms that apply to every resource; this specification is most often called a schema or model.

  (See "Abstraction in Resource Description")

- What are XML schemas used for?

XML schemas are often used to define web forms that capture resource instances, and are also used to describe the interfaces to web services and other computational resources.

(See "Abstraction in Resource Description")

- What is the value of standardization of resource description?

  When the task of resource description is standardized, the work can be distributed among many describers whose results are shared. This is the principle on which centralized bibliographic description has been based for a century.

  (See "Scope, Scale, and Resource Description")

- How does resource description support selection?

  Resource description can facilitate the discovery of resources, specify their capabilities and compatibility, authenticate them, and indicate their appraised value.

  (See "Resource Description to Support Selection")

- What are the four generic purposes for resource descriptions?

  The Functional Requirements for Bibliographic Records(FRBR) presents four purposes that apply generically: Finding, Identifying, Selecting, and Obtaining resources.

  (See "Resource Description to Support Interactions")

- What kinds of resource descriptions are most important in interactions with digital resources?

  The variety and functions of the interactions with digital resources depends on the richness of their structural, semantic, and format description.

  (See "Resource Description to Support Interactions")

- What's the relationship between sensemaking and organizing?

  Sensemaking is the foundation of organizing, as it is the basic human activity of making sense of the world. Sensemaking encompasses the range of organizing activities from the very informal and personal to systematic scientific processes.

  (See "Resource Description for Sensemaking and Science")

- What is the value of having multiple descriptions for the sane resource?

  Any particular resource might need many resource descriptions, all of which relate to different properties, depending on the interactions that need to be supported and the context in which they take place.

  (See "Identifying Properties")

- What are the two important dimensions for understanding resource properties?

  Two important dimensions for understanding and contrasting resource properties are whether the properties are intrinsically or extrinsically associated with the resource, and whether the properties are static or dynamic.

  (See "Identifying Properties")

- Why are some resource properties called "latent" ones?

  Recent advances in computing technology and *data science* techniques are making it possible to discover or create resource properties that are called "latent" because they are inferred rather than observed.

  (See the sidebar, Latent Feature Creation and Netflix Recommendations)

- What is a controlled vocabulary?

  A *controlled vocabulary* is a fixed or closed set of description terms in some domain with precise definitions that is used instead of the vocabulary that people would otherwise use. A controlled vocabulary reduces synonymy and homonymy.

- Who should create resource descriptions?

  Professionally created resource descriptions, author or user created descriptions, and computational or automated descriptions each have strengths and limitations that impose tradeoffs.

  (See "Creating Resource Descriptions")

- What are common criteria for evaluating resource descriptions?

  The most commonly used criteria for evaluating resource descriptions are accuracy, completeness, and consistency. Other typical criteria are timeliness, interoperability, and usability.

  (See "Evaluating Resource Descriptions")

- In what ways can computation support the description of non-text resources?

  Computational methods can describe and classify images, identify and classify sounds and music, and identify anomalous events in video.

  (See "Describing Non-text Resources")

PART VI

DESCRIBING RELATIONSHIPS AND STRUCTURES

Robert J. Glushko

Matthew Mayernik

Alberto Pepe

Murray Maloney

36. Introduction (VI)

We consider a family to be 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*.[1]

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

1. The Simpsons TV show began in 1989 and is now the longest running scripted TV show ever. The official website is www.thesimpsons.com. The show is dubbed into French, Italian and Spanish for viewers in Quebec, France, Italy, Latin America and Spain. The Simpson's Movie has been dubbed into Mandarin Chinese and Cantonese. For more information about Mandarin kinship terms see http://mandarin.about.com/od/vocabularylists/tp/family.htm. (Yes, we know that Bart actually calls his father by his first name.)

understanding *relationships* in organizing systems.[2] 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.

> Simpson Family Trees
>
> Because the Simpson family is known throughout the world, the Simpson family tree is often used to teach kinship terms to language learners.
>
> - A website for teaching Spanish
> - A website for teaching French
> - A website for teaching German

In a classic book called *Data and Reality*, William Kent defines a

2. Kinship can be studied from both anthropological and biological perspectives, which differ to the degree to which they emphasize social relationships and genetic ones. Kinship has been systematically studied since the nineteenth century: (Morgan 1871/ 1997) developed a system of kinship classification still taught today. A detailed interactive web tutorial developed by Brian Schwimer can be found at http://umanitoba.ca/faculties/arts/anthropology/kintitle.html.

relationship as an association among several things, with that association having a particular significance.[3] "The things being associated," the components of the relationship, are people in kinship relationships but more generally can be any type of resource (*Resources in Organizing Systems*), when we relate one resource instance to another. When we describe a resource (*Resource Description and Metadata*), 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 (*Categorization: Describing Resource Classes and Types*). 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. We apply these perspectives and vocabulary

3. Kent's *Data and Reality* was first published in 1978 with a second edition in 1998. Kent was a well-known and well-liked researcher in data modeling at IBM, and his book became a cult classic. In 2012, seven years after Kent's death, a third edition (Kent and Hoberman 2012) came out, slightly revised and annotated but containing essentially the same content as the book from 34 years earlier because its key issues about data modeling are timeless.

to analyze the most important types of relationships in organizing systems.

37. 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. Kent points out that when we describe a relationship we sometimes use whole phrases, such as "is-employed-by," if our language does not contain a single word that expresses the meaning of the relationship.

Navigating This Chapter

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

Semantic perspective

The semantic perspective is the most essential one; it characterizes the meaning of the association between resources. ("The Semantic Perspective")

Lexical perspective

The lexical perspective focuses on how the conceptual description of a relationship is expressed using words in a specific language. ("The Lexical Perspective")

Structural perspective

The structural perspective analyzes the actual patterns of association, arrangement, proximity, or connection between resources. (“The Structural Perspective”)

Architectural perspective

The architectural perspective emphasizes the number and abstraction level of the components of a relationship, which together characterize its complexity. (“The Architectural Perspective”)

Implementation perspective

The implementation perspective considers 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. (“The Implementation Perspective ”)

38. 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. This means that "Homer is married to Marge" is a semantic assertion, but "Homer is standing next to Marge" is not.[1]

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.[2]

1. "Semantic" is usually defined as "relating to meaning or language" and that does not seem helpful here.

2. For decades important and vexing questions have been raised about the specificity of these predicate-argument associations and how or when the semantic constraints they embody combine with syntactic and contextual constraints during the process of comprehending language. Consider how "While in the operating room, the surgeon used a knife to cut the ____" generates a

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 that is 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

different expectancy from the same predicate and agent in "While at the fancy restaurant, the surgeon used a knife to cut the _____." See (Elman 2009).

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.[3]

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.[4] Furthermore, for a computer to understand the

3. This book is not the place for the debate over the definition of marriage. We are not bigots; we just do not need this discussion here. If these definitions upset you here, you will feel better in "Degree".
4. Typically, when people use language they operate on 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.

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:

assumption that everyone shares their model of the world, providing the common ground that enables them to communicate. As we saw in *Resources in Organizing Systems* and *Resource Description and Metadata*, (because of the *vocabulary problem* and different purposes for using resources and language) this assumption is often wrong, This paves the way for serious misunderstandings, since what is assumed to be shared knowledge may not really be shared or understood the same way.

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."[5]

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.

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.

5. See (Chaffin and Herrmann 1984), (Storey 1993).

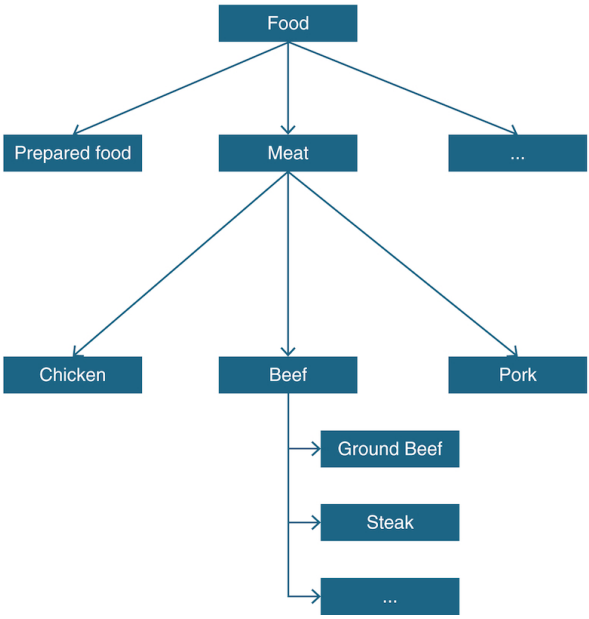
Meat → is-a → Food

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: A Partial Taxonomy of Food.

A PARTIAL TAXONOMY OF FOOD



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.

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 "Principles for Creating Categories", "Principles for Creating Categories."

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 make an assertion 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 *Classification: Assigning Resources to Categories*.)

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).[6] You might now remember the bibliographic class inclusion hierarchy we discussed in "Identity and Bibliographic Resources"; a specific physical *item* like your dog-eared copy of *Macbeth* is also a particular *manifestation* in some format or genre, and this *expression* is one of many for the abstract *work*.

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,

6. Which of these classifications is most relevant depends on the context. In addition, there might be other Homer Simpsons who are not cartoon characters or who are not married, so we might have to disambiguate this homonymy to make sure we referring to the intended Homer Simpson.

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.[7]

- *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 "Resources with Parts", "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 kind of 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

7. (Winston, Chaffin, and Herman 1987).

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

A seventh type of part-whole relationship called *Phase-Activity* was proposed by Storey.[8]

- *Phase-Activity* is similar to *feature-activity* except that the phases do not make sense as standalone activities without the context provided by the activity as a whole.

  Paying → is-part-of → Shopping

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

Attribution

In contrast to inclusion expressions that state relationships between resources, *attribution relationships* assert or assign values to properties for a particular resource. In *Resource Description and Metadata* 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 any object is an *attribute* of the object, and the *value* of that attribute might be "green."

8. (Storey 1993).

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:[9]

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

Dedre Gentner notes that this supposed distinction between one-predicate attributes and two-predicate relationships depends on context.[10]

For example, small can be viewed as an attribute, **X → is-small,** or as a relationship between X and some standard or reference Y, **X → is-smaller-than → Y.**

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

9. Martin is the animated gecko who is the advertising spokesman for Geico Insurance (http://www.geico.com/). Martin's wit and cockney accent make him engaging and memorable, and a few years ago he was voted the favorite advertising icon in the US.

10. (Gentner 1983).

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. This means that 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.

Finally, we note that we can express attribution relationships about other relationships, like the date a relationship was established. Homer and Marge Simpson's wedding anniversary is an attribute of their "is-married-to" relationship.

The semantic distinctions between attributes and other types of relationships are not strong ones, but they can be made clearer by implementation choices. For example, XML attributes are tightly coupled to a containing element, and their literal values are limited to atomic items of information. In contrast, inclusion relationships are expressed by literal containment of one XML element by another.

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.

Possession relationships also imply duration or *persistence*, and are often difficult to distinguish from relationships based on habitual location or practice. Miller and Johnson-Laird illustrate the complex nature of possession relationships with this sentence, which expresses three different types of them:[11]

> **He owns an umbrella but she's borrowed it, though she doesn't have it with her.**

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.

Symmetry

In most relationships the order in which the subject and object

11. (Miller and Johnson-Laird 1976, p 565).

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

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. [12]

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

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 generally

12. Some people have argued that meronomy is not transitive, but a closer look at their supposed counter-examples suggests otherwise. See Section 5 in (Winston, Chaffin, and Herman 1987).

substitutable for it. We make this explicit with an equivalence relationship.

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

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

Ontologies

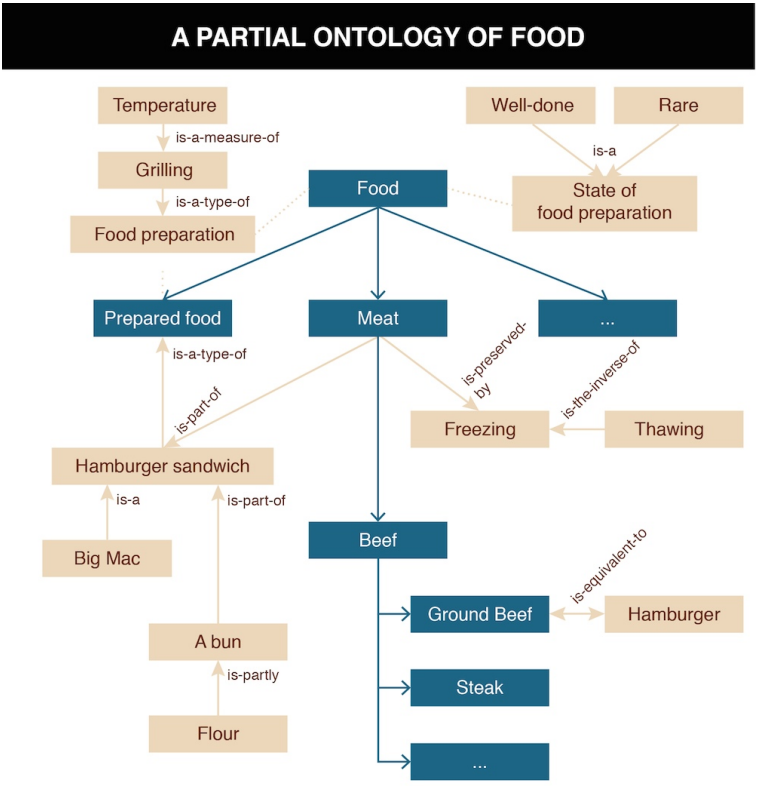
We now have described types and properties of semantic relationships in enough detail to return to the challenge we posed earlier: what information is required to fully understand relationships? This question has been asked and debated for decades and we will not pretend to answer it to any extent here.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



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."

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 just about everyone knows about food and related domains. A network of relationships

like these creates a resource that is called an *ontology*.[13] A visual depiction of the ontology illustrates this idea that it has a taxonomy as its conceptual scaffold. (See Figure: A Partial Ontology of Food.)

There are numerous formats for expressing ontologies, but many of them have recently converged to or are based on the *Web Ontology Language(*OWL), developed by the W3C. OWL ontologies use a formal logic-based language that builds on RDF ("Resource Description Framework (RDF)") to define resource classes and assign properties to them in rigorous ways, arrange them in a class hierarchy, establish their equivalence, and specify the properties of relationships.[14]

Ontologies are essential parts in some organizing systems, especially information-intensive ones where the scope and scale of the resources require an extensive and controlled description vocabulary. (See "The Process of Describing Resources".) The most extensive ontology ever created is Cyc, born in 1984 as an artificial intelligence research project. Three decades later, the latest version

13. *ontology* is a branch of philosophy concerned with what exists in reality and the general features and relations of whatever that might be (Hofweber 2009). Computer science has adopted "ontology" to refer to any computer-processable resource that represents the relationships among words and meanings in some knowledge domain. See (Gruber 1993), (Guarino 1998).

14. Web Ontology Language(OWL) http://www.w3.org/2004/OWL/.

of the Cyc ontology contains several hundred thousand terms and millions of assertions that interrelate them.[15]

15. http://www.cyc.com/.

39. 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.[1]

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

1. Languages and cultures differ in how they distinguish and describe kinship, so Bart might find the system of family organization easier to master in some countries and cultures and more difficult in others.

reassemble meaning from the two separate words. These "lexical gaps" differ from language to language, whereas "conceptual gaps"—the things we cannot think of or directly experience, like the pull of gravity— may be innate and universal. We revisit this issue as "linguistic relativity" in *Categorization: Describing Resource Classes and Types*. [2]

Earlier in this book we discussed the naming of resources ("The Problems of Naming") and the design of a vocabulary for resource description ("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. After our discussion about semantic relationships in this chapter, we now have a clearer sense of what is required to bring like things together, keep different things separate, and to satisfy any other goals for the organizing system.

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.[3]

2. (Bentivogli and Pianta 2000).

3. This example comes from (Fellbaum 2010, pages 236-237). German has a word *Kufenfahrzeug* for vehicle on runners.

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. ("Relationships among Word Meanings") and the most commonly used tool for describing them ("Thesauri").
- Second, we need to discuss relationships among the form of words. ("Relationships among Word Forms")

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 "The Semantic Perspective".

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 an exemplary 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 describe every property of robins, but it is sufficient to differentiate robins from bluebirds or eagles.[4]

Metonymy

Part-whole or meronymic semantic relationships have lexical analogues 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 top leaders reside 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..."

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."[5] This is an

4. (Miller 1998).

5. (Bolshakov and Gelbukh 2004), p, 314. The quote continues "The references to 'some class' and to 'insignificant change' make this definition rather vague, but we are not aware of any significantly stricter definition. Hence the creation of synonymy dictionaries,

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.

An unordered set of synonyms is often called a *synset*, a term first used by the WordNet "semantic dictionary" project started in 1985 by George Miller at Princeton.[6] Instead of using spelling as the

which are known to be quite large, is rather a matter of art and insight."

6. George Miller made many important contributions to the study of mind and language during his long scientific career. His most famous article, *The Magical Number Seven, Plus or Minus Two* (Miller 1956), was seminal in its proposals about information organization in human memory, even though it is one of the most misquoted scientific papers of all time. Relatively late in his career Miller began the WordNet project to build a semantic dictionary, which is now an essential resource in natural language processing applications. See http://wordnet.princeton.edu/.

primary organizing principle for words, WordNet uses their semantic properties and relationships to create a network that captures the idea that words and concepts are an inseparable system. Synsets are interconnected by both semantic relationships and lexical ones, enabling navigation in either space.[7]

Polysemy

We introduced the lexical relationship of *polysemy*, when a word has several different meanings or senses, in the context of problems with names ("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.[8]

Polysemy is represented in WordNet by including a word in multiple

7. This navigation is easiest to carry out using the commercial product called "The Visual Thesaurus" at http://www.visualthesaurus.com/.

8. These contrasting meanings for "bank" are clear cases of polysemy, but there are often much subtler differences in meaning that arise from context. The verb "save" seems to mean something different in "The shopper saved..." versus "The lifeguard saved..." although they overlap in some ways. (Fillmore and Atkins 2000) and others have proposed definitions of polysemy, but there is no rigorous test for determining when word meanings diverge sufficiently to be called different senses.

synsets. This enables WordNet to be an extremely useful resource for sense disambiguation in natural language processing research and applications. When a polysemous word is encountered, it and the words that are nearby in the text are looked up in WordNet. By following the lexical relationships in the synset hierarchy, a "synset distance" can be calculated. The smallest semantic distance between the words, which identifies their most semantically specific hypernym, can be used to identify the correct sense. For example, in the sentence:

Put the money in the bank

Two of the three WordNet senses for "money" are:

1) the most common medium of exchange

2) the official currency issued by a government or national bank

and the first two of the ten WordNet senses for "bank" are:

1) a financial institution that accepts deposits

2) sloping land, especially the slope beside a body of water

The synset hierarchies for the two senses of "money" intersect after a very short path with the hierarchy for the first sense of "bank," but do not intersect with the second sense of "bank" until they reach very abstract concepts.[9]

9. Many techniques for using WordNet to calculate measures of semantic similarity have been proposed. See (Budanitsky and Hirst 2006).

Antonymy

Antonymy is the lexical relationship between two words that have opposite meanings. *Antonymy* is a very salient lexical relationship, and for adjectives it is even more powerful than synonymy. In word association tests, when the probe word is a familiar adjective, the most common response is its antonym; a probe of "good" elicits "bad," and vice versa. Like synonymy, antonymy is sometimes exact and sometimes more graded.[10]

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.[11]

10. See (Gross and Miller, 1990).

11. This type of "lexical asymmetry" is called "markedness." The broader or dominant term is the unmarked one and the narrower one is the marked one. See (Battistella 1996).

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* ("The Vocabulary Problem") that becomes increasingly essential as the scope and scale of the organizing system grows. 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.[12]

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 *hypernomy* and *hyponymy* are usually encoded using `BT` ("broader term") and `NT` ("narrower term"):

Food BT Meat
Beef NT Meat

12. http://www.loc.gov/library/libarch-thesauri.html, http://www.getty.edu/research/tools/vocabularies/aat/index.html.

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. Many thesauri emphasize the cluster of relationships for these key words and de-emphasize the overall lexical hierarchy.

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

A thesaurus might employ `USE` as the inverse of the `UF` relationship to refer from a less preferred or variant term to a preferred one:

Victuals USE Food

Thesauri also use `RT` ("related term" or "see also") to indicate terms that are not synonyms but which often occur in similar contexts:

Food RT Cooking, Dining, Cuisine

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 called *morphology*.[13]

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* (discussed in more detail in *Interactions with Resources*) is morphological processing which removes 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. Many misspellings of common words (e.g., "pain") are words of lower frequency (e.g., "pane"), so adding "pane" to a list of misspelled words would occasionally identify it incorrectly. 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 popular photo-sharing site?

13. Languages differ a great deal in morphological complexity and in the nature of their morphological mechanisms. Mandarin Chinese has relatively few morphemes and few grammatical inflections, which leads to a huge number of homophones. English is pretty average on this scale. A popular textbook on morphology is (Haspelmath and Sims 2010).

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."[14] 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,"

14. These so-called endocentric compounds essentially mean what the morphemes would have meant separately. But if a "birdcage" is exactly a "bird cage," what is gained by creating a new word? This question has long been debated in subject classification, where it is framed as the contrast between "pre-coordination" and "post-coordination." For example, is it better to pre-classify some resources as about "Sports Gambling" or should such resources be found by intersecting those classified as about "Sports" and about "Gambling." See (Svenonius 2000, pages 187-192).

"-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.

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.[15]

15. English nouns have plural (book/books) and possessive forms (the professor's book), adjectives have comparatives and superlatives (big/bigger/biggest), and regular verbs have only four inflected forms (see http://cla.calpoly.edu/~jrubba/morph/morph.over.html). In contrast, in Classical Greek each noun can have 11 word forms, each adjective 30, and every regular verb over 300 (Anderson 2001).

40. 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.[1]

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

1. Of the five perspectives on relationships in this chapter, the structural one comes closest to the meaning of "relation" in mathematics and computer science, where a relation is a set of ordered elements ("tuples") of equal degree ("Degree"). A binary relation is a set of element pairs, a ternary relation is a set of 3-tuples, and so on. The elements in each tuple are "related" but they do not need to have any "significant association" or "relationship" among them.

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, Markoff and Sengupta describe a 2011 study using Facebook data that computed the average "degree of separation" of any two people in the Facebook world to be 4.74.[2]

> Stop and Think: Kevin Bacon Numbers
>
> See http://oracleofbacon.org/ for a web-based demonstration of "Kevin Bacon Numbers," which measure the average degrees of separation among more than 2.6 million actors in more than 1.9 million movies. Its name reflects the parlor game "Six Degrees of Kevin Bacon," a pun on "six degrees of separation" that is often associated with Travers and Milgram's work; the game relies on the remarkable variety of Bacon's roles, and hence the number of fellow actors in his movies (two actors in the same movie have one degree of separation). Bacon's Bacon Number is 2.994, but it turns out that more than 300 actors are closer to the center of the movie universe than Bacon. Try some famous

2. See (Travers and Milgram 1969) and (Markoff and Sengupta 2011).

actors and see if their Bacon Numbers are greater or smaller than Bacon's. (Hint: older actors have been in more movies.)

Many types of resources have internal structure in addition to their structural relationships with other resources. Of course, we have to remember (as we discussed in "Resource Identity") 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," something we consider in depth in *Interactions with Resources*.

Business Structures

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. The strict management hierarchy at Foxconn is needed to enable close oversight of large numbers of low level employees in the manufacturing industry, with workers organized by physical location.

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. A consulting firm's matrix structure might emphasize an employee's functional role (e.g., "process engineering consultant") and disassociate it from the employee's home location, which is why consultants spend so much time traveling on airplanes from project to project.

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. We acknowledged in "The Concept of "Intentional Arrangement"" that there is information in the piles of debris left after a tornado or tsunami and in the strata of the Grand Canyon. These structural patterns might be of interest to meteorologists, geologists, or others but because they were not

created by an identifiable agent following one or more organizing principles, they are not our primary focus.

> Stop and Think:
> Intentional, Implicit, or Explicit Structure?
>
> Find a map of the states (or provinces or other divisions) in your country. You probably think of some set of these as members of a collection. Other than their literal arrangement (e.g., "x is next to y, y is east of z"), how could you describe their relationships to each other within the collection? Are these relationships based on natural or unintentional properties or intentional ones? Example: in the United States, California, Oregon, and Washington are considered the "West Coast" and the Pacific Ocean determines their western boundaries. Some of the borders between the states are natural, determined by rivers, and other borders are more intentional and arbitrary.

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."[3]

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.

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

3. This seems like an homage to Jimi Hendrix based on the title from a 1967 song, Third Stone from the Sun http://en.wikipedia.org/wiki/Third_Stone_from_the_Sun.

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*.

> Mixed Content
>
> Mixed content distinguishes XML from other data representation languages. It is this structural feature, combined with the fact that child nodes in the XML Infoset ("XML Information Set") are ordered, that makes it possible for XML documents to function both as human reader-oriented, textual documents and as structured data formats. It allows us to use natural language in writing descriptions while still enabling us to identify content by type by embedding markup to enclose "semantic nuggets" in otherwise undifferentiated text.[4]

4. The subfield of natural language processing called "named entity recognition" has as its goal the creation of mixed content by identifying people, companies,

> The Guidelines for Electronic Text Encoding and Interchange, produced by the *Text Encoding Initiative(TEI)*, for example, includes a set of elements and attributes for *Names, Dates, People and Places*.[5]

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.

The internal parts of XML documents can be described, found and selected using the XPath language, which defines the structures and patterns used by XML forms, queries, and transformations. The key idea used by XPath is that the structure of XML documents is a tree of information items called nodes, whose locations are described in terms of the relationships between nodes. The relationships built

organizations, dates, trademarks, stock symbols, and so on in unstructured text.

5. Text Encoding Initiative 13. Names, Dates, People, and Places.

into XPath, which it calls axes, include self, child, parent, following, and preceding, making it very easy to specify a structure-based query like "find all sections in Chapter 1 through Chapter 5 that have at least two levels of subsections."[6]

In addition, tools like Schematron take advantage of XPath's structural descriptions to test assertions about a document's structure and content. For example, a common editorial constraint might be that a numbered list must have at least three items.[7]

In more qualitative, less information-intensive and more experience-intensive domains, we move toward the narrative end of the Document Type Spectrum, and document instances become more heterogeneous because they are produced by and for people. (See the sidebar, The Document Type Spectrum in "Resource Domain".) 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>`.

> Structural Metadata
> Structural metadata, in the form of a schema for a

6. See (Holman 2001) or (Tidwell 2008).

7. See (van der Vlist 2007) and schematron.org for overviews. See (Hamilton and Wood 2012) for a detailed case study.

database or document, describes a class of information resources, and may also prescribe grammatical details of inclusion and attribution relationships among the components. For example, the chapters of this book contain four levels of subsections. Each of those sections contains a title, some paragraphs and other text blocks, and subordinate sections. The textual content of the paragraphs includes highlighted terms and phrases that are defined *in situ* and referenced again in the glossary and index; there are also bibliographic citations that are reflected in the bibliography and index. We can discover these characteristics of a book through observation, but we could also examine its structural metadata, in its schema.

Structural metadata allows us to describe and prescribe relations among database tables, within the chapters of a book, or among parts in an inventory management system. The schema for HTML, for example, informs us that the `<A>` element can be used to signal a hypertext link-end; whether that link-end is an anchor or a target, or both, depends on the combination of values assigned to attributes. In HTML, the optional `REL` attribute may contain a value that signals the purpose of a hypertext link, and any HTML element may include a `CLASS` attribute value that may be used as a CSS selector for the purposes of formatting or dynamic interactions.

The usefulness of any given schema is often a function of the precision with which we may make useful statements based upon the descriptions and

prescriptions it offers. Institutional schemas tend to be more prescriptive and restrictive, stressing professional orthodoxy and conformance to controlled vocabularies. Schemas for the information content in social and informal applications tend to be less prescriptive. Whether and how we use structural metadata is a tradeoff. Structural metadata is essential to enable quality control and maintenance in information collection and publishing processes, but someone has to do the work to create it.

The internal structural hierarchy in a resource 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 to only 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, a given resource may be a chapter in one book while being an appendix in another. Some tables of contents are created as a static structural description, but others are dynamically generated from the internal structures whenever the resource is accessed. In addition, other types of entry points can be generated from the names or descriptions of content components, like selectable lists of tables, figures, maps, or code examples.

DocBook Schema
The schema most commonly used for producing

technical books is called DocBook; it describes every XML element and attribute and prescribes their grammatical forms. The schema lets us know that a formal paragraph must include a title, and that a title may contain emphasis. A schema can also describe and prescribe the lexical value space of a postal code, or require that every list must have at least three items. The DocBook schema is well-documented and has been production-tested in institutional publishing contexts for over twenty years.[8]

Identifying the components and their structural relationships in documents is easier when they follow consistent rules for structure (e.g., every non-text component must have a title and caption) and presentation (e.g., *hypertext links* in web pages are underlined and change cursor shapes when they are "moused over") that reinforce the distinctions between types of information components. Structural and presentation features are often ordered on some dimension (e.g., type size, line width, amount of white space) and used in a correlated manner to indicate the importance of a content component.[9]

Many indexing algorithms treat documents as "bags of words" to

8. (Walsh 2010).

9. These layout and typographic conventions are well known to graphic designers (Williams 2012) but are also fodder for more academic treatment in studies of visual language or semiotics (Crow 2010).

compute statistics about the frequency and distribution of the words they contain while ignoring all semantics and structure. In *Interactions with Resources*, we contrast this approach with algorithms that use internal structural descriptions to retrieve more specific parts of documents.

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.[10]

10. (Page, Brin, Motwani, and Winograd 1999) describes Page Rank when its inventors were computer science graduate students at Stanford. It is not a coincidence that the technique shares a name with one of its inventors,

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.

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."[11]

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

Google co-founder and CEO Larry Page. (Langville and Meyer 2012) is an excellent textbook. The ultimate authority about how page rank works is Google; see https://www.google.com/insidesearch/howsearchworks/thestory/.

11. (Bush 1945). "Wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them..." See http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/.

we often need to consider links from other perspectives. (See the sidebar, Perspectives on Hypertext Links).

> Transclusion
>
> The inclusion, by hypertext reference, of a resource or part of a resource into another resource is called *transclusion*. Transclusion is normally performed automatically, without user intervention. The inclusion of images in web documents is an example of transclusion. Transclusion is a frequently used technique in business and legal document processing, where re-use of consistent and up-to-date content is essential to achieve efficiency and consistency.

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.[12]

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.*[13]

At about the same time, and without knowing about Nelson's work, Douglas Engelbart's *Augmenting the Human Intellect,* described a future world in which professionals equipped with interactive

12. (Nelson 1981).

13. See *Computer Lib/Dream Machines* (Nelson 1981) for an early example of Nelson's non-liner book style.

computer displays utilize an information space consisting of a cross-linked resources.[14]

In the 1960s, when computers lacked graphic displays and were primarily employed to solve complex mathematical and scientific problems that might take minutes, hours or even days to complete, Nelson's and Engelbart's visions of hypertext-based personal computing may have seemed far-fetched. In spite of this, by 1968, Engelbart and his team demonstrated human computer interface including the mouse, hypertext, and interactive media, along with a set of guiding principles.[15]

> 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*. More generally, rhetorical structure theory analyzes how different conventions or signals in texts indicate

14. (Engelbart 1963) Douglas Engelbart credits Bush's As We May Think article as his direct inspiration. Engelbart was in the US Navy, living in a hut in the South Pacific during the last stages of WWII when he read The Atlantic monthly magazine in which Bush's article was published.

15. Doug Engelbart's demonstration has been called the "Mother of All Demos" and can be seen in its entirety at http://sloan.stanford.edu/MouseSite/1968Demo.html.

relationships between texts or parts of them, like the subtle differences in polarity among "see," "see also," and "but see" as citation signals.[16]

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*.[17]

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

16. See (Lorch 1989), (Mann and Thomson 1988). For example, an author might use "See" as in "See (Glushko et al. 2013)" when referring to this chapter if it is consistent with his point of view. On the other hand, that same author could use "but" as a contrasting citation signal, writing "But see (Glushko et al. 2013)" to express the relationship that the chapter disagrees with him.

17. Before the web, most hypertexts implementations were in stand-alone applications like CD-ROM encyclopedias or in personal information management systems that used "cards" or "notes" as metaphors for the information units that were linked together, typically using rich taxonomies of *link types*. See (Conklin 1987), (Conklin and Begeman 1988), and (DeRose 1989).

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.[18]

(See "The Architectural Perspective")

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.[19]

18. Many of the pre-web hypertext designs of the 1980s and 1990s allowed for n-ary links. The Dexter hypertext reference model (Halasz and Schwartz 1994) elegantly describes the typical architectures. However, there is some ambiguity in use of the term binary in hypertext link architectures. One-to-one vs. one-to-many is a cardinality distinction, and some people reserve binary to discussion about degree.

19. See (Weinreich, Obendorf, and Lamersdorf 2001).

20

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*.[21]

(See “The Implementation Perspective ”)

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.[22]

They invented the methods for encoding and following *hypertext links* using the now popular HyperText Markup Language(HTML).[23]

20. Most designers use a variety of visual cues and conventions to distinguish hyperlinks (e.g., plain hyperlink, button, selectable menu, etc.) so that users can anticipate how they work and what they mean. A recent counter-trend called “flat design” —exemplified most notably by the user interfaces of Windows 8 and iOS 7— argues for a minimalist style with less variety in typography, color, and shading. Flat designs are easier to adapt across multiple devices, but convey less information.

21. See (Brailsford 1999), (Wilde and Lowe 2002).

22. (Gillies and Cailliau 2000).

23. Most web links are very simple in structure. The anchor

> 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.[24]
>
> By 1993, personal computers, with a graphic display, speakers and a mouse pointer, had become ubiquitous. NCSA Mosaic is widely credited with popularizing the World Wide Web and HTML in 1993,

text in the linking document is wrapped in `<A>` and `</A>` tags, with an `HREF` (hypertext reference) attribute that contains the URI of the link destination if it is in another page, or a reference to an ID attribute if the link is to a different part of the same page. HTML also has a `<LINK>` tag, which, along with `<A>` have `REL` (relationship) and `REV` (reverse relationship) attributes that enable the encoding of typed relationships in links. In a book context for example, link relationships and reverse relations include obvious candidates such as next, previous, parent, child, table of contents, bibliography, glossary and index.

24. Using hypertext links as interaction controls is the modern dynamic manifestation of cross references between textual commentary and illustrations in books, a mechanism that dates from the 1500s (Kilgour 1998). Hypertext links can be viewed as state transition controls in distributed collections of web-based resources; this design philosophy is known as *Representational State Transfer*(REST). See (Wilde and Pautasso 2011).

by introducing inline graphics, audio and video media, rather than having to link to media segments in a separate window.[25]

The ability to transclude images and other media would transform the World Wide Web from a text-only viewer with links to a "networked landscape" with hypertext signposts to guide the way. On 12 November 1993, the first full release of NCSA Mosaic on the world's three most popular operating systems (X Windows, Microsoft Windows, and Apple Macintosh) enabled the general public to access the network with a graphical browser.[26]

Since browsers made them familiar, hypertext links have been used in other computing applications as structure and navigation mechanisms.

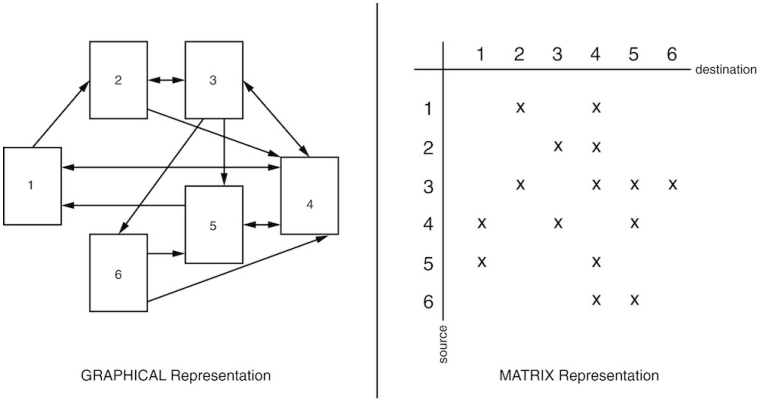
Analyzing Link Structures

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.

25. Mosaic was developed in Joseph Hardin's lab at the National Center for Supercomputing Applications(NCSA), hosted by the University of Illinois, at Urbana/Champaign by Marc Andreesen, Eric Bina and a team of student programmers. Mosaic was initially developed on the Unix X Window System. See http://www.ncsa.illinois.edu/Projects/mosaic.html.

26. (Schatz and Hardin 1994).

A graphical representation of link structure is shown on the left panel of figure Figure: 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.



Representing Link Structures

The structure of links between web resources can be represented graphically or in a matrix. The matrix representation is a more abstract one that can be analyzed by computers.

A matrix representation of the same link structure is shown on the right panel of Figure: 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.[27]

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.

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.[28]

27. Reachability is determined by calculating the transitive closure of the link matrix. A classic and well written explanation is (Agrawal, Borgida, and Jagadish 1989).

28. Eugene Garfield developed many of the techniques for studying scientific citation and he has been called the "grandfather of Google" (http://blog.lib.uiowa.edu/hardinmd/2010/07/12/eugene-garfield-librarian-grandfather-of-google/) because of Google's use of citation patterns to determine relevance. See (Garfield,

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.[29]

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

Cronin, and Atkins 2000) for a set of papers that review Garfield's many contributions. See (Bar-Ilan 2008) and (Neuhaus and Daniel 2008) for recent reviews of data sources and citation metrics.

29. Shepard first put adhesive stickers into case books, then published lists of cases and their citations. Shepardizing is a big business for Lexis/Nexis and Westlaw (where the technique is called "KeyCite").

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.[30]

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.[31]

30. See `http://altmetrics.org/manifesto/` for the original call for altmetrics. Altmetric.com and Plum Analytics are firms that provide altmetrics to authors, publishers, and academic institutions. In 2016 the National Information Standards Organization sought to standardize the definition and use cases for altmetrics, which should benefit everyone who cares about them. See also `http://www.niso.org/topics/tl/altmetrics_initiative/`

31. See (Watts 2004) for a detailed review of the theoretical foundations. See (Wu 2012) for applications in web-based social networks.

41. 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. Application and technology lifecycles have never been shorter, and vast amounts of new data are being created by increased tracking of online interactions and by all the active resources that are now part of the Internet of Things. Organizing systems built without clear architectural foundations cannot easily scale up in size and scope to handle these new requirements.

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 exactly the same, but its degree is now *unary* because only 1 entity type is needed to instantiate the two arguments:

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

Some relationships are best expressed as *ternary* ones that involve three different entity types. An example that appears in numerous data modeling books is one like this:

Supplier → provides → Part → assembled-in → Product

It is always possible to represent ternary relationships as a set of binary ones by creating a new entity type that relates to each of the others in turn. This new entity type is called a dummy in modeling practice.

Supplier → provides → DUMMY
Part → provided-for → DUMMY
DUMMY → assembled-in → Product

This transformation from a sensible ternary relationship to three binary ones involving a DUMMY entity type undoubtedly seems strange, but it enables all relationships to be binary while still preserving the meaning of the original ternary one. Making all relationships binary makes it easier to store relationships and combine them to discover new ones.

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 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

As we did with the ternary relationship in "Degree", 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.

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 "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.

42. 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 *The Forms of Resource Descriptions*.

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:

Subject-Predicate-Object 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:[1]

1. We are assuming a schema that establishes that the `name` attributes are of type ID and that the other attributes are of type IDREFS. This schema allows for polygamy, the possibility of multiple values for the `spouse` attribute. Restrictions on the number of spouses can be enforced

```
<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.

Syntax and Grammar

The *syntax* and *grammar* of a language consist 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

with Schematron. (Also see the sidebar, Inclusions and References).

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[2]

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

2. (Chomsky 1957) used these now famous sentences to motivate the distinction between syntax and semantics. He argued that since the probability in both cases that the words had previously occurred in this order was essentially zero, statistics of word occurrence could not be part of language knowledge. See. http://en.wikipedia.org/wiki/Colorless_green_ideas_sleep_furiously.

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.

43. Relationships in Organizing Systems

In the previous sections as we surveyed the five perspectives on analyzing relationships, we mentioned numerous examples where relationships had important roles in organizing systems. In this final section we examine three contexts for organizing systems where relationships are especially fundamental; the *Semantic Web* and Linked Data, bibliographic organizing systems, and situations involving system integration and interoperability.

The Semantic Web and Linked Data

In a classic 2001 paper, Tim Berners-Lee laid out a vision of a *Semantic Web* in which all information could be shared and processed by automated tools as well as by people.[1] The essential

1. (Berners-Lee, Hendler, and Lassila, 2001) is the classic paper, and (Shadbolt, Hall, and Berners-Lee 2006) is something of a revisionist history.

   Ironically, the web was not semantic originally because Berners-Lee implemented web documents using a presentation-oriented HTML markup language. Designing HTML to be conceptually simple and easy to implement led to its rapid adoption. HTML documents can make assertions and describe relationships using `REL`

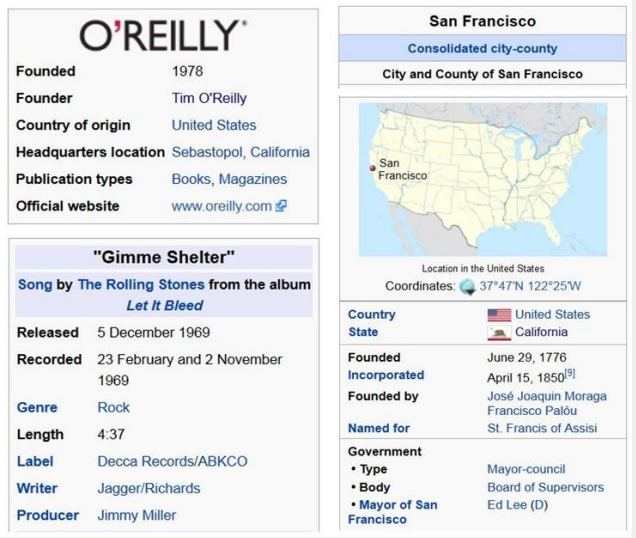
technologies for making the web more semantic and relationships among web resources more explicit are applications of XML, including RDF ("Resource Description Framework (RDF)"), and OWL ("Ontologies"). Many tools have been developed to support more semantic encoding, but most still require substantial expertise in semantic technologies and web standards.[2]

More likely to succeed are applications that aim lower, not trying to encode all the latent semantics in a document or web page. For example, some wiki and blogging tools contain templates for semantic annotation, and Wikipedia has thousands of templates and "infoboxes" to encourage the creation of information in content-encoded formats.

Wikipedia Infoboxes

and REV attributes, but browsers still do not provide useful interactions for link relations.

2. For example, Protégé a free, open-source platform with a suite of tools to construct domain models and knowledge-based applications with ontologies. (See http://protege.stanford.edu/)



Wikipedia encourages authors to augment their articles with "info boxes" that organize sets of structured information generically relevant to the type of resource that is the subject of the article. These three examples show parts of the info boxes for "Company," "Song," and "City."

(Collage of screenshots by R. Glushko.)

The "Linked Data" movement is an extension of the *Semantic Web* idea to reframe the basic principles of the web's architecture in more semantic terms. Instead of the limited role of links as simple untyped relationships between HTML documents, links between

resources described by RDF can serve as the bridges between islands of semantic data, creating a Linked Data network or cloud.[3]

Bibliographic Organizing Systems

Much of our thinking about relationships in organizing systems for information comes from the domain of bibliographic cataloging of library resources and the related areas of classification systems and descriptive thesauri. Bibliographic relationships provide an important means to build structure into library catalogs.[4]

Bibliographic relationships are common among library resources. Smiraglia and Leazer found that approximately 30% of the works in the *Online Computer Library Center(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. Moreover, "canonical" works that have strong cultural meaning and influence, such as "the plays of William Shakespeare" and The Bible, have very large and complex bibliographic families.[5]

3. See http://linkeddata.org/ and "Syntax".

4. Barbara Tillett has written extensively about the theory of bibliographic relationships; (Tillett 2001) is an especially useful resource because it is a chapter in a comprehensive discussion ambitiously titled *Relationships in the Organization of Knowledge* (Bean and Green 2001).

5. (Smiraglia and Leazer 1999).

Tillett's Taxonomy

Barbara Tillett, in a study of 19th and 20th-century catalog rules, found that many different catalog rules have existed over time to describe bibliographic relationships. She developed a taxonomy of bibliographic relationships that includes equivalence, derivative, descriptive, whole-part, accompanying, sequential or chronological, and shared characteristic. These relationship types span the relationship perspectives defined in this chapter; equivalence, derivative, and description are semantic types; whole-part and accompanying are part semantic and part structural types; sequential or chronological are part lexical and part structural types; and shared characteristics are part semantic and part lexical types.[6]

Smiraglia expanded on Tillett's derivative relationship to create seven subtypes: simultaneous derivations, successive derivations, translations, amplifications, extractions, adaptations, and performances.[7]

In "Identity and Bibliographic Resources", "Identity and Bibliographic Resources," we briefly mentioned the four-level abstraction hierarchy for resources introduced in the Functional Requirements for Bibliographic Records report. FRBR was highly influenced by Tillett's studies of bibliographic relationships, and prescribes how the relationships among resources at different levels are to be expressed (work-work, expression-expression, work-expression, expression-manifestation, and so on).

6. (Tillett 1991, 1992, 2003).

7. (Smiraglia 1994).

Resource Description and Access (RDA)

Many cataloging researchers have recognized that online catalogs do not do a very good job of encoding bibliographic relationships among items, both due to catalog display design and to the limitations of how information is organized within catalog records.[8] 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. Similarly, MARC records can be formatted and displayed in web library catalogs, but the data within the records are not available for re-use, re-purposing, or re-arranging by researchers, patrons, or librarians.

The Resource Description and Access(RDA) next-generation cataloging rules are attempting to bring together disconnected resource descriptions to provide more complete and interconnected data about works, authors, publications, publishers, and subjects.

RDA uses RDF to assert relationships among bibliographic materials.[9]

RDA and the Semantic Web

The move in RDA to encode bibliographic data in RDF stems from the desire to make library catalog data more web-accessible. As web-based data mash-ups, application programming interfaces

8. (Tillett 2005).

9. See Section 8.1.3.1.

(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.[10]

In addition to simply making library data more web accessible, 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*(FOAF) 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. RDA's process of moving to RDF is well underway.[11]

10. See (Coyle 2010a).

11. The FRBR entities, RDA data elements, and RDA value vocabularies have been defined in alignment with RDF using the Simple Knowledge Organization System (SKOS). SKOS is an "RDF-compliant language specifically designed for term lists and thesauri" (Coyle 2010b). The SKOS website 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 FRBR and RDA vocabularies.

Integration and Interoperability

Integration is the controlled sharing of information between two (or more) business systems, applications, or services within or between firms. *Integration* means that one party can extract or obtain information from another one, it does not imply that the recipient can make use of the information.

Interoperability goes beyond integration to mean that systems, applications, or services that exchange information can make sense of what they receive. *Interoperability* can involve identifying corresponding components and relationships in each system, transforming them syntactically to the same format, structurally to the same granularity, and semantically to the same meaning.

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 are not 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. *Interactions with Resources* will dive deeper into

interoperability issues, especially those that arise in business contexts.

44. Key Points in Chapter Six

- What is a relationship?

  A relationship is "an association among several things, with that association having a particular significance."

  (See "Introduction")

- Why is it essential to include the type of association in a specification of a relationship?

  Just identifying the resources involved is not enough because several different relationships can exist among the same resources.

  (See "Describing Relationships: An Overview")

- What is the most typical grammatical model for expressing a relationship?

  Most relationships between resources can be expressed using a subject-predicate-object model.

  (See "The Semantic Perspective" and "Choice of Implementation")

- What knowledge does a computer need to be able to understand relational expressions?

  For a computer to understand relational 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.

  (See "The Semantic Perspective")

- What are three broad categories of semantic relationships?

Three broad categories of semantic relationships are inclusion, attribution, and possession.

(See "Types of Semantic Relationships")

- What is a taxonomy?

  A set of interconnected class inclusion relationships creates a hierarchy called a taxonomy.

  (See "Inclusion")

- What kind of semantic relationship is expressed by a classification?

  Classification is a class inclusion relationship between an instance and a class.

  (See "Inclusion")

- What kinds of inferences are possible when relationships are transitive?

  Ordering and inclusion relationships are inherently transitive, enabling inferences about class membership and properties.

  (See "Transitivity")

- What is an ontology?

  Class inclusion relationships form a framework to which other kinds of relationships attach, creating a network of relationships called an ontology.

  (See "Ontologies")

- What is hyponymy?

  When words encode the semantic distinctions expressed by

class inclusion, the more specific class is called the hyponym; the more general class is the hypernym.

(See "Hyponymy and Hyperonymy")

- What is a practical application of morphological analysis?

  Morphological analysis of how words in a language are created from smaller units is heavily used in text processing.

  (See "Relationships among Word Forms")

- What are the two types of structural relationships?

  Many types of resources have internal structure in addition to their structural relationships with other resources.

  (See "Structural Relationships within a Resource" and "Structural Relationships within a Resource")

- What is link analysis?

  Using the pattern of links between documents to understand the structure of knowledge and the structure of the intellectual community that creates it is an idea that is nearly a century old.

  (See "Structural Relationships between Resources")

- When are hypertext links merely structural?

  Many hypertext links are purely structural because there is no explicit representation of the reason for the relationship.

  (See the sidebar, Perspectives on Hypertext Links)

- What aspects of relationships between resources does the architectural perspective emphasize?

  The architectural perspective on resources emphasizes the number and abstraction level of the components of a

relationship; three important issues are degree, cardinality, and directionality.

(See "The Architectural Perspective")

- What are the essential semantic web technologies?

  The essential technologies for making the web more semantic and relationships among web resources more explicit are XML, RDF, and OWL.

  (See "The Semantic Web and Linked Data")

- What is the origin of the study of relationships in organizing systems?

  Much of our thinking about relationships in organizing systems for information comes from the domain of bibliographic cataloging of library resources and the related areas of classification systems and descriptive thesauri.

  (See "Bibliographic Organizing Systems")

- What is RDA?

  The Resource Description and Access(RDA) next-generation cataloging rules are attempting to bring together disconnected resource descriptions.

  (See "Resource Description and Access (RDA)")

- What is integration?

  Integration is the controlled sharing of information between two (or more) business systems, applications, or services within or between firms.

  (See "Integration and Interoperability")

- How is interoperability different from integration?

Interoperability goes beyond integration to mean that systems, applications, or services that exchange information can make sense of what they receive.

(See "Integration and Interoperability")

PART VII

CATEGORIZATION: DESCRIBING RESOURCE CLASSES AND TYPES

Robert J. Glushko

Rachelle Annechino

Jess Hemerly

Robyn Perry

Longhao Wang

45. Introduction (VII)

For nearly two decades, a TV game show called Pyramid aired in North America. The show featured two competing teams, each team consisting of two contestants: an ordinary civilian contestant and a celebrity. In the show's first round, both teams' members viewed a pyramid-shaped sign that displayed six category titles, some straightforward like "Where You Live" and others less conventional like "Things You Need to Feed." Each team then had an opportunity to compete for points in 30-second turns. The goal was for one team member to gain points by identifying a word or phrase related to the category from clues provided by the other team member. For example, a target phrase for the "Where You Live" category might be "zip code," and the clue might be "Mine is 94705." "Things you Need to Feed" might include both "screaming baby" and "parking meter."

The team that won the first round advanced to the "Winner's Circle," where the game was turned around. This time, only the clue giver was shown the category name and had to suggest concepts or instances belonging to that category so that the teammate could guess the category name. Clues like "alto," "soprano," and "tenor" would be given to prompt the teammate to guess "Singing Voices" or "Types of Singers."

As the game progressed, the categories became more challenging. It was interesting and entertaining to hear the clue receiver's initial guess and how subsequent guesses changed with more clues. The person giving clues would often become frustrated, because to them their clues seemed obvious and discriminating but would seem not to help the clue receivers in identifying the category. Viewers enjoyed sharing in these moments of vocabulary and category confusion.

The Pyramid TV game show developers created a textbook example for teaching about categories—groups or classes of things, people,

processes, events or anything else that we treat as equivalent—and categorization—the process of assigning instances to categories. The game is a useful analog for us to illustrate many of the issues we discuss in this chapter. The Pyramid game was challenging, and sometimes comical, because people bring their own experiences and biases to understanding what a category means, and because not every instance of a category is equally typical or suggestive. How we organize reflects our thinking processes, which can inadvertently reveal personal characteristics that can be amusing in a social context. Hence, the popularity of the Pyramid franchise, which began on CBS in 1973 and has been produced in 20 countries.

Many texts in library science introduce categorization via cataloging rules, a set of highly prescriptive methods for assigning resources to categories that some describe and others satirize as "mark 'em and park 'em." Many texts in computer science discuss the process of defining the categories needed to create, process, and store information in terms of programming language constructs: "here's how to define an abstract type, and here's the data type system." Machine learning and *data science* texts explain how categories are created through statistical analysis of the correlations among the values of features in a collection or dataset. We take a very different approach in this chapter, but all of these different perspectives will find their place in it.[1]

1. Cataloging and programming are important activities that need to be done well, and prescriptive advice is often essential. However, we believe that understanding how people create psychological and linguistic categories can help us appreciate that cataloging and information systems design are messier and more intellectually challenging activities than we might otherwise think.

Navigating This Chapter

In the following sections, we discuss how and why we create categories, reviewing some important work in philosophy, linguistics, and cognitive psychology to better understand how categories are created and used in organizing systems. We discuss how the way we organize differs when we act as individuals or as members of social, cultural, or institutional groups ("The What and Why of Categories"); later we share principles for creating categories("Principles for Creating Categories"), design choices ("Category Design Issues and Implications"), and implementation experience ("Implementing Categories"). Throughout the chapter, we will compare how categories created by people compare with those created by computer algorithms. As usual, we close the chapter with a summary of the key points ("Key Points in Chapter Seven").

46. The What and Why of Categories

The What and Why of Categories

Categories are *equivalence classes*, sets or groups of things or abstract entities that we treat the same. This does not mean that every instance of a category is identical, only that from some perspective, or for some purpose, we are treating them as equivalent based on what they have in common. When we consider something as a member of a category, we are making choices about which of its properties or roles we are focusing on and which ones we are ignoring. We do this automatically and unconsciously most of the time, but we can also do it in an explicit and self-aware way. When we create categories with conscious effort, we often say that we are creating a model, or just modeling. You should be familiar with the idea that a model is a set of simplified descriptions or a physical representation that removes some complexity to emphasize some features or characteristics and to de-emphasize others.[1]

1. Cognitive science mostly focuses on the automatic and unconscious mechanisms for creating and using categories. This disciplinary perspective emphasizes the activation of category knowledge for the purpose of making inferences and "going beyond the information

When we encounter objects or situations, recognizing them as members of a category helps us know how to interact with them. For example, when we enter an unfamiliar building we might need to open or pass through an entryway that we recognize as a door. We might never have seen that particular door before, but it has properties and affordances that we know that all doors have; it has a doorknob or a handle; it allows access to a larger space; it opens

given," to use Bruner's classic phrase (Bruner 1957). In contrast, the discipline of organizing focuses on the explicit and self-aware mechanisms for creating and using categories because by definition, organizing systems serve intentional and often highly explicit purposes. Organizing systems facilitate inferences about the resources they contain, but the more constrained purposes for which resources are described and arranged makes inference a secondary goal.

Cognitive science is also highly focused on understanding and creating computational models of the mechanisms for creating and using categories. These models blend data-driven or bottom-up processing with knowledge-driven or top-down processing to simulate the time course and results of categorization at both fine-grained scales (as in word or object recognition) and over developmental time frames (as in how children learn categories). The discipline of organizing can learn from these models about the types of properties and principles that organizing systems use, but these computational models are not a primary concern to us in this book.

and closes. By mentally assigning this particular door to the "doors" category we distinguish it from "windows," a category that also contains objects that sometimes have handles and that open and close, but which we do not normally pass through to enter another space. Categorization judgments are therefore not just about what is included in a class, but also about what is excluded from a class. Nevertheless, the category boundaries are not sharp; a "Dutch door" is divided horizontally in half so that the bottom can be closed like a door while the top can stay open like a window.

Categories are *cognitive and linguistic models* for applying prior knowledge; creating and using categories are essential human activities. Categories enable us to relate things to each other in terms of similarity and dissimilarity and are involved whenever we perceive, communicate, analyze, predict, or classify. Without categories, we would perceive the world as an unorganized blur of things with no understandable or memorable relation to each other. Every wall-entry we encounter would be new to us, and we would have to discover its properties and supported interactions as though we had never before encountered a door. Of course, we still often need to identify something as a particular instance, but categories enable us to understand how it is equivalent to other instances. We can interchangeably relate to something as specific as "the wooden door to the main conference room" or more generally as "any door."

All human languages and cultures divide up the world into categories. How and why this takes place has long been debated by philosophers, psychologists and anthropologists. One explanation for this differentiation is that people recognize structure in the world, and then create categories of things that "go together" or are somehow similar. An alternative view says that human minds make sense of the world by imposing structure on it, and that what goes together or seems similar is the outcome rather than a cause of categorization. Bulmer framed the contrast in a memorable way by

asking which came first, the chicken (the objective facts of nature) or the egghead (the role of the human intellect).[2]

A secondary and more specialized debate going on for the last few decades among linguists, cognitive scientists, and computer scientists concerns the extent to which the cognitive mechanisms involved in category formation are specialized for that purpose rather than more general learning processes.[3]

2. However, even the way this debate has been framed is a bit controversial. Bulmer's chicken, the "categories are in the world" position, has been described as empirical, environment-driven, bottom-up, or objectivist, and these are not synonymous. Likewise, the "egghead" position that "categories are in the mind" has been called rational, constructive, top-down, experiential, and embodied—and they are also not synonyms. See (Bulmer 1970). See also (Lakoff 1990), (Malt 1995).

3. Is there a "universal grammar" or a "language faculty" that imposes strong constraints on human language and cognition? (Chomsky 1965) and (Jackendoff 1996) think so. Such proposals imply cognitive representations in which categories are explicit structures in memory with associated instances and properties. In contrast, generalized learning theories model category formation as the adjustment of the patterns and weighting of connections in neural processing networks that are not specialized for language in any way. Computational simulations of semantic networks can reproduce the

> Even before they can talk, children behave in ways that suggest they have formed categories based on shape, color, and other properties they can directly perceive in physical objects.[4]
>
> People almost effortlessly learn tens of thousands of categories embodied in the culture and language in which they grow up. People also rely on their own experiences, preferences, and goals to adapt

experimental and behavioral results about language acquisition and semantic judgments that have been used as evidence for explicit category representations without needing anything like them. (Rogers and McClelland 2008) thoroughly review the explicit category models and then show how relatively simple learning models can do without them.

4. The debates about human category formation also extend to issues of how children learn categories and categorization methods. Most psychologists argue that category learning starts with general learning mechanisms that are very perceptually based, but they do not agree whether to characterize these changes as "stages" or as phases in a more complex dynamical system. Over time more specific learning techniques evolve that focus on correlations among perceptual properties (things with wings tend to have feathers), correlations among properties and roles (things with eyes tend to eat), and ultimately correlations among roles (things that eat tend to sleep). See (Smith and Thelen 2003).

these *cultural categories* or create entirely individual ones that they use to organize resources that they personally arrange. Later on, through situational training and formal education, people learn to apply systematic and logical thinking processes so that they can create and understand categories in engineering, logistics, transport, science, law, business, and other institutional contexts.

These three contexts of *cultural*, *individual*, and *institutional categorization* share some core ideas but they emphasize different processes and purposes for creating categories, so they are a useful distinction.[5] Cultural categorization can be understood as a natural human cognitive ability that serves as a foundation for both informal and formal organizing systems. Individual categorization tends to grow spontaneously out of our personal activities. Institutional categorization responds to the need for formal coordination and cooperation within and between companies, governments, and other goal-oriented enterprises.

5. These three contexts were proposed by (Glushko, Maglio, Matlock, and Barsalou 2008), who pointed out that cognitive science has focused on cultural categorization and largely ignored individual and institutional contexts. They argue that taking a broader view of categorization highlights dimensions on which it varies that are not apparent when only cultural categories are considered. For example, institutional categories are usually designed and maintained using prescriptive methods that have no analogues with cultural categories. There is a difference between institutional categories created for people, and categories created in institutions by computers in the predictive analytics, data mining sense.

In contrast to these three categorization contexts in which categories are created by people, *computational* categories are created by computer programs for information retrieval, machine learning, predictive analytics, and other applications. Computational categories are similar to those created by people in some ways but differ substantially in other ways.

Cultural Categories

Cultural categories are the archetypical form of categories upon which individual and institutional categories are usually based. Cultural categories tend to describe our everyday experiences of the world and our accumulated cultural knowledge. Such categories describe objects, events, settings, internal experiences, physical orientation, relationships between entities, and many other aspects of human experience. Cultural categories are learned primarily, with little explicit instruction, through normal exposure of children with their caregivers; they are associated with language acquisition and language use within particular cultural contexts.

Two thousand years ago Plato wrote that living species could be identified by "carving nature at its joints," the natural boundaries or discontinuities between types of things where the differences are the largest or most salient. Plato's metaphor is intuitively appealing because we can easily come up with examples of perceptible properties or behaviors of physical things that go together that make some ways of categorizing them seem more natural than others.[6]

6. This quote comes from Plato's *Phaedrus* dialogue, written around 370 BCE. Contemporary philosophers and

Natural languages rely heavily on nouns to talk about categories of things because it is useful to have a shorthand way of referring to a set of properties that co-occur in predictable ways.[7] For example, in English (borrowed from Portuguese) we have a word for "banana" because a particular curved shape, greenish-yellow or yellow color, and a convenient size tend to co-occur in a familiar edible object, so it became useful to give it a name. The word "banana" brings together this configuration of highly interrelated perceptions into a

cognitive scientists commonly invoke it in discussions about whether "natural kinds" exist. . For example, see (Campbell, O'Rourke, and Slater 2011), and (Hutchins 2010), (Atran 1987), and others have argued that the existence of perceptual discontinuities is not sufficient to account for category formation. Instead, people assume that members of a biological category must have an essence of co-occurring properties and these guide people to focus on the salient differences, thereby creating categories. Property clusters enable inferences about causality, which then builds a framework on which additional categories can be created and refined. For example, if "having wings" and "flying" are co-occurring properties that suggest a "bird" category, wings are then inferred as the causal basis of flying, and wings become more salient.

7. Pronouns, adjectives, verbs, adverbs, prepositions, conjunctions, particles, and numerals and other "parts of speech" are also grammatical categories, but nouns carry most of the semantic weight.

unified concept so we do not have to refer to bananas by listing their properties.[8]

Languages differ a great deal in the words they contain and also in more fundamental ways that they require speakers or writers to attend to details about the world or aspects of experience that another language allows them to ignore. This idea is often described as *linguistic relativity*. (See the sidebar, Linguistic Relativity.)

8. In contrast, the set of possible interactions with even a simple object like a banana is very large. We can pick, peel, slice, smash, eat, or throw a banana, so instead of capturing this complexity in the meaning of banana it gets parceled into the verbs that can act on the banana noun. Doing so requires languages to use verbs to capture a broader and more abstract type of meaning that is determined by the nouns with which they are combined. Familiar verbs like "set," "put," and "get" have dozens of different senses as a result because they go with so many different nouns. We set fires and we set tables, but fires and tables have little in common. The intangible character of verbs and the complexity of multiple meanings make it easier to focus instead on their associated nouns, which are often physical resources, and create organizing systems that emphasize the latter rather than the former. We create organizing systems that focus on verbs when we are categorizing actions, behaviors, or services where the resources that are involved are less visible or less directly involved in the supported interactions.

Linguistic Relativity

Linguistic diversity led Benjamin Whorf, in the mid-20th century, to propose an overly strong statement of the relationships among language, culture, and thought. Whorf argued that the particularities of one's native language determine how we think and what we can think about. Among his extreme ideas was the suggestion that, because some Native American languages lacked words or grammatical forms that refer to what we call "time" in English, they could not understand the concept. More careful language study showed both parts of the claim to be completely false.

Nevertheless, even though academic linguists have discredited strong versions of Whorf's ideas, less deterministic versions of *linguistic relativity* have become influential and help us understand cultural categorization. The more moderate position was crisply characterized by Roman Jakobson, who said that "languages differ essentially in what they *must* convey and not in what they *may* convey." In English one can say "I spent yesterday with a neighbor." In languages with grammatical gender, one must choose a word that identifies the neighbor as male or female.[9]

9. Many languages have a system of grammatical gender in which all nouns must be identified as masculine or feminine using definite articles (*el* and *la* in Spanish, *le* and *la* in French, and so on) and corresponding pronouns. Languages also contrast in how they describe

> For example, speakers of the Australian aboriginal language, Guugu Yimithirr, do not use concepts of left and right, but rather use cardinal directions. Where in English we might say to a person facing north, "Take a step to your left," they would use their term for west. If the person faced south, we would change our instruction to "right," but they would still use their term for west. Imagine how difficult it would be for a speaker of Guugu Yimithirr and a speaker of English to collaborate in organizing a storage room or a closet.[10]

time, spatial relationships, and in which things are treated as countable objects (one ox, two oxen) as opposed to substances or mass nouns that do not have distinct singular and plural forms (like water or dirt). (Deutscher 2011) carefully reviews and discredits the strong Whorfian view and makes the case for a more nuanced perspective on linguistic relativity. He also reviews much of Lera Boroditsky's important work in this area. George Lakoff's book with the title *Women, Fire, and Dangerous Things* (Lakoff 1990) provocatively points out differences in gender rules among languages; in an aboriginal language called Dyirbal many dangerous things, including fire have feminine gender, meanwhile "fire" is masculine in Spanish (*el feugo*) and French (*le feu*).

10. This analysis comes from (Haviland 1998). More recently, Lera Boroditsky has done many interesting studies and experiments about linguistic relativity. See (Boroditksy 2003) for an academic summary and (Boroditsky 2010, 2011) for more popular treatments.

It is not controversial to notice that different cultures and language communities have different experiences and activities that give them contrasting knowledge about particular domains. No one would doubt that university undergraduates in Chicago would think differently about animals than inhabitants of Guatemalan rain forests, or even that different types of "tree experts" (taxonomists, landscape workers, foresters, and tree maintenance personnel) would categorize trees differently.[11]

On the other hand, despite the wide variation in the climates, environments, and cultures that produce them, at a high level "folk taxonomies" that describe natural phenomena are surprisingly consistent around the world. Half a century ago the sociologists Emile Durkheim and Marcel Mauss observed that the language and structure of folk taxonomies mirrors that of human family relationships (e.g., different types of trees might be "siblings," but animals would be part of another family entirely). They suggested that framing the world in terms of familiar human relationships allowed people to understand it more easily.[12]

11. (Medin et al. 1997).

12. This was ultimately reflected in complex mythological systems, such as Greek mythology, where genealogical relationships between gods represented category relationships among the phenomena with which they were associated. As human knowledge grew and the taxonomies became more comprehensive and complex, Durkheim and Mauss argued, they lay the groundwork for scientific classifications and shed their mythological roots. (Durkheim 1963).

Anthropologist Brent Berlin, a more recent researcher, concurs with Durkheim and Mauss's observation that kinship relations and folk taxonomies are related, but argues that humans patterned their family structures after the natural world, not the other way around.[13]

Invoking the Whorfian Hypothesis in a Clothing Ad

An advertisement for the "66 North" clothing brand invokes the Whorfian hypothesis to suggest that even though Icelanders have more than a hundred words for snow there is only one kind of winter clothing that matters to them; the kind that carries this brand name.

(Photo by R. Glushko. Taken in the Reykjavik airport.)

13. (Berlin 2014)

Individual Categories

Individual categories are created in an organizing system to satisfy the *ad hoc* requirements that arise from a person's unique experiences, preferences, and resource collections. Unlike cultural categories, which usually develop slowly and last a long time, individual categories are created by intentional activity, in response to a specific situation, or to solve an emerging organizational challenge. As a consequence, the categories in individual organizing systems generally have short lifetimes and rarely outlive the person who created them.[14]

Individual categories draw from cultural categories but differ in two important ways. First, individual categories sometimes have an imaginative or metaphorical basis that is meaningful to the person who created them but which might distort or misinterpret cultural categories. Second, individual categories are often specialized or synthesized versions of cultural categories that capture particular experiences or personal history. For example, a person who has lived in China and Mexico, or lived with people from those places, might have highly individualized categories for foods they like and dislike that incorporate characteristics of both Chinese and Mexican cuisine.

Individual categories in organizing systems also reflect the idiosyncratic set of household goods, music, books, website bookmarks, or other resources that a person might have collected

14. The personal archives of people who turn out to be famous or important are the exception that proves this rule. In that case, the individual's organizing system and its categories are preserved along with their contents.

over time. The organizing systems for financial records, personal papers, or email messages often use highly specialized categories that are shaped by specific tasks to be performed, relationships with other people, events of personal history, and other highly individualized considerations. Put another way, individual categories are used to organize resource collections that are likely not representative samples of all resources of the type being collected. If everyone had the same collection of music, books, clothes, or toys the world would be a boring place.

Traditionally, *individual categorization* systems were usually not visible to, or shared with, others, whereas, this has become an increasingly common situation for people using web-based organizing system for pictures, music, or other personal resources. On websites like the popular Flickr, Instagram, and YouTube sites for photos and videos, people typically use existing cultural categories to tag their content as well as individual ones that they invent.[15]

15. The typical syntactic constraint that tags are delimited by white space encourages the creation of new categories by combining existing category names using concatenation and camel case conventions; photos that could be categorized as "Berkeley" and "Student" are sometimes tagged as "BerkeleyStudent." Similar generative processes for creating individual category names are used with Twitter "hashtags" where tweets about events are often categorized with an ad hoc tag that combines an event name and a year identifier like "#NBAFinals16."

Institutional Categories

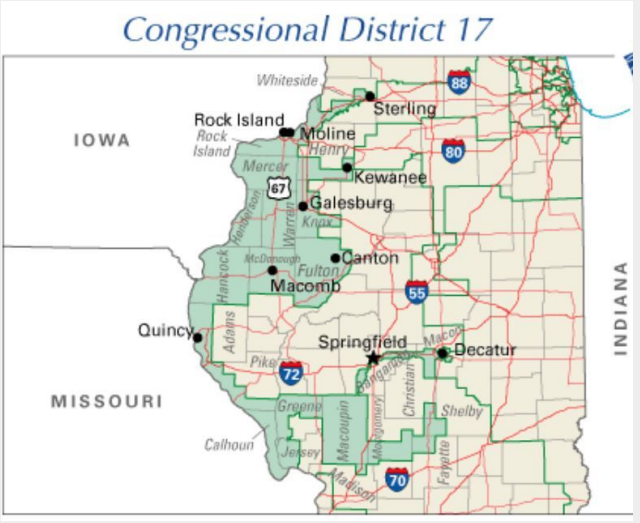
In contrast to cultural categories that are created and used implicitly, and to individual categories that are used by people acting alone, *institutional categories* are created and used explicitly, and most often by many people in coordination with each other. Institutional categories are most often created in abstract and information-intensive domains where unambiguous and precise categories are needed to regulate and systematize activity, to enable information sharing and reuse, and to reduce transaction costs. Furthermore, instead of describing the world as it is, institutional categories are usually defined to change or control the world by imposing semantic models that are more formal and arbitrary than those in cultural categories. Laws, regulations, and standards often specify institutional categories, along with decision rules for assigning resources to new categories, and behavior rules that prescribe how people must interact with them. The rigorous definition of institutional categories enables *classification:* the systematic assignment of resources to categories in an organizing system.[16]

16. Consider how the cultural category of "killing a person" is refined by the legal system to distinguish manslaughter and different degrees of murder based on the amount of intentionality and planning involved (e.g., first and second degree murder) and the roles of people involved with the killing (accessory). In general, the purpose of laws is to replace coarse judgments of categorization based on overall similarity of facts with rule-based categorization based on specific dimensions or properties.

Creating institutional categories by more systematic processes than cultural or individual categories does not ensure that they will be used in systematic and rational ways, because the reasoning and rationale behind institutional categories might be unknown to, or ignored by, the people who use them. Likewise, this way of creating categories does not prevent them from being biased. Indeed, the goal of institutional categories is often to impose or incentivize biases in interpretation or behavior. There is no better example of this than the practice of gerrymandering, designing the boundaries of election districts to give one political party or ethnic group an advantage.[17] (See the sidebar, Gerrymandering in Illinois.)

Gerrymandering in Illinois

17. The word was invented in 1812 in a newspaper article critical of Massachusetts governor Elbridge Gerry, who oversaw the creation of biased electoral districts. One such district was so contorted in shape, it was said to look like a salamander, and thus was called a Gerrymander. The practice remains widespread, but nowadays sophisticated computer programs can select voters on any number of characteristics and create boundaries that either "pack" them into a single district to concentrate their voting power or "crack" them into multiple districts to dilute it.



The 17th Congressional District in Illinois was dubbed "the rabbit on a skateboard" from 2003 through 2013 because of its highly contorted shape. The bizarre boundary was negotiated to create favorable voting constituencies for two incumbent legislators from opposing parties.

(Picture from nationatlas.gov. Not protectable by copyright (17 USC Sec. 105).)

Institutional categorization stands apart from individual categorization primarily because it invariably requires significant efforts to reconcile mismatches between existing individual categories, where those categories embody useful working or

contextual knowledge that is lost in the move to a formal institutional system.[18]

Institutional categorization efforts must also overcome the vagueness and inconsistency of cultural categories because the former must often conform to stricter logical standards to support inference and meet legal requirements. Furthermore, institutional categorization is usually a process that must be accounted for in a budget and staffing plans. While some kinds of institutional categories can be devised or discovered by computational processes, most of them are created through the collaboration of many individuals, typically from various parts of an organization or from different firms. For example, with the gerrymandering case we just discussed, it is important to emphasize that the inputs to these programs and the decisions about districting are controlled

18. The particularities or idiosyncrasies of individual categorization systems sometimes capture user expertise and knowledge that is not represented in the institutional categories that replace them. Many of the readers of this book are information professionals whose technological competence is central to their work and which helps them to be creative. But for a great many other people, information technology has enabled the routinization of work in offices, assembly lines, and in other jobs where new institutionalized job categories have "downskilled" or "deskilled" the nature of work, destroying competence and engendering a great deal of resistance from the affected workers.

by people, which is why the districts are institutional categories; the programs are simply tools that make the process more efficient. [19]

> Stop and Think: Color
>
> Think of the very broad category of "color." What are a few examples of a "cultural" category of color? How about an "individual" one? And an "institutional" one?

The different business or technical perspectives of the participants are often the essential ingredients in developing robust categories that can meet carefully identified requirements. And as requirements change over time, institutional categories must often

19. Similar technical concerns arise in within-company and multi-company standardization efforts, but the competitive and potentially anti-competitive character of the latter imposes greater complexity by introducing considerations of business strategy and politics. Credible standards-making in multi-company contexts depends on an explicit and transparent process for gathering and prioritizing requirements, negotiating specifications that satisfy them, and ensuring conformant implementations—without at any point giving any participating firm an advantage. See the OASIS Technical Committee Process for an example (https://www.oasis-open.org/policies-guidelines/tc-process) and (Rosenthal et al. 2004) for an analysis of best practices.

change as well, implying version control, compliance testing, and other formal maintenance and governance processes.

Some institutional categories that initially had narrow or focused applicability have found their way into more popular use and are now considered cultural categories. A good example is the periodic table in chemistry, which Mendeleev developed in 1869 as a new system of categories for the chemical elements. The periodic table proved essential to scientists in understanding their properties and in predicting undiscovered ones. Today the periodic table is taught in elementary schools, and many things other than elements are commonly arranged using a graphical structure that resembles the periodic table of elements in chemistry, including sci-fi films and movies, desserts, and superheroes.[20]

A "Categorization Continuum"

As we have seen, the concepts of cultural, individual, and institutional categorization usefully distinguish the primary processes and purposes when people create categories. However, these three kinds of categories can fuse, clash, and recombine with

20. Unfortunately, in this transition from science to popular culture, many of these so-called periodic tables are just *ad hoc* collections that ignore the essential idea that the rows and columns capture explanatory principles about resource properties that vary in a periodic manner. A notable exception is Andrew Plotkin's Periodic Table of Dessert. See (Suehle 2012) and Plotkin's table at (Periodic Table of Dessert).

each other. Rather than viewing them as having precise boundaries, we might view them as regions on a continuum of categorization activities and methods.

Consider a few different perspectives on categorizing animals as an example. Scientific institutions categorize animals according to explicit, principled classification systems, such as the Linnaean taxonomy that assigns animals to a phylum, class, order, family, genus and species. Cultural categorization practices cannot be adequately described in terms of a master taxonomy, and are more fluid, converging with principled taxonomies sometimes, and diverging at other times. While human beings are classified within the animal kingdom in biological classification systems, people are usually not considered animals in most cultural contexts. Sometimes a scientific designation for human beings, *homo sapiens* is even applied to human beings in cultural contexts, since the genus-species taxonomic designation has influenced cultural conceptions of people and (other) animals over the years.

Animals are also often culturally categorized as pets or non-pets. The category "pets" commonly includes dogs, cats, and fish. A pet cat might be categorized at multiple levels that incorporate individual, cultural, and institutional perspectives on categorization—as an "animal" (cultural/institutional), as a "mammal" (institutional), as a "domestic short-hair" (institutional) as a "cat" (cultural), and as a "troublemaker" or a "favorite" (individual), among other possibilities, in addition to being identified individually by one or more pet names. Furthermore, not everyone experiences pets as just dogs, cats and fish. Some people have relatively unusual pets, like pigs. For individuals who have pet pigs or who know people with pet pigs, "pigs" may be included in the "pets" category. If enough people have pet pigs, eventually "pigs" could be included in mainstream culture's pet category.

Categorization skewed toward cultural perspectives incorporate relatively traditional categories, such as those learned implicitly

from social interactions, like mainstream understandings of what kinds of animals are "pets," while categorization skewed toward institutional perspectives emphasizes explicit, formal categories, like the categories employed in biological classification systems.

CAFE Standards: Blurring the Lines Between Categorization Perspectives

The *Corporate Average Fuel Economy*(CAFE) standards sort vehicles into "passenger car" and "light truck" categories and impose higher minimum fuel efficiency requirements for cars because trucks have different typical uses.

When CAFE standards were introduced, the vehicles classified as light trucks were generally used for "light duty" farming and manufacturing purposes. "Light trucks" might be thought of as a "sort of" in-between category—a light truck is not really a car, but sufficiently unlike a prototypical truck to qualify the vehicle's categorization as "light." Formalizing this sense of in-between-ness by specifying features that define a "car" and a "light truck" is the only way to implement a consistent, transparent fuel efficiency policy that makes use of informal, graded distinctions between vehicles.

A manufacturer whose average fuel economy for all the vehicles it sells in a year falls below the CAFE standards has to pay penalties. This encourages them to produce "sport utility vehicles" (SUVs) that adhere to the CAFE definitions of light trucks but which most people use as passenger cars. Similarly, the PT Cruiser, a retro-styled hatchback produced by Chrysler from 2000-2010, strikes many people as a car. It looks like a

car; we associate it with the transport of passengers rather than with farming; and in fact it is formally classified as a car under emissions standards. But like SUVs, in the CAFE classification system, the PT Cruiser is a light truck.

CAFE standards have evolved over time, becoming a theater for political clashes between holistic cultural categories and formal institutional categories, which plays out in competing pressures from industry, government, and political organizations. Furthermore, CAFE standards and manufacturers' response to them are influencing cultural categories, such that our cultural understanding of what a car looks like is changing over time as manufacturers design vehicles like the PT Cruiser with car functionality in unconventional shapes to take advantage of the CAFE light truck specifications.[21]

21. The Corporate Average Fuel Economy(CAFE) standards have been developed by the United States National Highway Traffic Safety Administration (http://www.nhtsa.gov/fuel-economy) since 1975. For a careful and critical assessment of CAFE, including the politics of categorization for vehicles like the PT Cruiser, see the 2002 report from the Committee on the Effectiveness and Impact of Corporate Average Fuel Economy (CAFE) Standards, National Research Council.

Computational Categories

Computational categories are created by computer programs when the number of resources, or when the number of descriptions or observations associated with each resource, are so large that people cannot think about them effectively. Computational categories are created for information retrieval, predictive analytics, and other applications where information scale or speed requirements are critical. The resulting categories are similar to those created by people in some ways but differ substantially in other ways.

The simplest kind of computational categories can be created using descriptive statistics (see "Organizing With Descriptive Statistics"). Descriptive statistics do not identify the categories they create by giving them familiar cultural or institutional labels. Instead, they create implicit categories of items according to how much they differ from the most typical or frequent ones. For example, in any dataset where the values follow the normal distribution, statistics of central tendency and dispersion serve as standard reference measures for any observation. These statistics identify categories of items that are very different or statistically unlikely outliers, which could be signals of measurement errors, poorly calibrated equipment, employees who are inadequately trained or committing fraud, or other problems. The "Six Sigma" methodology for process improvement and quality control rests on this idea that careful and consistent collection of statistics can make any measurable operation better.

Many text processing methods and applications use simple statistics to categorize words by their frequency in a language, in a collection of documents, or in individual documents, and these categories are exploited in many information retrieval applications (see "Interactions Based on Instance Properties" and "Interactions Based on Collection Properties").

Supervised and Unsupervised Learning

Two subfields of *machine learning* that are relevant to organizing systems are *supervised* and *unsupervised* learning. In *supervised learning*, a machine learning program is trained with sample items or documents that are labeled by category, and the program learns to assign new items to the correct categories. In *unsupervised learning*, the program gets the same items but has to come up with the categories on its own by discovering the underlying correlations between the items; that is why unsupervised learning is sometimes called *statistical pattern recognition*.

Categories that people create and label also can be used more explicitly in computational algorithms and applications. In particular, a program that can assign an item or instance to one or more existing categories is called a classifier. The subfield of computer science known as *machine learning* is home to numerous techniques for creating classifiers by training them with already correctly categorized examples. This training is called *supervised learning*; it is supervised because it starts with instances labeled by category, and it involves learning because over time the classifier improves its performance by adjusting the weights for features that distinguish the categories. But strictly speaking, supervised learning techniques do not learn the categories; they implement and apply categories that they inherit or are given to them. We will further discuss the computational implementation of categories created by people in "Implementing Categories".

In contrast, many computational techniques in machine learning can analyze a collection of resources to discover statistical regularities or correlations among the items, creating a set of

categories without any labeled training data. This is called *unsupervised learning* or *statistical pattern recognition*. As we pointed out in "Cultural Categories", we learn most of our cultural categories without any explicit instruction about them, so it is not surprising that computational models of categorization developed by cognitive scientists often employ unsupervised statistical learning methods.

Many computational categories are like individual categories because they are tied to specific collections of resources or data and are designed to satisfy narrow goals. The individual categories you use to organize your email inbox or the files on your computer reflect your specific interests, activities, and personal network and are surely different than those of anyone else. Similarly, your credit card company analyzes your specific transactions to create computational categories of "likely good" and "likely fraudulent" that are different for every cardholder.

This focused scope is obvious when we consider how we might describe a computational category. "Fraudulent transaction for cardholder 4264123456780123" is not lexicalized with a one-word label as familiar cultural categories are. "Door" and "window" have broad scopes that are not tied to a single purpose. Put another way, the "door" and "window" cultural categories are highly reusable, as are institutional categories like those used to collect economic or health data that can be analyzed for many different purposes. The definitions of "door" and "window" might be a little fuzzy, but institutional categories are more precisely defined, often by law or regulation. Examples are the *North American Industry Classification System*(NAICS) from the US Census Bureau and the *United Nations Standard Products and Services Code*(UNSPC).

A final contrast between categories created by people and those created computationally is that the former can almost always be inspected and reasoned about by other people, but only some of the latter can. A computational model that categorizes loan applicants

as good or poor credit risks probably uses properties like age, income, home address, and marital status, so that a banker can understand and explain a credit decision. However, many other computational categories, especially those that created by clustering and deep learning techniques, are inseparable from the mathematical model that learned to use them, and as a result are uninterpretable by people.

A machine learning algorithm for classifying objects in images creates a complex multi-layer neural network whose features have no clear relationship to the categories, and this network has no other use. Put another way, machine learning programs are very general because they can be employed in any domain with high dimensional data, but what they learn cannot be applied in any other domain.

47. Principles for Creating Categories

"The What and Why of Categories" explained what categories are and the contrasting cultural, individual, and institutional contexts and purposes for which categories are created. In doing so, a number of different principles for creating categories were mentioned, mostly in passing.

We now take a systematic look at principles for creating categories, including enumeration, single properties, multiple properties and hierarchy, probabilistic, similarity, and theory- and goal-based categorization. These ways of creating categories differ in the information and mechanisms they use to determine category membership.

Enumeration

The simplest principle for creating a category is *enumeration*; any resource in a finite or countable set can be deemed a category member by that fact alone. This principle is also known as *extensional definition*, and the members of the set are called the *extension*. Many institutional categories are defined by enumeration as a set of possible or legal values, like the 50 United States or the ISO currency codes (ISO 4217).

Enumerative categories enable membership to be unambiguously determined because a value like a state name or currency code is either a member of the category or it is not. However, this clarity has a downside; it makes it hard to argue that something not explicitly mentioned in an enumeration should be considered a

member of the category, which can make laws or regulations inflexible. Moreover, there comes a size when enumerative definition is impractical or inefficient, and the category either must be sub-divided or be given a definition based on principles other than enumeration.[1]

> Too Many Planets to Enumerate:
> Keeping up with Kepler
>
> Kepler is a space observatory launched by NASA in 2009 to search for Earth-like planets orbiting other stars in our own Milky Way galaxy. Kepler has already discovered and verified a few thousand new planets, and these results have led to estimates that there may be at least as many planets as there are stars, a few hundred billion in the Milky Way alone. Count fast.

For example, for millennia we earthlings have had a cultural

1. Legal disputes often reflect different interpretations of category membership and whether a list of category members is exhaustive or merely illustrative. The legal principle of "implied exclusion"—*expressio unius est exclusio alterius* —says that if you "expressly name" or "designate" an enumeration of one or more things, any thing that is not named is excluded, by implication. However, prefacing the list with "such as," "including," or "like" implies that it is not a strict enumeration because there might be other members.

category of "planet" as a "wandering" celestial object, and because we only knew of planets in our own solar system, the planet category was defined by enumeration: Mercury, Venus, Earth, Mars, Jupiter, and Saturn. When the outer planets of Uranus, Neptune, and Pluto were identified as planets in the 18^{th}-20^{th} centuries, they were added to this list of planets without any changes in the cultural category. But in the last couple of decades many heretofore unknown planets outside our solar system have been detected, making the set of planets unbounded, and definition by enumeration no longer works.

The *International Astronomical Union(IAU)* thought it solved this category crisis by proposing a definition of planet as "a celestial body that is (a) in orbit around a star, (b) has sufficient mass for its self-gravity to overcome rigid body forces so that it assumes a hydrostatic equilibrium (nearly round) shape, and (c) has cleared the neighborhood around its orbit." Unfortunately, Pluto does not satisfy the third requirement, so it no longer is a member of the planet category, and instead is now called an "inferior planet."

Changing the definition of a significant cultural category generated a great deal of controversy and angst among ordinary non-scientific people. A typical headline was "Pluto's demotion has schools spinning," describing the outcry from elementary school students and teachers about the injustice done to Pluto and the disruption on the curriculum. [2]

2. International Astronomical Union(IAU) (iau.org) published its new definition of planet in August 2006. A public television documentary in 2011 called The Pluto Files retells the story (Tyson 2011).

Single Properties

It is intuitive and useful to think in terms of properties when we identify instances and when we are describing instances (as we saw in "Resource Identity" and in *Resource Description and Metadata*). Therefore, it should also be intuitive and useful to consider properties when we analyze more than one instance to compare and contrast them so we can determine which sets of instances can be treated as a category or *equivalence class*. Categories whose members are determined by one or more properties or rules follow the principle of *intensional definition*, and the defining properties are called the *intension*.

You might be thinking here that enumeration or extensional definition of a category is also a property test; is not "being a state" a property of California? But statehood is not a property precisely because "state" is defined by extension, which means the only way to test California for statehood is to see if it is in the list of states.[3]

Any *single property* of a resource can be used to create categories, and the easiest ones to use are often the intrinsic static properties. As we discussed in *Resource Description and Metadata*, intrinsic static properties are those inherent in a resource that never change. The material of composition of natural or manufactured objects is an intrinsic and static property that can be used to arrange physical resources. For example, an organizing system for a personal collection of music that is based on the intrinsic static property of

3. The distinction between intension and extension was introduced by Gottlob Frege, a German philosopher and mathematician (Frege 1892).

physical format might use categories for CDs, DVDs, vinyl albums, 8-track cartridges, reel-to-reel tape, and tape cassettes.[4]

Using a single property is most natural to do when the properties can take on only a small set of discrete values like music formats, and especially when the property is closely related to how the resources are used, as they are with the music collection where each format requires different equipment to listen to the music. Each value then becomes a subcategory of the music category.

The author, date, and location where an intellectual resource was created cannot be directly perceived but they are also intrinsic static properties. The subject matter or purpose of a resource, its "what it is about" or "what it was originally for," are also intrinsic static properties that are not directly perceivable, especially for information resources.

The name or identifier of a resource is often arbitrary but once assigned normally does not change, making it an extrinsic static

4. The number of resources in each of these categories depends on the age of the collection and the collector. We could be more precise here and say "single atomic property" or otherwise more carefully define "property" in this context as a characteristic that is basic and not easily or naturally decomposable into other characteristics. It would be possible to analyze the physical format of a music resource as a composition of size, shape, weight, and material substance properties, but that is not how people normally think. Instead, they treat physical format as a single property as we do in this example.

property. Any collection of resources with alphabetic or numeric identifiers as an associated property can use sorting order as an organizing principle to arrange spices, books, personnel records, etc., in a completely reliable way. Some might argue whether this organizing principle creates a category system, or whether it simply exploits the ordering inherent in the identifier notation. For example, with alphabetic identifiers, we can think of alphabetic ordering as creating a recursive category system with 26 (A-Z) top-level categories, each containing the same number of second-level categories, and so on until every instance is assigned to its proper place.[5]

Some resource properties are both extrinsic and dynamic because they are based on usage or behaviors that can be highly context-dependent. The current owner or location of a resource, its frequency of access, the joint frequency of access with other resources, or its current rating or preference with respect to alternative resources are typical extrinsic and dynamic properties that can be the basis for arranging resources and defining categories.

These properties can have a large number of values or are continuous measures, but as long as there are explicit rules for using property values to determine category assignment the resulting categories are still easy to understand and use. For example, we naturally categorize people we know on the basis of

5. We need to think of alphabetic ordering or any other organizing principle in a logical way that does not imply any particular physical implementation. Therefore, we do not need to consider which of these alphabetic categories exist as folders, files, or other tangible partitions.

their current profession, the city where they live, their hobbies, or their age. Properties with a numerical dimension like "frequency of use" are often transformed into a small set of categories like "frequently used," "occasionally used," and "rarely used" based on the numerical property values.[6]

There is an infinite number of logically expressible properties for any resource, but most of them would not lead to categories that would be interpretable and useful for people. If people are going to use the categories, it is important to base them on properties that are psychologically or pragmatically relevant for the resource domain being categorized. Whether something weighs more or less than 5000 pounds is a poor property to apply to things in general, because it puts cats and chairs in one category, and buses and elephants in another.[7]

To summarize: The most useful single properties to use for creating categories for an organizing system used by people are those that are formally assigned, objectively measurable and orderable, or tied

6. Another example: rules for mailing packages might use either size or weight to calculate the shipping cost, and whether these rules are based on specific numerical values or ranges of values, the intent seems to be to create categories of packages.

7. If you try hard, you can come up with situations in which this property is important, as when the circus is coming to the island on a ferry or when you are loading an elevator with a capacity limit of 5000 pounds, but it just is not a useful or psychologically salient property in most contexts.

to well-established cultural categories, because the resulting categories will be easier to understand and describe.

If only a single property is used to distinguish among some set of resources and to create the categories in an organizing system, the choice of property is critical because different properties often lead to different categories. Using the age property, Bill Gates and Mark Zuckerberg are unlikely to end up in the same category of people. Using the wealth property, they most certainly would. Furthermore, if only one property is used to create a system of categories, any category with a large numbers of items in it will lack coherence because differences on other properties will be too apparent, and some category members will not fit as well as the others.

Multiple Properties

Organizing systems often use multiple properties to define categories. There are three different ways in which to do this that differ in the scope of the properties and how essential they are in defining the categories.

Multi-Level or Hierarchical Categories

If you have many shirts in your closet (and you are a bit compulsive or a "neat freak"), instead of just separating your shirts from your pants using a single property (the part of body on which the clothes are worn) you might arrange the shirts by style, and then by sleeve length, and finally by color. When all of the resources in an organizing system are arranged using the same sequence of resource properties, this creates a *logical hierarchy*, a multi-level category system.

If we treat all the shirts as the collection being organized, in the

shirt organizing system the broad category of shirts is first divided by style into categories like "dress shirts," "work shirts," "party shirts," and "athletic or sweatshirts." Each of these style categories is further divided until the categories are very narrow ones, like the "white long-sleeve dress shirts" category. A particular shirt ends up in this last category only after passing a series of property tests along the way: it is a dress shirt, it has long sleeves, and it is white. Each test creates more precise categories in the intersections of the categories whose members passed the prior property tests.

Put another way, each subdivision of a category takes place when we identify or choose a property that differentiates the members of the category in a way that is important or useful for some intent or purpose. Shirts differ from pants in the value of the "part of body" property, and all the shirt subcategories share this "top part" value of that property. However, shirts differ on other properties that determine the subcategory to which they belong. Even as we pay attention to these differentiating properties, it is important to remember the other properties, the ones that members of a category at any level in the hierarchy have in common with the members of the categories that contain it. These properties are often described as "inherited" or "inferred" from the broader category.[8] For example, just as every shirt shares the "worn on top part of body" property, every item of clothing shares the "can be worn on the body" property, and every resource in the "shirts" and "pants" category inherits that property.

8. Many information systems, applications, and programming languages that work with hierarchical categories take advantage of this logical relationship to infer inherited properties when they are needed rather than storing them redundantly.

Each differentiating property creates another level in the category hierarchy, which raises an obvious question: How many properties and levels do we need? In order to answer this question, we must reflect upon the shirt categories in our closet. Our organizing system for shirts arranges them with the three properties of style, sleeve length, and color; some of the categories at the lowest level of the resulting hierarchy might have only one member, or no members at all. You might have yellow or red short-sleeved party shirts, but probably do not have yellow or red long-sleeved dress shirts, making them empty categories. Obviously, any category with only one member does not need any additional properties to tell the members apart, so a category hierarchy is logically complete if every resource is in a category by itself.

However, even when the lowest level categories of our shirt organizing system have more than one member, we might choose not to use additional properties to subdivide it because the differences that remain among the members do not matter to us for the interactions the organizing system needs to support. Suppose we have two long-sleeve white dress shirts from different shirt makers, but whenever we need to wear one of them, we ignore this property. Instead, we just pick one or the other, treating the shirts as completely equivalent or substitutable. When the remaining differences between members of a category do not make a difference to the users of the category, we can say that the organizing system is pragmatically or practically complete even if it is not yet logically complete. That is to say, it is complete "for all intents and purposes." Indeed, we might argue that it is desirable to stop subdividing a system of categories while there are some small differences remaining among the items in each category because this leaves some flexibility or logical space in which to organize new items. This point might remind you of the concept of overfitting, where models with many parameters can very accurately fit their training data, but as a result generalize less well to new data. (See "Resource Description for Sensemaking and Science".)

On the other hand, consider the shirt section of a big department store. Shirts there might be organized by style, sleeve length, and color as they are in our home closet, but would certainly be further organized by shirt maker and by size to enable a shopper to find a Marc Jacobs long-sleeve blue dress shirt of size 15/35. The department store organizing system needs more properties and a deeper hierarchy for the shirt domain because it has a much larger number of shirt instances to organize and because it needs to support many shirt shoppers, not just one person whose shirts are all the same size.

Classifying Hawaiian "Boardshorts"

The swimsuits worn by surfers, called "boardshorts," have evolved from purely functional garments to symbols of extreme sports and the Hawaiian lifestyle. A 2012 exhibition at the Honolulu Museum of Art captured the diversity of boardshorts on three facets: their

material, how they fastened around the surfer's fly and waist, and their length.

(Photo by R. Glushko.)

Different Properties for Subsets of Resources

A different way to use multiple resource properties to create categories in an organizing system is to employ different properties for distinct subsets of the resources being organized. This contrasts with the strict multi-level approach in which every resource is evaluated with respect to every property. Alternatively, we could view this principle as a way of organizing multiple domains that are conceptually or physically adjacent, each of which has a separate set of categories based on properties of the resources in that domain. This principle is used for most folder structures in computer file systems and by many email applications; you can create as many folder categories as you want, but any resource can only be placed in one folder.

The contrasts between intrinsic and extrinsic properties, and between static and dynamic ones, are helpful in explaining this method of creating organizing categories. For example, you might organize all of your clothes using intrinsic static properties if you keep your shirts, socks, and sweaters in different drawers and arrange them by color; extrinsic static properties if you share your front hall closet with a roommate, so you each use only one side of that closet space; intrinsic dynamic properties if you arrange your clothes for ready access according to the season; and, extrinsic

dynamic properties if you keep your most frequently used jacket and hat on a hook by the front door.[9]

If we relax the requirement that different subsets of resources use different organizing properties and allow any property to be used to describe any resource, the loose organizing principle we now have is often called *tagging*. Using any property of a resource to create a description is an uncontrolled and often unprincipled principle for creating categories, but it is increasingly popular for organizing photos, web sites, email messages in gmail, or other web-based resources. We discuss tagging in more detail in "Tagging of Web-based Resources".

A Supermarket Map

9. Similarly, clothing stores use intrinsic static properties when they present merchandise arranged according to color and size; extrinsic static properties when they host branded displays of merchandise; intrinsic dynamic properties when they set aside a display for seasonal merchandise, from bathing suits to winter boots; and extrinsic dynamic properties when a display area is set aside for "Today's Special."

A typical supermarket embodies a surprisingly complex classification system. Each section of the store employs a different set of properties to arrange its resources, and some properties such as perishability and onsite preparation are important in more than one section.

(Photo by R. Glushko.)

Necessary and Sufficient Properties

A large set of resources does not always require many properties and categories to organize it. Some types of categories can be defined precisely with just a few *essential* properties. For example, a prime number is a positive integer that has no divisors other than 1 and itself, and this category definition perfectly distinguishes prime and not-prime numbers no matter how many numbers are being

categorized. “Positive integer” and “divisible only by 1 and itself” are *necessary* or *defining* properties for the prime number category; every prime number must satisfy these properties. These properties are also *sufficient* to establish membership in the prime number category; any number that satisfies the necessary properties is a prime number. Categories defined by necessary and sufficient properties are also called *monothetic*. They are also sometimes called *classical categories* because they conform to Aristotle’s theory of how categories are used in logical deduction using syllogisms.[10] (See the sidebar, The Classical View of Categories.)

Theories of categorization have evolved a great deal since Plato and Aristotle proposed them over two thousand years ago, but in many ways we still adhere to classical views of categories when we create organizing systems because they can be easier to implement and maintain that way.

An important implication of necessary and sufficient category definition is that every member of the category is an equally good member or example of the category; every prime number is equally prime. Institutional category systems often employ necessary and sufficient properties for their conceptual simplicity and straightforward implementation in *decision trees*, database *schemas*,and programming language *classes*.

The Classical View of Categories

10. Aristotle did not call them classical categories. That label was bestowed about 2300 years later by (Smith and Medin 1981).

The classical view is that categories are defined by necessary and sufficient properties. This theory has been enormously influential in Western thought, and is embodied in many organizing systems, especially those for information resources. However, as we will explain, we cannot rely on this principle to create categories in many domains and contexts because there are not necessary and sufficient properties. As a result, many psychologists, cognitive scientists, and computer scientists who think about categorization have criticized the classical theory.

We think this is unfair to Aristotle, who proposed what we now call the classical theory primarily to explain how categories underlie the logic of deductive reasoning: All men are mortal; Socrates is a man; Therefore, Socrates is mortal. People are wrong to turn Aristotle's thinking around and apply it to the problem of inductive reasoning, how categories are created in the first place. But this is not Aristotle's fault; he was not trying to explain how natural cultural categories arise.

Consider the definition of an address as requiring a street, city, governmental region, and postal code. Anything that has all of these *information components* is therefore considered to be a valid address, and anything that lacks any of them will not be considered to be a valid address. If we refine the properties of an address to require the governmental region to be a state, and specifically one of the United States Postal Service's list of official state and territory codes, we create a subcategory for US addresses that uses an enumerated category as part of its definition. Similarly, we could create a subcategory for Canadian addresses by exchanging the

name "province" for state, and using an enumerated list of Canadian province and territory codes.

The Limits of Property-Based Categorization

Property-based categorization works tautologically well for categories like "prime number" where the category is defined by necessary and sufficient properties. Property-based categorization also works well when properties are conceptually distinct and the value of a property is easy to perceive and examine, as they are with man-made physical resources like shirts.

Historical experience with organizing systems that need to categorize information resources has shown that basing categories on easily perceived properties is often not effective. There might be indications "on the surface" that suggest the "joints" or boundaries between types of information resources, but these are often just presentation or packaging choices, That is to say, neither the size of a book nor the color of its cover are reliable cues for what it contains. Information resources have numerous descriptive properties like their title, author, and publisher that can be used more effectively to define categories, and these are certainly useful for some kinds of interactions, like finding all of the books written by a particular author or published by the same publisher. However, for practical purposes, the most useful property of an information resource is its *aboutness*, which may not be objectively perceivable and which is certainly hard to characterize.[11] Any collection of

11. We all use the word "about" with ease in ordinary discourse, but "aboutness" has generated a surprising amount of theoretical commentary about its typically

information resources in a library or document filing system is likely to be about many subjects and topics, and when an individual resource is categorized according to a limited number of its content properties, it is at the same time not being categorized using the others.

Classifying the Web: Yahoo! in 1996

implicit definition, starting with (Hutchins 1977) and (Maron 1977) and relentlessly continued by (Hjørland 1992, 2001).

Their goal was to manually assign every web page to a category.

(Screenshot by R. Glushko. Source: Internet Archive wayback machine.)

When the web first started, there were many attempts to create categories of web sites, most notably by Yahoo! As the web grew, it became obvious that search engines would be vastly more useful because their near real-time text indexes obviate the need for *a priori* assignment of web pages to categories. Rather, web search

engines represent each web page or document in a way that treats each word or term they contain as a separate property.

Considering every distinct word in a document stretches our notion of property to make it very different from the kinds of properties we have discussed so far, where properties were being explicitly used by people to make decisions about category membership and resource organization. It is just not possible for people to pay attention to more than a few properties at the same time even if they want to, because that is how human perceptual and cognitive machinery works. But computers have no such limitations, and algorithms for information retrieval and machine learning can use huge numbers of properties, as we will see later in this chapter and in *Classification: Assigning Resources to Categories* and *Interactions with Resources*.

Probabilistic Categories and "Family Resemblance"

As we have seen, some categories can be precisely defined using necessary and sufficient features, especially when the properties that determine category membership are easy to observe and evaluate. Something is either a prime number or it isn't. A person cannot be a registered student and not registered at the same time.

However, categorization based on explicit and logical consideration of properties is much less effective, and sometimes not even possible for domains where properties lack one or more of the characteristics of separability, perceptibility, and necessity. Instead, we need to categorize using properties in a probabilistic or statistical way to come up with some measure of resemblance or similarity between the resource to be categorized and the other members of the category.

Consider a familiar category like "bird." All birds have feathers, wings, beaks, and two legs. But there are thousands of types of birds, and they are distinguished by properties that some birds have that other birds lack: most birds can fly, most are active in the daytime, some swim, some swim underwater; some have webbed feet. These properties are correlated or clustered, a consequence of natural selection that conveys advantages to particular configurations of characteristics, and there are many different clusters; birds that live in trees have different wings and feet than those that swim, and birds that live in deserts have different colorations and metabolisms that those that live near water. So instead of being defined by a single set of properties that are both necessary and sufficient, the bird category is defined probabilistically, which means that decisions about category membership are made by accumulating evidence from the properties that are more or less characteristic of the category.

Categories of information resources often have the same probabilistic character. The category of spam messages is suggested by the presence of particular words (beneficiary, pharmaceutical) but these words also occur in messages that are not spam. A spam classifier uses the probabilities of each word in a message in spam and non-spam contexts to calculate an overall likelihood that the message is spam.

There are three related consequences for categories when their characteristic properties have a probabilistic distribution:

- The first is an effect of *typicality* or *centrality* that makes some members of the category better examples than others. Membership in probabilistic categories is not all or none, so even if they share many properties, an instance that has more of the characteristic properties will be judged as better or more

typical.[12] Try to define "bird" and then ask yourself if all of the things you classify as birds are equally good examples of the category (look at the six birds in Family Resemblance and Typicality). This effect is also described as *gradience* in category membership and reflects the extent to which the most characteristic properties are shared.

- A second consequence is that the sharing of some but not all properties creates what we call *family resemblances* among the category members; just as biological family members do not necessarily all share a single set of physical features but still are recognizable as members of the same family. This idea was first proposed by the 20th-century philosopher Ludwig Wittgenstein, who used "games" as an example of a category whose members resemble each other according to shifting property subsets.[13]

- The third consequence, when categories do not have necessary features for membership, is that the boundaries of the category are not fixed; the category can be stretched and new members assigned as long as they resemble incumbent members. Personal video games and multiplayer online games like World

12. Typicality and centrality effects were studied by Rosch and others in numerous highly influential experiments in the 1970s and 1980s (Rosch 1975). Good summaries can be found in (Mervis and Rosch 1981), (Rosch 1999), and in Chapter 1 of (Rogers and McClelland 2008).

13. An easy to find source for Wittgenstein's discussion of "game" is (Wittgenstein 2002) in a collection of core readings for cognitive psychology (Levitin 2002).

of Warcraft did not exist in Wittgenstein's time but we have no trouble recognizing them as games and neither would Wittgenstein, were he alive. Recall that in *Foundations for Organizing Systems* we pointed out that the cultural category of "library" has been repeatedly extended by new properties, as when Flickr is described as a web-based photo-sharing library. Categories defined by family resemblance or multiple and shifting property sets are termed *polythetic*.

What Is a Game?

Ludwig Wittgenstein (1889-1951) was a philosopher who thought deeply about mathematics, the mind, and language. In 1999, his *Philosophical Investigations* was ranked as the most important book of 20th-century philosophy in a poll of philosophers.[14] In that book, Wittgenstein uses "game" to argue that many concepts have no defining properties, and that instead there is a "complicated network of similarities overlapping and criss-crossing: sometimes overall similarities, sometimes similarities of detail." He contrasts board games, card games, ball games, games of skill, games of luck, games with competition, solitary games, and games for amusement. Wittgenstein notes that not all games are equally good examples of the category, and jokes about teaching children a gambling game with dice because he knows that this is not the kind of game that

14. The philosopher's poll that ranked Wittgenstein's book #1 is reported by (Lackey 1999).

the parents were thinking of when they asked him to teach their children a game.[15]

We conclude that instead of using properties one at a time to assign category membership, we can use them in a composite or integrated way where together a co-occurring cluster of properties provides evidence that contributes to a *similarity* calculation. Something is categorized as an A and not a B if it is more similar to A's best or most typical member rather than it is to B's.[16]

15. It might be possible to define "game,"but it requires a great deal of abstraction that obscures the "necessary and sufficient" tests. "To play a game is to engage in activity directed toward bringing about a specific state of affairs, using only means permitted by specific rules, where the means permitted by the rules are more limited in scope than they would be in the absence of the rules, and where the sole reason for accepting such limitation is to make possible such activity." (Suits 1967)

16. The exact nature of the category representation to which the similarity comparison is made is a subject of ongoing debate in cognitive science. Is it a *prototype*, a central tendency or average of the properties shared by category members, or it one or more *exemplars*, particular members that typify the category. Or is it neither, as argued by connectionist modelers who view categories

Family Resemblance and Typicality

These six animals have some physical features in common but not all of them, yet they resemble each other enough to be easily recognizable as birds. Most people consider a pigeon to be a more typical bird than a penguin.

A penguin, a pigeon, a swan, a stork, a flamingo, and a frigate bird. (Clockwise from top-left.)

(Photos by R. Glushko.)

as patterns of network activation without any explicitly stored category representation? Fortunately, these distinctions do not matter for our discussion here. A recent review is (Rips, Smith, and Medin 2012).

Similarity

Similarity is a measure of the resemblance between two things that share some characteristics but are not identical. It is a very flexible notion whose meaning depends on the domain within which we apply it. Some people consider that the concept of similarity is itself meaningless because there must always be some basis, some unstated set of properties, for determining whether two things are similar. If we could identify those properties and how they are used, there would not be any work for a similarity mechanism to do.[17]

To make similarity a useful mechanism for categorization we have to specify how the similarity measure is determined. There are four psychologically-motivated approaches that propose different functions for computing similarity: feature- or property-based, geometry-based, transformational, and alignment- or analogy-based. The big contrast here is between models that represent items as sets of properties or discrete conceptual features, and

17. Another situation where similarity has been described as a "mostly vacuous" explanation for categorization is with abstract categories or metaphors. Goldstone says "an unrewarding job and a relationship that cannot be ended may both be metaphorical prisons... and may seem similar in that both conjure up a feeling of being trapped... but this feature is almost as abstract as the category to be explained." (Goldstone 1994), p. 149.

those that assume that properties vary on a continuous metric space.[18]

Feature-based Models of Similarity

An influential model of feature-based similarity calculation is Amos Tversky's contrast model, which matches the features or properties of two things and computes a similarity measure according to three sets of features:

- those features they share,
- those features that the first has that the second lacks, and
- those features that the second has that the first lacks.

The similarity based on the shared features is reduced by the two sets of distinctive ones. The weights assigned to each set can be adjusted to explain judgments of category membership. Another commonly feature-based similarity measure is the Jaccard coefficient, the ratio of the common features to the total number of them. This simple calculation equals zero if there are no overlapping features and one if all features overlap. Jaccard's measure is often used to calculate document similarity by treating each word as a feature.[19]

18. (Medin, Goldstone, and Gentner 1993) and (Tenenbaum and Griffiths 2001).

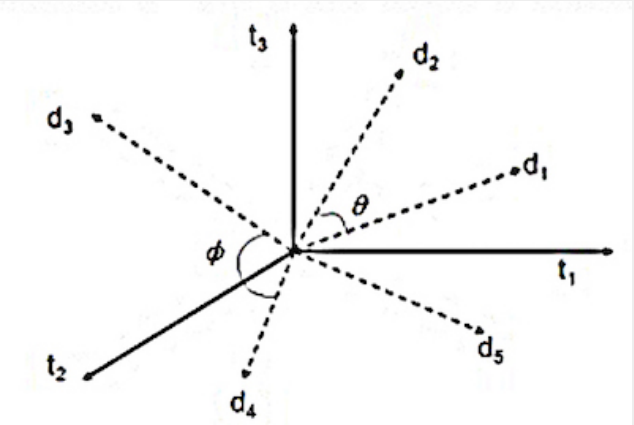
19. Because Tversky's model separately considers the sets of non-overlapping features, it is possible to accurately capture similarity judgments when they are not

We often use a heuristic version of feature-based similarity calculation when we create multi-level or hierarchical category systems to ensure that the categories at each level are at the same level of abstraction or breadth. For example, if we were organizing a collection of musical instruments, it would not seem correct to have subcategories of "woodwind instruments," "violins," and "cellos" because the feature-based similarity among the categories is not the same for all pairwise comparisons among the categories; violins and cellos are simply too similar to each other to be separate categories given woodwinds as a category.

Geometric Models of Similarity

Geometric models are a type of similarity framework in which items whose property values are metric are represented as points in a multi-dimensional feature- or property-space. The property values are the coordinates, and similarity is calculated by measuring the distance between the items.

symmetric, i.e., when A is judged more similar to B than B is to A. This framing effect is well-established in the psychological literature and many machine learning algorithms now employ asymmetric measures. (Tversky 1974)



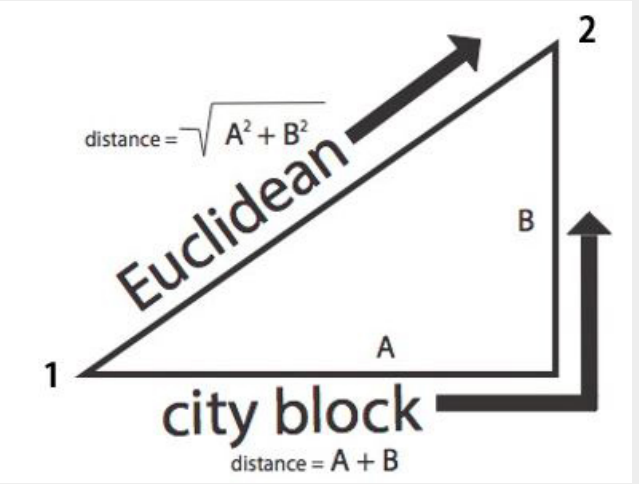
Documents represented as vectors in term space, with the angles between them as a measure of their similarity.

Geometric similarity functions are commonly used by search engines; if a query and document are each represented as a vector of search terms, relevance is determined by the distance between the vectors in the "term space." The simplified diagram in the sidebar, Document Similarity, depicts four documents whose locations in the term space are determined by how many of each of three terms they contain. The document vectors are normalized to length 1, which makes it possible to use the cosine of the angle between any two documents as a measure of their similarity. Documents d1 and d2 are more similar to each other than documents d3 and d4, because angle between the former pair (Θ) is smaller than the angle between the latter (Φ). We will discuss how this works in greater detail in *Interactions with Resources*.

If the vectors that represent items in a multi-dimensional property

space are of different lengths, instead of calculating similarity using cosines we need to calculate similarity in a way that more explicitly considers the differences on each dimension.

Geometric Distance Functions



The distance between points 1 and 2 depends on how the distance function combines the differences in values (A and B) on each dimension.

The diagram in the sidebar, Geometric Distance Functions shows two different ways of calculating the distance between points 1 and 2 using the differences A and B. The Euclidean distance function takes the square root of the sum of the squared differences on each dimension; in two dimensions, this is the familiar Pythagorean Theorem to calculate the length of the hypotenuse of a right triangle, where the exponent applied to the differences is 2. In contrast, the City Block distance function, so-named because it is the natural way to measure distances in cities with "gridlike" street

plans, simply adds up the differences on each dimension, which is equivalent to an exponent of 1.

We can interpret the exponent as a weighting function that determines the relative contribution of each property to the overall distance or similarity calculation. The choice of exponent depends on the type of properties that characterize a domain and how people make category judgments within it. The exponent of 1 in the City Block function ensures that each property contributes its full amount. As the exponent grows larger, it magnifies the impact of the properties on which differences are the largest.

The Chebyshev function takes this to the limit (where the exponent would be infinity) and defines the distance between two items as the difference of their values on the single property with the greatest difference. What this means in practice is that two items could have similar or even identical values on most properties, but if they differ much on just one property, they will be treated as very dissimilar. We can make an analogy to stereotyping or prejudice when a person is just like you in all ways except for the one property you view as negative, which then becomes the only one that matters to you.

At the other extreme, if the exponent is reduced to zero, this treats each property as binary, either present or absent, and the distance function becomes a count of the number of times that the value of the property for one item is different from the value for the other one. This is called the "Hamming distance."

Transformational Models of Similarity

Transformational models assume that the similarity between two things is inversely proportional to the complexity of the transformation required to turn one into the other. The simplest transformational model of similarity counts the number of

properties that would need to change their values. More generally, one way to perform the *name matching* task of determining when two different strings denote the same person, object, or other named entity is to calculate the "edit distance" between them; the number of changes required to transform one into the other.

The simplest calculation just counts the number of insertion, deletion, and substitution operations and is called the Levenshtein distance; for example, the distance between "bob" and "book" is two: insert "o" and change the second "b" to "k". Two strings with a short edit distance might be variant spellings or misspellings of the same name, and transformational models that are sensitive to common typing errors like transposed or duplicated letters are very effective at spelling correction. Transformational models of similarity are also commonly used to detect plagiarism and duplicate web pages.[20]

20. For a detailed explanation of distance and transformational models of similarity, see (Flach 2012), Chapter 9. There are many online calculators for Levenshein distance; http://www.let.rug.nl/kleiweg/lev/ also has a compelling visualization. The "strings" to be matched can themselves be transformations. The "soundex" function is very commonly used to determine if two words could be different spellings of the same name. It "hashes" the names into phonetic encodings that have fewer characters than the text versions. See (Christen 2006) and http://www.searchforancestors.com/utility/soundex.html to try it yourself.

Alignment or Analogy Models of Similarity

None of the previous types of similarity models works very well when comparing things that have lots of internal or relational structure. In these cases, calculations based on matching features is insufficient; you need to compare features that align because they have the same role in structures or relationships. For example, a car with a green wheel and a truck with a green hood both share the feature green, but this matching feature does not increase their similarity much because the car's wheel does not align with the truck's hood. On the other hand, analogy lets us say that an atom is like the solar system. They have no common properties, but they share the relationship of having smaller objects revolving around a large one.

This kind of analogical comparison is especially important in problem solving. You might think that experts are good at solving problems in their domain of expertise because they have organized their knowledge and experience in ways that enable efficient search for and evaluation of possible solutions. For example, it is well known that chess masters search their memories of previous winning positions and the associated moves to decide what to play. However, top chess players also organize their knowledge and select moves on the basis of abstract similarities that cannot be explained in terms of specific positions of chess pieces. This idea that experts represent and solve problems at deeper levels than novices do by using more abstract principles or domain structure has been replicated in many areas. Novices tend to focus more on surface properties and rely more on literal similarity.[21]

21. This explanation for expert-novice differences in categorization and problem solving was proposed in (Chi

Goal-Derived Categories

Another psychological principle for creating categories is to organize resources that go together in order to satisfy a goal. Consider the category "Things to take from a burning house," an example that cognitive scientist Lawrence Barsalou termed an *ad hoc* or *goal-derived* category.[22]

Things Used at the Gym

A hand towel, a music player with headphones, and a

et al 1981). See (Linhares 2007) for studies of abstract reasoning by chess experts.

22. (Barsalou 1983).

bottle of water have no properties in common but they go together because they are members of the "things used at the gym when working out" category. This type of ad hoc or goal-derived category gave contestants trouble on the Pyramid game show.

(Photo by R. Glushko.)

What things would you take from your house if a fire threatened it?? Possibly your cat, your wallet and checkbook, important papers like birth certificates and passports, and grandma's old photo album, and anything else you think is important, priceless, or irreplaceable—as long as you can carry it. These items have no discernible properties in common, except for being your most precious possessions. The category is derived or induced by a particular goal in some specified context.

Theory-Based Categories

A final psychological principle for creating categories is organizing things in ways that fit a theory or story that makes a particular categorization sensible. A *theory-based category* can win out even if probabilistic categorization, on the basis of *family resemblance* or *similarity* with respect to visible properties, would lead to a different category assignment. For example, a theory of phase change explains why liquid water, ice, and steam are all the same chemical compound even though they share few visible properties.

Theory-based categories based on origin or causation are especially important with highly inventive and computational resources

because unlike natural kinds of physical resources, little or none of what they can do or how they behave is visible on the surface (see "Affordance and Capability"). Consider all of the different appearances and form factors of the resources that we categorize as "computers" —their essence is that they all compute, an invisible or theory-like principle that does not depend on their visible properties.[23]

23. The emergence of theory-based categorization is an important event in cognitive development that has been characterized as a shift from "holistic" to "analytic" categories or from "surface properties" to "principles." See (Carey and Gelman 1991) (Rehder and Hastie 2004).

48. Category Design Issues and Implications

We have previously discussed the most important principles for creating categories: resource properties, similarity, and goals. When we use one or more of these principles to develop a system of categories, we must make decisions about its depth and breadth. Here, we examine the idea that some levels of abstraction in a system of categories are more basic or natural than others. We also consider how the choices we make affect how we create the organizing system in the first place, and how they shape our interactions when we need to find some resources that are categorized in it.

Category Abstraction and Granularity

We can identify any resource as a unique instance or as a member of a class of resources. The size of this class—the number of resources that are treated as equivalent—is determined by the properties or characteristics we consider when we examine the resources in some domain. The way we think of a resource domain depends on context and intent, so the same resource can be thought of abstractly in some situations and very concretely in others. As we discussed in *Resource Description and Metadata*, this influences the nature and extent of resource description, and as we have seen in this chapter, it then influences the nature and extent of categories we can create.

Consider the regular chore of putting away clean clothes. We can consider any item of clothing as a member of a broad category whose members are any kind of garment that a person might wear.

Using one category for all clothing, that is, failing to distinguish among the various items in any useful or practical way would likely mean that we would keep our clothes in a big unorganized pile.

However, we cannot wear any random combination of clothing items—we need a shirt, a pair of pants, socks, and so on. Clearly, our indiscriminate clothing category is too broad for most purposes. So instead, most people organize their clothes in more fine-grained categories that fit the normal pattern of how they wear clothes.

This tendency to use specific categories instead of broader ones is a general principle that reflects how people organize their experience when they see similar, but not identical, examples or events. This "size principle" for concept learning, as cognitive scientist Josh Tenenbaum describes it, is a preference for the most specific rules or descriptions that fit the observations. For example, if you visit a zoo and see many different species of animals, your conception of what you saw is different than if you visited a kennel that only contained dogs. You might say "I saw animals at the zoo," but would be more likely to say "I saw dogs at the kennel" because using the broad "animal" category to describe your kennel visit conveys less of what you learned from your observations there.[1]

In "Single Properties" we described an organizing system for the shirts in our closet, so let us talk about socks instead. When it comes to socks, most people think that the basic unit is a pair because they always wear two socks at a time. If you are going to need to find socks in pairs, it seems sensible to organize them into pairs when you are putting them away. Some people might further separate their dress socks from athletic ones, and then sort these socks by

1. (Tenenbaum 2000) argues that this preference for the most specific hypothesis that fits the data is a general principle of Bayesian learning with random samples.

color or material, creating a hierarchy of sock categories analogous to the shirt categories in our previous example.

Questions of resource abstraction and granularity also emerge whenever the information systems of different firms, or different parts of a firm, need to exchange information or be merged into a single system. All parties must define the identity of each thing in the same way, or in ways that can be related or mapped to each other either manually or electronically.

For example, how should a business system deal with a customer's address? Printed on an envelope, "an address" typically appears as a comprehensive, multi-line text object. Inside an information system, however, an address is best stored as a set of distinctly identifiable information components. This fine-grained organization makes it easier to sort customers by city or postal codes, for sales and marketing purposes. Incompatibilities in the abstraction and granularity of these information components, and the ways in which they are presented and reused in documents, will cause interoperability problems when businesses need to share information.[2]

The *Universal Business Language*(*UBL*) (mentioned briefly in

2. Consider what happens if two businesses model the concept of "address" in a customer database with different granularity. One may have a coarse "Address" field in the database, which stores a street address, city, state, and Zip code all in one block, while the other stores the components "StreetAddress," "City," and "PostalCode" In separate fields. The more granular model can be automatically transformed into the less granular one, but not vice versa (Glushko and McGrath 2005).

"Institutional Semantics") is a library of information components designed to enable the creation of business document models that span a range of category abstraction. UBL comes equipped with XML schemas that define document categories like orders, invoices, payments, and receipts that many people are familiar with from their personal experiences of shopping and paying bills. However, UBL can also be used to design very specific or subordinate level transactional document types like "purchase order for industrial chemicals when buyer and seller are in different countries," or document types at the other end of the abstraction hierarchy like "fill-in-the-blank" legal forms for any kind of contract.

Bowker and Star point out that there is often a pragmatic tradeoff between precision and validity when defining categories and assigning resources to them, particularly in scientific and other highly technical domains. More granular categories make more precise classification possible in principle, but highly specialized domains might contain instances that are so complex or hard to understand that it is difficult to decide where to organize them.[3]

As an example of this real-world messiness that resists precise classification, Bowker and Star turn to medicine and the World Health Organization's International Classification of Diseases (ICD), a system of categories for cause-of-death reporting. The ICD requires that every death be assigned to one and only one category out of thousands of possible choices, which facilitates important uses such as statistical reporting for public health research.

In practice, however, doctors often lack conclusive evidence about the cause of a particular death, or they identify a number of contributing factors, none of which could properly be described as the sole cause. In these situations, less precise categories would

3. (Bowker and Star 2000)

better accommodate the ambiguity, and the aggregate data about causes of death would have greater validity. But doctors have to use the ICD's precise categories when they sign a death certificate, which means they sometimes record the wrong cause of death just to get their work done.

It might seem counterintuitive, but when a system of human-generated categories is too complex for people to interpret and apply reliably, computational classifiers that compute statistical similarity between new and already classified items can outperform people.[4]

4. Statistician and baseball fan Nate Silver rejected a complex system that used twenty-six player categories for predicting baseball performance because "it required as much art as science to figure out what group a player belonged in." (Silver 2012, p, 83). His improved system used the technique of "nearest neighbor" analysis to identify current baseball players whose minor league statistics were most similar to the current minor league players being evaluated. (See "Categories Created by Clustering").

   Silver later became famous for his extremely accurate predictions of the 2008 US presidential elections. He is the founder and editor of the FiveThirtyEight blog, so named because there are 538 senators and representatives in the US Congress.

Basic or Natural Categories

Category abstraction is normally described in terms of a hierarchy of superordinate, basic, and subordinate category levels. "Clothing," for example, is a superordinate category, "shirts" and "socks" are basic categories, and "white long-sleeve dress shirts" and "white wool hiking socks" are subordinate categories. Members of basic level categories like "shirts" and "socks" have many perceptual properties in common, and are more strongly associated with motor movements than members of superordinate categories. Members of subordinate categories have many common properties, but these properties are also shared by members of other subordinate categories at the same level of abstraction in the category hierarchy. That is, while we can identify many properties shared by all "white long-sleeve dress shirts," many of them are also properties of "blue long-sleeve dress shirts" and "black long-sleeve pullover shirts."

Psychological research suggests that some levels of abstraction in a system of categories are more basic or natural than others. Anthropologists have also observed that folk taxonomies invariably classify natural phenomena into a five- or six-level hierarchy, with one of the levels being the psychologically basic or "real" name (such as "cat" or "dog"), as opposed to more abstract names (e.g. "mammal") that are used less in everyday life. An implication for organizing system design is that basic level categories are highly efficient in terms of the cognitive effort they take to create and use. A corollary is that classifications with many levels at different abstraction levels may be difficult for users to navigate effectively.[5]

5. (Rosch 1999) calls this the principle of cognitive economy, that "what one wishes to gain from one's categories is a great deal of information about the environment while

The Recall / Precision Tradeoff

The abstraction level we choose determines how precisely we identify resources. When we want to make a general claim, or communicate that the scope of our interest is broad, we use superordinate categories, as when we ask, "How many animals are in the San Diego Zoo?" But we use precise subordinate categories when we need to be specific: "How many adult emus are in the San Diego Zoo today?"

If we return to our clothing example, finding a pair of white wool hiking socks is very easy if the organizing system for socks creates fine-grained categories. When resources are described or arranged with this level of detail, a similarly detailed specification of the resources you are looking for yields precisely what you want. When you get to the place where you keep white wool hiking socks, you find all of them and nothing else. On the other hand, if all your socks are tossed unsorted into a sock drawer, when you go sock hunting you might not be able to find the socks you want and you will encounter lots of socks you do not want. But you will not have put time into sorting them, which many people do not enjoy doing; you can spend time sorting or searching depending on your preferences.

If we translate this example into the jargon of information retrieval, we say that more fine-grained organization reduces *recall*, the number of resources you find or retrieve in response to a query, but

conserving finite resources as much as possible. [...] It is to the organism's advantage not to differentiate one stimulus from another when that differentiation is irrelevant to the purposes at hand." (Pages 3-4.)

increases the *precision* of the recalled set, the proportion of recalled items that are relevant. Broader or coarse-grained categories increase recall, but lower precision. We are all too familiar with this hard bargain when we use a web search engine; a quick one-word query results in many pages of mostly irrelevant sites, whereas a carefully crafted multi-word query pinpoints sites with the information we seek. We will discuss recall, precision, and evaluation of information retrieval more extensively in *Interactions with Resources*.

This mundane example illustrates the fundamental tradeoff between organization and retrieval. A tradeoff between the investment in organization and the investment in retrieval persists in nearly every organizing system. The more effort we put into organizing resources, the more effectively they can be retrieved. The more effort we are willing to put into retrieving resources, the less they need to be organized first. The allocation of costs and benefits between the organizer and retriever differs according to the relationship between them. Are they the same person? Who does the work and who gets the benefit?

Category Audience and Purpose

The ways in which people categorize depend on the goals of categorization, the breadth of the resources in the collection to be categorized, and the users of the organizing system. Suppose that we want to categorize languages. Our first step might be determining what constitutes a language, since there is no widespread agreement on what differentiates a language from a dialect, or even on whether such a distinction exists.

What we mean by "English" and "Chinese" as categories can change depending on the audience we are addressing and what our purpose

is, however.[6] A language learning school's representation of "English" might depend on practical concerns such as how the school's students are likely to use the language they learn, or which teachers are available. For the purposes of a school teaching global languages, and one of the standard varieties of English (i.e., those associated with political power), or an amalgamation of several standard varieties, might be thought of as a single instance ("English") of the category "Languages."

Similarly, the category structure in which "Chinese" is situated can vary with context. While some schools might not conceptualize "Chinese" as a category encompassing multiple linguistic varieties,

6. For example, some linguists think of "English" as a broad category encompassing multiple languages or dialects, such as "Standard British English," "Standard American English," and "Appalachian English."

If we are concerned with linguistic diversity and the survival of minority languages, we might categorize some languages as endangered in order to mobilize language preservation efforts. We could also categorize languages in terms of shared linguistic ancestors ("Romance languages," for example), in terms of what kinds of sounds they make use of, by how well we speak them, by regions they are commonly spoken in, whether they are signed or unsigned, and so on. We could also expand our definition of the languages category to include artificial computer languages, or body language, or languages shared by people and their pets—or thinking more metaphorically, we might include the language of fashion.

but rather as a single instance within the "Languages" category, another school might teach its students Mandarin, Wu, and Cantonese as dialects within the language category "Chinese," that are unified by a single standard *writing system*. In addition, a linguist might consider Mandarin, Wu, and Cantonese to be mutually unintelligible, making them separate languages within the broader category "Chinese" for the purpose of creating a principled language classification system.

If people could only categorize in a single way, the Pyramid game show, where contestants guess what category is illustrated by the example provided by a clue giver, would pose no challenge. The creative possibilities provided by categorization allow people to order the world and refer to interrelationships among conceptions through a kind of allusive shorthand. When we talk about the language of fashion, we suggest that in the context of our conversation, instances like "English," "Chinese," and "fashion" are alike in ways that distinguish them from other things that we would not categorize as languages.

49. Implementing Categories

Categories are conceptual constructs that we use in a mostly invisible way when we talk or think about them. When we organize our kitchens, closets, or file cabinets using shelves, drawers, and folders, these physical locations and containers are visible implementations of our personal category system, but they are not the categories. This distinction between category design and implementation is obvious when we follow signs and labels in libraries or grocery stores to find things, search a product catalog or company personnel directory, or analyze a set of economic data assembled by the government from income tax forms. These institutional categories were designed by people prior to the assignment of resources to them.

This separation between category creation and category implementation prompts us to ask how a system of categories can be implemented. We will not discuss the implementation of categories in the literal sense of building physical or software systems that organize resources. Instead, we will take a higher-level perspective that analyzes the implementation problem to be solved for the different types of categories discussed in "Principles for Creating Categories", and then explain the logic followed to assign resources correctly to them.

Implementing Enumerated Categories

Categories defined by enumeration are easy to implement. The members or legal values in a set define the category, and testing an item for membership means looking in the set for it. Enumerated category definitions are familiar in drop-down menus and form-filling. You scroll through a list of all the countries in the world to

search for the one you want in a shipping address, and whatever you select will be a valid country name, because the list is fixed until a new country is born. Enumerated categories can also be implemented with associative arrays (also known as hash tables or dictionaries). With these data structures, a test for set membership is even more efficient than searching, because it takes the same time for sets of any size (see "Kinds of Structures").

Implementing Categories Defined by Properties

The most conceptually simple and straightforward implementation of categories defined by properties adopts the *classical view of categories* based on necessary and sufficient features. Because such categories are prescriptive with explicit and clear boundaries, classifying items into the categories is objective and deterministic, and supports a well-defined notion of *validation* to determine unambiguously whether some instance is a member of the category. Items are classified by testing them to determine if they have the required properties and property values. Tests can be expressed as rules:

- If instance X has property P, then X is in category Y.
- If a home mortgage loan in San Francisco exceeds $625,000, then it is classified as a "jumbo" loan by the US Office of Federal Housing Oversight.
- For a number to be classified as prime it must satisfy two rules: It must be greater than 1, and have no positive divisors other than 1 and itself.

This doesn't mean the property test is always easy; validation might require special equipment or calculations, and tests for the property might differ in their cost or efficiency. But given the test results, the

answer is unambiguous. The item is either a member of the category or it isn't.[1]

A system of hierarchical categories is defined by a sequence of property tests in a particular order. The most natural way to implement multi-level category systems is with *decision trees*. A simple *decision tree* is an algorithm for determining a decision by making a sequence of logical or property tests. Suppose a bank used a sequential rule-based approach to decide whether to give someone a mortgage loan.

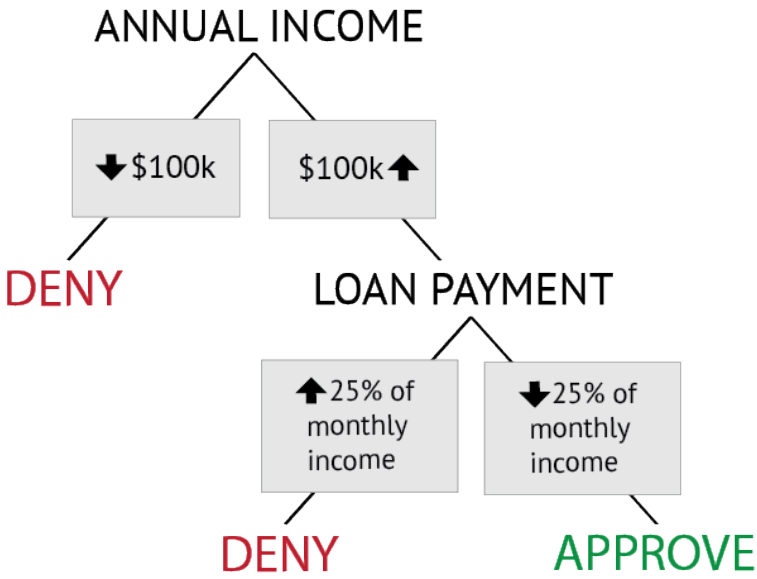
- If an applicant's annual income exceeds $100,000, and if the monthly loan payment is less than 25% of monthly income, approve the mortgage application.
- Otherwise, deny the loan application.

1. For example, you can test whether a number is prime by dividing it by every number smaller than its square root, but this algorithm is ridiculously impractical for any useful application. Many cryptographic systems multiply prime numbers to create encryption keys, counting on the difficulty of factoring them to protect the keys; so, proving that ever larger numbers are prime is very important. See (Crandall and Pomerance 2006).

   If you are wondering why prime numbers aren't considered an enumerative category given that every number that is prime already exists, it is because we have not found all of them yet, and we need to test through to infinity.

This simple decision tree is depicted in Figure: Rule-based Decision Tree. The rules used by the bank to classify loan applications as "Approved" or "Denied" have a clear representation in the tree. The easy interpretation of decision trees makes them a common formalism for implementing classification models.

Rule-based Decision Tree



In this simple decision tree, a sequence of two tests for the borrower's annual income and the percentage of monthly income required to make the loan payment classify the applicants into the "deny" and "approve" categories.

Nevertheless, any implementation of a category is only interpretable to the extent that the properties and tests it uses in its definition and implementation can be understood. Because natural language is inherently ambiguous, it is not the optimal representational format for formally defined institutional

categories. Categories defined using natural language can be incomplete, inconsistent, or ambiguous because words often have multiple meanings. This implementation of the bank's procedure for evaluating loans would be hard to interpret reliably:

- If applicant is wealthy, and then if the monthly payment is an amount that the applicant can easily repay, then applicant is approved.

To ensure their interpretability, decision trees are sometimes specified using the controlled vocabularies and constrained syntax of "simplified writing" or "business rule" systems.

Artificial languages are a more ambitious way to enable precise specification of property-based categories. An artificial language expresses ideas concisely by introducing new terms or symbols that represent complex ideas along with syntactic mechanisms for combining and operating on them. Mathematical notation, programming languages, schema languages that define valid document instances (see "Specifying Vocabularies and Schemas"), and regular expressions that define search and selection patterns (see "Controlling Values") are familiar examples of artificial languages. It is certainly easier to explain and understand the Pythagorean Theorem when it is efficiently expressed as "$H^2 = A^2 + B^2$" than with a more verbose natural language expression: "In all triangles with an angle such that the sides forming the angle are perpendicular, the product of the length of the side opposite the angle such that the sides forming the angle are perpendicular with itself is equal to the sum of the products of the lengths of the other two sides, each with itself."[2]

2. This example comes from (Perlman 1984), who introduced the idea of "natural artificial languages" as

Artificial languages for defining categories have a long history in philosophy and science. (See the sidebar, Artificial Languages for Description and Classification). However, the vast majority of institutional category systems are still specified with natural language, despite its ambiguities because people usually understand the languages they learned naturally better than artificial ones. Sometimes this is even intentional to allow institutional categories embodied in laws to evolve in the courts and to accommodate technological advances.[3]

Artificial Languages for Description and Classification

those designed to be easy to learn and use because they employ mnemonic symbols, suggestive names, and consistent syntax.

3. When the US Congress revised copyright law in 1976 it codified a "fair use" provision to allow for some limited uses of copyrighted works, but fair use in the digital era is vastly different today; website caching to improve performance and links that return thumbnail versions of images are fair uses that were not conceivable when the law was written. A law that precisely defined fair uses using contemporary technology would have quickly become obsolete, but one written more qualitatively to enable interpretation by the courts has remained viable. See (Samuelson 2009).

John Wilkins was one of the founders of the British Royal Society. In 1668 he published *An Essay towards a Real Character and a Philosophical Language* in which he proposed an artificial language for describing a universal taxonomy of knowledge that used symbol composition to specify a location in the category hierarchy. There were forty top level genus categories, which were further subdivided into differences within the genus, which were then subdivided into species. Each genus was a monosyllable of two letters; each difference added a consonant, and each species added a vowel.

This artificial language conveys the meaning of categories directly from the composition of the category name. For instance, *zi* indicates the genus of beasts, *zit* would be "rapacious beasts of the dog kind" whereas *zid* would be "cloven-footed beast." Adding for the fourth character an *a* for species, indicating the second species in the difference, would give *zita* for dog and *zida* for sheep.

In *The Analytical Language of John Wilkins,* Jorge Luis Borges remarks that Wilkins has many "ambiguities, redundancies and deficiencies" in the language and presents as a foil and parody an imagined "Celestial Empire of Benevolent Knowledge."

> In its remote pages it is written that the animals are divided into: (a) belonging to the emperor, (b) embalmed, (c) tame, (d) sucking pigs, (e) sirens, (f)

> fabulous, (g) stray dogs, (h) included in the present classification, (i) frenzied, (j) innumerable, (k) drawn with a very fine camel hair brush, (l) et cetera, (m) having just broken the water pitcher, (n) that from a long way off look like flies.
>
> Borges compliments Wilkins for inventing names that might signify in themselves some meaning to those who know the system, but notes that "it is clear that there is no classification of the Universe not being arbitrary and full of conjectures."[4]

Data schemas that specify data entities, elements, identifiers, attributes, and relationships in databases and XML document types on the transactional end of the Document Type Spectrum ("Resource Domain") are implementations of the categories needed for the design, development and maintenance of information organization systems. Data schemas tend to rigidly define categories of resources. [5]

4. (Wilkins 1668) and (Borges 1952)

5. "Rigid" might sound negative, but a rigidly defined resource is also precisely defined. Precise definition is essential when creating, capturing, and retrieving data and when information about resources in different organizing systems needs to be combined or compared. For example, in a traditional relational database, each

> In object-oriented programming languages, *classes* are schemas that serve as templates for the creation of objects. A class in a programming language is analogous to a database schema that specifies the structure of its member instances, in that the class definition specifies how instances of the class are constructed in terms of data types and possible values. Programming classes may also specify whether data in a member object can be accessed, and if so, how.[6]

table contains a field, or combination of fields, known as a primary key, which is used to define and restrict membership in the table. A table of email messages in a database might define an email message as a unique combination of sender address, recipient address, and date/time when the message was sent, by enforcing a primary key on a combination of these fields. Similar to category membership based on a single, monothetic set of properties, membership in this email message table is based on a single set of required criteria. An item without a recipient address cannot be admitted to the table. In categorization terms, the item is not a member of the "email message" class because it does not have all the properties necessary for membership.

6. Like *data schemas*, programming classes specify and enforce rules in the construction and manipulation of data. However, programming classes, like other implementations that are characterized by specificity and rule enforcement, can vary widely in the degree to which rules are specified and enforced. While some class

Unlike transactional document types, which can be prescriptively defined as *classical categories* because they are often produced and consumed by automated processes, narrative document types are usually descriptive in character. We do not classify something as a novel because it has some specific set of properties and content types. Instead, we have a notion of typical novels and their characteristic properties, and some things that are considered novels are far from typical in their structure and content.[7]

Nevertheless, categories like narrative document types can sometimes be implemented using document schemas that impose only a few constraints on structure and content. A schema for a purchase order is highly prescriptive; it uses *regular expressions*, strongly data typed content, and enumerated code lists to validate the value of required elements that must occur in a particular order. In contrast, a schema for a narrative document type would have much optionality, be flexible about order, and expect only text in its sections, paragraphs and headings. Even very lax document definitions are very rigid, others are more flexible. Some languages have abstract types that have no instances but serve to provide a common ancestor for specific implemented types.

7. The existence of chapters might suggest that an item is a novel; however, a lack of chapters need not automatically indicate that an item is not a novel. Some novels are hypertexts that encourage readers to take alternative paths. Many of the writings by James Joyce and Samuel Beckett are “stream of consciousness” works that lack a coherent plot, yet they are widely regarded as novels.

schemas can be useful in making content management, reuse, and formatting more efficient.

Implementing Categories Defined by Probability and Similarity

Many categories cannot be defined in terms of required properties, and instead must be defined probabilistically, where category membership is determined by properties that resources are likely to share. Consider the category "friend." You probably consider many people to be your friends, but you have longtime friends, school friends, workplace friends, friends you see only at the gym, and friends of your parents. Each of these types of friends represents a different cluster of common properties. If someone is described to you as a potential friend or date, how accurately can you predict that the person will become a friend? (See the sidebar, Finding Friends and Dates: Lessons for Learning Categories)

> Finding Friends and Dates:
> Lessons for Learning Categories
>
> Online dating or matchmaking sites use many of the same features to describe people, but also have additional features to make more accurate matches for their targeted users. As the number of features grows, there are exponentially more combinations of shared properties. For example, the matchmaking site eHarmony employs 29 "Dimensions of Compatibility" and more than 200 questions to create a user profile. Even if the 29 dimensions were Boolean (would you describe yourself as x?) this yields 2^{29} or over

500,000,000 different combinations. Using these complex resource descriptions to predict the probability of a good match requires matchmaking sites to use proprietary machine learning algorithms to propose matches, which are ranked with unexplained measures and precision (what does an 80% match mean?). Not surprisingly, many people who try online dating give up after less success than they expected.

With such a large number of features in user profiles, any matching algorithm confronts what machine learning calls the curse of dimensionality. With high-dimensional data, there can never be enough instances to learn which features are really the most important. Neither you nor the online dating algorithm will ever meet enough different kinds of people to reliably predict the outcome of a possible match.

But all is not hopeless. Machine learning programs attack the curse of dimensionality using statistical techniques that use correlations among features to combine them or adjust the weights given to features to reflect their value in making predictions or classifications. For example, OKCupid asks people to rate how much importance they assign to match questions. You might prefer cats to dogs, and you might either never consider dating a dog lover or you might not care at all.

Another way to reduce the number of features needed to classify accurately is to reduce the scope of the category being learned. The matchmaking model for sites that target people with particular professions, religions, or political views would be less complex than

the eHarmony one, because the former will have fewer relevant features, and hence fewer random correlations and noise that will undermine its accuracy. All other things being equal, the lower the variability in a set of examples, the better a model that learns from that data will perform.

Probabilistic categories can be challenging to define and use because it can be difficult to keep in mind the complex feature correlations and probabilities exhibited by different clusters of instances from some domain. Furthermore, when the category being learned is broad with a large number of members, the sample from which you learn strongly shapes what you learn. For example, people who grow up in high-density and diverse urban areas may have less predictable ideas of what an acceptable potential date looks like than someone in a remote rural area with a more homogeneous population.

More generally, if you are organizing a domain where the resources are active, change their state, or are measurements of properties that vary and co-occur probabilistically, the sample you choose strongly affects the accuracy of models for classification or prediction. In *The Signal and the Noise*, statistician Nate Silver explains how many notable predictions failed because of poor sampling techniques. One common sampling mistake is to use too short a historical window to assemble the training dataset; this is often a corollary of a second mistake, an over reliance on recent data because it is more available. For example, the collapse of housing prices and the resulting financial crisis of 2008 can be explained in part because the models that lenders used to predict mortgage foreclosures were based on data from 1980-2005, when house prices tended to grow higher. As a result, when mortgage

foreclosures increased rapidly, the results were "out of sample" and were initially misinterpreted, delaying responses to the crisis.

Samples from dynamic and probabilistic domains result in models that capture this variability. Unfortunately, because many forecasters want to seem authoritative, and many people do not understand probability, classifications or predictions that are inherently imprecise are often presented with certainty and exactness even though they are probabilistic with a range of outcomes. Silver tells the story of a disastrous 1997 flood caused when the Red River crested at 54 feet when the levees protecting the town of Grand Forks were at 51 feet. The weather service had predicted a crest between 40 and 58 feet, but emphasized the midpoint of the range, which was 49 feet. Unfortunately, most people interpreted this probabilistic prediction as if it were a binary classification, "flood" versus "no flood," ignored the range of the forecast, and failed to prepare for a flood that had about a 35% chance of occurring.[8]

Probabilistic Decision Trees

In "Implementing Categories Defined by Properties", we showed how a rule-based decision tree could be used to implement a strict property-based classification in which a bank uses tests for the properties of "annual income" and "monthly loan payment" to

8. See (Silver 2012). Over reliance on data that is readily available is a decision-making heuristic proposed by (Tversky and Kahneman 1974), who developed the psychological foundations for behavioral economics. (See the sidebar, Behavioral Economics.)

classify applicants as approved or denied. We can adapt that example to illustrate probabilistic decision trees, which are better suited for implementing categories in which category membership is probabilistic rather than absolute.

Banks that are more flexible about making loans can be more profitable because they can make loans to people that a stricter bank would reject but who still are able to make loan payments. Instead of enforcing conservative and fixed cutoffs on income and monthly payments, these banks consider more properties and look at applications in a more probabilistic way. These banks recognize that not every loan applicant who is likely to repay the loan looks exactly the same; "annual income" and "monthly loan payment" remain important properties, but other factors might also be useful predictors, and there is more than one configuration of values that an applicant could satisfy to be approved for a loan.

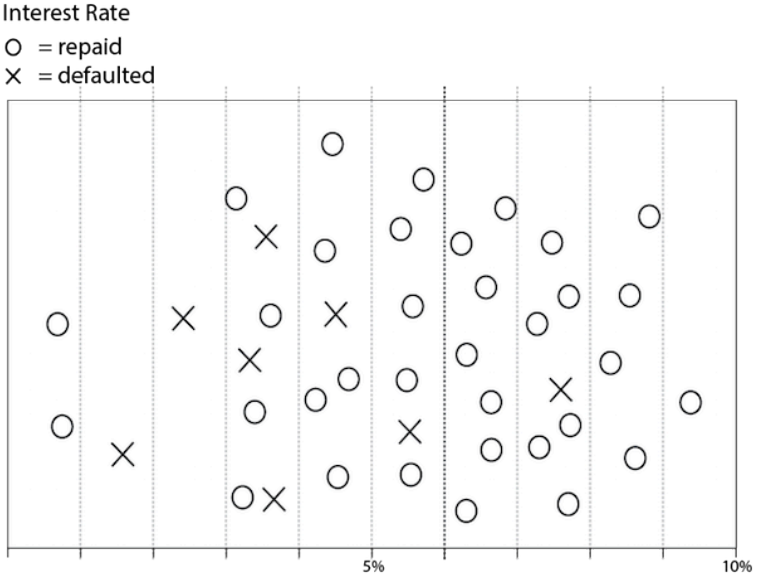
Which properties of applicants best predict whether they will repay the loan or default? A property that predicts each at 50% isn't helpful because the bank might as well flip a coin, but a property that splits the applicants into two sets, each with very different probabilities for repayment and defaulting, is very helpful in making a loan decision.

A data-driven bank relies upon historical data about loan repayment and defaults to train algorithms that create decision trees by repeatedly splitting the applicants into subsets that are most different in their predictions. Subsets of applicants with a high probability of repayment would be approved, and those with a high probability of default would be denied a loan. One method for selecting the property test for making each split is calculating the "information gain" (see the sidebar Using "Information Theory" to Quantify Organization). This measure captures the degree to which each subset contains a "pure" group in which every applicant is classified the same, as likely repayers or likely defaulters.

For example, consider the chart in Figure: Historical Data: Loan

Repayment Based on Interest Rate which is a simplified representation of the bank's historical data on loan defaults based on the initial interest rate. The chart represents loans that were repaid with "o" and those that defaulted with "x." Is there an interest rate that divides them into "pure" sets, one that contains only "o" loans and the other that contains only "x" loans?

Historical Data: Loan Repayment Based on Interest Rate



The "o" symbol represents loans that were repaid by the borrower; "x" represents loans on which the borrower defaulted. A 6% rate (darker vertical line) best divides the loans into subsets that differ in the payment outcome.

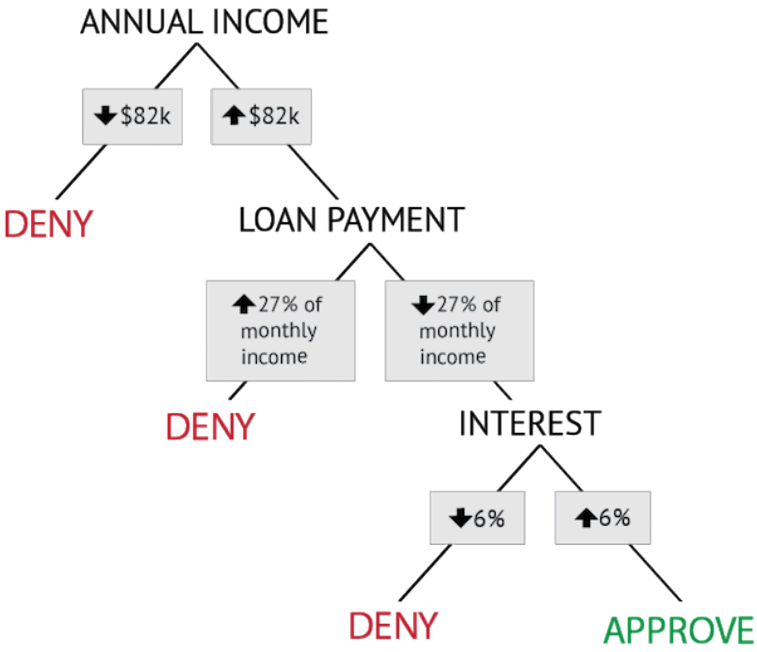
You can see that no interest rate divides these into pure sets. So the best that can be done is to find the interest rate that divides them

so that the proportions of defaulters are most different on each side of the line.[9]

This dividing line at the 6% interest rate best divides those who defaulted from those who repaid their loan. Most people who borrowed at 6% or greater repaid the loan, while those who took out loans at a lower rate were more likely to default. This might seem counter-intuitive until you learn that the lower-interest rate loans had adjustable rates that increased after a few years, causing the monthly payments to increase substantially. More prudent borrowers were willing to pay higher interest rates that were fixed rather than adjustable to avoid radical increases in their monthly payments.

Probabilistic Decision Tree

9. To be precise, this "difference of proportions" calculation uses an algorithm that also uses the logarithm of the proportions to calculate entropy, a measure of the uncertainty in a probability distribution. An entropy of zero means that the outcome can be perfectly predicted and entropy increases as outcomes are less predictable. The information gain for an attribute is how much it reduces entropy after it is used to subdivide a dataset.



In this probabilistic decision tree, the sequence of property tests and the threshold values in each test divide the loan applicants into categories that differ in how likely they are to repay the loan.

This calculation is carried out for each of the attributes in the historical data set to identify the one that best divides the applicants into the repaid and defaulted categories. The attributes and the value that defines the decision rule can then be ordered to create a decision tree similar to the rule-based one we saw in "Implementing Categories Defined by Properties". In our hypothetical case, it turns out that the best order in which to test the properties is Income, Monthly Payment, and Interest Rate, as shown in Figure: Probabilistic Decision Tree. The end result is still a set of rules, but behind each decision in the tree are probabilities based on historical data that can more accurately predict whether an applicant will repay or default. Thus, instead of the arbitrary cutoffs at $100,000 in income and 25% for monthly payment, the bank can offer loans to

people with lower incomes and remain profitable doing so, because it knows from historical data that $82,000 and 27% are the optimal decision points. Using the interest rate in their decision process is an additional test to ensure that people can afford to make loan payments even if interest rates go up.[10]

Because decision trees specify a sequence of rules that make property tests, they are highly interpretable, which makes them a very popular choice for data scientists building models much more complex than the simple loan example here. But they assume that every class is a conjunction of all the properties used to define them. This makes them susceptible to over-fitting because if they grow very deep with many property conjunctions, they capture exactly the properties that describe each member of the training set, effectively memorizing the training data. In other words, they capture both what is generally true beyond the set and what is particular to the training set only, when the goal is to build a model

10. Unfortunately, this rational data-driven process for classifying loan applications as "Approved" or "Denied" was abandoned during the "housing bubble" of the early 2000s. Because lending banks could quickly sell their mortgages to investment banks who bundled them into mortgage-backed securities, applicants were approved without any income verification for "subprime" loans that initially had very low adjustable interest rates. Of course, when the rates increased substantially a few years later, defaults and foreclosures skyrocketed. This sad story is told in an informative, entertaining, but depressing manner in "The Big Short" (Lewis, 2010) and in a 2015 movie with the same name.

that captures only what is generally true. Overfitting in decision trees can be prevented by pruning back the tree after it has perfectly classified the training set, or by limiting the depth of the tree in advance, essentially pre-pruning it.

Naïve Bayes Classifiers

Another commonly used approach to implement a classifier for probabilistic categories is called Naïve Bayes. It employs Bayes' Theorem for learning the importance of a particular property for correct classification. There are some common sense ideas that are embodied in Bayes' Theorem:

- When you have a hypothesis or prior belief about the relationship between a property and a classification, new evidence consistent with that belief should increase your confidence.
- Contradictory evidence should reduce confidence in your belief.
- If the base rate for some kind of event is low, do not forget that when you make a prediction or classification for a new specific instance. It is easy to be overly influenced by recent information.

Now we can translate these ideas into calculations about how learning takes place. For property A and classification B, Bayes' Theorem says:

$$P(A \mid B) = P(B|A)\, P(A) \,/\, P(B)$$

The left hand side of the equation, P (A | B), is what we want to estimate but can't measure directly: the probability that A is the correct classification for an item or observation that has property B.

This is called the conditional or posterior probability because it is estimated after seeing the evidence of property B.

P (B | A) is the probability that any item correctly classified as A has property B. This is called the likelihood function.

P (A) and P (B) are the independent or prior probabilities of A and B; what proportion of the items are classified as A? How often does property B occur in some set of items?

Using Bayes' Theorem to Calculate Conditional Probability

Your personal library contains 60% fiction and 40% nonfiction books. All of the fiction books are in ebook format, and half of the nonfiction books are ebooks and half are in print format. If you pick a book at random and it is in ebook format, what is the probability that it is nonfiction?

Bayes' Theorem tells us that:

P (nonfiction | ebook) = P (ebook |nonfiction) x P (no nfiction) / P (ebook).

We know: P (ebook | nonfiction) = .5 and P (nonfiction) = .4

We compute P (ebook) using the law of total probability to compute the combined probability of all the independent ways in which an ebook might be sampled. In this example there are two ways:

P (ebook) = P (ebook | nonfiction) x P (nonfiction) + P (ebook | fiction) x P (fiction)
= (.5 x .4) + (1 x .6) = .8

Therefore: P (nonfiction | ebook) = (.5 x .4) / .8 = .25

Now let's apply Bayes' Theorem to implement email spam filtering. Messages are classified as SPAM or HAM (i.e., non-SPAM); the former are sent to a SPAM folder, while the latter head to your inbox.

1. Select Properties. We start with a set of properties, some from the message metadata like the sender's email address or the number of recipients, and some from the message content. Every word that appears in messages can be treated as a separate property[11]

11. Machine learning algorithms differ in which properties they use in how they select them. A straightforward method is to run the algorithms using different sets of properties, and select the set that yields the best result. However, it can be very computationally expensive to run algorithms multiple times, especially when the number of properties is large. A faster alternative is to select or filter features based on how well they predict the classification. The information gain calculation discussed in "Probabilistic Decision Trees" is an example of a filter method.

    Naïve Bayes classifiers make the simplifying assumption that the properties are independent, an assumption that is rarely correct, which is why the

2. Assemble Training Data. We assemble a set of email message that have been correctly assigned to the SPAM and HAM categories. These labeled instances make up the training set.

3. Analyze the Training Data. For each message, does it contain a particular property? For each message, is it classified as SPAM? If a message is classified as SPAM, does it contain a particular property? (These are the three probabilities on the right side of the Bayes equation).

4. Learn. The conditional probability (the left side of the Bayes equation) is recalculated, adjusting the predictive value of each property. Taken together, all of the properties are now able to correctly assign (most of) the messages into the categories they belonged to in the training set.

5. Classify. The trained classifier is now ready to classify uncategorized messages to the SPAM or HAM categories.

approach is called naïve. For example, a document that contains the word "insurance" is also likely to contain "beneficiary," so their presence in messages is not independent.

Nevertheless, even though the independence assumption is usually violated, Naive Bayes classifiers often perform very well. Furthermore, treating properties as independent means that the classifier needs much less data to train than if we had to calculate the conditional probabilities of all combination of properties. Instead, we just have to count separately the number of times each property occurs with each of the two classification outcomes.

6. Improve. The classifier can improve its accuracy if the user gives it feedback by reclassifying SPAM messages as HAM ones or vice versa. The most efficient learning occurs when an algorithm uses "active learning" techniques to choose its own training data by soliciting user feedback only where it is uncertain about how to classify a message. For example, the algorithm might be confident that a message with "Cheap drugs" in the subject line is SPAM, but if the message comes from a longtime correspondent, the algorithm might ask the user to confirm that the classification.[12]

Categories Created by Clustering

In the previous two sections, we discussed how probabilistic decision trees and naïve Bayes classifiers implement categories that are defined by typically shared properties and similarity. Both are examples of supervised learning because they need correctly classified examples as training data, and they learn the categories they are taught.

In contrast, clustering techniques are unsupervised; they analyze a collection of uncategorized resources to discover statistical

12. See (Blanzieri and Bryl 2009) for a review of the spam problem and the policy and technology methods for fighting it. (Upsana and Chakravarty 2010) is somewhat more recent and more narrowly focused on text classification techniques.

    A very thorough yet highly readable introduction to Active Learning is (Settles 2012).

regularities or structure among the items, creating a set of categories without any labeled training data.

Clustering techniques share the goal of creating meaningful categories from a collection of items whose properties are hard to directly perceive and evaluate, which implies that category membership cannot easily be reduced to specific property tests and instead must be based on similarity. For example, with large sets of documents or behavioral data, clustering techniques can find categories of documents with the same topics, genre, or sentiment, or categories of people with similar habits and preferences.

Because clustering techniques are unsupervised, they create categories based on calculations of similarity between resources, maximizing the similarity of resources within a category and maximizing the differences between them. These statistically-learned categories are not always meaningful ones that can be named and used by people, and the choice of properties and methods for calculating similarity can result in very different numbers and types of categories. Some clustering techniques for text resources suggest names for the clusters based on the important words in documents at the center of each cluster. However, unless there is a labeled set of resources from the same domain that can be used as a check to see if the clustering discovered the same categories, it is up to the data analyst or information scientist to make sense of the discovered clusters or topics.

There are many different distance-based clustering techniques, but they share three basic methods.

- The first shared method is that clustering techniques start with an initially uncategorized set of items or documents that are represented in ways that enable measures of inter-item similarity can be calculated. This representation is most often a vector of property values or the probabilities of different

properties so that items can be represented in a multidimensional space and similarity calculated using a distance function like those described in "Geometric Models of Similarity".[13]

- The second shared method is that categories are created by putting items that are most similar into the same category. Hierarchical clustering approaches start with every item in its own category. Other approaches, notably one called "K-means clustering," start with a fixed number of K categories initialized with a randomly chosen item or document from the complete set.

- The third shared method is refining the system of categories by iterative similarity recalculation each time an item is added to a category. Approaches that start with every item in its own category create a hierarchical system of categories by merging the two most similar categories, recomputing the similarity between the new category and the remaining ones, and repeating this process until all the categories are merged into a single category at the root of a category tree. Techniques that start with a fixed number of categories do not create new ones but instead repeatedly recalculate the "centroid" of the

13. In particular, documents are usually represented as vectors of frequency-weighted terms. Other approaches start more directly with the similarity measure, obtained either by direct judgments of the similarity of each pair of items or by indirect measures like the accuracy in deciding whether two sounds, colors, or images are the same or different. The assumption is that the confusability of two items reflects how similar they are.

category by adjusting its property representation to the average of all its members after a new member is added.[14]

It makes sense that the algorithms that create clusters or categories of similar items can be later used as classifiers by using the same similarity measures to compare the unclassified items against items that are labeled by category. There are different choices about which items to compare with the unclassified one:

- The centroid: a prototypical or average item calculated on the properties of all the category members. However, the centroid might not correspond to any actual member (see the sidebar Median versus Average), and this can make it hard to interpret the classification.

- Items that actually exist: Because the items in categories defined by similarity are not equally typical or good members, it is more robust to test against more than one exemplar. Classifiers that use this approach are called nearest-neighbor techniques, and they essentially vote among themselves and the majority category is assigned to the new item.

- The edge cases: These are instances that are closest to the

14. Unlike hierarchical clustering methods that have a clear stopping rule when they create the root category, k-means clustering methods run until the centroids of the categorize stabilize. Furthermore, because the k-means algorithm is basically just hill-climbing, and the initial category "seed" items are random, it can easily get stuck in a local optimum. So it is desirable to try many different starting configurations for different choices of K.

boundary between two categories, so there needs to be at least two of them, one in each category. Because they are not typical members of the category, they are the hardest to classify initially, but using them in classifiers emphasizes the properties that are the most discriminating. This is the approach taken by support vector machines, which are not clustering algorithms but are somewhat like nearest-neighbor algorithms in that they calculate the similarity of an unclassified item to these edge cases. Their name makes more sense if you think of the vectors that represent the "edge cases" being used to "support" the category boundary, which falls between them.

Neural networks

Among the best performing classifiers for categorizing by similarity and probabilistic membership are those implemented using neural networks, and especially those employing deep learning techniques. Deep learning algorithms can learn categories from labeled training data or by using autoencoding, an unsupervised learning technique that trains a neural network to reconstruct its input data. However, instead of using the properties that are defined in the data, deep learning algorithms devise a very large number of features in hidden hierarchical layers, which makes them uninterpretable by people. The key idea that made deep learning possible is the use of "backpropagation" to adjust the weights on features by working backwards from the output (the object classification produced by the network) all the way back to the input. The use of deep learning to classify images was mentioned in "Describing Images".[15]

15. In addition, the complex feature representations of neural networks compute very precise similarity

Implementing Goal-Based Categories

Goal-based categories are highly individualized, and are often used just once in a very specific context. However, it is useful to consider that we could implement model goal-derived categories as rule-based decision trees by ordering the decisions to ensure that any sub-goals are satisfied according to their priority. We could understand the category "Things to take from a burning house" by first asking the question "Are there living things in the house?" because that might be the most important sub-goal. If the answer to that question is "yes," we might proceed along a different path than if the answer is "no." Similarly, we might put a higher priority on things that cannot be replaced (Grandma's photos) than those that can (passport).

Implementing Theory-Based Categories

Theory-based categories arise in domains in which the items to be categorized are characterized by abstract or complex relationships with their features and with each other. With this model an entity need not be understood as inherently possessing features shared in common with another entity. Rather, people project features from one thing to another in a search for congruities between things, much as clue receivers in the second round of the Pyramid game search for congruities between examples provided by the clue giver in order to guess the target category. For example, a clue like "screaming baby" can suggest many categories, as can "parking

measurements, which enable searches for specific images or that find duplicate ones.

meter." But the likely intersection of the interactions one can have with babies and parking meters is that they are both "Things you need to feed."

Theory-based categories are created as cognitive constructs when we use analogies and classify because things brought together by analogy have abstract rather than literal similarity. The most influential model of analogical processing is Structure Mapping, whose development and application have been guided by Dedre Gentner for over three decades.

The key insight in Structure Mapping is that an analogy "a T is like B" is created by matching relational structures and not properties between the base domain B and a target domain T. We take any two things, analyze the relational structures they contain, and align them to find correspondences between them. The properties of objects in the two domains need not match, and in fact, if too many properties match analogy goes away and we have literal similarity:

- Analogy: The hydrogen atom is like our solar system
- Literal Similarity: The X12 star system in the Andromeda galaxy is like our solar system

Structure Mapping theory was implemented in the Structure-Mapping Engine (SME), which both formalized the theory and offered a computationally-tractable algorithm for carrying out the process of mapping structures and drawing inferences.[16]

16. Structure Mapping theory was proposed in (Gentner 1983), and the Structure Mapping Engine followed a few years later (Falkenhainer et al 1989). The SME was criticized for relying on hand-coded knowledge

representations, a limitation overcome by (Turney 2008), who used text processing techniques to extract the semantic relationships used by Structure Mapping.

50. Key Points in Chapter Seven

- What are categories?

  Categories are *equivalence classes*: sets or groups of things or abstract entities that we treat the same.

  (See "The What and Why of Categories")

- What determines the size of the equivalence class?

  The size of the equivalence class is determined by the properties or characteristics we consider.

  (See "The What and Why of Categories")

- Why do we contrast cultural, individual, and institutional categorization?

  Cultural, individual, and institutional categorization share some core ideas but they emphasize different processes and purposes for creating categories.

  (See "The What and Why of Categories")

- What distinguishes individual categories?

  Individual categories are created by intentional activity that usually takes place in response to a specific situation.

  (See "Individual Categories")

- What distinguishes institutional categories?

  Institutional categories are most often created in abstract and

information-intensive domains where unambiguous and precise categories are needed.

(See "Institutional Categories")

- What is the relation between categories and classification?

  The rigorous definition of institutional categories enables *classification*, the systematic assignment of resources to categories in an organizing system.

  (See "Institutional Categories")

- When is it necessary to create categories by computational methods rather than by people?

  Computational categories are created by computer programs when the number of resources, or when the number of descriptions or observations associated with each resource, are so large that people cannot think about them effectively.

  (See "Computational Categories")

- What is the difference between supervised and unsupervised learning?

  In supervised learning, a machine learning program is trained by giving it sample items or documents that are labeled by category. In unsupervised learning, the program gets the samples but has to come up with the categories on its own.

  (See Supervised and Unsupervised Learning)

- Why does it matter if every resource in a collection has a sortable identifier?

  Any collection of resources with sortable identifiers (alphabetic or numeric) as an associated property can benefit from using sorting order as an organizing principle.

(See "Single Properties")

- What is the concern when only a single property is used to assign category membership?

  If only a single property is used to distinguish among some set of resources and to create the categories in an organizing system, the choice of property is critical because different properties often lead to different categories.

  (See "Single Properties")

- What is a hierarchical category system?

  A sequence of organizing decisions based on a fixed ordering of resource properties creates a *hierarchy*, a multi-level category system.

  (See "Multi-Level or Hierarchical Categories")

- What can one say about any member of a classical category in terms of how it represents the category?

  An important implication of necessary and sufficient category definition is that every member of the category is an equally good member or example of the category.

  (See "Necessary and Sufficient Properties")

- What is aboutness?

  For most purposes, the most useful property of information resources for categorizing them is their *aboutness*, which is not directly perceivable and which is hard to characterize.

  (See "The Limits of Property-Based Categorization")

- When it is necessary to adopt a probabilistic or statistical view of properties in defining categories?

In domains where properties lack one or more of the characteristics of separability, perceptibility, and necessity, a probabilistic or statistical view of properties is needed to define categories.

(See "Probabilistic Categories and "Family Resemblance"")

- What is family resemblance?

  Sharing some but not all properties is akin to *family resemblances* among the category members.

  (See "Probabilistic Categories and "Family Resemblance"")

- What is similarity?

  Similarity is a measure of the resemblance between two things that share some characteristics but are not identical.

  (See "Similarity")

- What are the four psychologically-motivated approaches that propose different functions for computing similarity?

  Feature- or property-based, geometry-based, transformational, and alignment- or analogy-based approaches are psychologically-motivated approaches that propose different functions for computing similarity.

  (See "Similarity")

- What are so-called "classical categories"?

  Classical categories can be defined precisely with just a few *necessary and sufficient* properties.

  (See "Basic or Natural Categories")

- How does the breadth of a category affect the recall/precision tradeoff?

Broader or coarse-grained categories increase *recall*, but lower *precision*.

(See "The Recall / Precision Tradeoff")

- What is a decision tree?

  A simple *decision tree* is an algorithm for determining a decision by making a sequence of logical or property tests.

  (See "Implementing Categories Defined by Properties")

- What is the practical benefit of defining categories according to necessary and sufficient features?

  The most conceptually simple and straightforward implementation of categories in technologies for organizing systems adopts the classical view of categories based on necessary and sufficient features.

  (See "Implementing Categories Defined by Properties")

- How do artificial languages like mathematical notation and programming languages enable precise specification of categories?

  An artificial language expresses ideas concisely by introducing new terms or symbols that represent complex ideas along with syntactic mechanisms for combining and operating on them.

  (See "Implementing Categories Defined by Properties")

- How do Naïve Bayes classifiers learn?

  Naïve Bayes classifiers learn by revising the conditional probability of each property for making the correct classification after seeing the base rates of the class and property in the training data and how likely it is that a member of the class has the property.

(See "Naïve Bayes Classifiers")

- How do clustering techniques create categories?

  Because clustering techniques are unsupervised, they create categories based on calculations of similarity between resources, maximizing the similarity of resources within a category and maximizing the differences between them.

  (See "Categories Created by Clustering")

PART VIII

CLASSIFICATION: ASSIGNING RESOURCES TO CATEGORIES

Robert J. Glushko

Jess Hemerly

Vivien Petras

Michael Manoochehri

Longhao Wang

Jordan Shedlock

Daniel Griffin

51. Introduction (VIII)

In *Describing Relationships and Structures* we discussed different types of semantic relationships and contrasted abstract relationships between categories that define a semantic hierarchy like

Meat → is-a → Food

with concrete relationships involving specific people like members of the Simpson family:

Homer Simpson → is-a → Husband

When we make an assertion that a particular instance like Homer Simpson is a member of class, we are *classifying* the instance.

Classification, the systematic assignment of resources to intentional categories, is the focus of this chapter. In *Categorization: Describing Resource Classes and Types*, we described categories created by people as cognitive and linguistic models for applying prior knowledge and we discussed a set of principles for creating categories and category systems. We explained how cultural categories serve as the foundations upon which individual and institutional categories are based. Institutional categories are most often created in abstract and information-intensive domains where unambiguous and precise categories enable *classification* to be purposeful and principled. Computational categories inherited by supervised learning techniques are usually as interpretable as those created by people, but categories created by unsupervised machine learning techniques are statistical patterns that might or might not be interpretable.

A system of categories and its attendant rules or access methods is typically called a *classification scheme* or just the *classifications*.

A system of categories captures the distinctions and relationships among its resources that are most important in a domain and for a particular context of use, creating a reference model or conceptual roadmap for its users. This classification creates the structure and support for the interactions that human or computational agents perform. For example, research libraries and bookstores do not use the same classifications to organize books, but the categories they each use are appropriate for their contrasting types of collections and the different kinds of browsing and searching activities that take place in each context. Likewise, the scientific classifications for animals used by biologists contrast with those used in pet stores because the latter have no need for the precise differentiation enabled by the former.

Navigating This Chapter

Most of the chapter is a survey of topics that span the broad range of how classifications are used in organizing systems. These include enumerative classification ("Bibliographic Classification"), faceted classification ("Faceted Classification"), activity-based

classification (“Classification by Activity Structure”), and computational classification (“Computational Classification”). Because classification and standardization are closely related, we also analyze standards and standards making as they apply to organizing systems. Throughout, we observe how personal, institutional, cultural, linguistic, political, religious, and even artistic biases can affect otherwise principled and purposeful classification schemes. We finish the chapter with “Key Points in Chapter Eight”.

Classification vs. Categorization

Classification requires a system of categories, so not everyone distinguishes classification from categorization. Batley, for example, says classification is “imposing some sort of structure on our understanding of our environment,” a vague definition that applies equally well to categorization.[1]

In the discipline of organizing, the definition of classification is narrower and more formal. The contrasts among cultural, individual, and institutional categories in “The What and Why of Categories” yield a precise definition of classification: *The systematic assignment of resources to a system of intentional categories, often institutional ones.* This definition highlights the intentionality behind the system of categories, the systematic

1. (Batley 2005 p. 1).

processes for using them, and implies the greater requirements for *governance* and *maintenance* that are absent for cultural categories and most individual ones.

Classification vs. Tagging

Precise and reliable classification is possible when the shared properties of a collection of resources are used in a principled and systematic manner. This method of classification is essential to satisfy institutional and commercial purposes. However, this degree of rigor might be excessive for personal classifications and for classifications of resources in social or informal contexts.

Instead, a weaker approach to organizing resources is to use any property of a resource and any vocabulary to describe it, regardless of how well it differentiates it from other resources to create a system of categories. This method of organizing resources is most often called *tagging* ("Tagging of Web-based Resources"), but it has also been called *social classification*.[2]

Tagging is often used in personal organizing systems, but is social when it serves goals to convey information, develop a community, or manage reputation. Regardless of its name, however, tagging is popular for organizing and rating photos, websites, email messages,

2. (Hammond et al. 2004) note that the "unstructured (or better, free structured) approach to classification with users assigning their own labels is variously referred to as a folksonomy, folk classification, ethnoclassification, distributed classification, or social classification."

or other web-based resources or web-based descriptions of physical resources like stores and restaurants.

The distinction between classification and tagging was blurred when Thomas Vander Wal coined the term "folksonomy" —combining "folk" and "taxonomy" (which is a classification; see "Inclusion") —to describe the collection of tags for a particular web site or application.[3] Folksonomies are often displayed in the form of a *tag cloud*, where the frequency with which the tag is used throughout the site determines the size of the text in the tag cloud. The tag cloud emerges through the bottom-up aggregation of user tags and is a statistical construct, rather than a semantic one.[4]

Tagging seems insufficiently principled to be considered classification. Tagging a photo as "red" or "car" is an act of resource description, not classification, because the other tags that would serve as the alternative classifications are unspecified. Furthermore, when tagging principles are followed at all, they are likely to be idiosyncratic ones that were not pre-determined or arrived at through an analysis of goals and requirements.

Noticeably, some uses of tags treat them as category labels, turning tagging into classification. Many websites and resources encourage

3. Thomas Vander Wal invented the term "folksonomy" in 2004, and the term quickly gained traction. His personal account of the creation and dispersion of the term is (Vander Wal 2007).

4. See (Halvey and Keane 2007), (Sinclair and Cardrew-Hall 2007)) for analyses of the usability of different presentations, and (Kaser and Lemire 2007) for algorithms for drawing tag clouds.

users to assign “Like” or “+1” tags to them, and because these tags are pre-defined, they are category choices in an implied classification system; for example, we can consider “Like” as an alternative to a “Not liked enough” category.

When users or communities establish sets of principles to govern their tagging practices, tagging is even more like classification. Such a tagging system can be called a *tagsonomy*, a neologism we have invented to describe more systematic tagging. For example, a tagsonomy could predetermine tags as categories to be assigned to particular contents of a blog post, or specify the level of abstraction and granularity for assigning tags without predetermining them (“Category Design Issues and Implications”). Some people use multiple user accounts for the same application to establish distinct personas or contexts (e.g., personal vs. business photo collections) as a way to make their tagsonomies more distinct.

Making these decisions about tagging content and form and applying them in the tagging process transforms an *ad hoc* set of tags into a principled tagsonomy. When tagging is introduced in a business setting, more pragmatic purposes and more systematic tagging—for example, by using tags from lists of departments or products—also tends to create tagsonomic classification.[5]

“Tagging documents by computer,” or multi-label classification, is a glib way to describe topic modeling, an unsupervised learning technique for organizing and summarizing collections of unstructured documents by discovering patterns or clusters in the words they contain. The basic intuition behind topic modeling is that the words in a document are probabilistic indications of what the document is about; a document that contains words like

5. See (Millen, Feinberg, and Kerr 2006), (John and Seligmann 2006).

"election, government, and candidate" is probably about the "politics" topic, while words like "adore, wedding, and marriage" are good indications of a "love" topic. Topic models are not quite tagging because the words they identify to describe documents are not atomic tags or labels explicitly assigned to individual documents. Instead, topics are more like themes that different documents are more or less likely to contain.

Topic models have been used to implement user interfaces for browsing large document collections because they let a user explore using themes instead of specific search terms. In digital humanities, topic models have been used to discover changes in "what's written about" by some author or resource (like a newspaper) over time. Web commerce companies use topic models to organize books or products for their recommendation engines.[6]

Classification vs. Physical Arrangement

We have often stressed the principle in the discipline of organizing that logical issues must be separated from implementation issues. (See "The Concept of "Organizing Principle"", "Designing the Description Form", and "The Implementation Perspective ") With classification we separate the conceptual act of assigning a resource to a category from the subsequent but often incidental act of putting it in some physical or digital storage location. This focus

6. The statistical techniques used in topic models are intimidating; to vastly oversimplify, topic models start with a document x term matrix and extract topics by reducing the dimensionality through linear algebra techniques. (Blei 2012) is a relatively easy introduction.

on the logical essence of classification is elegantly expressed in a definition by Gruenberg: Classification is "a higher order thinking skill requiring the fusion of the naturalist's eye for relationships... with the logician's desire for structured order... the mathematician's compulsion to achieve consistent, predictable results... and the linguist's interest in explicit and tacit expressions of meaning."[7]

Taking a conceptual or cognitive perspective on classification contrasts with much conventional usage in library science, where classification is mostly associated with arranging tangible items on shelves, emphasizing the "parking" function that realizes the "marking" function of identifying the category to which the resource belongs.[8]

7. Gruenberg wrote this definition over a decade ago as a University of Illinois PhD student in an unpublished paper titled Faceted Classification, Facet Analysis, and the Web that was found by a web search by the first author of this chapter in 2005. When this chapter was being written several years later, the paper was no longer on the web, but a copy was located at Illinois by Matthew Beth on a backup disk.

8. This is reflected in library call numbers, which assign a unique number to books to designate the order in which they are shelved. Most American libraries use a classification system as part of their call number, composing it from a class number of the classification and a unique identifier (derived from the author name and title), which identifies the book within the class, often using a system called Cutter numbers. See

From a library science or collection curation perspective, it seems undeniable that when the resources being classified are physical or tangible things such as books, paintings, animals, or cooking pots, the end result of the classification activity is that some resource has been placed in some physical location. Moreover, the placement of physical resources can be influenced by the physical context in which they are organized. Once placed, the physical context often embodies some aspects of the organization when similar or related resources are arranged in nearby locations. In libraries and bookstores, this adjacency facilitates the serendipitous discovery of resources, as anyone well knows who has found an interesting book by browsing the shelves.

It might seem natural to identify storage locations with the classes used by the classification system. Just as we might think of a location in the zoo as the "lion habitat," we can put a "QC" sign on a particular row of shelves in a library where books about physics are arranged.

However, once we broaden the scope of organizing to include digital resources, it is clear that we rely on their logical classifications when we interact with them, not whether they reside on a computer in Berkeley or Bangalore. It is better to emphasize that a classification system is foremost a specification for the logical arrangement of resources because there are usually many possible and often arbitrary mappings of logical references to physical locations.

http://www.itsmarc.com/crs/mergedProjects/cutter/cutter/general_information_cutter.htm.

Classification Schemes

A classification scheme is a realization of one or more organizing principles. Physical resources are often classified according to their tangible or perceivable properties. As we discussed in "Single Properties" and "Multiple Properties", when properties take on only a small set of discrete values, a classification system naturally emerges in which each category is defined by one property value or some particular combination of property values. Classification schemes in which all possible categories to which resources can be assigned are defined explicitly are *enumerative*. For example, the *enumerative classification* for a personal collection of music recorded on physical media might have categories for CDs, DVDs, vinyl albums, 8-track cartridges, reel-to-reel tape, and tape cassettes; every music resource fits into one and only one of these categories.

When multiple resource properties are considered in a fixed sequence, each property creates another level in the system of categories and the classification scheme is *hierarchical* or *taxonomic*. (See "Inclusion".)

For information resources, their *aboutness* is usually more important than their physical properties. For example, a professor planning a new course might organize candidate articles for the syllabus in a fixed set of categories, one for each potential lecture topic. But it is more challenging to enumerate all the subjects or topics that a larger collection of resources might be about. The Library of Congress Classification(LCC) is a hierarchical and enumerative scheme with a very detailed set of subject categories because books can be about almost anything. We discuss the LCC more in "Bibliographic Classification".

In addition to or instead of their *aboutness*, information resources are sometimes organized using intrinsic properties like author

names or creation dates. Our professor might primarily organize his collection of articles by author name, and when he plans a new course, he might put those he selects for the syllabus into a classification system with one category for every scheduled lecture.

Because names and dates can take on a great many values, an organizing principle like *alphabetical* or *chronological* ordering is unlikely to enumerate in advance an explicit category for each possible value. Instead, we can consider these organizing principles as creating an *implicit or latent* classification system in which the categories are generated only as needed. For example, the Q category only exists in an alphabetical scheme if there is a resource whose name starts with Q.

Many resource domains have multiple properties that might be used to define a classification scheme. For example, wine can be classified by type of grape (varietal), color, flavor, price, winemaker, region of origin (appellation), blending style, and other properties. Furthermore, people differ in their knowledge or preferences about these properties; some people choose wine based on its price and varietal, while others studiously compare winemakers and appellations. Each order of considering the properties creates a different hierarchical classification, and using all of them would create a very deep and unwieldy system. Moreover, many different hierarchies might be required to satisfy divergent preferences. An alternative classification scheme for domains like these is *faceted* classification, a type of classification system that takes a set of resource properties and then generates only those categories for combinations that actually occur.

The most common types of facets are enumerative (mutually exclusive); Boolean (yes or no); hierarchical or taxonomic (logical containment); and spectrum (a range of numerical values). We discuss *faceted classification* in detail (in "Faceted Classification") because it is very frequently used in online classifications. Faceted schemes enable easier search and browsing of large resource

collections like those for retail sites and museums than hierarchical enumerative schemes. In library science a classification system that builds categories by combination of facets is sometimes also called *analytico-synthetic*.

The *Dewey Decimal Classification*(DDC) is a highly enumerative classification system that also uses faceted properties; we will discuss it more in "Bibliographic Classification".

Classification and Standardization

Classifications impose order on resources. Standards do the same by making distinctions, either implicitly or explicitly, between "standard" and "nonstandard" ways of creating, organizing, and using resources. Classification and standardization are not identical, but they are closely related. Some classifications become standards, and some standards define new classifications. Institutional categories ("Institutional Categories") are of two broad types.

Institutional Taxonomies

Institutional taxonomies are classifications designed to make it more likely that people or computational agents will organize and interact with resources in the same way. Among the thousands of standards published by the *International Organization for Standardization*(ISO) are many institutional taxonomies that govern the classification of resources and products in agriculture, aviation,

construction, energy, healthcare, information technology, transportation, and almost every industry sector.[9]

Institutional taxonomies are especially important in libraries and knowledge management. The Dewey Decimal Classification(DDC) and Library of Congress Classification(LCC) enable different libraries to arrange books in the same categories, and the *Diagnostic and Statistical Manual of Mental Disorders*(DSM) in clinical psychology enables different doctors to assign patients to the same diagnostic and insurance categories.[10]

9. The most "standard" of all standards organization is the International Organization for Standardization(ISO), whose members are themselves national standards organizations, which as a result gives the nearly 20,000 ISO standards the broadest and most global coverage. See http://ISO.org. In addition, there are scores of other national and industry-specific standards bodies whose work is potentially relevant to organizing systems of the sorts discussed in this book. We encounter these kinds of standards every day in codes for countries, currencies, and airports, in file formats, in product barcodes, and in many other contexts.

10. Dewey Decimal Classification: http://www.oclc.org/dewey/DDC.

    Similarly, the DSM is maintained and published by the American Psychiatric Association(APA) and it earns the many millions of dollars a year.

Institutional Semantics

Systems of *institutional semantics* offer precisely defined abstractions or *information components* ("Identity and Information Components") needed to ensure that information can be efficiently exchanged and used. Organizing systems that use different information models often cannot share and combine information without tedious negotiation and excessive rework.

Automating transactions with suppliers and customers in a supply chain requires that all the parties use the same data format or formats that can be transformed to be interoperable. Retrofitting or replacing these applications to enable efficient interoperability is often possible, and it is usually desirable for the firm to develop or adopt enterprise standards for information exchange models rather than pay the recurring transaction costs to integrate or transform incompatible formats.

Standard semantics are especially important in industries or markets that have significant network effects where the value of a product depends on the number of interoperable or compatible products—these include much of the information and service economies.

An example of a system of institutional semantics is the Universal Business Language(UBL) a library of about 2000 semantic "building blocks" for common concepts like "Address," "Item," "Payment," and "Party" along with nearly 100 document types assembled from the standard components. UBL is widely used to facilitate the automated exchange of transactional documents in procurement, logistics, inventory management, collaborative planning and forecasting, and payment.[11]

11. (OASIS 2006). All the finished work of OASIS is freely

Specifications vs. Standards

Implementing an organizing system of significant scope and complexity in a robust and maintainable fashion requires precise descriptions of the resources it contains, their formats, the classes, relations, structures and collections in which they participate, and the processes that ensure their efficient and effective use. Rigorous descriptions like these are often called "specifications" and there are well-established practices for developing good ones.

There is a subtle but critical distinction between "specifications" and "standards." Any person, firm, or *ad hoc* group of people or firms can create a specification and then use it or attempt to get others to use it.[12] In contrast, a standard is a published specification

available at https://www.oasis-open.org; the UBL committee is at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl.

12. A small number of people can often informally agree on an organizing system that meets the needs of each participant. But each new person often brings new requirements and it is not feasible to resolve every disagreement between every pair of participants. Instead, for a large-scale organizing system, decisions are usually made by entities that have the authority to coordinate actions and prevent conflicts by imposing a single solution on all the participants. (Rosenthal, Seligman, and Renner 2004) call this the "person-concept" tradeoff, which we can paraphrase as "a few people can agree on a lot, but a lot of people can only agree on a little."

that is developed and maintained by consensus of all the relevant stakeholders in some domain by following a defined and transparent process, usually under the auspices of a recognized standards organization.[13] In addition, implementations of standards often are

This authority can come from many different sources, but they can be roughly categorized as "authority from power" and "authority from consensus." Often the economic dominance of a firm allows it to control how business gets done in its industry. One key part of that is establishing specifications for data formats and classification schemes in organizing systems, which usually means requiring other firms to use the ones developed by the dominant firm for its own use. This ensures the continued efficiency of their own business processes while making it harder for other firms to challenge their market power.

In contrast, consensus is the authority mechanism embodied in the workings of the open source community, where the freedom to view and change data formats and code that uses them encourages cooperation and adoption. Consensus also underlies the authority of voluntary standards activities, where firms work together under the auspices of a standards body and agree to follow its procedures for creating, ratifying, and implementing standards.

13. International and national standards bodies derive their

subject to conformance tests that establish the completeness and accuracy of the implementation. This means that users can decide either to implement the specification themselves or choose from other conforming implementations.

The additional rigor and transparency when specifications are developed and maintained through a standards process often makes them fairer and gives them more legitimacy. Governments often require or recommend these *de jure* standards, especially those that are "open" or "royalty free" because they are typically supported by multiple vendors, minimizing the cost of adoption and maximizing their longevity.

For example, work on UBL has gone on for over a decade in a technical committee under the auspices of a standards

authority from the authority of the governments that created them. But standards organizations arguably derive most of their authority from the collective power of their members, because many influential standards organizations like OASIS, W3C, OMG, and IETF are not chartered or sponsored by governments. In addition, firms often create *ad hoc* "quasi-standards" organizations or "communities of interest" to facilitate relatively short-term cooperative standards-making activities that in the former case would otherwise be prohibited by anti-trust considerations. Finally, at the extreme "lightweight" end of the standards-making continuum, the codification of simple and commonly used information models as "microformats" depends on authority that emerges from the collaboration of individuals rather than firms.

development consortium called the Organization for the Advancement of Structured Information Standards(OASIS), which has developed scores of standards for web services and information-intensive industries.

Despite these important distinctions between "specifications" and "standards," however, in conventional usage "standard" is often simply a synonym for "dominant or widely-adopted specification." These *de facto* standards, in contrast with the *de jure* standards created by standards organizations, are typically created by the dominant firm or firms in an industry, by a new firm that is first to use a new technology or innovative method, or by a non-profit entity like a foundation that focuses on a particular domain.[14]

De facto standards and *ad hoc* standards often co-exist and compete in "standards wars," especially in information-intensive domains and industries with rapid innovation. Standards "wars" tend to occur when different firms or groups of firms develop two or more standards that tend to address the same needs. Not surprisingly, the competing standards are often incompatible on purpose. At first this lets each standard attract customers with features not enabled by the other, but it ends up locking them in by imposing switching costs. Current examples include Google vs. Apple on mobile phones and Kindle versus Apple on ebook readers.[15]

14. Often a standard evolves from an existing specification submitted to a standards organization by the firm that created it. In other cases, the specifications used by a dominant firm becomes a *de facto* standard by other firms in its industry, and it is never submitted to a formal standards-making process.

15. See (Shapiro and Varian, 1998).

For example, the Dewey Decimal Classification(DDC) is the world's most widely used library classification system, and most people treat it as a standard. In fact, the DDC is proprietary and it is maintained and licensed for use by the Online Computer Library Center(OCLC). Similarly, the DSM is maintained and published by the *American Psychiatric Association(APA)* and it earns the APA many millions of dollars a year.

In contrast, *de jure* standards include the Library of Congress Classification(LCC), developed under the auspices of the US government, the familiar MARC record format used in online library catalogs (ISO 2709), and its American counterpart ANSI Z39.2.[16]

As a result, even though it would be technically correct to argue that "while all standards are specifications, not all specifications are standards," this distinction is hard to maintain in practice.

Mandated Classifications

Standards are often imposed by governments to protect the interests of their citizens by coordinating or facilitating activities that might otherwise not be possible or safe. Some of them primarily concern public or product safety and are only tangentially relevant to systems for organizing information. Others are highly relevant, especially those that specify the formats and content of

16. Even so, the LCC is not "open" standard. You can browse the classifications on the LOC site, but to get them packaged as a book or complete digital resource you have to pay for them.

information exchange; many European governments require firms doing business with the government to adopt UBL.[17]

Other government standards that are important in organizing systems are those that express requirements for classification and

17. Governments have inherently long time horizons for their actions, they need to serve all citizens fairly and without discrimination, and they (should seek to) minimize cost to taxpayers. Each of these principles is an independent argument for standards and taken together they make a very strong one. Indeed, one the founding goals in the US Constitution is to protect the public interest, and this is enabled in Article I, Section 8 by granting Congress the power to set standards "of Weights and Measures" to facilitate commerce. Setting standards is a key role of the National Institute of Standards and Technology(NIST), part of the Department of Commerce, and other departments have similar standards-setting responsibilities and agencies, like the *Food and Drug Administration(FDA)* in the Department of Health and Social Services. In addition, independent government agencies like the *Federal Communications Commission(FCC)* and *Federal Trade Commission(FTC)* set numerous standards that are relevant to information organizing systems. And of course, the Library of Congress(LOC) maintains procedures and standards needed "to sustain and preserve a universal collection of knowledge... for future generations" (LOC.gov/about).

retention of auditing information for financial activities, such as the Sarbanes-Oxley Act, or for non-retention of personal information, such as HIPAA and FERPA.[18]

18. The Sarbanes-Oxley Act is US Public Law 107-204, http://www.sec.gov/about/laws/soa2002.pdf.

    The definitive source for the *Health Insurance Portability and Accountability Act*(HIPAA) is the US Department of Health & Human Services, http://www.hhs.gov/ocr/privacy/hipaa/understanding/index.html.

    The definitive source for the *Family Educational Rights and Privacy Act*(FERPA) is the US Department of Education, http://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html.

    Complying with government regulations like these can be expensive and difficult, and many companies, especially smaller ones, complain about the cost. On the other hand, the argument can be made that investing in a rigorous system for organizing information can provide competitive advantages, turning the compliance burden into a competitive weapon (Taylor 2006).

52. Understanding Classification

Classifications arrange resources to support discovery, selection, combination, integration, analysis, and other purposeful activity in every organizing system. A classification of diseases facilitates diagnosis and development of medical procedures, as well as accounting and billing. In addition, classifications facilitate understanding of a domain by highlighting the important resources and relationships in it, supporting the training of people who work in the domain and their acquisition of specialized skills for it.

We consider classification to be systematic when it follows principles that govern the structure of categories and their relationships. However, being systematic and principled does not necessarily ensure that a classification will be unbiased or satisfy all users' requirements. For example, the zoning, environmental, economic development, and political district classifications that overlay different parts of a city determine the present and future allocation of services and resources, and over time influence whether the city thrives or decays. These classifications reflect tradeoffs and negotiations among numerous participants, including businesses, lobbyists, incumbent politicians, donors to political parties, real estate developers, and others with strong self-interests.

Classification Is Purposeful

Categories often arise naturally, but by definition, classifications do not because they are systems of categories that have been intentionally designed for some purpose. Every classification brings

together resources that go together, and in doing so differentiates among them. However, bringing resources together would be pointless without reasons for finding, accessing, and interacting with them later.

Classifications Are Reference Models

A classification creates a semantic or conceptual roadmap to a domain by highlighting the properties and relationships that distinguish the resources in it. This reference model facilitates learning, comprehension, and the use of organizing systems within the domain. Standard classifications like those used in libraries enable people to rely on one system that they can use to locate resources in many libraries. Standard business, job, and product classifications enable the reliable collection, analysis, and interchange of economic data and resources.

Another important use of standard classifications created by people is as a "gold standard" for comparison with unsupervised computational classifications carried out on the same collection of resources or in the same domain. Presumably, no unsupervised classifier could exactly reproduce the classifications created by careful experts.

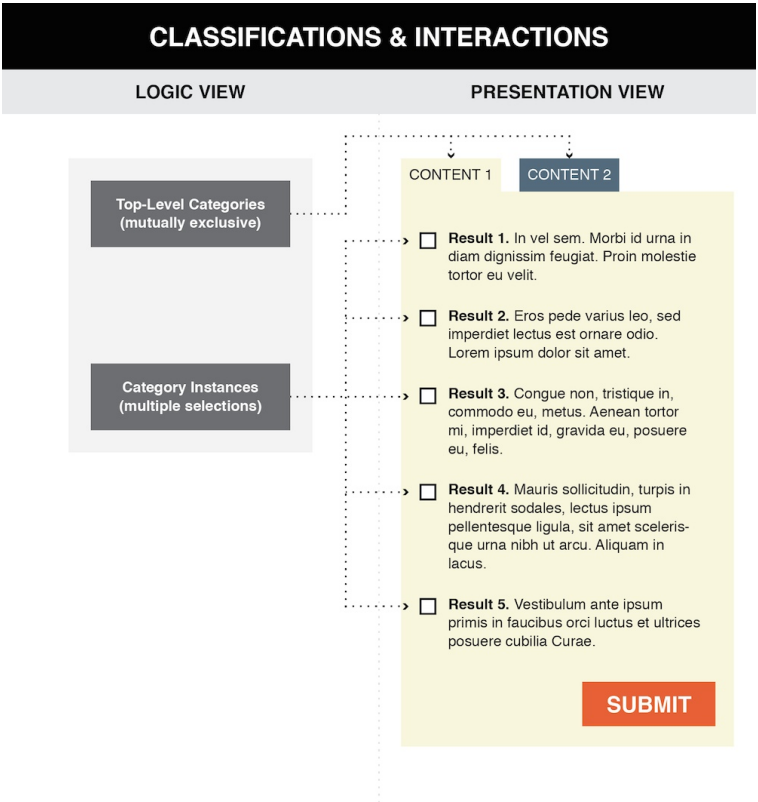
Classifications Support Interactions

A classification creates structure in the organizing system that increases the variety and capability of the interactions it can support. With physical resources, classification increases useful co-location; in kitchens, for example, keeping resources that are used together near each other (e.g., baking ingredients) makes cooking and cleanup more efficient (see "activity-based" classification in "Classification by Activity Structure").

Classification makes systems more usable when it is manifested in the arrangement of resource descriptions or controls in user interface components like list boxes, tabs, buttons, function menus, and structured lists of search results.[1]

A typical mapping between the logic of a classification scheme and a user interface is illustrated in Figure: Classification and Interactions.

1. The application of classification and organizing principles more generally to the design of user interfaces to facilitate information access, navigation, and use is often called "Information Architecture." See (Morville and Rosenfeld 2006).



Good user interface design creates a clear mapping between the logic of a classification scheme and the selection methods and arrangements presented to users. Categories that are mutually exclusive imply different tabs or other visualizations that imply a single selection, for example.

How a business classifies its product or service strongly influences whether a customer can find it; this is the essential task of marketing. The business of "search engine optimization" exists to help a firm with a web presence choose the categories and

descriptive terms that will improve its ranking in search results and attract the number of types of customer it desires.[2]

How a customer interacts with a supplier is influenced by how the supplier classifies its offerings in its shopping aisles or catalogs; the "science of shopping" uses creative classifications and co-location of goods to shape browsing behavior and encourage impulse buying.[3]

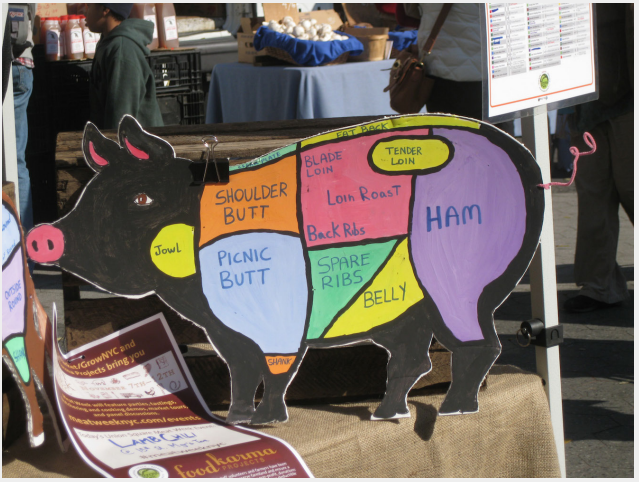
In business-to-business contexts, standard classifications for business processes and their application interfaces enable firms to more easily build and maintain supply chains and distribution networks that interconnect many business partners.[4]

2. (Grappone and Couzin 2011) is a search engine optimization "cookbook" for do-it-yourselfers. See (Malaga 2008) for a critique of typical SEO practices.

3. See (Gladwell 1996), (Schwartz 2005), (Underhill 2008).

4. The RosettaNet standards are used by thousands of firms as specifications and implementations of business-to-business processes in several industries, especially component manufacturing and electronics. The specifications are defined using a three-level hierarchy of process clusters, segments, and partner interface processes (PIPs) to enable firms to find a level of process abstraction that works best for them. See http://RosettaNet.org.

Classification In A Novel User Interface



The meat from animals used as food is classified into numerous "cuts" based on its origin. In the US, these classifications are standardized by the Department of Agriculture to ensure that meat is labeled correctly. The most natural way to convey the classification system is to label the parts of the animal in a diagram, because this binds each logical category to the "user interface."

(Photo by R. Glushko. Taken in 2011 at the Union Square Greenmarket in New York City.)

Classification Is Principled

"Principles for Creating Categories" explained principles for

creating categories, including enumeration, single properties, multiple properties and hierarchy, probabilistic co-occurrence of properties, theory and goal-based categorization. It logically follows that the principles considered in designing categories are embodied in classifications that use those categories. However, when we say, "classification is principled," we are going further to say that the processes of assigning resources to categories and maintaining the classification scheme over time must also follow principles.

The design and use of a classification system involves many choices about its purposes, scope, scale, intended lifetime, extensibility, and other considerations. Principled classification means that once those design choices are made they should be systematically and consistently followed.

Principled does not necessarily equate to "good," because many of the choices can be arbitrary and others may involve tradeoffs that depend on the nature of the resources, the purposes of the classification, the amount of effort available, the complexity of the domain, and the capabilities of the people doing the classification and of the people using it (see "Category Design Issues and Implications"). Every classification system is biased in one way or another (see "Bibliographic Classification").

Consider the classifications of resources in a highly-organized kitchen. (See "Organizing a Kitchen"). Tableware, dishes, pots and pans, spices and food provisions, and other resources have dedicated locations determined by a set of intersecting requirements and organizing principles. There is no written specification, and other people organize their kitchens differently.

On the other hand, complex institutional classification systems like those used in libraries or government agencies are implemented with detailed specifications, methods, protocols, and guidelines. The people who apply these methods in the field have studied the protocols in school or they have received extensive on-the-job

training to ensure that they apply them correctly, consistently, and in accordance with the specifications and guidelines.

Principles Embodied in the Classification Scheme

Some of the most important principles that lead us to say that classification is principled are those that guide the design of the classification scheme in the first place. These principles are fundamental in the discipline of library science but they apply more broadly to other domains.

The *warrant* principle concerns the justification for the choice of categories and the names given to them. The principle of *literary warrant* holds that a classification must be based only on the specific resources that are being classified. In the library context, this *ad hoc* principle that builds a classification from a particular collection principle is often posed in opposition to a more philosophical or epistemological perspective, first articulated by Francis Bacon in the seventeenth century, that a classification should be universal and must handle all knowledge and all possible resources.[5]

The principle of *scientific warrant* argues that only the categories recognized by the scientists or experts in a domain should be used in a classification system, and it is often opposed by the principle of *use* or *user warrant*, which chooses categories and descriptive

5. See (Gaukroger 2001) and (Weinberger 1985) for an introduction to Bacon's philosophy, and (Miksa 1984) for an analysis of Bacon's influence on systems of library classification.

terms according to their frequency of use by everyone, not just experts.[6]

With classifications of physical resources like those in a kitchen, we see *object warrant*, where similar objects are put together, but more frequently the justifying principle will be one of use warrant, where resources are organized based on how they are used.

> Starbucks Coffee Sizes: "Anti-User" Warrant?
>
> The Starbucks coffee chain seemingly goes out of its way to confuse its customers by calling the smallest (twelve ounces) of its three coffee sizes the "tall" size, calling its sixteen-ounce size a "*grande*," and calling its largest a "*venti*," which is Italian for twenty (ounces). Outside of Starbucks, something that is "tall" is never also considered "small." Ironically, despite having more than five thousand coffeehouses in over fifty countries, Starbucks has none in Italy where *venti* would be in the local language.

A second principle embodied in a classification scheme concerns the breadth and depth of the category hierarchy. We discussed this in "Category Design Issues and Implications" but in the context of classification this principle has additional implications and is framed as the extent to which the scheme is *enumerative* ("Classification vs. Physical Arrangement"). The decision to classify broadly or precisely depends largely on the variety or heterogeneity of the resources that the system of categories has been designed to organize. Because of the diversity of resources for a sale in a department

6. (Svenonius 2000, Ch. 8).

store, a broad classification is necessary to accommodate everything in the store. Kitchen goods will be grouped together in a few aisles on a single floor. But a specialty kitchen store or a wholesale kitchen supply store for restaurants would classify much more precisely because of the restricted resource domain and the greater expertise of those who want to buy things there. An entire section might be dedicated just to knives, organized by knife type, manufacturer, quality of steel, and other categories that are not used in the kitchen section of the department store.[7]

The precision or enumerativeness of a classification scheme increases the similarity of resources that are assigned to the same category and sharpens the distinctions between resources in different categories. However, when different classifications must be combined, mismatches in their precision or granularity can create challenges (see "Reorganizing Resources for Interactions").

Principles for Assigning Resources to Categories

The *uniqueness principle* means the categories in a classification scheme are mutually exclusive. Thus, when a logical concept is assigned to a particular category, it cannot simultaneously be assigned to another category. Resources, however, can be assigned to several categories if they embody several concepts represented by those different categories. This can present a challenge when a physical storage solution is based on storing resources according to its assigned category in a logical classification system. This is not a serious problem for resource types like technical equipment

7. Very detailed classification of knives are at http://www2.knifecenter.com/knifecenter/kitchen/ and http://kitchenknives.com/.

or tools, for which the properties used to classify them are highly salient, and that have very narrow and predictable contexts of use. It is also not a problem for highly-specialized information resources like scientific research reports or government economic data, which might end up in only one specialized class. However, many resources are inherently more difficult to classify because they have less salient properties or because they have many more possible uses.

We face this kind of problem all the time. For example, should we store a pair of scissors in the kitchen or in the office? One solution is to buy a second pair of scissors so that scissors can be kept in both locations where they are typically used, but this is not practical for many types of resources and this principle would be difficult to apply in a systematic manner.

Many books are about multiple subjects. A self-help book about coping with change in a business setting might reasonably be classified as either about applied psychology or about business. It is not helpful that book titles are often poor clues to their content; *Who Moved My Cheese?* is in fact a self-help book about coping with change in a business setting. Its Library of Congress Classification is BF 637, "Applied Psychology," and at UC Berkeley it is kept in the business school library.

The general solution to satisfying the *uniqueness principle* in library classifications when resources do not clearly fit in a single category is to invent and follow a detailed set of often arbitrary rules. Usually, the primary subject of the book is used for assigning a category, which will then determine the book's place on a shelf.

However, another rule might state that if a book treats two subjects equally, the subject that is covered first determines the classification. For some classifications a "table of preference" can trump other rules at the last minute. Not surprisingly, the rules for

categorizing books take a long time to learn and are not always easy to apply.[8]

Principles for Maintaining the Classification over Time

Most personal classifications are created in response to a specific situation to solve an emerging organizational challenge. As a consequence, personal classification systems change in an *ad hoc* or opportunistic manner during their limited lifetimes. For example, the classification schemes in your kitchen or closet are deconstructed and disappear when you move and take your possessions to a different house or apartment. Your efforts to re-implement the classifications will be influenced by the configuration of shelves and cabinets in your new residence, so they will not be exactly the same.

In contrast, the institutional classification schemes for many library resources, culturally or scientifically-important artifacts, and much of the information created or collected by businesses, governments and researchers might have useful lives of decades or centuries. Classification systems like these can only be changed incrementally to avoid disruption of the work flows of the organization. We described maintaining resources as an activity in all organizing

8. For example, the introductory text for the Dewey Decimal Classification(DDC) is 38 pages long (http://www.oclc.org/dewey/resources/scholar.htm). A full set of online training modules "focused on the needs of experienced librarians needing Dewey application training" runs 30 hours (http://www.oclc.org/dewey/resources/teachingsite/courses/default.htm).

systems (“Maintaining Resources”) and the issues of persistence, effectivity, authenticity, and provenance that emerge with resources over time (“Resources over Time”). Much of this previous discussion applies in a straightforward manner to maintaining classifications over time.

However, some additional issues arise with classifications over time. The warrant principle (“Principles Embodied in the Classification Scheme”) implicitly treats the justification for designing and naming categories as a one-time decision. This is reasonable if you are organizing a collection of bibliographic resources or common types of physical resources like printed books, clothing or butterflies. However, in domains where the resources are active, change their state or implementation, or otherwise have a probabilistic character it might be necessary to revisit warrant and the decisions based on it from time to time. Put another way, if the world that you are sampling from or describing has some randomness or change in it, the categories and descriptions you imposed on it probably need to change as well. It often happens that the meaning of an underlying category can change, along with its relative and absolute importance with respect to the other categories in the classification system. Categories sometimes change slowly, but they can also change quickly and radically as a result of technological, process, or geopolitical innovation or events. Entirely new types of resources and bodies of knowledge can appear in a short time. Consider what the categories of “travel,” “entertainment,” “computing,” and “communication” mean today compared to just a decade or two ago.

Changes in the meaning of the categories in a classification threaten its *integrity*, the principle that categories should not move within the structure of the classification system.[9]

9. (Taylor and Joudrey 2009, p. 392) define integrity as the stability of notations (class identifiers) in a classification

One way to maintain integrity while adapting to the dynamic and changing nature of knowledge is to define a new version of a classification system while allowing earlier ones to persist, which preserves resource assignments in the previous version of the classification system while allowing it to change in the new one. If we adopt a logical perspective on classification ("Classification vs. Tagging") that dissociates the conceptual assignment of resources to categories from their physical arrangement, there is no reason why a resource cannot have contrasting category assignments in different versions of a classification.

However, the conventional library with collections of physical resources cannot easily abandon its requirement to use a classification to arrange books on shelves in specific places so they can be located, checked out, and returned to the same location.

This constraint does not preclude the versioning of library classifications, but it increases the inertia and limits the degree of change when revisions are made because of the cost and coordination considerations of rearranging books in all the world's libraries.

A related principle about maintaining classifications over time is *flexibility*, the degree to which the classification can accommodate new categories. Computer scientists typically describe this principle as *extensibility*, and library scientists sometimes describe

so that resources are never given new notations when the category meaning changes. This is especially pertinent in a physical world where class notations are affixed to resources (books in a traditional library, for example) and where the changing of meaning would necessitate the changing of many numbers.

it as *hospitality*. In any case the concern is the same and we are all familiar with it. When you buy a bookshelf, clothes wardrobe, file cabinet, or computer, it makes sense to buy one that has some extra space to accommodate the books, clothes, or files you will acquire over some future time frame. As with other choices that need to be made about organizing systems, how much extra space and "organizing room" you will acquire involves numerous tradeoffs.

Classification schemes can increase their flexibility by creating extra "logical space" when they are defined. Library classifications accomplish this by using naming or numbering schemes for classification that can be extended easily to create new subcategories.[10]

Classification schemes in information systems can also anticipate the evolution of document or database schemas.[11]

10. For example, the *Universal Decimal Classification*(UDC) intentionally left the main class 4 blank in order to have space for currently unknown subjects on the highest hierarchy level. (http://www.udcc.org/udcsummary/php/index.php). The Library of Congress Classification(LCC) also left space on the highest hierarchical level by not using all letters in the alphabet. Classifications also leave spaces in the enumeration of more specific classes.

11. (Rahm and Bernstein 2006) provide a crisp introduction to the challenges and approaches for changing deployed schemas in databases, conceptual models, ontologies, XML schemas, and software application interfaces. They operate an online bibliography on schema evolution that

Classification Is Biased

The discipline of organizing is fundamentally about choices of properties and principles for describing and arranging resources. We discussed choices about describing resources in "The Process of Describing Resources", choices for creating resource categories in "Principles for Creating Categories", and choices for creating classifications in this chapter. The choices made reflect the purposes, experiences, professions, politics, values, and other characteristics and preferences of the people making them. As a result, every system of classification is biased because it takes a point of view that is a composite of all of these influences.

> Statistical Bias and Variance
>
> Statistical bias is the systematic error in measurements introduced by miscalibration of the measurement instrument, by ineffective measurement techniques, an algorithm that makes incorrect assumptions, or some environment interference, all of which distort the measured value in a predictable way. Measurement bias contrasts with the variability or variance of a measurement, the amount of dispersion around an average or expected value, most often due to random factors. Some variance arises because the property being measured is not the same for all instances, as we would expect for measurements of the

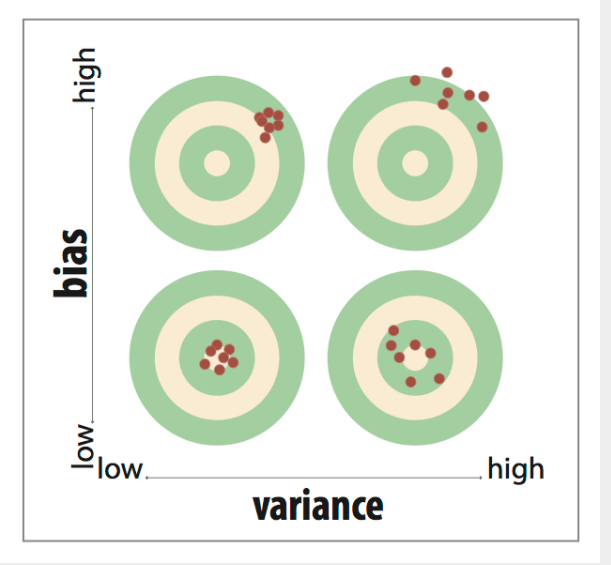
contains several hundred sources. See http://se-pubs.dbs.uni-leipzig.de/.

weight of a random sample of people, or in the set of tags or topics assigned to a random sample of news articles by people or algorithms. By analyzing a large enough set of instances it is possible to determine the most likely values of the property and also to estimate the amount of random error.

High variance in the measurements for a sample of resources when we expect all of them to have more similar values can be a quality problem. High bias, on the other hand, might be less of a quality problem, because systematic sources of inaccuracy might be easier to correct.

But first we need to point out that there are at least two quite different senses of "bias" that people reading this book are likely to encounter. The colloquial sense of bias we discuss in this section reflects value-based decisions in organizing systems that implicitly or explicitly favor some interactions or users over others. In contrast, statistical bias is systematic error or distortion in a measurement. (See the sidebar, Statistical Bias and Variance.)

Bias and Variance on Dartboards



Precise and accurate dart throws demonstrate low bias and low variance (lower left in the figure). Precise but inaccurate darts reflect high bias and low variance (upper left). Imprecise but accurate ones have low bias but high variance (lower right). Finally, a lack of accuracy and precision shows both high bias and high variance (upper right).

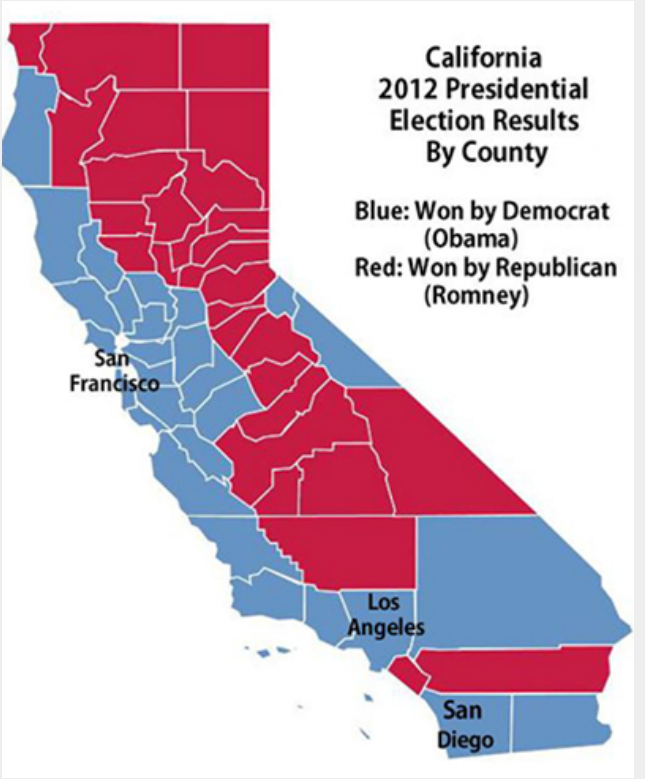
The claim that classification is biased might seem surprising, because many classification systems are formal and institutional, created by governments or firms participating in standards organizations. We expect these classifications to be impartial and objective. However, consider the classification of people as "employed" or "unemployed." Many people think that any employable person who is not currently employed would be counted as unemployed. But the US government's Department of Labor only counts someone as unemployed if they have actively

looked for work in the past month, effectively removing anyone who has given up on finding work from the unemployed category by assigning them to a "discouraged worker" category. In 2012 this classification scheme allowed the government to report that unemployment was about 8% and falling, when in fact it was closer to 20% and rising. The political implications of this classification are substantial.[12]

Classification bias is often intentionally or unintentionally shown in data visualizations, including choropleth maps, in which map regions are colored, patterned, or otherwise distinguished according to a statistical variable being displayed on the map. Choropleths are commonly used to display election results, with the districts or states won by each candidate shown in different colors; in the United States, the convention is to show those won by Democratic Party candidates in blue, and those won by Republicans in Red. These election choropleths are often misleading because coloring an entire state in the winner's colors ignores population density and the regional concentrations of votes that differ from the majority.

12. See How the Government Measures Unemployment, http://www.bls.gov/cps/cps_htgm.htm from the Department of Labor's Bureau of Labor Statistics, and a critical commentary about the measurement scheme titled Making 9 Million Jobless Vanish: How the Government Manipulates Unemployment Statistics at http://danielamerman.com/articles/2012/WorkC.html.

California Election Map:
A “Blue” State with More “Red” Counties



California voters are reliably “blue” as a whole, but with election results divided by county, it is clear that this majority is amassed in the large cities along the coast, and inland and rural counties are more reliably “red” in their voting.

A more subtle way in which choropleths encode bias reflects the decisions made to organize the data into the categories that are represented by different colors or patterns. Choropleth categories might present data divided into equal range intervals, into sets with the same number of observations, or into categories that reflect clusters or natural breaks in the observed data. Small changes in the data ranges or proportions that are then assigned to each category can communicate entirely different stories with the same data. To learn "how to lie with maps" or how to prevent being lied to, refer to the classic book with that title by Mark Monmonier.[13]

Friedman and Nissenbaum's *Bias In Computer Systems* offers a framework for conceptualizing the various types of bias that may be present in technical systems. Friedman and Nissenbaum define bias as "a system that systematically and unfairly discriminates against individuals or groups of individuals in favor of others" Their taxonomy includes pre-existing, technical, and emergent bias.[14]

Pre-existing bias is the type people are most familiar with: it occurs when an organizing system's design embodies personal or societal biases that exist at the time of its creation, either intentionally or inadvertently, and sometimes despite one's best intentions to prevent it.

Technical bias arises from limitations and constraints of technical systems that result in unfairness when the system is applied to the real world. Automated decision-making is especially ripe for this

13. (Monmonier 1996) is a highly-readable treatment of intentional and inadvertent bias in mapmaking. A web search for "lying with maps" yields a large number of examples. See also When Maps Lie" by Wiseman

14. (Friedman and Nissenbaum 1996)

sort of bias: alphabetical ordering, processes that rely on pseudo-random number generation, and other automated ways of sorting or grouping resources may systematically create different opportunities for different user groups (e.g., people or companies whose names begin with "A").

Emergent bias is related to the interplay between actual users and a technical system. Problems of this type arise when, due to the designer's incomplete understanding of the user population, or a change in that population over time, there is a mismatch between users and the system. User interfaces are especially susceptible to this form of bias, given their need to reflect the habits and capacities of intended users. Unfairness can emerge when an unexpected user group uses the system, or as new societal knowledge arises that the system is not able to incorporate or respond to.

Both pre-existing and emergent bias may be difficult to assess accurately; the former may be difficult for the biased to see or admit to, and the latter, arising due to unanticipated circumstances after implementation, is hard to predict.

Bowker and Star have written extensively about biases in classification systems but acknowledge that many people do not see them:

> *Information scientists work every day on the design, delegation and choice of classification systems and standards, yet few see them as artifacts embodying moral and aesthetic choices that in turn craft people's identities, aspirations and dignity.*[15]
> (Bowker and Star 2000)

15. (Bowker and Star 2000, p. 4).

Bowker and Star describe many examples where seemingly neutral and benign classifications implement controversial assumptions. A striking example is found in the ethnic classifications of the United States Census and the categories to which US residents are required to assign themselves. These categories have changed nearly every decade since the first census in 1790 and strongly reflect political goals, prevailing cultural sensitivities or lack thereof, and non-scientific considerations. Some recent changes included a "multi-racial" category, which some people viewed as empowering, but which was attacked by African-American and Hispanic civil rights groups as diluting their power.[16]

A more positive way to think about bias in classification is that the choices made in an organizing system about resource selection, description, and arrangement come together to convey the values of the organizers. This makes a classification a rhetorical or communicative vehicle for establishing credibility and trust with those who interact with the resources in the classification. Seen in this light, an objective or neutral classification is not only unrealistic as a goal; it may also consume valuable time and energy when instead it might be more desirable to seize the opportunity to interpret the resources in a creative way to communicate a particular message to a particular user group. Melanie Feinberg makes the point that "fair trade" or "green" supermarkets differentiate themselves by a relatively small proportion of the

16. See the Wikipedia entry Race and ethnicity in the United States census, http://en.wikipedia.org/wiki/Race_and_ethnicity_in_the_United_States_Census, and (Lee 1993) for arguments against any racial categorization because of the "political motivations and non-scientific character of the classifications."

goods they offer compared with ordinary stores, but these particular items signal the values that their customers care most about.[17]

Bias is clearly evident in the most widely used bibliographic classifications, the Library of Congress and the Dewey Decimal, which we discuss next.

17. (Feinberg 2012).

53. Bibliographic Classification

Much of our thinking about classification comes from the bibliographic domain. Libraries and the classification systems for the resources they contain have been evolving for millennia, shaped by the intellectual, social, and technological conditions of the societies that created them. As early as the third millennium BCE, there were enough written documents—papyrus scrolls or clay tablets—that the need arose to organize them. Some of the first attempts, by Mesopotamian scribes, were simple lists of documents in no particular order. The ancient Greeks, Romans, and Chinese created more principled systems, both sorting works by features such as language and alphabetical order, and placing them into semantically significant categories such as topic or genre. Medieval European libraries were tightly focused on Christian theology, but as secular books and readers proliferated thanks to new technologies and increased literacy, bibliographic classifications grew broader and more complex to accommodate them. Modern classification systems are highly nuanced systems designed to encompass all knowledge; however, they retain some of the same features and biases of their forebears.[1]

1. One of the earliest known libraries—at Nippur in Mesopotamia—was small enough that its catalog needed no particular organization: the list of titles in the collection fit onto two easily scanned clay tablets. As collections grew, scribes made it easier to browse the contents of a collection by adding "colophons," brief

descriptions containing a document's title, author, and place in a sequence of tablets (Casson 2002). A further step was the sorting of works into categories. A temple in the ancient Egyptian city of Edfu placed books into different trunks based on their topics, including royal duties, temple management, and timekeeping, as well as two trunks each for astronomy and protection from crocodiles. The fabled library of Alexandria in ancient Greece used categories based on Aristotle's three modes of thought: theoretical (e.g. mathematics, physics, metaphysics), practical (ethics, politics, economics), and poetic (poetry, music, and art), plus a fourth "meta-category," logic, that applied to all of them. Callimachus, one of the library's directors, created the Pinakes, a library catalog whose top-level distinction was between poetry and prose (followed by genre, author, and work). A few centuries later, librarians in the Chinese Wei and Jin dynasties (third-fifth centuries CE) settled on four major categories—classics, philosophy, history, and literature—that lasted well into the twentieth century. (Shamurin 1955) Unlike the Greek system, which classified authors, the Chinese system classified individual works; some authors have suggested that this reflects Western cultures' greater emphasis on the individual. Medieval libraries adapted ancient practices for their own needs: monastery libraries had separate cabinets for topics such as Bibles, Church history, and

We will briefly describe the most important systems for bibliographic classification, especially the Dewey Decimal Classification(DDC) and *Library of Congress Classification(LCC)* systems. However, there are several important ways in which bibliographic classification is distinctive and we will discuss those first:

Scale, Complexity, and Degree of Standardization:

Department stores and supermarkets typically offer tens of

Christian poets, and divided their collections into Christian and secular literature (meanwhile, scholars in the intellectually flourishing Muslim world classified knowledge into Muslim and non-Muslim sciences) (Christ 1984). Today's classification systems reflect both their debt to earlier systems and the biases of their own cultures: the first category of the Universal Decimal Classification, just like Aristotle's "logic" category, is a meta-category covering organization, documentation, and information science, while the first top-level category of the Chinese Classification System is "Marxism, Leninism, Maoism, and Deng Xiaoping theory."

(Taylor and Joudrey 2009, Ch. 3) is a historical review of library classification. (Svenonius 2000) reviews the evolution of the theoretical foundations. (Kilgour 1998) focuses on the evolution of the book and the story of the co-evolution of libraries and classification comes along for the ride.

thousands of different items (as measured by the number of "stock keeping units" or SKUs), and popular online commerce sites like Amazon.com and eBay are of similar scale. However, the standard product classification system for supermarkets has only about 300 categories.[2] The classifications for online stores are typically deeper than those for physical stores, but they are highly idiosyncratic and non-standard. In contrast, scores of university libraries have five million or more distinct items in their collections, and they almost all use the same standard bibliographic classification system that has about 300,000 distinct categories.[3]

Legacy of Physical Arrangement, User Access, and Re-Shelving:

A corollary to the previous one that distinguishes bibliographic classification systems is that they have long been shaped and continue to be shaped by the legacy of physical arrangement, user access to the storage locations, and re-shelving that they support. These requirements constrain the evolution and extensibility of bibliographic classifications, making them less able to keep pace with changing concepts and new bodies of knowledge. Amazon classifies the products it sells in huge

2. Supermarkets typically carry anywhere from 15,000 to 60,000 SKUs (depending on the size of the store), and may offer a service deli, a service bakery, and/or a pharmacy. 300 standard product categories (http://www.fmi.org/research-resources/supermarket-facts).

3. http://www.ala.org/tools/libfactsheets/alalibraryfactsheet22.

warehouses, but its customers do not have to pick out their purchases there, and most goods never return to the warehouse. Amazon can add new product categories and manage the resources in warehouses far more easily than libraries can.

With digital libraries, constraints of scale and physical arrangement are substantially eliminated, because the storage location is hidden from the user and the resources do not need to be returned and re-shelved. However, when users can search the entire content of the library, as they have learned to expect from the web, they are less likely to use the bibliographic classification systems that have painstakingly been applied to the library's resources.

The Dewey Decimal Classification

The Dewey Decimal Classification(DDC) is the world's most widely used bibliographic system, applied to books in over 200,000 libraries in 135 countries. It is a proprietary and *de facto* standard, and it must be licensed for use from the Online Computer Library Center(OCLC).[4]

In 1876, Melvil Dewey invented the DDC when he was hired to manage the Amherst College library immediately after graduating. Dewey was inspired by Bacon's attempt to create a universal classification for all knowledge and considered the DDC as a numerical overlay on Bacon with 10 main classes, each divided into 10 more, and so on. Despite his explicit rejection of literary warrant,

4. Dewey Decimal Classification: http://www.oclc.org/dewey/.

however, Dewey's classification was strongly influenced by the existing Amherst collection, which reflected Amherst's focus on the time on the "education of indigent young men of piety and talents for the Christian ministry."[5]

The resulting nineteenth-century Western bias in the DDC's classification of religion seems almost startling today, where it persists in the 23rd revision (see Figure: "Religion" in Dewey Decimal Classification.). "Religion" is one of the 10 main classes, the 200 class, with nine subclasses, Six of these nine subclasses are topics with "Christian" in the name; one class is for the Bible alone; and another section is entitled "Natural theology." Everything else related to the world's many religions is lumped under 290, "Other religions."

"Religion" in Dewey Decimal Classification

```
200 Religion
  210 Natural Theology
  220 Bible
  230 Christian theology
  240 Christian moral and devotional theology
  250 Christian orders and local church
  260 Christian social theology
  270 Christian church history
  280 Christian sects and denominations
  290 Other religions
```

The notational simplicity of a decimal system makes the DDC easy to use and easy to subdivide existing categories, So-called subdivision tables allow facets for language, geography or format to

5. https://www.amherst.edu/aboutamherst/history. Today Amherst is aggressively co-ed and secular.

be added to many classes, making the classification more specific. But the overall system is not very hospitable to new areas of knowledge.

The Library of Congress Classification

The US Library of Congress is the largest library in the world today, but it got off to a bad start after being established in 1800. In 1814, during the War of 1812, British troops burned down the US Capitol building where the library was located and the 3000 books in the collection went up in flames.[6] The library was restarted a year later

6. That was not a typo. The "War of 1812" lasted well into 1815. The persistence of an inaccurate name for this war reflects its unique characteristics. Wars (in the English language) are generally named for the location of the fighting or the enemy being fought (the Mexican-American War, the Korean War, the Vietnam War, the Iraq War), or for a particular ideal or ambition (the Revolutionary War, the Civil War). The War of 1812 does not satisfy any of these naming conventions; the war was fought across a huge range of geography from eastern Canada to Louisiana, between a diverse range of groups from Canadians and Native American tribes, with national armies getting involved very late in the war. While nominally fought over freedom the seas, the war quickly morphed into one about territorial ambition in North America. Of course, if the world were a place

when Congress purchased the personal library of former president Thomas Jefferson, which was over twice the size of the collection that the British burned. Jefferson was a deeply intellectual person, and unlike the narrow historical and legal collection of the original library, Jefferson's library reflected his "comprehensive interests in philosophy, history, geography, science, and literature, as well as political and legal treatises."[7]

Restarting the Library of Congress around Jefferson's personal collection and classification had an interesting implication. When Herbert Putnam formally created the Library of Congress Classification (LCC) in 1897, he meant it not as a way to organize all the world's knowledge, but to provide a practical way to organize and later locate items within the Library of Congress's collection. However, despite Putnam's commitment to literary warrant, the breadth of Jefferson's collection made the LCC more intellectually ambitious than it might otherwise had been, and probably contributed to its dominant adoption in university libraries.

The LCC has 21 top-level categories, identified by letters instead of using numbers like the DDC (see Figure: Top Level Categories in the Library of Congress Classification.). Each top-level category is divided into about 10-20 subclasses, each of which is further subdivided. The complete LCC and supporting information takes up 41 printed volumes.

where people could agree on naming standards for wars, it is likely we would no longer have wars. See http://en.wikipedia.org/wiki/List_of_wars_involving_the_United_States.

7. (Miksa 1984, p. 3).

Top Level Categories in the
Library of Congress Classification

```
A — GENERAL WORKS
B — PHILOSOPHY. PSYCHOLOGY. RELIGION
C — AUXILLARY SCIENCES OF HISTORY (GENERAL)
D — WORLD HISTORY (EXCEPT AMERICAN HISTORY)
E — HISTORY: AMERICA
F — HISTORY: AMERICA
G — GEOGRAPHY. ANTHROPOLOGY. RECREATION
H — SOCIAL SCIENCE
J — POLITICAL SCIENCE
K — LAW
L — EDUCATION
M — MUSIC
N — FINE ARTS
P — LANGUAGE AND LITERATURE
Q — SCIENCE
R — MEDICINE
S — AGRICULTURE
T — TECHNOLOGY
U — MILITARY SCIENCE
V — NAVAL SCIENCE
Z — BIBLIOGRAPHY. LIBRARY SCIENCE
```

Bias is apparent in the LCC as it is in the DDC, but is somewhat more subtle. A library for the US emphasizes its own history. "Naval science" was vastly more important in the 1800s when it was given its own top level category, separated from other resources about "Military science" (which had a subclass for "Cavalry").[8]

The LCC is highly enumerative, and along with the uniqueness

8. For additional examples, (Shirky 2005).

principle, this creates distortions over time and sometimes requires contortions to incorporate new disciplines. For example, it might seem odd today that a discipline as broad and important as computer science does not have its own second level category under the Q category of science, but because computer science was first taught in math departments, the LCC has it as the QA76 subclass of mathematics, which is QA.[9]

The BISAC Classification

A very different approach to bibliographic classification is represented in the *Book Industry Standards Advisory Committee classification*(BISAC). BISAC is developed by the *Book Industry Study Group*(*BISG*), a non-profit industry association that "develops, maintains, and promotes standards and best practices that enable the book industry to conduct business more efficiently." The BISAC classification system is used by many of the major businesses within the North American book industry, including Amazon, Baker & Taylor, Barnes & Noble, Bookscan, Booksense, Bowker, Indigo, Ingram and most major publishers.[10]

9. Cognitive Science has an even harder time finding its proper place in the LCC because it emerged as the intersection of psychology, linguistics, computer science, and other disciplines. Cognitive science books can be found scattered throughout the LCC, with concentrations in BF, P, and QA.

10. The Book Industry Study Group(BISG) first and foremost is focused on resource description and classification as

The BISAC classifications are used by publishers to suggest to booksellers how a book should be classified in physical and online bookstores. Because of its commercial and consumer focus, BISAC follows a principle of use warrant, and its categories are biased toward common language usage and popular culture. Some top-level BISAC categories, including Law, Medicine, Music, and Philosophy, are also top-level categories in the LCC. However, BISAC also has top-level categories for Comics & Graphic Novels. Cooking, Pets, and True Crime.

The differences between BISAC and the LCC are understandable because they are used for completely different purposes and generally have little need to come into contact. This changed in 2004, when Google began its ambitious project to digitize the majority of the world's books. (See the sidebar, What Is a Library?). To the dismay of many people in the library and academic community, Google initially classified books using BISAC rather than the LCC.[11]

In addition, some new public libraries have adopted BISAC rather than the DDC because they feel the former makes the library friendlier to its users. Some librarians believe that their online catalogs need to be more like web search engines, so a less precise

means to business ends; this purpose contrasts with goals of DDC or LOC. BISG classifications are used for barcodes and shipping labels to support supply chain and inventory management, marketing, and promotion activities. See http://www.bisg.org/.

11. See (Pope and Holley 2011), (Samuelson 2010).

classification that uses more familiar category terms seems like a good choice.[12]

12. What some call the "Perry Rebellion" or the "Dewey Dilemma" began in 2007 when the new Perry Branch Library in Gilbert, Arizona opened with its books classified using the BISAC rather than Dewey classifications. (Fister 2009). This is a highly inflamed controversy that pits advocates of customer service and usability against the library establishment, which despises the idea of turning to retailing as inspiration when designing and operating a library. Even if BISAC gets more widely adopted in public libraries it is unimaginable that it can be used in research libraries.

54. Faceted Classification

We have noted several times that strictly enumerative classifications constrain how resources are assigned to categories and how the classification can evolve over time. *Faceted classifications* are an alternative that overcome some of these limitations. In a *faceted classification* system, each resource is described using properties from multiple facets, but a person searching for resources does not need to consider all of the properties (and consequently the facets) and does not need to consider them in a fixed order, which an enumerative hierarchical classification requires.

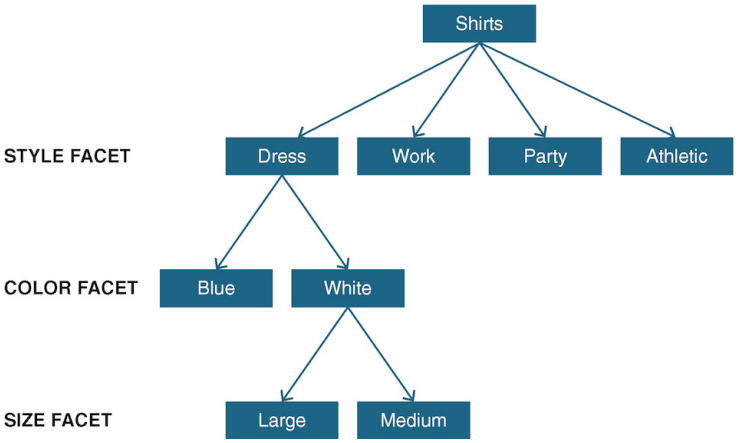
Faceted classifications are especially useful in web user interfaces for online shopping or for browsing a large and heterogeneous museum collection. The process of considering facets in any order and ignoring those that are not relevant implies a dynamic organizational structure that makes selection both flexible and efficient. We can best illustrate these advantages with a shopping example in a domain that we are familiar with from "Multiple Properties".

If a department store offers shirts in various styles, colors, sizes, brands, and prices, shoppers might want to search and sort through them using properties from these facets in any order. However, in a physical store, this is not possible because the shirts must be arranged in actual locations in the store, with dress shirts in one area, work shirts in another, and so on.

Assume that the shirt store has shirts in four styles: dress shirts, work shirts, party shirts, and athletic shirts. The dress shirts come in white and blue, the work shirts in white and brown, and the party and athletic shirts come in white, blue, brown, and red. White dress shirts come in large and medium sizes.

Suppose we are looking for a white dress shirt in a large size. We can think of this desired shirt in two equivalent ways, either as a member of a category of "large white dress shirts" or a shirt with "dress," "white," and "large" values on style, color, and size facets. Because of the way the shirts are arranged in the physical store, our search process has to follow a hierarchical structure of categories. We go to the dress shirt section, find white shirts, and then look for a large one. This process corresponds to the hierarchy shown in Figure: Enumerative Classification with Style Facet Followed by Color Facet.

Enumerative Classification with
Style Facet Followed by Color Facet



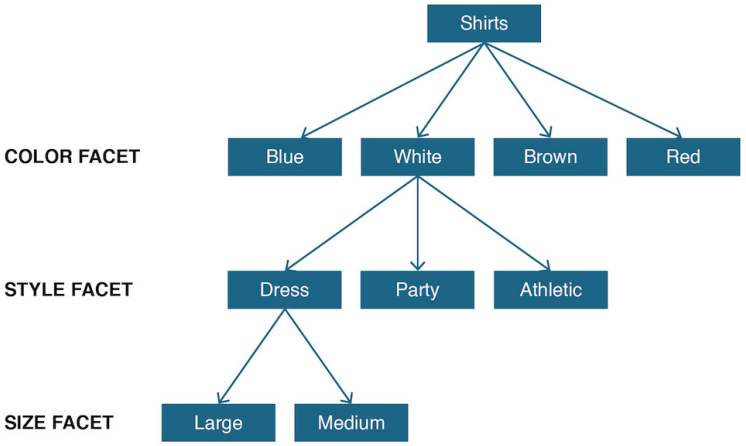
In an enumerative classification system the order of the facets determines the classification hierarchy. For example, a store might classify shirts first using a style facet, next with a color facet, and finally with a size facet. This ordering could result in two piles of dress shirts, one blue and one white, in which each pile contains shirts of large and medium sizes.

Although unlikely, a store might choose to organize its shirts by

color. In our search for a "white dress shirt in a large size," if we consider the color first, because shirts come in four colors, there are four color categories to choose from. When we choose the white shirts, there is no category for work shirts because there are no work shirts that come in white. We then choose the dress shirts, and then finally find the large one. (Figure: Enumerative Classification with Color Facet Followed by Style Facet.)

This department store example shows that for a physical organization, one property facet guides the localization of resources; all other facets are subordinated under the primary organizing property. In hierarchical enumerative classifications, this means that the primary organizing facet determines the primary form of access. The shirts are either organized by style and then color, or by color then style, which enforces an inflexible query strategy (style first or color first).

Enumerative Classification with
Color Facet Followed by Style Facet



An alternative ordering of the same shirt facets changes the

classification hierarchy. If the first facet considered is color, style is next, and finally size, this ordering could result in two piles of white shirts, one for dress shirts and one for athletic shirts, in which each pile contains shirts of large and medium sizes.

In an online store, however, descriptions of the shirts are being searched and sorted instead of the real shirts, and different organizations are possible. When the shirts are described using a faceted classification system, we treat all facets independently (i.e., they can all be the primary facet).

We can enumerate all the properties needed to assign resources appropriately, but we create the categories (i.e., union of properties from different facets) only as needed to sort resources with a particular combination of properties.

An additional aspect of the flexibility of faceted classification is that a facet can be left out of a resource description if it is not needed or appropriate. For example, because party shirts are often multi-colored with exotic patterns, it is not that useful to describe their color. Likewise, certain types of athletic shirts might be very loose-fitting, and as a result not be given a size description, but their color is important because it is tied to a particular team. Figure: Faceted Classification. shows how these two resource types can be classified with the faceted Shirt classification. Resource 1 describes a party shirt in medium; resource 2 describes an athletic shirt in blue without information about size.

Faceted Classification

In a pure faceted classification, not every facet needs to apply to every resource, and there is no requirement for a predetermined order in which the facets are considered.

A faceted classification scheme like that shown in Figure: Faceted Classification. eliminates the requirement for predetermining a combination and ordering of facets like those in Figure: Enumerative Classification with Style Facet Followed by Color Facet. and Figure: Enumerative Classification with Color Facet Followed by Style Facet. Instead, imagine a shirt store where you decide when you begin shopping which facets are important to you ("show me all the medium party shirts," "show me the blue athletic shirts") instead of having to adhere to whatever predetermined (pre-combined) enumerative classification the store invented. In a digital organizing system, faceted classification enables highly flexible access because prioritizing different facets can dynamically reorganize how the collection is presented.

Foundations for Faceted Classification

In library and information science texts it is common to credit the idea of faceted classification to S.R. Ranganathan, a Hindu mathematician working as a librarian. Ranganathan had an almost mystical motivation to classify everything in the universe with a single classification system and notation, considering it his dharma (the closest translation in English would be "fundamental duty" or "destiny"). Facing the limitations of Dewey's system, where an item's essence had to first be identified and then the item assigned to a category based on that essence, Ranganathan believed that all bibliographic resources could be organized around a more abstract variety of aspects.

In 1933 Ranganathan proposed that a set of five facets applied to all knowledge:

Personality

The type of thing.

Matter

The constituent material of the thing.

Energy

The action or activity of the thing.

Space

Where the thing occurs.

Time

When the thing occurs.

This classification system is known as colon classification (or

PMEST) because the notation used for resource identifiers uses a colon to separate the values on each facet. These values come from tables of categories and subcategories, making the call number very compact. Colon classification is most commonly used in libraries in India.[1]

For example, a book on "research in the cure of tuberculosis of lungs by x-ray conducted in India in 1950" has a Personality facet value of Medicine, a Matter facet value of Lungs with tuberculosis, an Energy facet value of Treatment using X-rays, a Space facet value of India, and a Time facet value of 1950. When the alphanumeric codes for these values are looked up in the classification tables, the composed call number is L,45;421:6;253:f.44'N5.[2]

Ranganathan deserves credit for implementing the first faceted classification system, but people other than librarians generally credit the idea to Nicolas de Condorcet, a French mathematician and philosopher. About 140 years before Ranganathan, Condorcet was concerned that "systems of classification that imposed a given interpretation upon Nature… represented an insufferable obstacle to… scientific advance." Condorcet thus proposed a flexible classification scheme for "arranging a large number of subjects in a system so that we may straightway grasp their relations, quickly perceive their combinations, and readily form new combinations."[3]

1. (Ranganathan 1967). (Satija 2001). See (Svenonius 2000, p. 174–176) for a quick introduction.

2. Wikipedia article at http://en.wikipedia.org/wiki/Colon_classification.

3. (Baker 1962). The first quote is on page 104; the second one is on page 100. This article contains Condorcet's 1805

Condorcet's system was based on five major facet categories, divided into 10 terms each, yielding 10^5 or 100,000 combinations:

Objects

domains of study.

Methods

for studying objects and describing the knowledge gained.

Points of view

for studying objects.

Uses and utility

of knowledge.

Ways

in which knowledge can be acquired.[4]

essay in French, but fortunately for us Baker's analysis is in English, This motivation of Condorcet's classification scheme sounds like the description of a data warehouse or business intelligence system in which transactional data can be "sliced and diced" into new combinations to answer questions in support of strategic decision-making. See (Watson and Wixon 2007).

4. See Joacim Hansson, Condorcet and the Origins of Faceted Classification, http://documentationandlibrarianship.blogspot.com/2011/02/condorcet-and-origins-of-faceted.html.

Condorcet and Ranganathan proposed different facets, but both hoped that their five top-level facets would be sufficient for a universal classification system. People have generally rejected the idea of universal facets, but Ranganathan's proposals continue to influence the development of the Library of Congress Subject Headings (LCSH).[5]

Faceted classification is most commonly used in narrow domains, each with its own specific facets. This makes intuitive sense because even if resources can be distinguished with a general classification, doing so requires lengthy notations, and it is much harder to add to a general classification than to a classification created specifically for a single subject area. We could probably describe shirts using the PMEST facets, but style, color, and size seem more natural.

Faceted Classification in Description

Elaine Svenonius defines facets as "groupings of terms obtained by the first division of a subject discipline into homogeneous or semantically cohesive categories."[6] The relationships between these facets results in a controlled vocabulary ("Identity, Identifiers, and Names") governing the resources we are organizing. From this controlled vocabulary we can generate many descriptions that are

5. LCSH uses facets for Topic, Place, Time, and Form (but they can be ordered in a variety of ways, not as rigidly as PMEST. (Anderson and Hoffman 2006) argue for a fully faceted syntax in LCSH.

6. (Svenonius 2000, p. 140).

complex but formally structured, enabling us to describe things for which terms do not yet exist.

Getty's Art & Architecture Thesaurus(AAT) is a robust and widely used *controlled vocabulary* consisting of generic terms to describe artifacts, objects, places and concepts in the domains of "art, architecture, and material culture."[7]

AAT is a thesaurus with a faceted hierarchical structure. The AAT's facets are "conceptually organized in a scheme that proceeds from abstract concepts to concrete, physical artifacts:"

Associated Concepts

> Concepts, philosophical and critical theory, and phenomena, such as "love" and "nihilism."

Physical Attributes

> Material characteristics that can be measured and perceived, like "height" and "flexibility."

Styles and Periods

> Artistic and architectural eras and stylistic groupings, such as "Renaissance" and "Dada."

Agents

> Basically, people and the various groups and organizations with which they identify, whether based on physical, mental, socio-economic, or political characteristics—e.g., "stonemasons" or "socialists."

7. The Getty AAT is online at http://www.getty.edu/research/tools/vocabularies/aat/index.html.

Activities

> Actions, processes, and occurrences, such as "body painting" and "drawing." These are different from the "Objects" facet, which may also contain "body painting," in terms of the actual work itself, not the creation process.

Materials

> Concerned with the actual substance of which a work is made, like "metal" or "bleach." "Materials" differ from "Physical Attributes" in that the latter is more abstract than the former.

Objects

> The largest facet, *objects* contains the actual works, like "sandcastles" and "screen prints."

Within each facet is a strict hierarchical structure drilling down from broad term to very specific instance.

"Patent Leather" in the
Art & Architecture Thesaurus

Hierarchical Position


Materials Facet


...Materials (Hierarchy Name) (G)


.....materials (matter) (G)


.....<materials by origin> (G)


.....<biological material> (G)


.....animal material (G)


.....<processed animal material> (G)


.....leather (G)


.....<leather by process> (G)


.....patent leather (G)

The Art and Architecture Thesaurus has a faceted hierarchical structure. For example, the materials facet distinguishes the material of "patent leather" according to the process applied to processed animal material, which is a type of biological material, and so on.

Figure: "Patent Leather" in the Art & Architecture Thesaurus. shows how a particular instance may be described on a number of dimensions for the purpose of organizing the item and retrieving information about it. And by using a standard *controlled vocabulary*, catalogers and indexers make it easier for users to understand and

adapt to the way things are organized for the purpose of finding them.[8]

A Classification for Facets

There are four major types of facets.

Enumerative facets

Have mutually exclusive possible values. In our online shirt store, "Style" is an enumerative facet whose values are "dress," "work," "party," and "athletic."

Boolean facets

Take on one of two values, yes (true) or no (false) along some dimension or property. On a sportswear website, "Waterproof" would be a Boolean facet because an item of clothing is either waterproof or it is not.

Hierarchical facets

Organize resources by logical inclusion ("Inclusion"). At Williams-Sonoma's website, the top-level facet includes "Cookware," "Cooks' Tools," and "Cutlery." At wine.com the "Region" facet has values for "US," "Old World," and "New

8. This section of the thesaurus comes from http://www.getty.edu/vow/AATFullDisplay?find=leather&logic=AND&note=&english=N&prev_page=1&subjectid=300193362.

World," each of which is further divided geographically. [9] Also see *taxonomic facets*.

Spectrum facets

Assume a range of numerical values with a defined minimum and maximum. Price and date are common spectrum facets. The ranges are often modeled as mutually exclusive regions (potential price facet values might include "$0—$49," "$50—$99," and "$100—$149").

Designing a Faceted Classification System

It is important to be systematic and principled when designing a faceted classification. In some respects the process and design concerns overlap with those for describing resources, and much of the advice in "The Process of Describing Resources" is relevant here.[10]

9. You might have thought that the US was in the new world, but according to wine.com, the new world of wine includes Australia, New Zealand, Argentina, Chile, and South Africa. The geography under the US facet is equally distorted by the uneven distribution of quality wine making regions, so the values of that facet are California. Oregon, Washington, and Other US.

10. Denton, William. How to Make a Faceted Classification and Put It On the Web Nov. 2003.

Design Process for Faceted Classification

We advocate a five step process for designing a faceted classification system.

1. Define the purposes of the classification (“Determining the Purposes”, “Classification Is Purposeful”) and specify the collection of concepts or resources to be classified.

2. For each facet, determine its logical type (“A Classification for Facets”) and possible values. Specify the order of the values for each facet so that they make sense to users; useful orderings are alphabetical, chronological, procedural, size, most popular to least popular, simple to complex, and geographical or topological.

3. Analyze and describe a representative sample of resource instances to identify properties or dimensions as candidate facets (See “Identifying Properties”).

4. Examine the relationships between the facets to create sub-facets if necessary. Determine how the facets will be combined to generate the classifications.

5. Test the classification on new instances, and revise the facets, facet values, and facet grammar as needed.

`http://www.miskatonic.org/library/facet-web-howto.html`. See also (Spiteri 1998).

Here is some more specific advice about selecting and designing facets and facet values:

Orthogonality

> Facets should be independent dimensions, so a resource can have values of all of them while only having one value on each of them. In an online kitchen store, one facet might be "Product" and another might be "Brand." A particular item might be classified as a "Saucepan" in the "Product" facet and as "Calphalon" in the "Brand" one. Other saucepans might have other brands, and other Calphalon products might not be saucepans, because Product and Brand are orthogonal.

Semantic Balance

> Top-level facets should be the properties that best differentiate the resources in the classification domain. The values should be of equal semantic scope so that resources are distributed among the subcategories. Subfacets of "Cookware" like "Sauciers and Saucepans" and "Roasters and Brasiers" are semantically balanced as they are both named and grouped by cooking activity.[11]

Coverage

> The values of a facet should be able of classifying all instances within the intended scope.

11. Should remind you of issues of lexical gap in "The Lexical Perspective".

Scalability

Facet values must accommodate potential additions to the set of instances. Including an "Other" value is an easy way to ensure that a facet is flexible and hospitable to new instances, but it not desirable if all new instances will be assigned that value.

Objectivity

Although every classification has an explicit or implicit bias ("Classification Is Biased"), facets and facet values should be as unambiguous and concrete as possible to enable reliable classification of instances.

Normativity

To make a faceted classification as useful by as many people as possible, the terms used for facets and facet values should not be idiosyncratic, metaphorical, or require special knowledge to interpret.[12]

As we will see in "Computational Classification", classification can sometimes be done by computers rather than by people. Computer algorithms can analyze resource properties and descriptions to identify dimensions on which resources differ and the most frequent descriptive terms, which can then be used to design a faceted classification scheme. Resources can then be assigned to the appropriate categories, either without human intervention or in

12. Semantic balance is a bit hard to define, but you can often tell when facet values are not balanced. A cookware facet whose values include saucepans, frying pans, stock pots, and pizza pans will not evenly distribute resources across the facets.

collaboration with a human who trains the algorithm with classified instances.

55. Classification by Activity Structure

Institutional classification systems are often strongly hierarchical and taxonomic because their many users come to them for diverse purposes, making a context-free or semantic organization the most appropriate. However, in narrow domains that offer a more limited variety of uses it can be much more effective to classify resources according to the tasks or activities they support. A task or activity-based classification system is called a *taskonomy*, a term invented by anthropologists Janet Dougherty and Charles Keller after their ethnographic study of how blacksmiths organized their tools. Instead of keeping things together according to their semantic relationships in what Donald Norman called "hardware store organization," the blacksmiths arranged tools in locations where they were used— "fire tools," "stump tools," "drill press rack tools," and so on.[1]

1. See (Dougherty and Keller 1985) for the ethnography of blacksmithing, and also (Norman 2006), who extends the taskonomy idea to the design of user interfaces for cell phones and other computing devices. You probably have not worked as blacksmith, but you have certainly used taskonomic classification. For example, a student writing a term paper or doing a course project checks out books from the library's taxonomic classification system (or prints them out from the web) and then organizes them in piles on a desk or on the floor according to the plan for

Personal organizing systems are often taskonomic. Think about the way you cook when you are following a recipe. Do you first retrieve all the ingredients from their storage places, and arrange them in activity-based groups in the preparation area?[2]

Looking at the relationship between tasks and tools in this way can help a cook determine the best way to organize tools in a kitchen. Cutting items would necessarily be kept together near a prep area; having to run across the kitchen to another area where a poultry knife is kept with, say, chicken broth would be detrimental to the cook's workflow. It would make far more sense to have all of the items for the task of cutting in a single area.

The intentional arrangement of tools in a working kitchen might look something like Table: A cook's taskonomy:

A cook's taskonomy

the paper or project. Some of the original classification might persist, but the emphasis clearly shifts toward getting work done. When the task is completed the books go back to the library and are put back into the context-free taxonomy.

2. See (Kirsh 1995) for theoretical motivation and a classification scheme for the "intelligent use of space," and (de Leon 2003) for an example of cooking ethnography.

Prep	Oven	Stove
Poultry knife	Oven mitts	Pots and pans
Paring knife	Baking sheets	Wooden spoons
Vegetable knife	Aluminum foil	Wok
Cutting board	Parchment paper	
	Roasting pan	

Stop and Think: Office Taskonomy

Think about your personal office space. It may be an interesting hybrid space—it probably contains documents that could be classified in a hierarchical system, but it is also a work space that could lend itself to "taskonomy" organization. Which does it more closely resemble? How have any conflicts between hierarchy and "taskonomy" been resolved?

56. Computational Classification

Because of its importance, ubiquity, and ease of processing by computers, it should not be surprising that a great many computational classification problems involve text. Some of these problems are relatively simple, like identifying the language in which a text is written, which is solved by comparing the probability of one, two, and three character-long contiguous strings in the text against their probabilities in different languages. For example, in English the most likely strings are “the”, “and”, “to”, “of”, “a”, “in”, and so on. But if the most likely strings are “der”, “die”, “und”, and “den” the text is German and if they are “de”, “la”, “que”, “el”, and “en” the text is Spanish.

More challenging text classification problems arise when more features are required to describe each instance being classified and where the features are less predictable. The unknown author of a document can sometimes be identified by analyzing other documents known to be written by him to identify a set of features like word frequency, phrase structure, and sentence length that create a “writeprint” analogous to a fingerprint that uniquely identifies him. This kind of analysis was used in 2013 to determine that *Harry Potter* author J. K. Rowling had written a crime fiction novel entitled *The Cuckoo’s Calling* under the pseudonym Robert Galbraith.[1]

Another challenging text classification problem is sentiment

1. (SeeLi, Zheng, and Chen 2006), (Juola 2014), (Rowling 1997-2007), (Rowling as “Galbraith” 2013),

analysis, determining whether a text has a positive or negative opinion about some topic. Much academic and commercial research has been conducted to understand the sentiment of Twitter tweets, Facebook posts, email sent to customer support applications, and other similar contexts. Sentiment analysis is hard because messages are often short so there is not much to analyze, and because and because sarcasm, slang, clichés, and cultural norms obscure the content needed to make the classification.

A crucial consideration whenever supervised learning is used to train a classifier is ensuring that the training set is appropriate. If we were training a classifier to detect spam messages using email from the year 2000, the topics of the emails, the words they contain, and perhaps even the language they are written in would be substantially different than messages from this year. Up to date training data is especially important for the classification algorithms used by Twitter, Facebook, YouTube, and similar social sites that classify and recommend content based on popularity trends.

When the relevant training data is constantly changing and there is a great deal of it, there is a risk that by the time a model can learn to classify correctly it is already out of date. This challenge has led to the development of streaming algorithms that operate on data as it comes in, using it as a live data source rather than as a static training set. Streaming algorithms are essential for tackling datasets that are too large to store or for models that must operate under intense time pressure. Streaming approaches complement rather than replace those that work with historical datasets because they make different tradeoffs between accuracy and speed. The streaming system might provide real-time alerting and recommendations, while historical analyses are made on the batch-oriented system that works with the entire data collection.[2]

2. (Ellis 2014). A compelling demonstration of the need to

Stop and Think: Sentiment Analysis

Sometimes, a text message might seem complimentary, but really is not. Is the customer happy if he tweets "Nice job, United. You only lost one of my bags this time." Think of some other short messages where sarcasm or slang makes sentiment analysis difficult. How would you write a product or service review that is unambiguously positive, negative, or neutral? How would you write a review whose sentiment is difficult to determine?

How a computational classifier "learns" depends on the specific machine learning algorithm. Decision trees, Naive Bayes, support vector machines, and neural net approaches were briefly described in "Implementing Categories".

sample big data streams to ensure against bias is (Morstatter et al 2013).

57. Key Points in Chapter Eight

- What is classification?

  Classification is the systematic assignment of resources to a system of intentional categories, often institutional ones.

  (See "Introduction")

- What is a classification system?

  A classification system is foremost a specification for the logical arrangement of resources because there are usually many possible and often arbitrary mappings of logical locations to physical ones.

  (See "Classification vs. Physical Arrangement")

- How does classification affect the potential interactions in an organizing system?

  A classification creates structure in the organizing system that increases the variety and capability of the interactions it can support.

  (See "Classifications Support Interactions")

- Why are classifications always biased in some way?

  Classifications are always biased by the purposes, experiences, professions, politics, values, and other characteristics and preferences of the people making them.

  (See "Classification Is Biased")

- According to Friedman and Nissenbaum, what are the three types of bias in technical systems?

  Three types of bias in technical systems are pre-existing, technical, and emergent bias.

  (See "Classification Is Biased")

- What are enumerative classification schemes?

  Classification schemes in which all possible categories to which resources can be assigned are defined explicitly are called *enumerative*.

  (See "Classification Schemes")

- What is a taxonomic classification scheme?

  When multiple resource properties are considered in a fixed sequence, each property creates another level in the system of categories and the classification scheme is *hierarchical* or *taxonomic*.

  (See "Classification Schemes")

- What is the relationship between classification and standardization?

  Classification and standardization are not identical, but they are closely related. Some classifications become standards, and some standards define new classifications.

  (See "Classification and Standardization")

- What is a standard?

  A standard is a published specification that is developed and maintained by consensus of all the relevant stakeholders in some domain by following a defined and transparent process.

(See "Specifications vs. Standards")

- Why are standard semantics important?

  Standard semantics are especially important in industries or markets that have significant network effects where the value of a product depends on the number of interoperable or compatible products.

  (See "Institutional Semantics")

- What is literary warrant?

  The principle of *literary warrant* holds that a classification must be based only on the specific resources that are being classified.

  (See "Principles Embodied in the Classification Scheme")

- What is the uniqueness principle?

  The *uniqueness principle* means the categories in a classification scheme are mutually exclusive. Thus, when a logical concept is assigned to a particular category, it cannot simultaneously be assigned to another category.

  (See "Principles for Assigning Resources to Categories")

- How is the uniqueness principle followed when resources do not clearly fit in a single category?

  The general solution to satisfying the uniqueness principle in library classifications when resources do not clearly fit in a single category is to invent and follow a detailed set of often-arbitrary rules.

  (See "Principles for Assigning Resources to Categories")

- What motivates category change?

  Categories sometimes change slowly, but they can also change

quickly and radically as a result of technological, process, or geopolitical innovation or events.

(See "Principles for Maintaining the Classification over Time")

- What is the relationship among flexibility, extensibility and hospitality in a classification system?

  Flexibility, *extensibility*, and *hospitality* are synonyms for the degree to which the classification can accommodate new resources.

  (See "Principles for Maintaining the Classification over Time")

- What distinguishes bibliographic classification?

  Bibliographic classification is distinctive because of a legacy of physical arrangement and its scale and complexity.

  (See "Bibliographic Classification")

- What distinguishes faceted classification systems?

  Faceted classification systems enumerate all the categories needed to assign resources appropriately, but instead of combining them in advance in a fixed hierarchy, they are applied only if they are needed to sort resources with a particular combination of properties.

  (See "Faceted Classification")

- What is orthogonality in a faceted classification system?

  Facets should be independent dimensions, so a resource can have values of all of them while only having one value on each of them.

  (See "Design Principles and Pragmatics")

- What is semantic balance in a faceted classification system?

Top-level facets should be the properties that best differentiate the resources in the classification domain. The values should be of equal semantic scope so that resources are distributed among the subcategories. Subfacets of "Cookware" like "Sauciers and Saucepans" and "Roasters and Brasiers" are semantically balanced as they are both named and grouped by cooking activity.

(See "Design Principles and Pragmatics")

- What is scalability in a faceted classification system?

  Facet values must accommodate potential additions to the set of instances. Including an "Other" value is an easy way to ensure that a facet is flexible and hospitable to new instances, but it not desirable if all new instances will be assigned that value.

  (See "Design Principles and Pragmatics")

- What is the relationship between classification and tagging?

  Most tagging seems insufficiently principled to be considered classification, except when tags are treated as category labels or when decisions that make tagging more systematic turn a set of tags into a *tagsonomy*.

  (See "Classification vs. Tagging")

- What is a taskonomy?

  A task or activity-based classification system is called a *taskonomy*.

  (See "Classification by Activity Structure")

- What is the relationship between supervised learning and classification?

  Supervised learning techniques start with a designed

classification scheme and then train computers to assign new resources to the categories.

(See "Computational Classification")

PART IX

THE FORMS OF RESOURCE DESCRIPTIONS

Ryan Shaw

Murray Maloney

58. Introduction (IX)

Throughout this book, we have emphasized the importance of separately considering fundamental organizing principles, application-specific concepts, and details of implementation. The three-tier architecture we introduced in "The Concept of "Organizing Principle"" is one way to conceptualize this separation. In "The Implementation Perspective ", we contrasted the implementation-focused perspective for analyzing relationships with other perspectives that focus on the meaning and abstract structure of relationships. In this chapter, we present this contrast between conceptualization and implementation in terms of separating the *content* and *form* of resource descriptions.

In the previous chapters, we have considered principles and concepts of organizing in many different contexts, ranging from personal organizing systems to cultural and institutional ones. We have noted that some organizing systems have limited scope and expected lifetime, such as a task-oriented personal organizing system like a shopping list. Other organizing systems support broad uses that rely on standard categories developed through rigorous processes, like a product catalog.

By this point you should have a good sense of the various conceptual issues you need to consider when deciding how to describe a resource in order to meet the goals of your organizing system. Considering those issues will give you some sense of what the content of your descriptions should be. In order to focus on the conceptual issues, we have deferred discussion of specific implementation issues. Implementation involves choosing the specific form of your descriptions, and that is the topic of this chapter.

We can approach the problem of how to form resource descriptions from two perspectives: structuring and writing. From one

perspective, resource descriptions are things that are *used* by both people and computational agents. From this perspective, choosing the form of resource descriptions is a kind of design. This is easy to see for certain kinds of resource descriptions, notably signs and maps found in physical environments like airport terminals, public libraries, and malls. In these spaces, resource descriptions are quite literally designed to help people orient themselves and find their way. But any kind of resource description, not just those embedded in the built environment, can be viewed as a designed object. Designing an object involves making decisions about how it should be structured so that it can best be used for its intended purpose. From a design perspective, choosing the form of a resource description means making decisions about its *structure*.

In "The Structural Perspective", we took a structural perspective on resources and the relationships among them. In this chapter, we will take a structural perspective on resource *descriptions*. The difference is subtle but important. A structural perspective on resource *relationships* focuses on how people or computational processes associate, arrange, and connect those resources. A structural perspective on resource *descriptions* focuses on how those associations, arrangements, and connections are explicitly represented or implemented in the descriptions we create. Mismatches between the structure imposed on the resources being organized and the structure of the descriptions used to implement that organization could result in an organizing system that is complex, inefficient, and difficult to maintain, as you will see in our first example (Example: Description structured as a dictionary).

The structures of resource descriptions enable or inhibit particular ways of interacting with those descriptions, just as the descriptions themselves enable or inhibit particular ways of interacting with the described resources. (See "Designing Resource-based Interactions", and *Interactions with Resources*) Keep in mind that resource descriptions are themselves information resources, so much of what we will say in this chapter is applicable to the structures and

forms of information resources in general. Put another way, the structure and form of information resources informs the design of resource descriptions.

From another perspective, creating resource descriptions is a kind of writing. I may describe something to you orally, but such a description might not be very useful to an organizing system unless it were transcribed. Organizing systems need persistent descriptions, and that means they need to be written. In that sense, choosing the form of a resource description means making decisions about *notation* and *syntax*.

Modern Western culture tends to make a sharp distinction between designing and writing, but there are areas where this distinction breaks down, and the creation of resource descriptions in organizing systems is one of them. In the following sections, we will use designing and writing as two lenses for looking at the problem of how to choose the form of resource descriptions. Specifically, we will examine the spectrum of options we have for structuring descriptions, and the kinds of syntaxes we have for writing those descriptions.

59. Structuring Descriptions

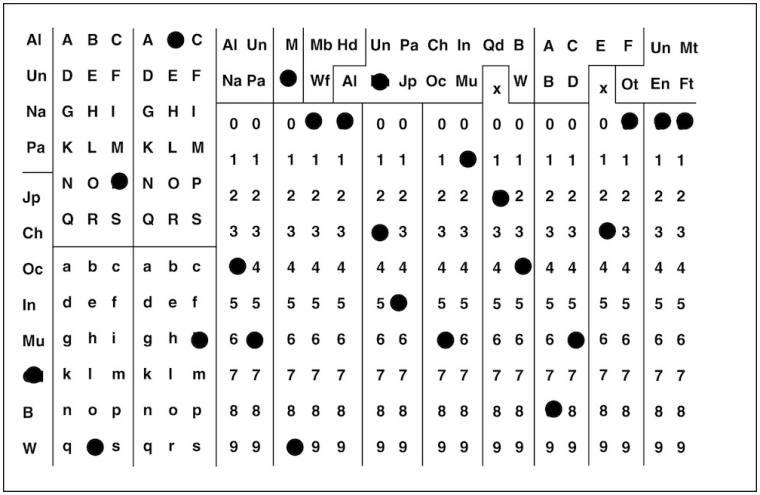
Choosing how to structure resource descriptions is a matter of making principled and purposeful design decisions in order to solve specific problems, serve specific purposes, or bring about some desirable property in the descriptions. Most of these decisions are specific to a *domain*: the particular context of application for the organizing system being designed and the kinds of interactions with resources it will enable. Making these kinds of context-specific decisions results in a model of that domain. (See "Abstraction in Resource Description".)

Over time, many people have built similar kinds of descriptions. They have had similar purposes, desired similar properties, and faced similar problems. Unsurprisingly, they have converged on some of the same decisions. When common sets of design decisions can be identified that are not specific to any one domain, they often become systematized in textbooks and in design practices, and may eventually be designed into standard formats and architectures for creating organizing systems. These formally recognized sets of design decisions are known as *abstract models* or *metamodels*. *Metamodels* describe structures commonly found in resource descriptions and other information resources, regardless of the specific domain. While any designer of an organizing system will usually create a model of her specific domain, she usually will not create an entirely new metamodel but will instead make choices from among the metamodels that have been formally recognized and incorporated into existing standards. The resulting model is sometimes called a "domain-specific language." Reusing standard metamodels can bring great economical advantages, as developers can reuse tools designed for and knowledge about these metamodels, rather than having to start from scratch.

In the following sections, we examine some common kinds of

structures used as the basis for metamodels. But first, we consider a concrete example of how the structure of resource descriptions supports or inhibits particular uses. As we explained in *Foundations for Organizing Systems*, the concept of a resource de-emphasizes the differences between physical and digital things in favor of focusing on how things, in general, are used to support goal-oriented activity. Different kinds of books can be treated as information resources regardless of the particular mix of tangible and intangible properties they may have. Since resource descriptions are also information resources, we can similarly consider how their structures support particular uses, independent of whether they are physical, digital, or a mix of both.

A Batten Card



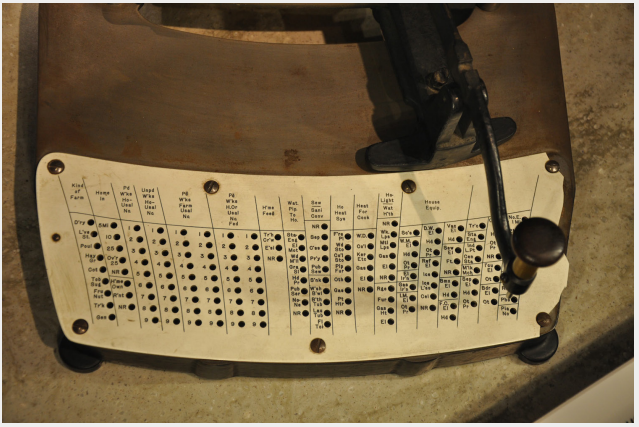
An example of a punch card used by Batten to describe a particular patent in a patent collection. Each card represented an individual description term, and each punch position on a card represented a particular patent.

During World War II, a British chemist named W. E. Batten developed a system for organizing patents.[1] The system consisted of a language for describing the product, process, use, and apparatus of a patent, and a way of using punched cards to record these descriptions. Batten used cards printed with matrices of 800 positions (see Figure: A Batten Card.). Each card represented a specific value from the vocabulary of the description language, and each position corresponded to a particular patent. To describe patent #256 as covering *extrusion of polythene to produce cable coverings*, one would first select the cards for the values *polythene*, *extrusion*, and *cable coverings*, and then punch each card at the 256th position. The description of patent #256 would thus extend over these three cards.

The advantage of this structure is that to find patents covering *extrusion of polythene* (for any purpose), one needs only to select the two cards corresponding to those values, lay one on top of the other, and hold them up to a light. Light will shine through wherever there is a position corresponding to a patent described using those values. Patents meeting a certain description are easily found due to the structure of the cards designed to describe the patents.

Punchcard Machine

1. This discussion of Batten's cards is based on (Lancaster 1968, pages 28–32). Batten's own explanation is in (Batten 1951).



Punchcards were an important information input and storage medium for decades, even before the invention of computers. The Hollerith keyboard punch was used to transcribe the information collected in the 1890 US census. The template being used in this photo is for recording information about a farm. The punch cards were tabulated by electromechanical machines. A merger of four tabulating machine companies in 1911 created a company whose current name is IBM.

This keyboard punch machine is in the collection of the Computer History Museum in Mountain View, California.

(Photo by R. Glushko.)

Of course, this system has clear disadvantages as well. Finding the concepts associated with a particular patent is tedious, because every card must be inspected. Adding a new patent is relatively easy as long as there is an index that allows the cards for specific

concepts to be located quickly. However, once the cards run out of space for punching holes, the whole set of cards must be duplicated to accommodate more patents: a very expensive operation. Adding new concepts is potentially easy: simply add a new card. But if we want to be able to find existing patents using the new concept, all the existing patents would have to be re-examined to determine whether their positions on the new card should be punched: also an expensive operation.

The structure of Batten's cards supported rapid selection of resources given a partial description. The kinds of structures we will examine in the following sections are not quite so elaborate as Batten's cards. But like the cards, each kind of structure supports more efficient mechanical execution of certain operations, at the cost of less efficient execution of others.

Kinds of Structures

Sets, lists, dictionaries, trees, and graphs are kinds of structures that can be used to form resource descriptions. As we shall see, each of these kinds is actually a family of related structures. These structures are *abstractions*: they describe formal structural properties in a general way, rather than specifying an exact physical or textual form. Abstractions are useful because they help us to see common properties shared by different specific ways of organizing information. By focusing on these common properties, we can more easily reason about the operations that different forms support and the affordances that they provide, without being distracted by less relevant details.

Blobs

The simplest kind of structure is no structure at all. Consider the following description of a book: *Sebald's novel uses a walking tour in East Anglia to meditate on links between past and present, East and West.*[2] This description is an unstructured text expression with no clearly defined internal parts, and we can consider it to be a *blob*. Or, more precisely, it has structure, but that structure is the underlying grammatical structure of the English language, and none of that grammatical structure is explicitly represented in a surface structure when the sentence is expressed. As readers of English we can interpret the sentence as a description of the subject of the book, but to do this mechanically is difficult.[3] On the other hand, such a written description is relatively easy to create, as the describer can simply use natural language.

A blob need not be a blob of text. It could be a photograph of a resource, or a recording of a spoken description of a resource. Like blobs of text, blobs of pixels or sound have underlying structure that any person with normal vision or hearing can understand easily.[4]

2. (Silman 1998). (Sebald 1995).

3. The technique of diagramming sentences was invented in the mid-19th century by Stephen W. Clark, a New York schoolmaster; (Clark2010) is an exact reprinting of a nearly 100 year old edition of his book *A Practical Grammar.* A recent tribute to Clark is (Florey 2012).

4. It is easy to underestimate the incredible power of the human perceptual and cognitive systems to apply neural computation and knowledge to enable vision and hearing

But we can treat these blobs as unstructured, because none of the underlying structure in the visual or auditory input is explicit, and we are concerned with the ways that the structures of resource descriptions support or inhibit mechanical or computational operations.[5]

Sets

The simplest way to structure a description is to give it parts and treat them as a *set*. For example, the description of Sebald's novel might be reformulated as a set of terms: *Sebald*, *novel*, *East Anglia*, *walking*, *history*. Doing this has lost much of the meaning, but something has been gained: we now can easily distinguish *Sebald* and *walking* as separate items in the description.[6] This makes it

to seem automatic. Computers are getting better at extracting features from visual and auditory signals to identify and classify inputs, but our point here is that none of these features are explicitly represented in the input "blob" or "stream."

5. As we commented earlier, an oral description of a resource may not be especially useful in an organizing system because computers cannot easily understand it. On the other hand, there are many contexts in which an oral description would be especially useful, such as in a guided tour of a museum where visitors can use audio headsets.

6. What was lost was the previously invisible structure

easier to find, for example, all the descriptions that include the term *walking*. (Note that this is different from simply searching through blob-of-text descriptions for the word *walking*. When treated as a set, the description *Fiji, fire walking, memoir* does not include the term *walking*, though it does include the term *fire walking*.)

Sets make it easy to find intersections among descriptions. Sets are also easy to create. In "Classification vs. Tagging" we looked at "folksonomies," organizing systems in which non-professional users create resource descriptions. In these systems, descriptions are structured as sets of "tags." To find resources, users can specify a set of tags to obtain resources having descriptions that intersect at those tags. This is more valuable if the tags come from a *controlled vocabulary*, making intersections more likely. But enforcing vocabulary control adds complexity to the description process, so a balance must be struck between maximizing potential intersections and making description as simple as practical.[7]

A set is a type or class of structure. We can refine the definition of different kinds of sets by introducing *constraints*. For example, we might introduce the constraint that a given set has a maximum number of items. Or we might constrain a set to always have the same number of items, giving us a fixed-size set. We can also remove constraints. Sets do not contain duplicate items (think of a tagging system in which it does not make sense to assign the same tag more than once to the same resource). If we remove this

provided by the grammar, which made us assign roles to each of these terms to create a semantic interpretation.

7. It is rarely practical to make things as simple as possible. According to Einstein, we should endeavor to "Make everything as simple as possible, but not simpler."

uniqueness constraint, we have a different structure known as a "bag" or "multiset."

Lists

Constraints are what distinguish lists from sets. A *list*, like a *set*, is a collection of items with an additional constraint: their items are ordered. If we were designing a tagging system in which it was important that the order of the tags be maintained, we would want to use lists, not sets. Unlike sets, *lists* may contain duplicate items. In a *list*, two items that are otherwise the same can be distinguished by their position in the ordering, but in a set this is not possible. For example, we might want to organize the tags assigned to a resource, listing the most used tag first, the least frequently used last, and the rest according to their frequency of use.

Again, we can introduce constraints to refine the definition of different kinds of *lists*, such as fixed-length lists. If we constrain a list to contain only items that are themselves lists, and further specify that these contained lists do not themselves contains lists, then we have a *table* (a list of lists of items). A spreadsheet is a list of lists.

Dictionaries

One major limitation of *lists* and *sets* is that, although items can be individually addressed, there is no way to distinguish the items except by comparing their values (or, in a list, their positions in the ordering). In a set of terms like *Sebald*, *novel*, *East Anglia*, *walking*, *history*, for example, one cannot easily tell that *Sebald* refers to the author of the book while *East Anglia* and *walking* refer to what it is about. One way of addressing this problem is to break each item in a set into two parts: a *property* and a *value*. So, for example, our

simple set of tags might become *author*: *Sebald*, *type*: *novel*, *subject*: *East Anglia*, *subject*: *walking*, *subject*: *history*. Now we can say that *author*, *type*, and *subject* are the properties, and the original items in the set are the values.

author

: Sebald

type

: novel

subject1

: East Anglia

subject2

: walking

subject3

: history

This kind of structure is called a *dictionary*, a *map* or an *associative array*. A *dictionary* is a set of property-value pairs or entries. It is a set of entries, not a list of entries, because the pairs are not ordered and because each entry must have a unique key.[8] Note that this specialized meaning of *dictionary* is different from the more common meaning of "dictionary" as an alphabetized list of terms accompanied by sentences that define them. The two meanings

8. This structural metamodel only allows one value for each property, which means it would not work for books with multiple authors or that discuss multiple subjects.

are related, however. Like a "real" dictionary, a *dictionary* structure allows us to easily find the value (such as a definition) associated with a particular property or *key* (such as a word). But unlike a real dictionary, which orders its keys alphabetically, a *dictionary* structure does not specify an order for its keys.[9]

Dictionaries are ubiquitous in resource descriptions. Structured descriptions entered using a form are easily represented as dictionaries, where the form items' labels are the properties and the data entered are the values. Tabular data with a "header row" can be thought of as a set of dictionaries, where column headers are the properties for each dictionary, and each row is a set of corresponding values. Dictionaries are also a basic type of data structure found in nearly all programming languages (referred to as associative arrays).

Again, we can introduce or remove constraints to define specialized types of dictionaries. A sorted dictionary adds an ordering over entries; in other words, it is a list of entries rather than a set. A *multimap* is a dictionary in which multiple entries may have the same key.

Trees

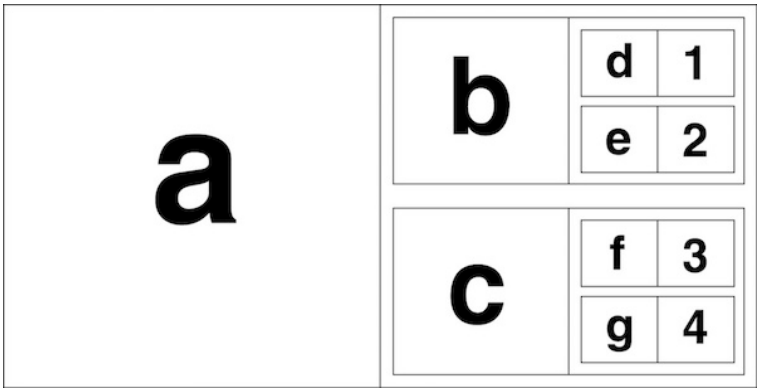
In dictionaries as they are commonly understood, properties are terms and values are their corresponding definitions. The terms and values are usually words, phrases, or other expressions that can be

9. Going the other direction is not so easy, however: just as real dictionaries do not support finding a word given a definition, neither do dictionary structures support finding a key given a value.

ordered alphabetically. But if generalize the notion of a dictionary as abstract sets of property-value pairs, the values can be anything at all. In particular, the values can themselves be dictionaries. When a dictionary structure has values that are themselves dictionaries, we say that the dictionaries are *nested*. Nesting is very useful for resource descriptions that need more structure than what a (non-nested) dictionary can provide.

Figure: Four Nested Dictionaries. presents an example of nested dictionaries. At the top level there is one dictionary with a single entry having the property *a*. The value associated with *a* is a dictionary consisting of two entries, the first having property *b* and the second having property *c*. The values associated with *b* and with *c* are also dictionaries.

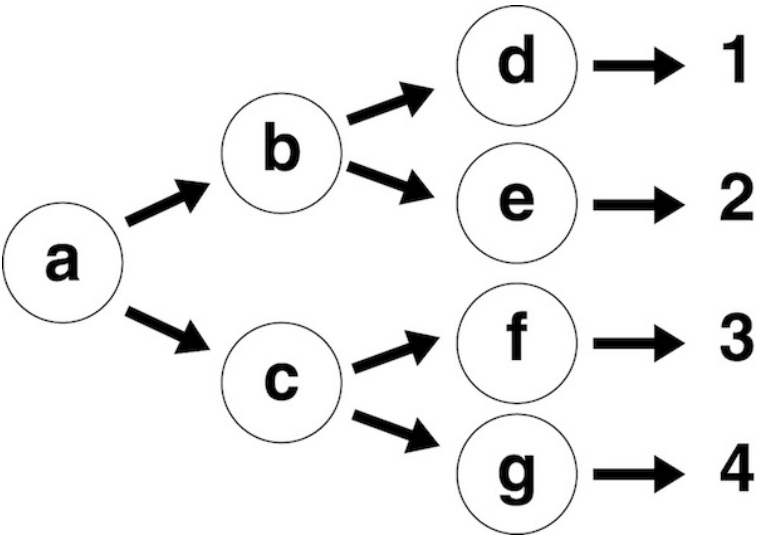
Four Nested Dictionaries



If we nest dictionaries like this, and our "top" dictionary (the one that contains all the others) has only one entry, then we have a kind of *tree* structure. Figure: A Tree of Properties and Values. shows the same properties and values as Figure: Four Nested Dictionaries., this

time arranged to make the tree structure more visible. *Trees* consist of *nodes* (the letters and numbers in Figure: A Tree of Properties and Values.) joined by *edges* (the arrows). Each node in the tree with a circle around it is a property, and the value of each property consists of the nodes below (to the right of) it in the tree. A node is referred to as the *parent* of the nodes below it, which in turn are referred to as the *children* of that node. The edges show these "parent of" relationships between the nodes. The node with no parent is called the *root* of the tree. Nodes with no children are called *leaf* nodes.

A Tree of Properties and Values



An alternative representation of nested dictionaries is as a tree. The lowest level or leaf nodes of the tree contain property values.

As with the other types of structures we have considered, we can

define different kinds of trees by introducing different types of constraints. For example, the predominant metamodel for XML is documents is a kind of tree called the *XML Information Set* or Infoset. [10]

The *XML Information Set* defines a specific kind of tree structure by adding very specific constraints, including ordering of child nodes, to the basic definition of a tree. The addition of an ordering constraint distinguishes XML trees from nested dictionaries, in which child nodes do not have any order (because dictionary entries do not have an ordering). Ordering is an important constraint for resource descriptions, since without ordering it is impossible to, for example, list multiple authors while guaranteeing that the order of

10. The XML Information Set (Cowan2004)

RDF/XML is one example where meta models meet. In *Document Design Matters*, (Wilde and Glushko 2008b) point out that "If the designer of an exchange format uses a non-XML conceptual metamodel because it seems to be a better fit for the data model, XML is only used as the physical layer for the exchange model. The logical layer in this case defines the mapping between the non-XML conceptual model, and any reconstruction of the exchange model data requires the consumer to be fully aware of this mapping. In such a case, it is good practice to make users of the API aware of the fact that it is using a non-XML metamodel. Otherwise they might be tempted to base their implementation on a too small set of examples, creating implementations which are brittle and will fail at some point in time."

authors will be maintained. Figure: A Tree of Properties and Values. depicts a kind of tree with a different set of constraints: all non-leaf nodes are properties, and all leafs are values. We could also define a tree in which every node has both a property and a value. Trees exist in a large variety of flavors, but they all share a common topology: the edges between nodes are directed (one node is the parent and the other is the child), and every node except the root has exactly one parent.

Trees provide a way to group statements describing different but related resources. For example, consider the description structured as a dictionary here:

Description Structured as a Dictionary

author given names → Winfried Georg
author surname → Sebald
title → Die Ringe des Saturn
pages → 371

The dictionary groups together four property-value pairs describing a particular book. (The arrows are simply a schematic way to indicate property-value relations. Later in the chapter we look at ways to "write" these relations using some specific syntax.)

But really the first two entries are not describing the book; they are describing the book's author. So, it would be better to group those two statements somehow. We can do this by nesting the entries describing the author within the book description, creating a tree structure:

Nesting an Author Description Within a Book Description

author →
given names → Winfried Georg
surname → Sebald
title → Die Ringe des Saturn
pages → 371

Using a tree works well in this case because we can treat the book as the primary resource being described, making it the root of our tree, and adding on the author description as a "branch."

We also could have chosen to make the author the primary resource, giving us a tree like the one in Example: Nesting book descriptions within an author description.

Nesting Book Descriptions Within an Author Description

given names → Winfried Georg
surname → Sebald
books authored →
1. title → Die Ringe des Saturn
pages → 371
2. title → Austerlitz
pages → 416

Note that in this dictionary, the value of the *books authored* property is a *list* of dictionaries. Making the author the primary or root resource allows us to include multiple book descriptions in the tree (but makes it more difficult to describe books having multiple authors). A tree is a good choice for structuring descriptions as long as we can clearly identify a primary resource. In some cases, however, we want to connect descriptions of related resources without having to designate one as primary. In these cases, we need a more flexible data structure.

Graphs

Suppose we were describing two books, where the author of one book is the subject of the other, as in Example: Two related descriptions:

Two Related Descriptions

1. author → Mark Richard McCulloch
title → Understanding W. G. Sebald
subject → Winfried Georg Sebald
2. author → Winfried Georg Sebald
title → Die Ringe des Saturn

By looking at these descriptions, we can guess the relationship between the two books, but that relationship is not explicitly represented in the structure: we just have two separate dictionaries and have inferred the relationship by matching property values. It is possible that this inference could be wrong: there might be two people named *Winfried Georg Sebald*. How can we structure these descriptions to explicitly represent the fact that the *Winfried Georg Sebald* that is the subject of the first book is the same *Winfried Georg Sebald* who authored the second?

One possibility would be to make *Winfried Georg Sebald* the root of a tree, similar to the approach taken in Example: Nesting book descriptions within an author description, adding a *book about* property alongside the *books authored* one. This solution would work fine if people were our primary resources, and it thus made sense to structure our descriptions around them. But suppose that we had decided that our descriptions should be structured around books, and that we were using a vocabulary that took this perspective (with properties such as *author* and *subject* rather than *books authored* and *books about*). We should not let a particular

structure limit the organizational perspective we can take, as Batten's cards did. Instead, we should consciously choose structures to suit our organizational perspective. How can we do this?

If we treat our two book descriptions as trees, we can join the two branches (subject and author) that share a value. When we do this, we no longer have a tree, because we now have a node with more than one parent (Figure: Descriptions Linked into a Graph.). The structure in Figure: Descriptions Linked into a Graph. is a *graph*. Like a *tree*, a *graph* consists of a set of nodes connected by edges. These edges may or may not have a direction ("Directionality"). If they do, the *graph* is referred to as a "directed graph." If a *graph* is directed, it may be possible to start at a node and follow edges in a path that leads back to the starting node. Such a path is called a "cycle." If a directed graph has no cycles, it is referred to as an "acyclic graph."

A tree is just a more constrained kind of *graph*. Trees are *directed* graphs because the "parent of" relationship between nodes is asymmetric: the edges are arrows that point in a certain direction. (See "Symmetry".) Furthermore, trees are *acyclic* graphs, because if you follow the directed edges from one node to another, you can never encounter the same node twice. Finally, trees have the constraint that every node (except the root) must have exactly one parent.[11]

In Figure: Descriptions Linked into a Graph. we have violated this constraint by joining our two book trees. The graph that results is

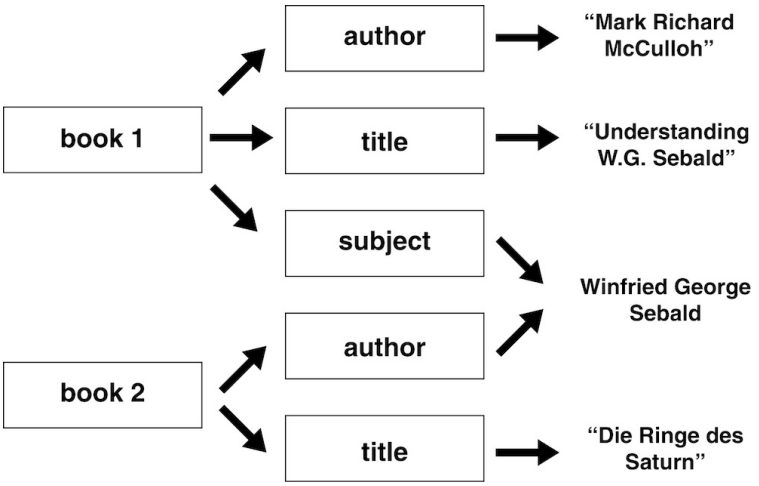
11. Technically, what is described here is referred to as "rooted tree" by mathematicians, who define trees more generally. Since trees used as data structures are always rooted trees, we do not make the distinction here.

still directed and acyclic, but because the *Winfried George Sebald* node now has two parents, it is no longer a tree.

> Stop and Think: Social Network Properties
>
> Compare the concept of "friend" in Facebook with that of "follower" in Twitter, in terms of the semantic properties discussed in "Properties of Semantic Relationships" and the graph properties discussed in this section.

Graphs are very general and flexible structures. Many kinds of systems can be conceived of as nodes connected by edges: stations connected by subway lines, people connected by friendships, decisions connected by dependencies, and so on. Relationships can be modeled in different ways using different kinds of *graph*s. For example, if we assume that friendship is symmetric (see "Symmetry"), we would use an undirected *graph* to model the relationship. However, in web-based social networks friendship is often asymmetric (you might "friend" someone who does not reciprocate), so a directed *graph* is more appropriate.

Descriptions Linked into a Graph



Descriptions can be linked to form a graph when the value assigned to two different properties is the same.

Often it is useful to treat a *graph* as a set of pairs of nodes, where each pair may or may not be directly connected by an edge. Many approaches to characterizing structural relationships among resources (see "Structural Relationships between Resources") are based on modeling the related resources as a set of pairs of nodes, and then analyzing patterns of connectedness among them. As we will see, being able to break down a *graph* into pairs is also useful when we structure resource descriptions as *graph*s.

In "The Document Processing World" we will use XML to model the graph shown in Figure: Descriptions Linked into a Graph. by using "references" to connect a book to its title, authors and subject. This will allow us to develop sophisticated graphs of knowledge within a

single XML document instance. (See also the sidebar, Inclusions and References)[12]

Comparing Metamodels: JSON, XML and RDF

Now that we are familiar with the various kinds of metamodels used to structure resource descriptions, we can take a closer look at some specific metamodels. A detailed comparison of the affordances of different metamodels is beyond the scope of this chapter. Here we will simply take a brief look at three popular metamodels—JSON, XML, and RDF—in order to see how they further specify and constrain the more general kinds of metamodels introduced above.

JSON

JavaScript Object Notation (JSON)

> *JavaScript Object Notation*(JSON) is a textual format for exchanging data that borrows its metamodel from the JavaScript programming language. Specifically, the JSON metamodel consists of two kinds of structures found in JavaScript: lists (called "arrays" in JavaScript) and dictionaries

12. This feature relies upon the existence of an XML schema. An XML schema can declare that certain attributes are of type ID, IDREF or IDREFS. Whether an XML DTD or one of the many schema languages that have been developed under the auspices of the W3C or ISO.

(called "objects" in JavaScript). Lists and dictionaries contain values, which may be strings of text, numbers, Booleans (true or false), or the null (empty) value. Again, these types of values are taken directly from JavaScript. Lists and dictionaries can be values too, meaning lists and dictionaries can be nested within one another to produce more complex structures such as tables and trees.

Lists, dictionaries, and a basic set of value types constitute the JSON metamodel. Because this metamodel is a subset of JavaScript, the JSON metamodel is very easy to work with in JavaScript. Since JavaScript is the only programming language that is available in all web browsers, JSON has become a popular choice for developers who need to work with data and resource descriptions on the web. (See "Writing Systems" later in this chapter.) Furthermore, many modern programming languages provide data structures and value types equivalent to those provided by JavaScript. So, data represented as JSON is easy to work with in many programming languages, not just JavaScript.

XML Information Set

The *XML Information Set* metamodel is derived from data structures used for document markup. (See "Metadata".) These markup structures—*elements* and *attributes*—are well suited for programmatically manipulating the structure of documents and data together.[13]

XML *Infoset*

The XML *Infoset* is a tree structure, where each node of the

13. http://www.w3.org/TR/xml-infoset/.

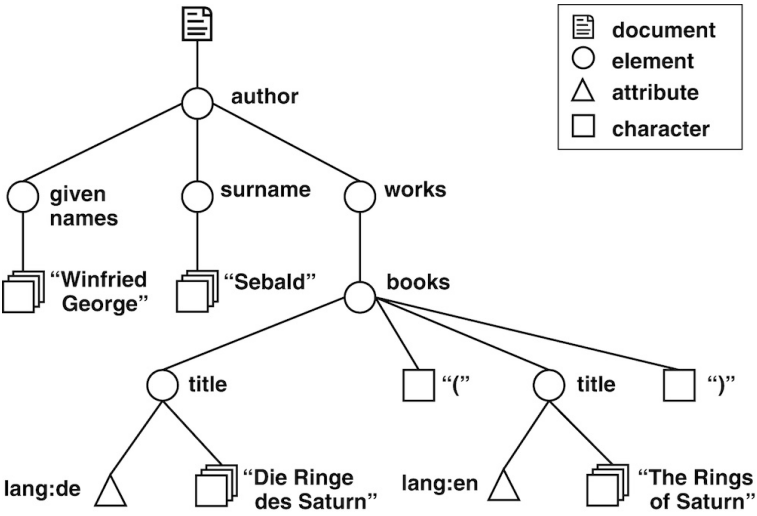
tree is defined to be an "information item" of a particular type. Each information item has a set of type-specific properties associated with it. At the root of the tree is a "document item," which has exactly one "element item" as its child. An *element* item has a set of *attribute* items, and a list of child nodes. These child nodes may include other element items, or they may be character items. (See "Kinds of Structures" below for more on characters.) *Attribute* items may contain character items, or they may contain typed data, such as name tokens, identifiers and references. Element identifiers and references (ID/IDREF) may be used to connect nodes, transforming a tree into a graph. (See the sidebar, Inclusions and References.)[14]

Figure: A Description Structure. is a graphical representation of how an XML document might be used to structure part of a description of an author and his works. This example demonstrates how we might use element items to model the domain of the description, by giving them names such as author and title. The character items

14. The *XML Infoset* is one of many metamodels for XML, including the DOM and XPath. Typically, an XML Infoset is created as a by-product of parsing a well-formed XML document instance. An XML document may also be informed by its DTD or schema with information about the types of attribute values, and their default values. Attributes of type ID, IDREF and IDREFs provide a mechanism for intra-document hypertext linking and transclusion. An XML document instance may contain entity definitions and references that get expanded when the document is parsed, thereby offering another form of transclusion.

that are the children of these elements hold the content of the description: author names, book titles, and so on. Attribute items are used to hold auxiliary information about this content, such as its language.

A Description Structure



An XML document can be described as a tree in which elements are nodes that can contain character content directly or attributes that contain character content.

This example also demonstrates how the XML Infoset supports mixed content by allowing element items and character items to be "siblings" of the same parent element. In this case, the Infoset structure allows us to specify that the book description can be displayed as a line of text consisting of the original title and the translated title in parentheses. The elements and attributes are used

to indicate that this line of text consists of two titles written in different languages, not a single title containing parentheses.

If not for mixed content, we could not write narrative text with *hypertext links* embedded in the middle of a sentence. It gives us the ability to identify the subcomponents of a sentence, so that we could distinguish the terms "Sebald," "walking" and "East Anglia" as an author and two subjects.

> Inclusions and References
>
> An XML Infoset is typically the result of processing a well-formed XML document instance.[15] Schemas associated with XML document instances "inform" the corresponding XML Infoset. Thus, the "truth value" of any XML Infoset is dependent upon its related schemas.[16] Traditionally, any documentation that is

15. A well-formed XML document instance, when processed, will yield an XML Information Set, as described here. Information sets may also be constructed by other means, such as transforming from another information set. See the section on Synthetic Infosets at http://www.w3.org/TR/xml-infoset/#intro.synthetic for details.

16. The Infoset contains knowledge of whether all related declarations have been read and processed, the base URI of the document instance, information about attribute

related to the schema is considered to be part of the schema definition and, at least notionally, informs human understanding and interpretation of corresponding documents.[17]

types, comments, processing instructions, unparsed entities and notations, and more.

A well-formed XML document instance for which there are associated schemas, such as a DTD, may contribute information to the Infoset. Notably, schemas may associate data types with element and attribute information items, and it may also specify default or fixed values for attributes. A DTD may define entities that are referenced in the document instance and are expanded in-place when processed. These contributions can affect the truth value of the document.

17. The SGML standard explicitly stated that documentation describing or explaining a DTD is part of the document type definition. The implication being that a schema is not just about defining syntax, but also semantics. Moreover, since DTDs do not make possible to describe all possible constraints, such as co-occurrence constraints, the documentation could serve as human-consumable guidance for implementers as well as content creators and consumers.

The XML family offers several mechanisms to create inclusion relationships: by employing element references; by way of entity definition and reference; by using XML *Inclusions(XInclude)* or XLink. These inclusions and references can also inform the XML Infoset, if they are processed.

Any XML node may refer to another node simply by referencing it by its assigned ID. Assuming attributes are declared, the Infoset exposes this information as a *references* property as an ordered list of element information items. That is to say that an element may contain other element nodes by subordination, or by reference.[18]

XInclude "specifies a processing model and syntax for general purpose inclusion. Inclusion is accomplished by

18. Attribute types may be declared in an XML DTD or schema. Attributes whose type is ID must have a valid XML name value that is unique within that XML document; an attribute of type IDREF whose value corresponds to a unique ID has a "references" property whose value is the element node that corresponds to the element with that ID. An attribute of type IDREFS whose value corresponds to a list of unique ID has a "references" property whose value is a list of element node(s) that corresponds to the element(s) with matching IDs.

merging a number of XML information sets into a single composite infoset." XInclude offers the most versatile mechanism for addressing whole documents, specific information items, ranges of information items, and even parts of information items, which has led to its widespread adoption in document processing.[19]

XLink "allows elements to be inserted into XML documents in order to create and describe links between resources. It uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of today's HTML, as well as more sophisticated links."[20]

Entities are similar to macros found in many programming languages; a value is assigned to a token, the token is referenced wherever the value is needed, and macro expansion happens when the XML document instance is read into the Infoset.[21] Entities are a handy

19. XML Inclusions (XInclude) is (Marsh, Orchard, and Veillard 2006).

20. XML Linking Language (XLink) is (DeRose, Maler, Orchard, and Walsh 2010).

21. Within the document's DTD, one simply declares the entity and its corresponding value, which could be anything from an entire document to a phrase and then it may be referenced in place within the XML document instance. The entity reference is replaced by the entity

feature, but since they are expanded on their way in, entities do not survive as information items in the XML Infoset. The ID/IDREF feature is more popular than the use of entities because it carries more information into the XML Infoset.

Using schemas to define data representation formats is a good practice that facilitates shared understanding and contributes to long-term maintainability in institutional or business contexts. An XML schema represents a contract among the parties subscribing to its definitions, whereas JSON depends on out-of-band communication among programmers. The notion that "the code is the documentation" may be fashionable among programmers, but modelers prefer to design at a higher level of abstraction and then implement.

The XML Infoset presents a strong contrast to JSON and does not always map in a straightforward way to the data structures used in popular web scripting languages. Whereas JSON's structures make it easier for object-oriented programmers to readily exchange data, they lack any formal schema language and cannot easily handle mixed content.

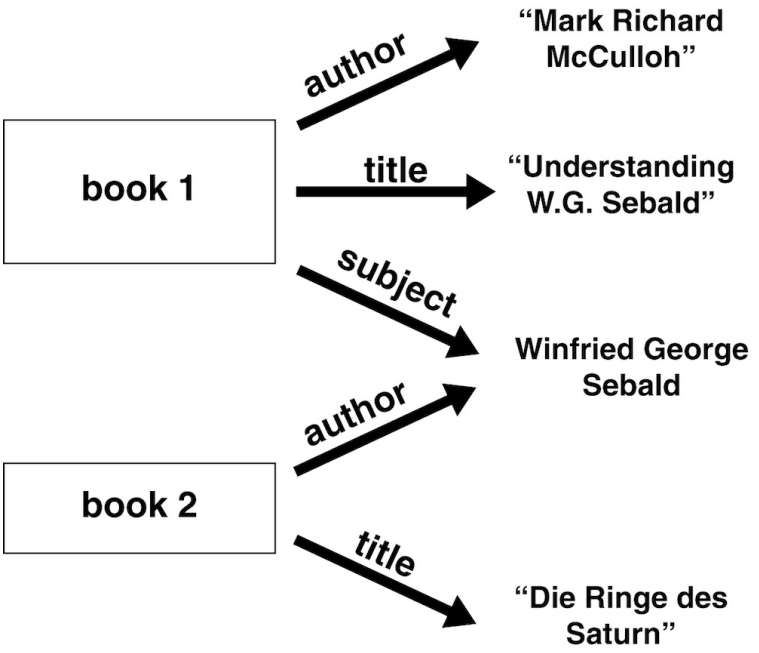
RDF

In Figure: Descriptions Linked into a Graph., we structured our

value in the XML Infoset. Entities, as nameable wrappers, effectively disappear on their way into the XML Infoset.

resource description as a graph by treating resources, properties, and values as nodes, with edges reflecting their combination into descriptive statements. However, a more common approach is to treat resources and values as nodes, and properties as the edges that connect them. Figure: Treating Properties as Edges Rather Than Nodes. shows the same description as Figure: Descriptions Linked into a Graph., this time with properties treated as edges. This roughly corresponds to the particular kind of graph metamodel defined by RDF. ("Resource Description Framework (RDF)")

Treating Properties as Edges Rather Than Nodes

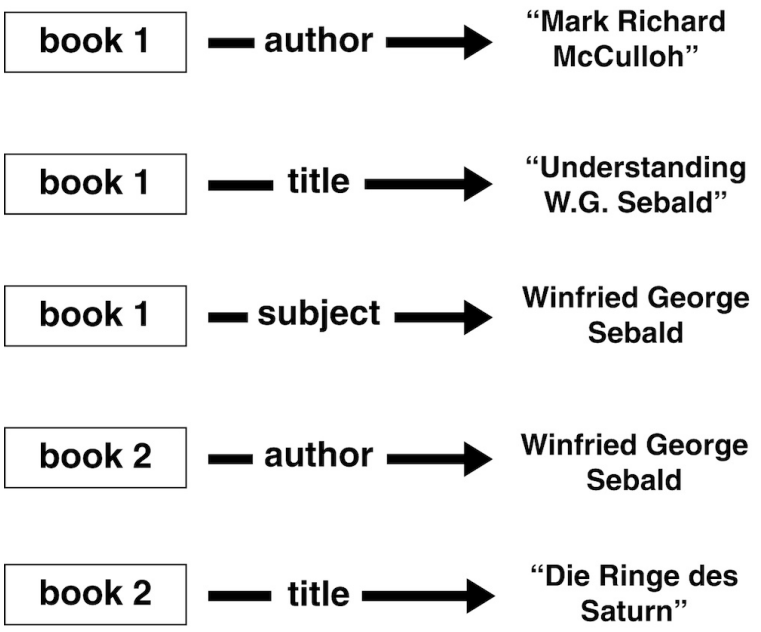


We can treat each component of a description as a pair of nodes (a resource and a value) with an edge (the property) linking them.

Here, we have two book resources that are related to four values through five properties. The single value node, “Winfried George Sebald” is the subject of one book while being the author of the second book. The books are depicted as boxes, the edges as labeled arrows and the values as text strings.

We have noted that we can treat a graph as a set of pairs of nodes, where each pair may be connected by an edge. Similarly, we can treat each component of the description in Figure: Treating Properties as Edges Rather Than Nodes. as a pair of nodes (a resource and a value) with an edge (the property) linking them. In the RDF metamodel, a pair of nodes and its edge is called a *triple*, because it consists of three parts (two nodes and one edge). The RDF metamodel is a directed graph, so it identifies one node (the one from which the edge is pointing) as the *subject* of the triple, and the other node (the one to which the edge is pointing) as its *object*. The edge is referred to as the *predicate* or (as we have been saying) *property* of the triple.

Listing Triples Individually



Lists each of the triples individually. Here, each statement relates one resource to one value through an edge. Thus, we have two distinct "Winfried George Sebald" value nodes. The books are depicted as boxes, the edges as labeled arrows and the values as text strings.

Figure: Listing Triples Individually. lists separately all the triples in Figure: Treating Properties as Edges Rather Than Nodes. However, there is something missing in Figure: Listing Triples Individually.. Figure: Treating Properties as Edges Rather Than Nodes. clearly indicates that the *Winfried George Sebald* who is the subject of book 1 is the same *Winfried George Sebald* who is the author of book 2. In Figure: Listing Triples Individually. this relationship is not clear. How can we tell if the *Winfried George Sebald* of the third triple is the same as the *Winfried George Sebald* of the triple statement? For that matter, how can we tell if the first three triples all involve the same

book 1? This is easy to show in a diagram of the entire description graph, where we can have multiple edges attached to a node. But when we disaggregate that graph into triples, we need some way of uniquely referring to nodes. We need identifiers ("Choosing Good Names and Identifiers"). When two triples have nodes with the same identifier, we can know that it is the same node. RDF achieves this by associating URIs with nodes. (See "Resource Description Framework (RDF)")

The need to identify nodes when we break down an RDF graph into triples becomes important when we want to "write" RDF graphs—create textual representations of them instead of depicting them—so that they can be exchanged as data. Tree structures do not necessarily have this problem, because it is possible to textually represent a tree structure without having to mention any node more than once. Thus, one price paid for the generality and flexibility of graph structures is the added complexity of recording, representing or writing those structures.

Choosing Your Constraints

This tradeoff between flexibility and complexity illustrates a more general point about constraints. In the context of managing and interacting with resource descriptions, constraints are a good thing. As discussed above, a tree is a graph with very specific constraints. These constraints allow you to do things with trees that are not possible with graphs in general, such as representing them textually without repeating yourself, or uniquely identifying nodes by the path from the root of the tree to that node. This can make managing descriptions and the resources they describe easier and more efficient—*if* a tree structure is a good fit to the requirements of the organizing system. For example, an ordered tree structure is a good fit for the hierarchical structure of the content of a book or book-like document, such as an aircraft service manual or an SEC

filing. On the other hand, the network of relationships among the people and organizations that collaborated to produce a book might be better represented using a graph structure. XML is most often used to represent hierarchies, but is also capable of representing network structures.

Modeling within Constraints

A metamodel imposes certain constraints on the structure of our resource descriptions. But in organizing systems, we usually need to further specify the content and composition of descriptions of the specific types of resources being organized. For example, when designing a system for organizing books, it is not sufficient to say that a book's description is structured using XML, because the XML metamodel constrains structure and not the content of descriptions. We need also to specify that a book description includes a list of contributors, each entry of which provides a name and indicates the role of that contributor. This kind of specification is a *model* to which our descriptions of books are expected to conform. (See "Abstraction in Resource Description".)

When designing an organizing system we may choose to reuse a standard model. For example, ONIX for Books is a standard model (conforming to the XML metamodel) developed by the publishing industry for describing books.[22]

22. Online Information Exchange(ONIX) is the international standard for representing and communicating book industry product information in electronic form: http://www.editeur.org/11/Books/.

If no such standard exists, or existing standards do not suit our needs, we may create a new model for our specific domain. But we will not usually create a new metamodel: instead we will make choices from among the metamodels, such as JSON, XML, or RDF, that have been formally recognized and incorporated into existing standards. Once we have selected a metamodel, we know the constraints we have to work with when modeling the resources and collections in our specific domain.[23]

Specifying Vocabularies and Schemas

Creating a model for descriptions of resources in a particular domain involves specifying the common elements of those descriptions, and giving those elements standard names. (See "The Process of Describing Resources") The model may also specify how these elements are arranged into larger structures, for example, how they are ordered into lists nested into trees. Metamodels vary in the tools they provide for specifying the structure and composition of domain-specific models, and in the maturity and robustness of the methods for designing them.[24] RDF and XML each

23. Do not take on the task of creating a new XML model lightly. Literally thousands of XML vocabularies have been created, and some represent hundreds or thousands of hours of effort. See (Bray 2005) for advice on how to reduce the risk of vocabulary design if you cannot find an existing one that satisfies your requirements.

24. See (Glushko and McGrath 2005) for a synthesis of best

provide different, metamodel-specific tools to define a model for a specific domain. But not every metamodel provides such tools.

In XML, models are defined in separate documents known as *schemas*. An XML schema defining a domain model provides a vocabulary of terms that can be used as element and attribute names in XML documents that adhere to that model. For example, Onix for Books schema specifies that an author of a book should be called a `Contributor`, and that the page count should be called an `Extent`. An XML schema also defines rules for how those elements, attributes, and their content can be arranged into higher-level structures. For example, the Onix for Books specifies that the description of a book must include a list of `Contributor` elements, that this list must have at least one element in it, and that each `Contributor` element must have a `ContributorRole` child element.

If an XML schema is given an identifier, XML documents can use that identifier to indicate that they use terms and rules from that schema. An XML document may use vocabularies from more than one XML schema.[25] Associating a schema with an XML instance

practices for creating domain-specific languages in technical publishing and business-to-business document exchange contexts. You need best practices for big problems, while small ones can be attacked with *ad hoc* methods.

25. Unless an XML instance is associated with a schema, it is fair to say that it does not have any model at all because there is no way to understand the content and structure of the information it contains. The assignment

enables *validation*: automatically checking that vocabulary terms are being used correctly.[26]

of a schema to an XML instance requires a "Document Type Declaration." If some of the same vocabulary terms occur in more than one XML schema, with different meanings in each, using elements from more than one schema in the same instance requires that they be distinguished using *namespaces.* For example, if an element named "title" means the "title of the book" in one schema and "the honorific associated with a person" in another, instances might have elements with namespace prefixes like <book:title>The Discipline of Organizing</book:title> and <hon:title>Professor</hon:title>. Namespaces are a common source of frustration in XML, because they seem like an overly complicated solution to a simple problem. But in addition to avoiding naming collisions, they are important in schema composition and organization.

26. What "correctly" means depends on the schema language used to encode the conceptual model of the document type. The XML family of standards includes several schema languages that differ in how completely they can encode a document type's conceptual model. The Document Type Definition(DTD) has its origins in publishing and enforces structural constraints well; it expresses strong data typing through associated documentation resources. XML *Schema Definition*

If two descriptions share the same XML schema and use only that schema, then combining them is straightforward. If not, it can be problematic, unless someone has figured out exactly how the two schemas should "map" to one another. Finding such a mapping is not a trivial problem, as XML schemas may differ semantically, lexically, structurally, or architecturally despite sharing a common implementation form. (See *Describing Relationships and Structures*.)

Tree structures can vary considerably while still conforming to the XML Infoset metamodel. Users of XML often specify rules for checking whether certain patterns appear in an XML document (document-level validation). This is less often done with RDF, because graphs that conform to the RDF metamodel all have the same structure: they are all sets of triples. This shared structure makes it simple to combine different RDF descriptions without worrying about checking structure at the document level. However, sometimes it is desirable to check descriptions at the document level, as when part of a description is required. As with XML, if consumers of those descriptions want to assert that they expect those descriptions to have a certain structure (such as a required property), they must check them at the document level.

Because the RDF metamodel already defines structure, defining a domain-specific model in RDF mainly involves specifying URIs and names for predicates. A set of RDF predicate names and URIs is known as an *RDF vocabulary*. Publication of vocabularies on the web and the use of URIs to identify and refer to predicate definitions are

Language(XSD) is better for representing transactional document types but its added expressive power tends to make it more complex.

key principles of Linked Data and the *Semantic Web*. (Also see "The Semantic Web and Linked Data", as well as later in this chapter.)[27]

For example, the Resource Description and Access(RDA) standard for cataloging library resources includes a set of RDF vocabularies defining predicates usable in cataloging descriptions. One such predicate is:

```
<http://rdvocab.info/Elements/extentOfText>
```

which is defined as "the number and type of units and/or subunits making up a resource consisting of text, with or without accompanying illustrations." The vocabulary further specifies that this predicate is a refinement of a more general predicate:

```
<http://rdvocab.info/Elements/extent>
```

which can be used to indicate, "the number and type of units and/or subunits making up a resource" regardless of whether it is textual or not.

JSON lacks any standardized way to define which terms can be used. That does not mean one cannot use a standard vocabulary when creating descriptions using JSON, only that there is no agreed-upon way to use JSON to communicate which vocabulary is being used, and no way to automatically check that it is being used correctly.

Controlling Values

So far, we have focused on how models specify vocabularies of terms and how those terms can be used in descriptions. But models

27. For example, see Linked Open Vocabularies at http://lov.okfn.org/dataset/lov/index.html.

may also constrain the values or content of descriptions. Sometimes, a single model will define both the terms that can be used for property names and the terms that can be used for property values. For example, an XML schema may enumerate a list of valid terms for an attribute value.[28]

Often, however, there are separate, specialized vocabularies of terms intended for use as property values in resource descriptions. Typically these vocabularies provide values for use within statements that describe what a resource is about. Examples of

28. Attribute values can be constrained in a schema by specifying a data type, a default value, and a list of potential values. Data types allow us to specify whether a value is supposed to be a name, a number, a date, a token or a string of text. Having established the data type, we can further constrain the value of an attribute by specifying a range of values, for a number or a date, for example. We can also use *regular expression* patterns to describe a data type such as a postal code, telephone number or ISBN number. Specifying default values and lists of legal values for attributes simplifies content creation and quality assurance processes. In Schematron, a rule-based XML schema language for making test assertions about XML documents, we can express constraints between elements and attributes in ways that other XML schema languages cannot. For example, we can express the constraint that if two `<title>` elements are provided, then each must contain a unique string value and different `language` attribute values.

such subject vocabularies include the Library of Congress Subject Headings(LOC-SH) and the *Medical Subject Headings*(*MeSH*).[29] Other vocabularies may provide authoritative names for people, corporations, or places. Classification schemes are yet another kind of vocabulary, providing the category names for use as the values in descriptive statements that classify resources.

Because different metamodels take different approaches to specifying vocabularies, there will usually be different versions of these vocabularies for use with different metamodels. For example the LCSH are available both as XML conforming to the *Metadata Authority Description Standard*(*MADS*) schema, and as RDF using the *Simple Knowledge Organization System*(*SKOS*) vocabulary.

Specifying a vocabulary is just one way models can control what values can be assigned to properties. Another strategy is to specify what types of values can be assigned. For example, a model for book descriptions may specify that the value of a *pages* property must be a positive integer. Or it could be more specific; a course catalog might give each course an identifier that contains a two-letter department code followed by a 1-3 digit course number. Specifying a data type like this with a *regular expression* narrows down the set of possible values for the property without having to enumerate every possible value. (See the sidebar.)

In addition to or in lieu of specifying a type, a model may specify an encoding scheme for values. An *encoding scheme* is a specialized *writing system* or syntax for particular types of values. For example, a model like Atom for describing syndicated web content requires a publication date. But there are many different ways to write dates: `9/2/76`, `2 Sept. 1976`, `September 2nd 1976`, etc. Atom also

29. See LOC-SH as http://id.loc.gov/authorities/subjects.html; MeSH at http://www.nlm.nih.gov/mesh/.

specifies an encoding scheme for date values. The encoding scheme is RFC3339, a standard for writing dates. When using RFC3339, one always writes a date using the same form: `1976-09-02`.[30]

> Regular Expressions
>
> *Regular expressions* have been used to describe patterns in text documents since the early days of computing and came into widespread use when Ken Thompson incorporated them into early UNIX text processing tools, such as *ed* and *grep*. There are too many variations of regular expression syntax for us to detail them here, but it is worthwhile to consider them briefly while we are on the subject of controlling values.[31]
>
> Regular expressions are employed by modern text processing tools for selection and retrieval purposes. In search and replace applications, one might search for the string "`Chapter [1-5]`" to express your intent to select chapters 1 through 5, or "`it['']?s`" to locate

30. The Atom Publishing Protocol is IETF RFC 5023, (https://tools.ietf.org/html/rfc5023); a good introduction is (Sayre 2005). IETF RFC is http://www.ietf.org/rfc/rfc3339.txt.
31. There is no single authority on the subject of regular expressions or their syntax. A good starting point is the Wikipedia article on the subject: http://en.wikipedia.org/wiki/Regular_expression.

every use of “it’s” and “its” in a manuscript; this capability is highly valued by anyone who has had to edit a book. Programmers and data modelers use regular expressions to describe expected encoding schemes when they design documents, data elements, databases, and encoding schemes. You experience regular expression processing when you enter a phone number or postal code into a Web-based form. Many data modeling, programming and XML schema languages employ regular expressions to control data entry and validation of values. In the context of controlling values, we can use regular expressions to describe data values as varied as identifiers, names, dates, telephone numbers, and postal codes. We can, likewise, define rules for white space handling and punctuation within a data value.

Encoding schemes are often defined in conjunction with standardized identifiers. (See “Make Names Informative”.) For example, International Standard Book Numbers(ISBN) are not just sequences of Arabic numerals: they are values written using the ISBN encoding scheme. This scheme specifies how to separate the sequence of numerals into parts, and how each of these parts should be interpreted. The ISBN `978-3-8218-4448-0` has five parts, the first three of which indicate that the resource with this identifier is 1) a product of the book publishing industry, 2) published in a German-speaking country, and 3) published by the publishing house Eichborn.

Encoding schemes can be viewed as very specialized models of particular kinds of information, such as dates or book identifiers. But because they specify not only the structure of this information,

but also how it should be written, we can also view them as specialized *writing systems*. That is, encoding schemes specify how to *textually represent* information.

In the second half of this chapter, we will focus on the issues involved in textually representing resource descriptions—writing them. Graphs, trees, dictionaries, lists, and sets are general types of structures found in different metamodels. Thinking about these broad types and how they fit or do not fit the ways we want to model our resource descriptions can help us select a specific metamodel. Specific metamodels such as the XML Infoset or RDF are formalized and standardized definitions of the more general types of structures discussed above. Once we have selected a metamodel, we know the constraints we have to work with when modeling the *resources* and collections in our specific domain. But because metamodels are abstract and exist only on a conceptual level, they can only take us so far. If we want to create, store, and exchange individual resource descriptions, we need to make the structures defined by our abstract metamodels concrete. We need to write them.

60. Writing Descriptions

Suppose that I am organizing books, and I have decided that it is important for the purposes of this organizing to know the title of each book and how many pages it has. Before me I have a book, which I examine to determine that its title is *Die Ringe des Saturn* and it has 371 pages. Example: Basic ways of writing part of a book description. lists a few of the ways to write this description. Let us examine these various forms of writing to see what they have in common and where they differ.

Basic Ways Of Writing Part Of A Book Description

The title is *Die Ringe des Saturn* and it has 371 pages.
`{ book: {"title":"Die Ringe des Saturn","pages":371} }`
`<book pages="371"> <title>Die Ringe des Saturn</title> </book>`
`<div class="book">The title is <span class="title">Die Ringe des Saturn</span> and it has <span class="pages">371 pages.</span> </div>`
`<http://lccn.loc.gov/96103072> <http://rdvocab.info/Elements/title> "Die Ringe des Saturn"@de ; <http://rdvocab.info/Elements/extentOfText> "371 p." .`

We examine the notations, writing systems and syntax of each of these description forms, and others, in the following sections.

Notations

First, let us look at the actual marks on the page. To write you must make marks or—more likely—select from a menu of marks using a keyboard. In either case, you are using a *notation*: a set of characters with distinct forms.[1] The Latin alphabet is a notation, as are Arabic numerals. Some more exotic notations include the symbols used for editorial markup and alchemical symbols.[2] The characters in a notation usually have an ordering. Arabic numerals are ordered 1 2 3 and so on. English-speaking children usually learn the ordering of the Latin alphabet in the form of an alphabet song.[3]

A character may belong to more than one notation. The examples in Example: Basic ways of writing part of a book description. use characters from a few different notations: the letters of the Latin alphabet, Arabic numerals, and a handful of auxiliary marks: `. { } " :< > / $` Collectively, all of these characters—alphabet, numerals, and auxiliary marks—also belong to a notation called the American Standard Code for Information Interchange(ASCII).[4]

1. The terminology here and in the following sections comes from (Harris 1996).

2. See http://unicode.org/charts/PDF/U1F700.pdf.

3. Entitled "The ABC," the song was copyrighted in 1835 by Boston music publisher Charles Bradlee. It is sung to a tune that was originally developed by Wolfgang Amadeus Mozart, and is commonly recognizable as Twinkle, Twinkle, Little Star.

4. http://tools.ietf.org/html/rfc20.

ASCII is an example of a notation that has been codified and standardized for use in a digital environment. A traditional notation like the Latin alphabet can withstand a certain degree of variation in the form of a particular mark. Two people might write the letter A rather differently, but as long as they can mutually recognize each other's marks as an "A," they can successfully share a notation. Computers, however, cannot easily accommodate such variation. Each character must be strictly defined. In the case of ASCII, each character is given a number from 0 to 127, so that there are 128 ASCII characters.[5] When using a computer to type ASCII characters, each key you press selects a character from this "menu" of 128 characters. A notation that has had numbers assigned to its characters is called a *character encoding*.

ASCII

5. Only 95 of these characters are actually "marks" in the sense of being visible and printable. The other 33 ASCII characters are "control codes" that indicate things like horizontal and vertical tabs, the ends of printed lines, form feeds, and transmission control. We can think of many of these as special auxiliary marks, similar to the kind of symbols editors and proofreaders use to annotate texts.

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	“	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	–	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

The most ambitious character coding in existence is Unicode, which as of version 6.0 assigns numbers to 109,449 characters.[6] Unicode makes the important distinction between *characters* and *glyphs*. A *character* is the smallest meaningful unit of a written language. In alphabet-based languages like English, *characters* are letters; in languages like Chinese, characters are ideographs. Unicode treats all of these *characters* as abstract ideas (*Latin capital* A) rather than

6. The Unicode standard is maintained by a global non-profit organization. Everything you need to know is at http://www.unicode.org/.

specific marks (A A A **A**). A specific mark that can be used to depict a *character* is a *glyph*. A *font* is a collection of glyphs used to depict some set of *characters*. A Unicode font explicitly associates each glyph with a particular number in the Unicode character encoding. The inability of computers to use contextual understanding to bridge the gap between various glyphs and the abstract *character* depicted by those glyphs turns out to have important consequences for organizing systems.

Different notations may include very similar marks. For example, modern music notation includes marks for indicating the pitch of note, known as accidentals. One of these music notation marks is ♯ ("sharp"). The sharp sign looks very much like the symbol used in English as an abbreviation for the word *number*, as in *We're #1!*[7] If you were to write a sharp sign and a number sign by hand, they would probably look identical. In a non-digital environment, we would rely on context to understand whether the written mark was being used as part of music notation, or mathematical notation, or as an English abbreviation.

Computers, however, have no such intuitive understanding of context. Unicode encodes the number sign and the sharp sign as two different characters. As far as a computer using Unicode is concerned, ♯ and # are completely different, and the fact that they have similar-looking glyphs is irrelevant. That is a problem if, for example, a cataloger has carefully described a piece of music by

7. The Chinese character 井 (water well) looks like the # character too. The # symbol was historically used to denote pounds, the Imperial unit of weight, as in 10# of potatoes. In the United Kingdom, the # character is called"hash." We could go on, but we will leave it to you to discover more.

correctly using the sharp sign, but a person looking for that piece of music searches for descriptions using the number sign (since that is what you get when you press the keyboard button with the symbol that most closely resembles a sharp sign).[8]

Writing Systems

A *writing system* employs one or more notations, and adds a set of rules for using them. Most writing systems assume knowledge of a particular human language. These writing systems are known as *glottic* writing systems. But there are many writing systems, such as mathematical and musical ones, that are not tied to human languages in this way. Many of the writing systems used for describing resources belong to this latter group, meaning that (at least in principle) they can be used with equal facility by speakers of any language.

Glottic writing systems, being grounded in natural human languages, are difficult to describe precisely and comprehensively. Non-glottic writing systems, on the other hand, can be described precisely and comprehensively using an abstract model. That is the connection between the structural perspective taken in the previous section, and the textual perspective taken in this section. A non-glottic writing system is described by a particular metamodel,

8. To add to the confusion, while the American standard (ASCII) places the # character at position 23, the British equivalent (BS 4730) places the currency symbol £ at the same position. As a result, improperly configured computers sometimes display # in place of £ and vice versa.

and structures that fit within the constraints of a given metamodel can be textually represented using one or more writing systems that are described by that metamodel.

Some writing systems are closely identified with specific metamodels. For example, XML and JSON are *both* 1) metamodels for structuring information *and* 2) writing systems for textually representing information. In other words, they specify both the abstract structure of a description and how to write it down. It is possible to conceive of other ways to textually represent the structure of these metamodels, but for each of these metamodels just one writing system has been standardized.[9]

RDF, on the other hand, is *only* a metamodel, not a writing system. RDF only defines an abstract structure, not how to write that structure. So how do we write information that is structured as RDF? It turns out that we have many choices. Unlike XML and JSON, several different writing systems for the RDF metamodel have been standardized, including N-Triples, Turtle, RDFa, and RDF/XML.[10] Each of these is a writing system that is abstractly described by the RDF metamodel.

9. Recently, an alternative writing system for XML-structured data has been standardized: Efficient XML Interchange(EXI). However it is not yet widely used.

10. RDF/XML is a bit confusing; it is a writing system that uses XML syntax to textually represent RDF structure. This means that while XML tools can read and write RDF/XML, they cannot manipulate the graph structures it represents, because they were designed to work with XML's tree structures.

Writing systems provide rules for arranging characters from a notation into meaningful structures. A character in a notation has no inherent meaning. Characters in a notation only take on meaning in the context of a writing system that uses that notation. For example: what does the letter *I* from the Latin alphabet mean? That question can only be answered by looking at how it is being used in a particular writing system. If the writing system is American English, then whether *I* has a meaning depends on whether it is grouped with other letters or whether it stands alone. Only in the latter case does it have an assignable meaning. However in the arithmetic writing system of ancient Rome, which also uses as a notation the letters of the Latin alphabet, *I* has a different meaning: *one*.

This example also serves to illustrate how the ordering of a notation can differ from the ordering of a writing system that uses that notation. According to the ordering of the Latin alphabet, the twelfth letter L comes before the twenty-second letter V. But in the Roman numeric writing system, V (the number 5) comes before L (the number 50). Unless we know which ordering we are using, we cannot arrange L and V "in order."[11]

Roman Numerals

11. Although we use alphabetic characters today to represent Roman numerals, originally they were represented by unique symbols.

Roman Number	Arabic Number
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

This kind of difference in ordering can arise in more subtle ways as well. When we alphabetically order names, we first compare the first character of each name, and arrange them according to the ordering of the writing system. The first known use of alphabetical ordering was in the Library of Alexandria about two thousand years ago, when Zenodotus arranged the collection according to the first letter of resource names.[12] If the first characters of two names are the same, we compare the second character, and so on. We can also apply this same kind of ordering procedure to sequences of numerals. If we do, then 334 will come before 67, because 3 (the first character of the first sequence) comes before 6 (the first character of the second sequence) according to the ordering of our notation (Arabic numerals). However, it is more common when ordering sequences of numerals to treat them as decimal numbers,

12. It took a few hundred years before alphabetization became recursive and applied to letters other than the first (Casson 2002, p. 37). Alphabetization relies on the ordering of the writing system, not the notation. For example, Swedish and German are two writing systems that assign different orderings to the same notation.

and thus to use the ordering imposed by the decimal system. In the decimal writing system, 67 precedes 334, since the latter is a greater number.

This difference is important for organizing systems. Computers will sort values differently depending on whether they are treating sequences of numerals as numbers or just as sequences. Some organizing systems mix multiple ways of ordering the same characters. For example, Library of Congress call numbers have four parts, and sequences of Arabic numerals can appear in three of them. In the second part, indicating a narrow subject area, and fourth part, indicating year of publication, sequences of numerals are treated as numbers and ordered according to the decimal system. In the third part, however, sequences of numerals are treated as sequences and ordered "notationally" as in the example above (334 before 67).

Differences in ordering demonstrate just one way that multiple writing systems may use the same notation differently. For example, the American English and British English writing systems both use the same Latin alphabet, but impose slightly different spelling rules.[13] The Japanese writing system employs a number of notations, including traditional Chinese characters (*kanji*) as well as the Latin alphabet (*rōmaji*). Often, writing systems do not share the same exact notation but have mostly overlapping notations. Many

13. For example, the American spelling of the words "center" and "color" contrasts slightly with the English spelling of "centre" and "colour." There are too many examples to include here. Wikipedia has a comprehensive analysis of American and British spelling differences at http://en.wikipedia.org/wiki/American_and_British_English_spelling_differences.

European languages, for example, extend the Latin alphabet with characters such as Å and Ü that add additional marks, known as diacritics, to the basic characters.[14]

In organizing systems it is often necessary to represent values from one writing system in another writing system that uses a different notation, a process known as *transliteration*. For example, early computer systems only supported the ASCII notation, so text from writing systems that extend the Latin alphabet had to be converted to ASCII, usually by removing (or sometimes transliterating) diacritics. This made the non-ASCII text usable in an ASCII-based computerized organizing system, at the expense of information loss.

Even in modern computer systems that support Unicode, however, transliteration is often needed to support organizing activities by users who cannot read text written using its original system. The Library of Congress and the American Library Association provide standard procedures for transliterating text from over sixty different writing systems into the (extended) Latin alphabet.

14. ASCII's 128 characters are insufficient to represent these more complex character sets, so a new family of character encodings was created, ISO-8859, in which each encoding enumerates 256 characters. Each encoding thus has more space to accommodate the additional characters of regionally-specific notations. ISO 8859-5, for example, has extensions to support the Cyrillic alphabet.

Syntax

The examples in Example: Basic ways of writing part of a book description. express the same information using different writing systems. The examples use the same notation (ASCII) but differ in their *syntax*: the rules that define how characters can be combined into words and how words can be combined into higher-level structures.[15]

- Consider the first entry: *The title is* Die Ringe des Saturn *and it has 371 pages*. The leading capital letter and the period ending this sequence of characters indicate to us that this is a sentence. This sentence is one way we might use the English writing system to express two statements about the book we are describing. A *statement* is one distinct fact or piece of information. In glottic writing systems like English, there is usually more than one sentence we could write to express the same statement. For example, instead of *it has 371 pages* we might have written *the number of pages is 371*. English writing also enables us to construct complex sentences that express more than one statement.[16]

15. In discussions of glottic writing systems, "syntax" usually refers only to the rules for combining words into sentences. In discussions of programming languages, "syntax" has the broader sense we use here.

16. Compound sentences contain two independent clauses joined by a conjunction, such as "and," "or," "nor," "but." For example: I went to the store and I bought a book." Complex sentences contain an independent clause joined

> In contrast, when we create descriptions of resources in an organizing system, we generally use non-glottic writing systems in which each sentence only expresses a single statement, and there is just one way to write a sentence that expresses a given statement.[17] These restrictions make these writing systems less expressive, but simplify their use. In particular, since there is a one-to-one correspondence between sentences and statements, we can drop the distinction and just talk about the statements of a description.
>
> Now we return to our example and look at the structure of the statement, *The title is* Die Ringe des Saturn *and it has 371 pages*. Spaces are used to separate the text into words, and English syntax defines the functions of those words. The verb *is* in this statement functions to link the word *title* to the phrase *Die Ringe des Saturn*. This is typical of the kind of

by one or more dependent clauses. For example: "I read the book that I bought at the store."

17. In truth, even non-glottic writing systems designed to encode resource descriptions unambiguously can have variant forms of the same statement. For example, XML permits some variation in the way the same Infoset may be textually represented. Often these variations involve the treatment of content that may under some circumstances be treated as optional, such as white space. The difference is that in writing systems designed for resource description, these variations can be precisely enumerated and rules developed to reconcile them, while this is not generally true for glottic writing systems.

statements found in a resource description. Each statement identifies and describes some aspect of the resource. In this case, the statement attributes the value *Die Ringe des Saturn* to the property *title*.

As we saw when we looked at description structures, we can analyze descriptions as involving properties of resources and their corresponding values or content. In a writing system like English, it is not always so straightforward to determine which words refer to properties and which refer to values. (This is why blobs are not ideal description structures.) Writing systems designed for expressing resource descriptions, on the other hand, usually define syntax that makes this determination easier. In our dictionary examples above, we used an arrow character → to indicate the relationship between properties and values.

This ease of distinguishing properties and values comes at a price, however. The syntax of English is forgiving: we can read a sentence with somewhat garbled syntax such as *371 pages it has* and often still make out its meaning.[18] This is usually not the case with writing systems intended for expressing resource descriptions. These systems strictly define their rules for how characters can be combined into higher-level structures. Structures that follow the rules are *well formed* according to that system.

- Take for example the second entry in Example: Basic ways of writing part of a book description..

```
{ book: {"title":"Die Ringe des Saturn","pages":371} }
```

18. Fortunately for Yoda. There are many web services for converting English to Yoda-speak; an example is http://www.yodaspeak.co.uk/.

This fragment is written in JSON. As explained earlier in this chapter, JSON is a metamodel for structuring information using lists and dictionaries. But JSON is also a writing system, which borrows its syntax from JavaScript. The JSON syntax uses brackets to textually represent lists `[1,2,3]` and braces to textually represent dictionaries `{title:"Die Ringe des Saturn", "pages":371}`. Within braces, the colon character `:` is used to link properties with their values, much as is was used in the previous example. So `"pages":371` is a statement assigning the value `371` to the property `pages`.

- The third fragment is written in XML.

```
<book pages="371"> <title>Die Ringe des Saturn</title> </boo
```

  Like JSON, XML is a metamodel and also a writing system. Here we have XML elements and attributes. XML elements are textually represented as *tags* that are marked using the special characters `<`, `>` and `/`. So, this fragment of XML consists of a `book` element with a child element, `title`, and a `pages` attribute, each of which has some text content. In this case, `pages="371"` is a statement assigning the value 371 to the property `pages`. The difference is syntax is subtle; quotation marks surround the value and equal sign = is used to assign the property to its value.

- The fourth is a fragment of HTML.

```
<div class="book">The title is  <span class="title">Die Ring
```

  The writing system that HTML employs is close enough to XML to ignore any differences in syntax. In this example, the CLASS attribute contains the property name and the property value is the element content.

- The fifth entry is a fragment of Turtle, one of the writing systems for RDF.

```
<http://lccn.loc.gov/96103072> <http://rdvocab.info/Elements
```

Turtle provides a syntax for writing down RDF *triples*. Each triple consists of a subject, predicate, and object separated by spaces. Recall that RDF uses URIs to identify subjects, predicates, and some objects; these URIs are written in Turtle by enclosing them in angle brackets `<` `>`. Triples are separated by period `.` characters, but triples that share the same subject can be written more compactly by writing the subject only once, and then writing the predicate and object of each triple, separated by a semicolon `;` character. This is what we see in Example: Nesting an author description within a book description: two triples that share a subject.

The two fragments in Example: Writing part of a book description in Semantic XML. demonstrate namespaces, terms from the Dublin Core and DocBook namespaces, and the facility with which XML embraces semantic encoding of description resources.

Writing Part Of A Book Description In Semantic XML

```
<book xmlns:dc="http://purl.org/dc/terms/"
dc:extent="371 p."> <dc:title>Die Ringe des
Saturn</title> ... </book>
```

```
<book xmlns:db="http://www.docbook.org/xml/4.5/
docbookx.dtd"> <bookinfo> <title>Die Ringe des
Saturn</title> <pagenums>371
p.</pagenums>...</bookinfo> ... </book>
```

- The first example extends the third fragment from Example: Basic ways of writing part of a book description.; the `xmlns:dc="..."` segment is a namespace declaration, which is associating `dc` with the quoted URI, which happens to be the Dublin Core Metadata Initiative(DCMI); the child `<dc:title>`

element and the attached `dc:extent="371"` tell us that the corresponding values are attributable to the title and extent properties, respectively, from the Dublin Core namespace.

- The next fragment employs DocBook DTD namespace; we now have a `<pagenums>` element for which the meaning is contextually obvious; the title is still a title; an extra layer of markup reflects the fact that it could be metadata in the source file of a book that is being edited, is in production or is on your favorite tablet right now.[19]

> Microformats, RDFa and Microdata
>
> When Tim Berners-Lee deployed HTML, its syntax contained the basic elements and attributes needed to make formal statements about the document as a whole by using `<LINK/>`, or about specific parts of the document by using the `<A>` element. Each of these elements have four attributes in common: the famous HREF attribute contains a URI that names an object resource; the NAME attribute allows the element to be

19. DocBook (Walsh 2010) is widely used to publish academic, commercial, industrial book, scientific, and computing book, papers and articles. The book that you are reading is encoded with DocBook markup; complete bibliographic information for the book is contained within the source files, ready to be extracted on the way into one of the latest ebook formats.

the target end of a link; the REL and REV attributes contain descriptions of the link relations.

Microformats, RDFa and Microdata are the latest generation of metadata extensions to HTML. Each approach is widely used on the web and by search engines. As such, they are potential targets when transforming into HTML from richer semantic formats.

Microformats are the simplest of the three. It uses controlled vocabularies of terms in REL/REV, and in the CLASS attribute, to declare high-level information types.

RDFa is RDF in Attributes. That is, RDFa is a formal specification for writing RDF expressions by using attributes in XML and HTML documents. It uses an ABOUT attribute to name the subject of the relation; the REL and REV attributes; HREF is joined by SRC and RESOURCE to name the object of the link; a TYPEOF attribute declares a type; PROPERTY and CONTENT attributes are used to *attribute* a value to an object's property.

Microdata is similar, inasmuch as it uses attributes extensively. The presence of an ITEMSCOPE attribute identifies an item while the ITEMTYPE attribute value identifies its type; ITEMID declares an items name or unique identifier; ITEMPROP is a name value pair, and; ITEMREF relates this item to other elements that are outside of the scope of the container element.

The two fragments in Example: Writing part of a book description in RDFa or microdata. demonstrate RDFa and microdata formats, which each rely upon specific attributes to establish the type of the

property values contained by the HTML elements. In each example, the book title is contained by a `<span>` element. Whereas RDFa relies upon the `property` attribute, the microdata example employs the `itemprop` attribute to specify that the contents of the element is, effectively, a "title" in exactly the same sense as we know that the contents of `<dc:title>` is a "title."

Writing Part of a Book Description in RDFa or Microdata

```
<div class="book">The title is <span
property="http://purl.org/dc/terms/title">Die
Ringe des Saturn</span> and it has <span
property="http://purl.org/dc/terms/extent">371
p.</span></div>
```

```
<div itemscope itemtype="book">The title is <span
itemprop="http://purl.org/dc/terms/title">Die
Ringe des Saturn</span> and it has <span
itemprop="http://purl.org/dc/terms/extent">371
p.</span></div>
```

61. Worlds of Description

In the previous two sections we have considered descriptions as designed objects with particular structures and as written documents with particular syntaxes. As we have seen, there are many possible choices of structure and syntax. But these choices are never made in isolation. Just as an architect or designer must work within the constraints of the existing built environment, and just as any author must work with existing writing systems, descriptions are always created as part of a pre-existing "world" over which any one of us has little control.

In the final part of this chapter, we will consider how choices of structure and syntax have converged historically into broad patterns of usage. For lack of a better term, we call these broad patterns "worlds." "World" is not a technical term and should not be taken too literally: the broad areas of application sketched here have considerable overlap, and there are many other ways one might identify patterns of description structure and syntax. That said, the three worlds described here do reflect real patterns of description form that influence tool and technology choices. In your own work creating and managing resource descriptions, it is likely that you will need to think about how your descriptions fit into one or more of these worlds.

The Document Processing World

The first world we will consider is concerned primarily with the creation, processing and management of hybrid narrative-transactional documents such as instruction manuals, textbooks, or annotated medieval manuscripts. (See The Document Type Spectrum). These are quite different kinds of documents, but they

all contain a mixture of narrative text and structured data, and they all can be usefully modeled as tree structures. Because of these shared qualities, tools as different as publishing software, supply-chain management software, and scholarly editing software have all converged on common XML-based solutions. ("The XML world" would be another appropriate name for the document-processing world.)

This convergence was no accident, because XML was designed specifically to address the problem of how to add structure and data to documents by "marking them up." XML is the descendant of Standard Generalized Markup Language(SGML), which in turn descended from International Business Machines(IBM)'s Generalized Markup Language, which was invented to enable the production and management of large-scale technical documentation. The explicitness of markup makes it well-suited for representing structure and content type distinctions in institutional contexts, where the scope, scale, and expected lifetime of organizing systems for information implies reuse by unknown people for unanticipated purposes.

The abstract data model underlying XML is called the XML Information Set or Infoset. The Infoset defines a document as a partially ordered tree of "information items." Every XML document can thus be understood as a specific kind of tree, although not every tree structure is expressible as an XML document.[1]

1. It should be noted that the content of the Infoset for a given document may be affected by knowledge of any related DTDs or schemas. That is to say that, upon examination of a given XML document instance, its Infoset may be augmented with some useful information,

As we discussed in Inclusions and References, XML has the ability to describe graphs by incorporating the use of ID and IDREF attribute types to create references among element information items within the same document. This modest form of hypertext linking allows us to present the following document fragment that approximates the graph we saw modeled in Figure: Descriptions Linked into a Graph.

XML Implementation of a Biblio-graph

```
<person id="WG.Sebald">Winfried George Sebald</person>
<person id="MR.McCulloch>Mark Richard McCulloch</person>

<book>
    <title>Understanding W.G. Sebald</title>
    <subject idref="WG.Sebald"/>
    <author idref="WG.Sebald"/>
    <author idref="MR.McCulloch"/>
</book>

<book pages="371">
    <title lang="de">Die Ringe des Saturne</title>
    <title lang="en">The Rings of Saturn</title>
    <author idref="WG.Sebald"/>
</book>

<book pages="416">
    <title lang="de">Austerlitz</title>
    <author idref="WG.Sebald"/>
</book>
```

As one might expect, tools and technologies in the document-

such as default attribute values and attribute types. (See Inclusions and References.)

processing world are optimized for manipulating and combining tree structures. A "toolchain" is set of tools intended to be used together to achieve some goal.

> The XML Toolchain
>
> The XML toolchain is quite comprehensive. It consists of tools for creating XML documents (XML editors), tools for expressing logical document and data models (DTD, XML Schema, *REgular LAnguage for XML Next Generation*(RELAX NG), Schematron), tools for transforming XML documents (XSLT), tools for describing document processing "pipelines" (XP*roc: An XML Pipeline Language*), and tools for storing and querying collections of XML documents (XML databases, queried using XML Query Language(XQuery)). Used together, these tools provide very powerful means of working with tree-structured documents. XML editors incorporate knowledge of DTDs, schemas, transformations, style sheets, queries, databases and pipelines. Pipelines choreograph the plumbing and inter-dependencies involved in processing a complex dataset and publishing a useful result in one or more output formats.

For programmers who do not to use the XML toolchain, other programming languages also provide libraries for working with XML. This fact has led some to propose, and others to believe, that XML is a kind of universal format for exchanging data among systems. However, programmers have observed that a random XML Infoset does not map easily to the data structures commonly found in many programming languages. "Working with XML" frequently means translating from XML tree structures to data structures

native to another language, usually meaning lists and dictionaries. This translation can be problematic and often means giving up many of the strengths of XML. By the same token, there are decades more practical experience working with markup languages and institutional publishing than there is with JSON and RDF.

XML is not a universal solution for every possible problem. That does not mean that it is not the best solution for a wide variety of problems, including yours. To gauge whether your resource descriptions are, or ought to be, part of the document-processing world, ask yourself the following questions:

- Do my resource descriptions contain mixtures of narrative text, hypertext, structured data and a variety of media formats?
- Can my descriptions easily be modeled using tree structures, hypertext links, and transclusion?
- Are the vocabularies I need or want to use made available using XML technologies?
- Do I need to work with a body of existing descriptions already encoded as XML?
- Do I need to interoperate with processes or partners that utilize the XML toolchain?
- Do I need to publish my resource descriptions in multiple formats from a single source?

If the answer to one or more of these questions is "yes," then chances are good that you are working within the document processing world, and you will need to become familiar with conceptualizing your descriptions as trees and working with them using XML tools.

The Web World

The second "world" emerged in the early 1990s with the creation of the World Wide Web. The web was developed to address a need for simple and rapid sharing of scientific data. Of course, it has grown far beyond that initial use case, and is now a ubiquitous infrastructure for all varieties of information and communication services. ("The browser world" would be another appropriate name for what we are calling the Web World.)

Documents, data, and services on the web are conceptualized as resources, identified using Uniform Resource Identifiers(URI), and accessible through *representations* transferred via Hypertext Transfer Protocol(HTTP). Representations are sequences of bytes, and could be HTML pages, JPEG images, tabular data, or practically anything else transferable via HTTP. No matter what they are, representations transferred over the web include descriptions of themselves. These descriptions take the form of property-value pairs, known as "HTTP headers." The HTTP headers of web representations are structured as dictionaries.

Dictionary structures appear many other places in web infrastructure. URIs may include a *query* component beginning with a `?` character. This component is used for purposes such as providing query parameters to search services. The query component is commonly structured as a dictionary, consisting of a series of property-value pairs separated by the `&` character. For example, the following URI:

```
https://www.google.com/search?q=sebald&tbs=qdr:m
```

includes the query component `q=sebald&tbs=qdr:m`. This is a dictionary with the properties `q` and `tbs`, respectively specifying the search term and temporal constraints on the search.

Data entered into an HTML form is also structured as a dictionary.

When an HTML form is submitted, the entered data is used either to compose the query component of a URI, or to create a new representation to be transferred to a web server. In either case, the data is structured as a set of properties and their corresponding values.

HTML documents are structured as trees, but descriptions embedded within HTML documents can also be structured as dictionaries. HTML documents may include a dictionary of metadata elements, each of which specifies a property and its value. Recently support for *microdata* was added to HTML, which is another method of adding dictionaries of property-value pairs to documents. Using *microdata*, authors can annotate web content with additional information, making it easier to automatically extract structured descriptions of that content.[2] *Microformats* are another method for doing this by mapping existing HTML attributes and values to (nested) dictionary structures.[3]

Dictionary structures are easy to work with in any programming language, and they pervade various popular frameworks for

2. Microdata is an invention of WHATWG and exists and part of what they call a "living standard." It was supported by Google, so it was widely used and there exist numerous controlled vocabularies, including those for creative works, persons, events and organizations. Support for microdata has since been withdrawn from Apple Safari and Google Chrome browsers.

3. Microformats is a non-standard that emerged from the community and has been sponsored by CommerceNet and Microformats.org.

programming the Web. In the programming languages used to implement web services, HTTP headers and query parameters are easily mapped to dictionary data structures native to those languages. On the client side, there is only one programming language that runs within all web browsers: JavaScript. The dictionary is the fundamental data structure within JavaScript as well.

Thus it is unsurprising that JSON, a dictionary-structured, JavaScript-based syntax, has become the *de facto* standard for application-to-application interchange of data on the web in contexts that do not involve business transactions. Web services providing structured data intended for programmatic use can make that data available as JSON, which is well-suited for use either by JavaScript programs running within browsers, or by programs written in other languages running outside of browsers (e.g., smart phone applications).

It is now commonly accepted that there are useful differences of approach between the document-processing world and the Web World. This does not mean that the two worlds do not have significant overlaps. Some very important web representation types are XML-based, such as the Atom syndication format. Trees will continue to be the structure of choice for web representations that consist primarily of narrative rather than transactional data. But for structured descriptions that are intended to be accessed and manipulated on the Web, dictionary structures currently rule.

To gauge whether your resource descriptions are or ought to be part of the Web world, ask yourself the following questions:

- Is the web the primary platform upon which I will be making my descriptions available?
- Are my resource descriptions primarily structured, transaction-oriented data?

- Can my descriptions easily be modeled as lists of properties and values (dictionaries)?
- Are the vocabularies I need or want to use made available primarily using HTML technologies such as microdata or microformats?
- Do I need to make my descriptions easily usable for use within a wide array of programming languages?

If the answer to one or more of these questions is "yes," then chances are good that you are working within the Web World, and you will need to become familiar with conceptualizing your descriptions as dictionaries and working with them using programming languages such as JavaScript.

The Semantic Web World

The last world we consider is still somewhat of a possible world, at least in comparison with the previous two. While the document processing world and the web world are well-established, the Semantic Web world is only starting to emerge, despite having been envisioned over a decade ago.

The vision of a *Semantic Web* world builds upon the web world, but adds some further prescriptions and constraints for how to structure descriptions. The Semantic Web world unifies the concept of a resource as it has been developed in this book, with the web notion of a resource as anything with a URI. On the Semantic Web, anything being described must have a URI. Furthermore, the descriptions must be structured as graphs, adhering to the RDF metamodel and relating resources to one another via their URIs.

Advocates of Linked Data further prescribe that those descriptions must be made available as representations transferred over HTTP.[4]

This is a departure from the web world. The web world is also structured around URIs, but it does not require that every resource being described have a URI. For example, in the web world a list of bibliographic descriptions of books by W.G. Sebald might be published at a specific URI, but the individual books themselves might not have URIs. In the Semantic Web world, in addition to the list having a URIs, each book would have a URI too, in addition to whatever other identifiers it might have.[5]

Making an HTTP request to an individual book URI may return a graph-structured description of that book, if best practices for Linked Data are being followed. This, too, is a departure from the web world, which is agnostic about the form representations or descriptions of resources should take (although as we have seen, dictionary structures are often favored on the web when the clients consuming those descriptions are computer programs). On the Semantic Web, all descriptions are structured as RDF graphs. Each description graph links to other description graphs by referring to

4. (Bizer, Heath, and Berners-Lee 2009).

5. It is worth noting that URIs are not required to have anything at their endpoints. Resolvability of URIs is evangelized as a best practice for Linked Data but not a requirement within the broader Semantic Web paradigm. Merely asserting that a URI is associated with a book is enough. If the URI can return a description or a resource, so much the better, but if not, at least you can talk about the book by referring to the same URI.

these related resources using their URIs. Thus, at least in theory, all description graphs on the Semantic Web are linked into a single massive graph structure. In practice, however, it is far from clear that this is an achievable, or even a desirable, goal.

Although the Semantic Web is in its infancy, a significant number of resource descriptions have already been made available in accordance with the principles outlined above. Descriptions published according to these principles are often referred to as "Linked Data." Prominent examples include: DBpedia, a graph of descriptions of subjects of Wikipedia articles; the *Virtual International Authority File*(VIAF), a graph of descriptions of names collected from various national libraries' name authority files; GeoNames, a graph of descriptions of places; and Data.gov.uk, a graph of descriptions of public data made available by the UK government.[6]

Despite the growing amount of Linked Data, tools for working with graph-structured data are still immature in comparison to the XML toolchain and Web programming languages. Although there is an XML syntax for RDF, using the XML toolchain to work with graph-structured data is generally a bad idea. And just as most programming languages do not support natively working with tree structures, most do not support natively working with graph structures either. Storing and querying graph-structured data efficiently requires a graph database or *triple store*.

Still, the Semantic Web world has much to recommend it. Having a common way of identifying resources (the URI) and a single shared metamodel (RDF) for all resource descriptions makes it much easier to combine descriptions from different sources. To gauge whether

6. Many more available datasets are listed at linkeddata.org.

your resource descriptions are or ought to be part of the Semantic Web world, ask yourself the following questions:

- Is the web the primary platform upon which I will be making my descriptions available?
- Is it important that I be able to easily and freely aggregate the elements of my descriptions in different ways and to combine them with descriptions created by others?
- Are my descriptions best modeled as graph structures?
- Have the vocabularies I need or want to use been created using RDF?
- Do I need to work with a body of existing descriptions that have been published as Linked Data?

If the answer to one or more of these questions is "yes," then chances are good that you should be working within the Semantic Web world, and you ought to become familiar with conceptualizing your descriptions as graphs and working with them using Semantic Web tools.

62. Key Points in Chapter Nine

- What are two perspectives on forming resource descriptions?

  We can approach the problem of how to form resource descriptions from two perspectives: structuring and writing.

  (See "Introduction")

- Are metamodels domain-specific?

  Metamodels describe structures commonly found in resource descriptions and other information resources, regardless of the specific domain.

  (See "Structuring Descriptions")

- What do blobs, sets, lists, dictionaries, trees, and graphs have in common?

  Blobs, sets, lists, dictionaries, trees, and graphs are all kinds of structures that can be used to form resource descriptions.

  (See "Kinds of Structures")

- What is a list?

  A *list*, like a set, is a collection of items with an additional constraint: their items are *ordered*.

  (See "Lists")

- What is a dictionary?

  A *dictionary*, also known as a *map* or an *associative array*, is a set of property-value pairs or *entries*.

(See “Dictionaries”)

- What is a nested dictionary?

  Nested dictionaries form a *tree*.

  (See “Dictionaries”)

- What is a tree?

  Trees consist of *nodes* joined by *edges*.

  (See “Trees”)

- What are the two kinds of data structures used by JSON?

  JSON consists of two kinds of structures: lists (called *arrays* in JavaScript) and dictionaries (called *objects* in JavaScript).

  (See “JSON”)

- What is the XML Infoset?

  The XML Infoset is a tree structure, where each node of the tree is defined to be an *information item* of a particular type.

  (See “XML Information Set”)

- What is the benefit of a data schema?

  Using schemas to define data representation formats is a good practice that facilitates shared understanding and contributes to long-term maintainability.

  (See “XML Information Set”)

- What is RDF?

  The RDF metamodel is a directed graph, so it identifies one node (the one from which the edge is pointing) as the *subject* of the triple, and the other node (the one to which the edge is

pointing) as its *object*. The edge is referred to as the *predicate* or (as we have been saying) *property* of the triple.

(See "RDF")

- What is an encoding scheme?

  An "encoding scheme" is a specialized writing system or syntax for particular types of values. Encoding schemes specify how to *textually represent* information.

  (See "Notations")

- What is a writing system?

  A *writing system* employs notations, and adds a set of rules for using them.

  (See "Writing Systems")

- How could one notation be used in multiple writing systems?

  Differences in ordering demonstrate just one way that multiple writing systems may use the same notation differently.

  (See "Writing Systems")

- What is syntax?

  Syntax is the rules that define how characters can be combined into words and how words can be combined into higher-level structures.

  (See "Syntax")

- What are the concerns of the document processing world?

  The document processing world is concerned primarily with the creation, processing and management of hybrid narrative-transactional documents.

(See "The Document Processing World")

- How are resources conceptualized in the Web world?

  In the web world, documents, data, and services are conceptualized as resources, identified using Uniform Resource Identifiers(URI), and accessible through *representations* transferred via the Hypertext Transfer Protocol (HTTP).

  (See "The Web World")

- What is a resource in Semantic Web terms?

  The Semantic Web world unifies the concept of a resource as it has been developed in this book, with the web notion of a resource as anything with a URI. Descriptions must be structured as graphs, adhering to the RDF metamodel and relating *resources* to one another via their URIs.

  (See "The Semantic Web World")

PART X

INTERACTIONS WITH RESOURCES

Vivien Petras

Robert J. Glushko

Ian MacFarland

Karen Joy Nomorosa

J.J.M. Ekaterin

Hyunwoo Park

Robyn Perry

Sean Marimpietri

63. Introduction (X)

Picture a dim room in the basement of a Detroit police station, lined with metal shelves: the shelves contain boxes and boxes of cold case files, evidence meticulously logged and categorized for no one to look at, documenting murders that will never be solved. Or the library of a small-town historical society in New Jersey: struggling with budget cuts, the board of directors has been forced to close its doors, locking its treasures inside, carefully curated and preserved but inaccessible to the public. Or a valuable data store encoded in an orphaned storage format: business records in a legacy database system that will not run on modern computers, census data on proprietary magnetic tape reels from the 1970s, your unfinished novel on a series of eight-inch floppy disks. You know the data is there, but you cannot interact with it.

An organizing system without interactions is a sad one indeed.

Interactions are the answer to two of the fundamental questions we posed back in *Foundations for Organizing Systems*: why and when are the resources organized?

The question of "why?" has been in the background (and often the foreground) of every chapter in this book thus far; whenever we select a resource for inclusion in an organizing system, describe it, or arrange it according to an organizing principle, we have an interaction in mind. We include a resource in our system because our users will need it; we assign a resource to one or more categories to help our users find it, understand it, and connect it with other resources in a meaningful way.

In this chapter we will pivot from design for interactions to the design of interactions—and to do this we must pause to consider the question of "when?" In "When Is It Being Organized?", we contrasted organization done "on the way in" with that done "on the way out,"

but this distinction is not always a particularly relevant one. Consider a bookshelf: if you do not organize its resources on the way in (i.e., when you put a book on the shelf), you cannot really organize them on the way out; you just have a disorganized bookshelf. When the time comes to retrieve a book, you'll have to employ a brute-force linear search algorithm—reading every spine until you find the one you want, and it will not make the remaining books on the shelf any more organized.

But digital resources and networked organizing systems are an entirely different story. In fact, we argue that they blur the traditional boundary between the academic disciplines of "information organization" and "information retrieval"; with the World Wide Web, ubiquitous digital information, and effectively unlimited processing, storage, and communication capability driven by cloud computing architecture and Moore's law, billions of people create and browse websites, blog, tag, tweet, and upload and download content of all media types without thinking "I am organizing now" or "I am retrieving now." When people use their smartphones to search the web or run applications, location information transmitted from their phone is used to filter and reorganize the information they retrieve. Arranging results to make them fit the user's location is a kind of computational curation, but because it takes place quickly and automatically we hardly notice it. Likewise, almost every application that once seemed predominantly about information retrieval is now increasingly combined with activities and functions that most would consider to be information organization.

Thus we come to the question of when a system's resources are organized: we may apply the techniques of computational information retrieval to a set of resources that simply are not organized the way we need them to be in order to support our desired interaction. Maybe the system was designed poorly or for a different purpose than the one we are pursuing; maybe we are attempting to collect or aggregate resources from multiple

organizing systems, each of which has its own separate purposes and design flaws. Regardless of the reasons, what we are essentially doing is reorganizing these resources on the fly, or "on the way out," following many of the same principles and procedures we've covered in the preceding eight chapters of this book.

Most Common Museum Interaction

Because museums often contain extremely rare or valuable resources that do not circulate, their most popular items are mobbed by visitors. The crowding often makes it impossible to get a good look at the rare item. This ironic situation is typified by the crowd control cordon that creates a 20-foot barrier around La Gioconda (aka "The Mona Lisa") at Musée du Louvre in Paris.

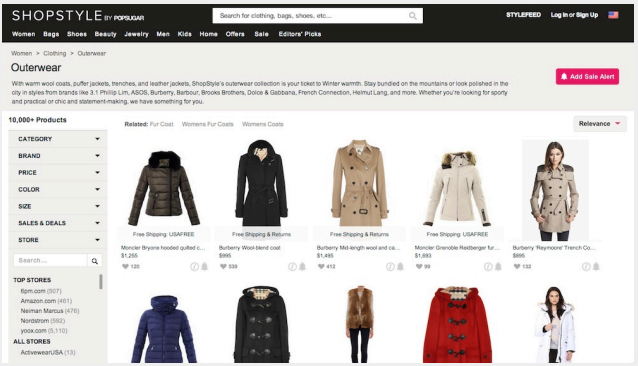
(Photo by R. Glushko.)

The fundamental interaction of any organizing system is accessing resources or resource descriptions, whether physically or digitally. Sometimes we must combine or merge resources or resource descriptions to access them effectively; this poses numerous strategy, design, and implementation challenges, as producers often use different identifiers, description or cataloging formats, and practices for similar resources. Different service providers use different technologies, have different information policies, and follow different processes developed in their separate organizing systems.

Some organizing systems have the power to determine the description standards that others must use. Walmart, the largest retailer in the United States, has devised an organizing system for its supply chain that supports access and movement of physical goods with maximal efficiency and effectiveness. This system saves the corporation money on inventory management and distribution, but to maximize savings, Walmart requires its suppliers to employ the same data model, follow company-set standards, and adopt new technologies such as bar codes and RFID tags that support the highly efficient interactions it requires.[1]

Browsing Merchandise Catalogs

1. Walmart uses its market power to impose technology and process decisions on its suppliers and partners. See (Fishman 2003), (Grean and Shaw 2005), (Wilbert 2006). Walmart's website for suppliers is http://walmartstores.com/Suppliers/248.aspx/



Shopstyle.com provides a transparent interface to the catalogs of hundreds of other online clothing retailers, aggregating their listings to allow users to browse them all from a single page.

(Screenshot by Ian MacFarland.)

Other organizing systems must adapt to whatever their counterparts develop. Online retailer Shopstyle.com presents a typical ecommerce interface, allowing shoppers to browse a multitude of fashion and beauty products organized into familiar categories. But behind the scenes, Shopstyle is aggregating the catalogs of more than 250 online stores and providing a seamless access interaction for all their merchandise. It does not actually sell anything: it directs shoppers to those third-party stores to make their purchases. Rather than moving physical resources like Walmart, Shopstyle's most important interactions involve moving and combining digital resource descriptions.

Still others choose to abide by what a standard-setting body decides, or participate in laborious, democratic processes to align

their organizing practices and interactions.[2] Libraries and museums

2. In order to more easily use and reuse content, as well as have the ability to integrate different learning tools into a single *Learning Management System (LMS)*, Global Learning Consortium, an organization composed of 140 members from leading educational institutions and education-related companies, has released specifications to make this possible. Called Common Cartridge and Learning Tools Interoperability (http://www.imsglobal.org/commoncartridge.html), the specifications provide a common format and guidelines to construct tools and create content that can be easily imported into learning management systems. Common Cartridge(CC) specifications give detailed descriptions of the directory structure, metadata and information models associated with a particular learning object. For example, a learning package from a provider from McGraw-Hill may contain content from a book, some interactive quizzes, and some multimedia to support the text. CC specifies how files would be organized within a directory, how links would be represented, how the package would communicate with a backend server, how to describe each of the components, and the like. This would enable a professor or a student using any capable learning management system to import a "cartridge" or learning material and have it appear in a consistent manner with all other learning materials within the LMS.

are the classic examples of this. The most important interaction in a library, of course, is borrowing: checking out a book to use it off the premises, and checking it back in when you're done. Patrons search descriptions in a catalog to find books on a certain topic, by a certain author, or with a certain title, and access them by fetching them from the stacks or asking a librarian to retrieve them. As institutions that serve the public interest, libraries adhere to

This means that content providers need not maintain multiple versions of the same content just to conform to the formats of different systems, allowing them to focus their resources on creating more content as opposed to maintaining the ones they already have. Looking at this in the context of the interoperability framework, we see that while information from providers are in a structured digital form, the main problem was that users were consuming the content using competing systems that had their own data formats by which to accept content. Huge publishers, wanting to increase distribution of their product, offered their content in all these different formats. While the specifications that the LMS created refer to the technical considerations in creating content and tools, the process of getting to that point involved a lot of organizational and political discussions. Internally, content and LMS providers needed to set aside the necessary resources to refactor their products to conform to the standards. Externally, competing providers had to collaborate with one another to create the specifications.

standards and democratic processes to ensure consistent and familiar user experiences for patrons, but also to enable powerful search interactions such as union catalogs, where resource descriptions from multiple libraries are merged before they are offered for search. Union catalogs allow patrons to find out with a single search whether a resource is available from any library that is accessible to them.[3]

Museums serve the public interest as well, and employ standards and democratic procedures for similar reasons as libraries, but their visitors generally look at their resources rather than borrowing them. Museums enable people to discover or experience resources by exhibiting artifacts in creative contexts, and when they implement this interaction digitally, as in a website, they vastly increase the opportunity for public access. Virtual collections are accessible to remote patrons who are unable to visit the physical museum, and they allow access to resources that are not currently on view.

The digitization of museum resources also allows visitors to experience them from a perspective that might not be possible in a physical museum. For example, in Google's Art Project, users can zoom in to view fine details of digitized paintings.[4] Museums are starting to leverage technology and the popularity of Web 2.0 features such as tagging and social networking to attract new audiences.[5]

Implemented in 2004, the MuseumFinland project aims to provide

3. http://www.worldcat.org/.

4. (Proctor 2011).

5. (Srinivasan, Boast, Furner, Becvar 2009).

a portal for publishing heterogeneous museum collections on the Semantic Web.[6] Institutions such as the Getty Information Institute and the International Committee for Documentation of the International Council of Museums have worked on standards that ensure worldwide consistency in how museums manage information about their collections.[7]

How can these differences be handled in order to provide seamless interactions within and across organizing systems? Which requirements have to be met in order to provide the interactions that are desired? How are different interaction types implemented? Finally, how can the quality of interactions be evaluated with respect to their requirements? These are the main questions for interactions that we will try to answer.

6. (Hyvönen et al. 2004). Museum visitors are presented with intelligent, content-based search and browsing services that offer a consolidated view across Finnish museums from the National Museum to the Lahti City Museum. To enable these goals, MuseumFinland mapped the variety of existing terms used by different museums onto shared ontologies, which now enable aggregated searching and browsing.

7. (Bower and Roberts 2001).

Navigating This Chapter

This chapter concentrates on the processes that develop interactions based on leveraging the resources of organizing systems to provide valuable services to their users (human or computational agents). It will discuss the determination of the appropriate interactions ("Determining Interactions"), the organization of resources for interactions ("Reorganizing Resources for Interactions"), the implementation of interactions ("Implementing Interactions"), and their evaluation and adaptation ("Evaluating Interactions"). Although the fundamental questions pertain to all types of organizing systems, this chapter focuses on systems that use computers to satisfy their goals.

64. Determining Interactions

Creating a strategy for successfully implementing interactions involves an intricate balance between the resources, the organizing system that arranges and manages them, its producers, and its intended users or consumers. The design of interactions is driven by user requirements and their impact on the choices made in the implementation process. It is constrained by resource and technical system properties and by social and legal requirements. Determining the scope and scale of interactions requires a careful analysis of these individual factors, their combination, and the consequences thereof.

> Stop and Think: Constraint vs Flexibility
>
> Think of an information organization project you were involved in. Can you recall ways in which you were constrained in representing an idea by the organizing system the project was implemented with? In what ways was the project negatively affected by the implementation? In what ways might the constraint have had a positive effect?

It is useful to distinguish decisions that involve choices, where multiple feasible alternatives exist, from decisions that involve constraints, where design choices have been eliminated or rendered infeasible by previous ones. The goal when creating an organizing system is to make design decisions that preserve subsequent choices or that create constraints that impose design decisions that would have been preferred anyway.

User Requirements

Users (human or computational agents) search or navigate resources in organizing systems not just to identify them, but also to obtain and further use the selected resources (e.g., read, cluster, annotate, buy, copy, distribute, adapt, etc.). How resources are used and by whom affects how much of the resource or its description is exposed, across which channels it is offered, and the *precision* and *accuracy* of the interaction.

An organizing system should enable interactions that allow users to achieve their goals. The more abstract and intermediated the interaction between a user and an organizing system becomes, the more precisely the requirements must be expressed. User requirements can be stated or implied, depending on the sophistication and functional capabilities of the system.

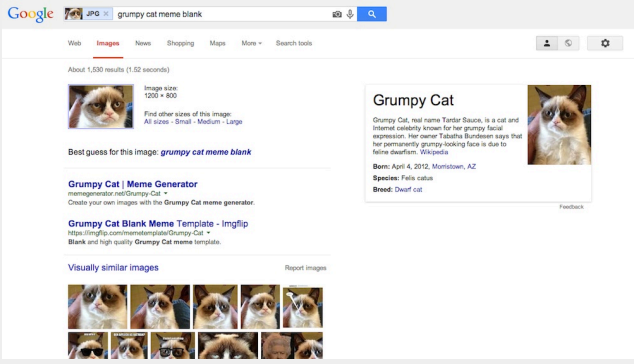
In a closet, which is a personal organizing system for physical resources, the person searching with an intent to find a particular shirt might think, "Where is my yellow Hawaiian shirt?" but does not need to communicate the search criteria to anyone else in an explicit way. In a business or institutional organizing system, however, the user needs to describe the desired resource and interact with the system to select from candidate resources. This interaction might involve a human intermediary like a salesperson or reference librarian, or a computational one like a search engine.

A user's information need usually determines the kind and content of resources required. User information needs are most often expressed in search queries (whatever is typed into a search box) or manifest themselves in the selection of one or more of the system categories that are offered for browsing. Queries can be as simple as a few keywords or very complex and specialized, employing different search fields or operators; they may even be expressed in a query language by expert users. Techniques such as spelling

correction, query expansion, and suggestion assist users in formulating queries. Techniques like breadcrumb navigation and faceted filtering assist users in browsing an organizing system's category system. Some systems allow the query to be expressed in natural language and then transform it into a description that is easier for the system to process. Queries for non-textual information like photos or videos are typically expressed as text, but some systems compute descriptions from non-textual queries such as images or audio files. For example, a user can hum a tune or draw or drag an image into an image query box.

Information needs of computational agents are determined by rules and criteria set by the creators of the agents (i.e., the function or goal of the agent). When a computational agent interacts with another computational agent or service by using its API, in the ideal case its output precisely satisfies those information needs.

Google Image Search



Google's Image Search tool can accept an image file as

an input rather than text, and will find visually similar images as well as making its best guess of the image's subject matter.

(Screenshot by Ian MacFarland.)

While search queries are explicitly stated user information needs, organizing systems increasingly attempt to solicit the user's context or larger work task in order to provide more suitable or precise interactions. Factors such as level of education, physical disabilities, location, time, or deadline pressure often specify and constrain the types of resources needed as well as the types of interactions the user is willing or able to engage in. Implicit information can be collected from user behavior, for example, search or buying history, current user location or language, and social or collaborative behavior (other people with the same context). Methods for explicitly soliciting user requirements include observation, surveys, focus groups, interviews, work task analysis and many more.[1]

Designers of organizing systems must recognize that people are

1. A conceptual framework for analyzing users and their work tasks for design requirements is (Fidel 2012). A general survey of design methods is given in (Hanington and Martin 2012). Designing particularly for successful interactions (services) is discussed in (Polaine, Løvlie and Reason). (Resmini and Rosati 2011) describe designing for engaged users using cross-channel, cross-media information architecture.

not perfectly capable and rational decision makers. Limited memory and attention capacities prevent people from remembering everything and make them unable to consider more than a few things or choices at once. As a result of these fundamental limitations, people consciously and unconsciously reduce the cognitive effort they make when faced with decisions.

Behavioral Economics

Classical economics assumes that humans are perfectly rational goal-oriented actors who act to achieve maximal satisfaction or utility. In contrast, behavioral economics recognizes the cognitive and emotional constraints on human behavior and assumes that people are biased and flawed decision makers.

Daniel Kahneman and Amos Tversky systematized the psychological foundations for behavioral economics, building on the work of Herbert Simon, who first proposed to understand people as "boundedly rational." Kahneman and Tversky identified the systematic biases that prevent people from making optimal decisions and the heuristics they use to save cognitive effort. Kahneman contrasts classical and behavioral economics as follows:

> Psychological theories of intuitive thinking cannot match the elegance and precision of formal normative models of belief and choice, but this is just another way of saying that rational models are psychologically unrealistic.
>
> (Kahneman 2003, p 1449)

Sunstein and Thaler popularized the application of behavioral economics as "libertarian paternalism," with the goal of encouraging the design of organizing systems and policies that maintain or increase freedom of choice but which at the same time influence people to make choices that they would judge as good ones. This perspective is nicely captured by the title of their best-selling book, *Nudge.* Many government agencies and businesses in the US and elsewhere are building "nudging" principles into policies and products in the areas of social services, healthcare, and financial services because of the complexity of their offerings.

Behavioral economics complements the discipline of organizing by offering insights into the thinking and behavior of typical users that can lead to classifications and choices that make them more effective and satisfied. However, the principles of behavioral economics can be used to design organizing systems that manipulate people into taking actions and making choices that they might not intend or that are not in their best interests. (See Dark Patterns.)[2]

One important way in which this affects how people behave demonstrates what Barry Schwartz calls The Paradox of Choice. You might think that people would prefer many options rather than just

2. (Simon 1982), (Tversky and Kahneman 1974), (Kahneman and Tversky 1979), (Kahneman 2003), and (Thaler 2008).

a few because that would better enable them to select a resource that best meets their requirements. In fact, because considering more choices requires more mental effort, this can cause stress and indecision and might cause people to give up. For example, when there were 24 different types of jam offered at an upscale market, more people stopped to taste than when only 6 choices were offered, but a greater percentage of people who were presented a smaller number of options actually made a purchase.[3]

We see the same phenomenon when we compare libraries and bookstores. A rational book seeker should prefer the detailed classification system used in libraries over the very coarse BISAC system used in bookstores. However, many people say that the detailed system makes them work too hard, leading to calls that new libraries adopt the bookstore organizing system. (See "The BISAC Classification")

People can avoid making choices if a system proposes or pre-selects an option for them that becomes a default choice if they do nothing. Often people will make a cursory assessment about how well the option satisfies a requirement and if it is good enough they will not consider any other alternatives.[4]

The study of the limits to human rationality in decision-making is the centerpiece of the discipline known as Behavioral Economics.

Organizing systems should plan for interactions based on non-

3. (Schwartz 2005) and (Iyengar 2000)

4. A comprehensive review of the power of defaults in software and other technical systems from the perspectives of law, computer science, and behavioral economics is (Kesan and Shah 2006)

purposeful user behavior. A user who does not have a particular resource need in mind might interact with an organizing system to see what it contains or to be entertained or educated. Imagine a user going to a museum to avoid the heat outside. Their requirement is to be out of the heat and—possibly—to see interesting things. A visitor to a zoo might go there to view a specific animal, but most of the time, visitors follow a more or less random path among the zoo's resources. Similarly, web surfing is random, non-information-need-driven behavior. This type of requirement cannot be satisfied by providing search capabilities alone; other interaction types (e.g., browsing, suggestions) must be provided as well.

Lastly, not all users are human beings, typing in search queries or browsing through catalogs. An organizing system should plan for interaction scenarios where computational agents access the system via APIs (application programming interfaces), which require heavily standardized access procedures and resource descriptions in order to enable interactions.

Socio-Political and Organizational Constraints

An important constraint for interaction design choices is the access policies imposed by the producers of organizing systems, as already described in "Access Policies". If resources or their descriptions are restricted, interactions may not be able to use certain properties and therefore cannot be supported.

Inter-organizational or socio-political constraints are imposed when certain parties in an interaction, or even producers of an organizing system, can exert power over other parties and therefore control the nature of the interaction (or even the nature of the resource descriptions). We can distinguish different types of constraints:

Information and economic power asymmetry

Some organizations are able to impose their requirements for interactions and their resource description formats upon their clients or customers. For example, Google and Apple each have the power to control the extent of interoperability attainable in products, services, or applications that utilize their numerous platforms through mandated APIs and the process by which third-party applications are approved. The asymmetry between these dominant players and the myriad of smaller entities providing peripheral support, services, or components can result in *de facto* standards that may pose significant burden for small businesses and reduce overall competition.

Standards

Industry-wide or community standards can be essential in enabling interoperability between systems, applications, and devices. A standard interface describes the data formats and protocols to which systems should conform.[5] Failure to adhere to standards complicates the merging of resources from different organizing systems. Challenges to standardization include organizational inertia; closed policies, processes, or development groups; intellectual property; credentialing; lack of specifications; competing standards; high implementation costs; lack of conformance metrics; lack of clarity or awareness; and abuse of standards as trade barriers.

Public policy

Beyond businesses and standards-setting organizations, the

5. (von Riegen 2006).

government sector wields substantial influence over the implementation and success of possible interactions in organizing systems. As institutions with large and inalienable constituents, governments and governmental entities have similar influences as large businesses due to their size and substantial impact over society at large. Different forms of government around the world, ranging from centrally planned autocracy to loosely organized nation-states, can have far-reaching consequences in terms of how resource description policies are designed. Laws and regulations regarding data privacy prevent organizing systems from recording certain user data, therefore prohibiting interactions based on this information.[6]

6. A good example for the importance of standards and interoperability rules is E-government. E-government refers to the ability to deliver government services through electronic means. These services can range from government-to-citizen, government-to-business, government-to-employees, government-to-government, and vice-versa (Guijarro 2007), (Scholl 2007). This could range from a government unit providing a portal where citizens can apply for a driver's license or file their taxes, to more complex implementations such as allowing different government agencies to share certain pertinent information with one another, such as providing information on driver's license holders to the police. Because the government interacts with heterogeneous entities and their various systems, e-government planners must consider how to integrate and

interoperate with different systems and data models. Countries belonging to the *Organization for Economic Cooperation and Development*(OECD) have continuously refined their strategies for e-government.

An example of a highly successful implementation of a business-to-government implementation is the use of the Universal Business Language(UBL) by the Government of Denmark. UBL is a "royalty-free library of standard electronic XML business documents such as purchase orders and invoices" [oasis-open.org]. The Government of Denmark localized these standards, and mandated all organizations wanting to do business with the government to use these formats for invoicing. By automating the matching process between an electronic order and an electronic invoice, the government expects total potential savings of about 160 million Euros per year [UBL case study], thus highlighting the need for a standard format by which businesses can send in orders and invoices electronically.

Recognizing that its position as government entails that all types of suppliers, big or small, must have equal opportunity to sell its products and services, the Government of Denmark not only set data format standards, it also gave several options by which information can be exchanged. Paper-based invoices would be sent to scanning agencies that would scan and create electronic versions to be submitted to the

Stop and Think: Standards

It is easy to take standards for granted, but without them our lives would run less smoothly because many products and services would not work very well or even be dangerous to use. If you search for the phrase "ISO Standard" along with almost anything, there is a good chance that you will find something. Try "currency," "food," "sunglasses," "tea," "water," "wine," and then a few of your own.

Even within the same firm or organization, constraints on interaction design may result from contradictory policies for organizing systems or even require the implementation of separate, disjoint systems that cannot be integrated without additional investment. Siloed business functions may be resistant to the merging of resources or resource descriptions in order to gain competitive advantage or command resources over other business functions.

Often characterized by different kinds of value contribution, different policies, processes, and practices, organizational units must clearly define and prioritize different interaction goals, align and coordinate processes, and build collaboration capabilities to achieve a high level of interoperability within the organizing system or between different organizing systems in the organization.

In addition to information exchange, organizational interoperability

government. This highlights the different organizational and consumption issues that the government of Denmark had to consider when designing the system.

also aims to provide services that are widely available, easily identifiable, and accessible across the enterprise.

Nevertheless, inter-organizational constraints are inherently less deterministic than intra-organizational ones, because it is possible that a decision-maker with broad authority can decide that some interaction is important enough to warrant the change of institutional policies, formats, or even category systems. (See "Institutional Categories".)

Regulatory Constraints: Right to be Forgotten

A controversial idea known as the "right to be forgotten" gained the force of law in the European Union in May 2014 after the EU's highest court ruled that people could ask search engines such as Google, which dominates the European market, to remove certain kinds of personal information from their search results.

The ruling had its foundations in the EU's 1995 Data Protection Directive, a data retention policy crafted in a time before the dominance of the Internet and search engines. While many privacy advocates hailed it as a victory, others in the technology and media firms have decried it as censorship. Either way, it has highlighted the need for the European Commission to update and modernize its data policy; a proposal has been before the European Parliament since 2012, and plans for its adoption were underway as of summer 2014. (Source: EC fact sheet on the "right to be forgotten" ruling.)

65. Reorganizing Resources for Interactions

Once the scope and range of interactions is defined according to requirements and constraints, the resources and the technology of the organizing system have to be arranged to enable the implementation of the desired interactions.

Commonly, interactions are determined at the beginning of a development process of the organizing system. It follows that most required resource descriptions (which properties of a resource are documented in an organizing system) need to be clarified at the beginning of the development process as well; that is, resource descriptions are determined based on the desired interactions that an organizing system should support. Most of these processes have been described in detail in *Resource Description and Metadata*, *Describing Relationships and Structures* and *The Forms of Resource Descriptions*.

Resources from different organizing systems are often aggregated to be accessed within one larger organizing system (warehouses, portals, search engines, union catalogs, cross-brand retailers), which requires resources and resource descriptions to be transformed in order to adapt to the new organizing system with its extended interaction requirements.[1] Elsewhere, legacy systems

1. Major library system vendors now market so-called discovery portals to their customers, which allow libraries to integrate their local catalogs with central indexes of journal and other full-text databases. The

often need to be updated to accommodate new standards, technologies, and interactions (e.g, mobile interfaces for digital libraries). That means that the necessary resources and resource descriptions for an interaction need to be identified, and, if necessary, changes have to be made in the description of the resources. Sometimes, resources are merged or transformed in order to perform new interactions.

Identifying and Describing Resources for Interactions

Individual and collection resource descriptions need to be carefully considered in order to record the necessary information for the designed interactions. (See *The Forms of Resource Descriptions*.) The type of interaction determines whether new properties need to be derived or computed with the help of external factors and whether these properties will be represented permanently in the organizing

advantages of discovery portals are the seamless access for patrons to all the library's electronic materials (including externally licensed databases) while maintaining a local and customized look and feel. By providing out-of-the-box solutions, vendors on the other hand bind libraries more closely to their products.

See for an example Exlibris Primo (http://www.exlibrisgroup.com/category/PrimoOverview/) or OCLC WorldCat Local (http://www.oclc.org/worldcatlocal/default.htm).

system (e.g., an extended topical description added due to a user comment) or created on the fly whenever a transaction is executed (e.g., a frequency count).

Determining which resources or resource descriptions will be used in an interaction is simple when all resources are included (e.g., in a simple search interaction over all resources in a data warehouse). Sometimes resources need to be identified according to more selective criteria such as resources exhibiting a certain property (e.g., all restaurants in your neighborhood with four stars on Yelp in an advanced search interaction).

Transforming Resources for Interactions

When an organizing system and its interactions are designed with resources or resource descriptions from legacy systems with outdated formats or from multiple organizing systems or when the new organizing systems has a different purpose and requires different resource properties, resources and their descriptions need to be transformed. The processing and transformation steps required to produce the expected modification can be applied at different layers:

Infrastructure or notation transformation

> When resources are aggregated, the organizing systems must have a common basic infrastructure to communicate with one another and speak the same language. This means that participating systems must have a common set of communication protocols and an agreed upon way of

representing information in digital formats, i.e., a notation ("Notations"), such as the Unicode encoding scheme.[2]

Writing system transformation

During a writing system transformation (*The Forms of Resource Descriptions*), the syntax or vocabulary—also called the data exchange format—of the resource description will be changed

2. While data encoding describes how information is represented, and data exchange formats describe how information is structured, communication protocols refer to how information is exchanged between systems. These protocols dictate how these documents are enclosed within messages, and how these messages are transmitted across the network. Things such as message format, error detection and reporting, security and encryption are described and considered. Nowadays, there are a number of communication protocols that are used over networks, including *File Transfer Protocol*(FTP), Hypertext Transfer Protocol(HTTP) commonly used in the Internet, *Post Office Protocol*(POP) commonly used for e-mail, and other protocols under the *Transmission Control Protocol/Internet Protocol*(TCP/IP) suite. Different product manufacturers normally also have more proprietary protocols that they employ, including Apple Computer Protocols Suite and Cisco Protocols. In addition, different types of networks would also have corresponding protocols, including Mobile Wireless Protocols and such.

to conform to another model, e.g., when library records are mapped from the MARC21 standard to the Dublin Core format in order to be aggregated, or when information in a business information system is transformed into an EDI or XML format so that it can be sent to another firm.[3] Sometimes customized

3. Electronic Data Interchange(EDI), is used to exchange formatted messages between computers or systems. Organizations use this format to conduct business transactions electronically without human intervention, such as in sending and receiving purchase orders or exchange invoice information and such. There are four main standards that have been developed for EDI, including the *UN/EDIFACT* standard recommended by the *United Nations*(UN), *ANSI ASC X12* standard widely used in the US, TRADACOMS standard that is widely used in the UK, and the ODETTE standard used in the European automotive industry. These standards include formats for a wide range of business activities, such as shipping notices, fund transfers, and the like. EDI messages are highly formatted, with the meaning of the information being transmitted being highly dependent on its position in the document. For instance, a line in an EDI document with `BEG*00*NE*MOG009364501**950910*CSW11096^` corresponds to a line in the X12 standard for Purchase Orders (standard 850). "BEG" specifies the start of a Purchase Order Transaction Set. The asterisk (*) symbol delineates between items in the line, with each value corresponding

vocabularies are used to represent certain types of properties. These vocabularies were probably introduced to reduce errors or ambiguity or abbreviate common organizational resource properties. These customized vocabularies need to be explained and agreed upon by organizations combining resources to prevent interoperability problems.

Semantic transformation

Agreeing on a category or classification system (*Categorization: Describing Resource Classes and Types* & *Classification: Assigning Resources to Categories*) is crucial so that organizing systems agree semantically—that is, so that resource properties and descriptions share not only technology but also meaning. For example, because the US Census has often changed its system of race categories, it is difficult to compare data from

to a particular field or information component described in the standard. "NE," for example, corresponds to the Purchase Order Type Code, which in this instance is "New Order." As can be seen in the example, the description of the information being transmitted is not readily available within the document. Instead, parties exchanging the information must agree on these formats beforehand, and need to ensure that the information instance is at the right position within the document so that the receiving party can correctly interpret it.

*EDI samples come from http://miscouncil.org.

American National Standards Association(ANSI) can be found at http://www.ansi.org.

different censuses without some semantic transformation to align the categories.[4]

Resource or resource description transformation

Resources or resource descriptions are often directly transformed, as when they are converted to another file format. In computer-based interactions like search engines, text resources are often pre-processed to remove some of the ambiguity inherent in natural language. These steps, collectively called *text processing*, include decoding, filtering, normalization, stopword elimination, and stemming. (See the sidebar, Text Processing)

Text Processing

Decoding

A digital resource is first a sequence of bits. Decoding transforms those bits into characters according to the encoding scheme used, extracting the text from its stored form. (See "Notations".)

Filtering

If a text is encapsulated by formatting or non-semantic markup, these characters are removed because this information is rarely used as the basis of further interactions.

4. This and more examples for difficult categorizations can be found in: (Bowker and Star 2000).

Tokenization

Segments the stream of characters (in an encoding scheme, a space is also a character) into textual components, usually words. In English, a simple rule-based system can separate words using spaces. However, punctuation makes things more complicated. For example, periods at the end of sentences should be removed, but periods in numbers should not. Other languages introduce other problems for tokenization; in Chinese, a space does not mark the divisions between individual concepts.

Normalization

Normalization removes superficial differences in character sequences, for example, by transforming all capitalized characters into lower-case. More complicated normalization operations include the removal of accents, hyphens, or diacritics and merging different forms of acronyms (e.g., U.N. and UN are both normalized to UN).

Stopword elimination

Stopwords are those words in a language that occur very frequently and are not very semantically expressive. Stopwords are usually articles, pronouns, prepositions, or conjunctions. Since they occur in every text, they can be removed because they cannot distinguish them. Of course, in some cases, removing stopwords might remove

semantically important phrases (e.g., "To be or not to be").

Stemming

These processing steps normalize inflectional and derivational variations in terms, e.g., by removing the "-ed" from verbs in the past tense. This homogenization can be done by following rules (*stemming*) or by using dictionaries (*lemmatization*). Rule-based stemming algorithms are easy to implement, but can result in wrongly normalized word groups, for example when "university" and "universe" are both stemmed to "univers."

Transforming Resources from Multiple or Legacy Organizing Systems

The traditional approach to enabling heterogeneous organizing systems to be accessed together has been to fully integrate them, which has allowed the "unrestricted sharing of data and business processes among any connected applications and data sources" in the organization.[5] This can be a strategic approach to improving the management of resources, resource descriptions, and organizing systems as a whole, especially when organizations have disparate systems and redundant information spread across different groups

5. (Linthicum 1999).

and departments. However, it can also be a costly approach, as integration points may be numerous, with vastly different technologies needed to get one system to integrate with another. Maintenance also becomes an issue, as changes in one system may entail changes in all systems integrating with it.[6]

6. Allowing unrestricted access to data and business processes also becomes a problem when working across organizations. Fully integrating systems between two companies, for instance, may entail the exposure of business intelligence and information that should be kept private. This type of exposure is too much for most businesses, regardless of whether the relationship with the other business is collaborative rather than competitive. There are security issues to be considered, as collaborating organizations would need to access private networks and secure servers. The heterogeneity in supporting organizing systems along with the need to quickly evolve with the rapid changes in an organization's competitive and collaborative environment has pushed organizations to shift from more vertical, isolated structures to a more loosely coupled, ecosystem paradigm This has led to more componentized and modularized systems that need only to exchange information or transform resources when an interaction requires it.

   The emerging paradigm then is to enable independent systems to interoperate, or to have "the ability of two or

Planning the transformation of resources from different organizing systems to be merged in an aggregation is called *data mapping* or alignment. In this process, aspects of the description layers (most often writing system or semantics) are compared and matched between two or more organizing systems. The relationship between each component may be unidirectional or bidirectional.[7] In

more systems or components to exchange information and to use the information that has been exchanged." Because the focus is in the exchange of resources or resource descriptions, independent systems need not necessarily know other systems' underlying logic or implementation, for example, how they store resources. What is important is knowing what kind of resource is expected and in what format (notation, writing system, semantics), and what kind of information is returned for a particular. This is a strategic approach to exchanging resources, as systems can remain highly independent of each other. Changes in one system need not necessarily affect how other systems work as long as the information that is sent and received through an interface stays the same. This allows greater adaptability, as changes to system logic or business processes can be done in self-contained modules without necessarily affecting others. The transformation then happens in an intermediate space where the agreements on resource descriptions are fixed.

7. To illustrate the difference between a unidirectional and

addition, resource properties and values that are semantically

bidirectional map, consider two systems, the *Systematized Nomenclature of Medicine – Clinical Terms*(SNOMED-CT) and the *International Classification of Diseases, Tenth Revision, Clinical Modification*(ICD-10-CM).

SNOMED-CT is a medical language system for clinical terminology maintained by the *International Health Terminology Standards Development Organization*(*IHTSDO*) and a designated electronic exchange standard for clinical health information for US Federal Government systems (http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html).

The ICD-10-CM, on the other hand, is an international diagnostic classification system for general epidemiological, health management, and clinical use maintained by the *World Health Organization*(*WHO*) and used for coding and classifying morbidity data from inpatient/outpatient records, physicians offices, and most National Center for Health Statistics(NCHS) surveys (http://www.who.int/classifications/icd/en/).

Because many different SNOMED-CT concepts can be mapped to a single ICD-9-CM code, a map in this direction cannot be used in reverse without introducing confusion and ambiguity.

equivalent might have different names (the *vocabulary problem* of "The Vocabulary Problem"). The purpose of mapping may vary from allowing simple exchanges of resource descriptions, to enabling access to longitudinal data, to facilitating standardized reporting.[8] The preservation of version histories of resource description elements and relations in both systems is vital for verifying the validity of the data map.

Similar to mapping, a straightforward approach to transformation is the use of *crosswalks*, which are equivalence tables that relate resource description elements, semantics, and writing systems from one organizing system to those of another.[9] Crosswalks not only enable systems with different resource descriptions to interchange information in real-time, but are also used by third-party systems, such as harvesters and search engines to generate union catalogs and perform queries on multiple systems as if they were one consolidated system.[10]

In the digital library space, WorldCat allows users to access many library databases to locate items in their community libraries and, depending on patron privileges, to request items through their local libraries from libraries all over the world. For this powerful tool to accurately locate holdings in each library, two resource description standards are involved. At the book publisher, wholesaler, and retailer end, the international standard Online Information Exchange(ONIX) is used to standardize books and serials metadata

8. (McBride et al. 2006).

9. (NISO 2004).

10. http://journal.code4lib.org/articles/54 (Section 1.), http://www.dlib.org/dlib/june06/chan/06chan.html.

throughout the supply chain.[11] ONIX is implemented in book suppliers' internal and customer-facing information systems to track products and to facilitate the generation of advance information sheets and supplier catalogs.[12] At the library end, the Machine-Readable Cataloging(MARC) formats manage and communicate bibliographic and related information.[13] When a member library acquires a title, information in ONIX format is sent from the supplier to the Online Computer Library Center(OCLC) where it is matched with a corresponding MARC record in the WorldCat database by using an ONIX to MARC *crosswalk*.[14] This enables WorldCat to provide accurate real-time holdings information of its member libraries.

As the number of organizing systems increases, crosswalks and mappings become increasingly impractical if each pair of organizing systems requires a separate crosswalk. A more efficient approach would be the use of one vocabulary or format as a switching mechanism (also called a pivot or *hub language*) for all other vocabularies to map towards.[15] Another possibility, which is often

11. (EDItEUR 2009a).

12. (EDiTEUR 2009b).

13. http://www.loc.gov/marc/.

14. (Godby, Smith, and Childress 2008), Sections 1 and 2.

15. *Toward element-level interoperability in bibliographic metadata* (Godby, Smith, and Childress 2008), Sec. 4.4, "Switching-Across." Consider how the Getty has created a crosswalk called Categories for the Description of Works of Art(CDWA) to switch between eleven metadata

used in asymmetric power relationships between organizing systems, is to force all systems to adhere to the format that is used by the most powerful party.

Modes of Transformation

The conceptual relationships between different descriptions can be mapped out manually when creating simple maps. This, however, becomes more difficult as maps become more complex, due to the number of properties being mapped or when there are more structural or granularity issues to consider.

The use of automatic tools to create these alignments become vital in ensuring their accuracy and robustness. Graphical mapping tools provide users with a graphical user interface to connect description elements from source to target by drawing a line from one to the

standards, including Machine-Readable Cataloging/ Anglo-American Cataloging Rules(MARC) and Dublin Core(DC). In this instance, the "Creation Date" element in CDWA is mapped to "260c Imprint — Date of Publication, Distribution, etc." in MARC/AACR and to "Date.Created" in DC. Although this creates a two-step look-up in real-time, a direct mapping of this element from MARC/AACR to DC is no longer necessary for systems to interoperate.

other.[16] Other tools perform automatic mappings based on predetermined rules and criteria.[17]

16. More commonly, graphical data mapping tools are included in an extract, transform, and load (ETL) database suite that provides additional powerful data transformation capabilities. Whereas data mapping is the first step in capturing the relationships between different systems, data transformation entails code generation that uses the resulting maps to produce an executable transformational program that converts the source data into target format. ETL databases *extract* the information needed from the outside sources, *transform* these into information that can be used by the target system using the necessary data mappings, and then *loads* it into the end system.

17. Languages such as XSLT and *Turing eXtender Language*(TXL) facilitate the ease of data transformation while various commercial data warehousing tools provide varying functionalities such as single/multiple source acquisition, data cleansing, and statistical and analytical capabilities. Based on XML, XSLT is a declarative language designed for transforming XML documents into other documents. For example, XSLT can be used to convert XML data into HTML documents for web display or PDF for print or screen display. XSLT processing entails taking an input document in XML format and one or more XSLT

We often perform manual run-time transformations for decisions that require consulting more than one organizing system in our daily lives. For example, when planning a vacation, we use a variety of systems to negotiate a wide set of *ad hoc* requirements such as our resources and time, our fellow travelers and their availability, and the bookings for hotel and transportation, as well as desirable destinations and their various offerings. We somehow reconcile the different descriptions used in each of the systems and match these against each other so that the relevant information can be combined and compared. Even though the systems use different formats, vocabularies and structures, they are targeted toward human users and are relatively easy to interpret. For automatic run-time transformations, which need to be handled computationally, designers face the challenge of creating more structured processes for merging information from different systems.[18]

The time of the transformation—at design time when organizing system resources are merged, or at run time when a certain interaction is performed— depends on the nature of the collaboration between organizing systems. Design-time transformations depend on highly cooperative environments where specific design requirements (like mapping rules and criteria) can be negotiated ahead of the system implementation. In cases where high-flexibility, *ad hoc* or real-time transformations would not be possible due to a lack of cooperation (such as the ShopStyle.com), run-time transformation processes may provide appropriate alternatives. Some low-level incompatibilities between organizing systems, such as the presence of syntactical, encoding, and

style sheets through a template-processing engine to produce a new document.

18. (Carney et al. 2005).

particular structural and content issues, can also be rectified by implementing run-time transformation techniques, creating more loosely-coupled interoperating systems.

Granularity and Abstraction

Within writing system and semantic transformations, issues of *granularity* and level of abstraction ("Determining the Scope and Focus" and "Category Abstraction and Granularity") pose the most challenges to cross-organizing system interoperability.[19] *Granularity* refers to the level of detail or precision for a specific information resource property. For instance, the postal address of a particular location might be represented as several different data items, including the number, street name, city, state, country and postal code (a high-granularity model). It might also be represented in one single line including all of the information above (a low-granularity model). While it is easy to create the complete address by aggregating the different *information components* from the high-granularity model, it is not as easy to decompose the low-granularity model into more specific information components.

This does not mean, however, that a high-granularity model is always the best choice, especially if the context of use does not require it, as there are corresponding tradeoffs in terms of efficiency and speed in assembling and processing the resource information. (See the sidebar, AccuWeather Request Granularity)

The level of abstraction is the degree to which a resource description is abstracted from the concrete use case in order to fit

19. For an in-depth discussion of interoperability challenges, see Chapter 6 of (Glushko and McGrath 2005).

a wider range of resources. For example, many countries have an address field called *state*, but in some countries, a similar regional division is called *province*. In order to accommodate both concepts, we can abstract from the original concrete concepts and establish a more abstract description of *administrative region*. Granularity and abstraction differences can occur at every resource property layer when resources need to be transformed; therefore, they need to be recognized and analyzed at every layer.

AccuWeather Request Granularity

Requests for AccuWeather data have exploded in the last years, due to automated requests from mobile devices to keep weather apps updated. The company has dealt with this challenge by truncating the GPS coordinates sent by the mobile device when it requests weather data (a transformation to lower granularity). If the request with the truncated coordinates is identical to one recently made, a cached version of the content is served, resulting in 300 million to 500 million fewer requests a day.[20]

Accuracy of Transformations

Automatic mapping tools can only be as accurate as the specifications and criteria that are included in the mapping guidelines. Intellectual checks and tests performed by humans are almost always necessary to validate the *accuracy* of the

20. (AT&T 2011).

transformation. Because description systems vary in expressive power and complexity, challenges to transformations may arise from differences in semantic definitions, rules regarding whether an element is required or requires multiple values, hierarchical or value constraints, and controlled vocabularies.[21] As a result of these complexities, absolute transformations that ensure exact mappings will result in a loss of precision if the source description system is substantially richer than the target system.

In practice, relative crosswalks where all elements in a source description are mapped to at least one target, regardless of semantic equivalence, are often implemented. This lowers the quality and *accuracy* of the mapping and can result in "down translation" or "dumbing down" of the system for resource description. As a result of mapping compromises due to different granularity or abstraction levels, transformations from different organizing systems usually result in less granular or specific resource descriptions. Consequently, whereas some interactions are now enabled (e.g., cross-organizing system search), others that were once possible can no longer be supported. For example, conflating geographical and person subject fields from one system (e.g., geographical subject = Alberta, person subject = Virginia) to a joint subject field (e.g. subject = Alberta, Virginia) to transform to the resource description of another system does not allow for searches that distinguish between these specific categories anymore.

> Stop and Think: Dumbing Down
> Can you think of an example where resource

21. (Chan and Zeng 2006). Section 4.3.

description elements from one system are available for interaction in another due to a transformation, where the target system does not retain all the details of the descriptions in the source?

66. Implementing Interactions

The next sections describe some common interactions in digital organizing systems. One way to distinguish among them is to consider the source of the algorithms that are used in order to perform them. We can mostly distinguish *information retrieval* interactions (e.g., search and browse), *machine learning* interactions (e.g., cluster, classify, extract) or *natural language processing* interactions (e.g., named entity recognition, summarization, sentiment analysis, anaphoric resolution). Another way to distinguish among interactions is to note whether resources are changed during the interaction (e.g., annotate, tag, rate, comment) or unchanged (search, cluster). Yet another way would be to distinguish interactions based on their absolute and relative complexity, i.e., on the progression of actions or steps that are needed to complete the interaction. Here, we will distinguish interactions based on the different resource description layers they act upon.

Activities in Organizing Systems, introduced the concept of *affordance* or behavioral repertoire—the inherent actionable properties that determine what can be done with resources. We will now look at affordances (and constraints) that resource properties pose for interaction design. The interactions that an individual resource can support depend on the nature and extent of its inherent and described properties and internal structure. However, the interactions that can be designed into an organizing system can be extended by utilizing collection properties, derived properties, and any combination thereof. These three types of resource properties can be thought of as creating layers because they build on each other.

The further an organizing system moves up the layers, the more functional capabilities are enabled and more interactions can be designed. The degree of possible interactions is determined by the extent of the properties that are organized, described, and created in an organizing system. This marks a correlation between the extent of organization and the range of possible interactions: *The more extensive the organization and the number of identifiable resource properties, the larger the universe of "affordable" interactions.*

Interactions can be distinguished by four layers:

Interactions based on properties of individual resources

Resource properties have been described extensively in *Resources in Organizing Systems* and *Resource Description and Metadata*. Any information or property that describes the resource itself can be used to design an interaction. If a property is not described in an organizing system or does not pertain to certain resources, an interaction that needs this information cannot be implemented. For example, a retail site like Shopstyle cannot offer to reliably search by color of clothing if this property is not contained in the resource description.

Interactions based on collection properties

Collection-based properties are created when resources are aggregated. (See *Foundations for Organizing Systems*.) An interaction that compares individual resources to a collection average (e.g., average age of publications in a library or average price of goods in a retail store) can only be implemented if the collection average is calculated.

Interactions based on derived or computed properties

Derived or computed properties are not inherent in the

resources or collections but need to be computed with the help of external information or tools. The popularity of a digital resource can be computed based on the frequency of its use, for example. This computed property could then be used to design an access interaction that searches resources based on their popularity. An important use case for derived properties is the analysis of non-textual resources like images or audio files. For these content-based interactions, intrinsic properties of the resources like color distributions are computationally derived and stored as resource properties. A search can then be performed on color distributions (e.g., a search for outdoor nature images could return resources that have a high concentration of blue in the upper half and a high concentration of green on the bottom: a meadow on a sunny day).

Interactions based on combining resources

Combining resources and their individual, collection or derived properties can be used to design interactions based on joint properties that a single organizing system and its resources do not contain. This can lead to interactions that individual organizing systems with their particular purposes and resource descriptions cannot offer.

Whether a desired interaction can be implemented depends on the layers of resource properties that have been incorporated into the organizing system. How an interaction is implemented (especially in digital organizing systems) depends also on the algorithms and technologies available to access the resources or resource descriptions.

In our examples, we write primarily about textual resources or resource descriptions. Information retrieval of physical goods (e.g., finding a favorite cookie brand in the supermarket) or non-textual multimedia digital resources (e.g., finding images of the UC Berkeley

logo) involves similar interactions, but with different algorithms and different resource properties.

Interactions Based on Instance Properties

Interactions in this category depend only on the properties of individual resource instances. Often, using resource properties on this lower layer coincides with basic action combinations in the interaction.

Boolean Retrieval

In a Boolean search, a query is specified by stating the information need and using operators from Boolean logic (AND, OR, NOT) to combine the components. The query is compared to individual resource properties (most often terms), where the result of the comparison is either TRUE or FALSE. The TRUE results are returned as a result of the query, and all other results are ignored. A Boolean search does not compare or rank resources so every returned resource is considered equally relevant. The advantage of the Boolean search is that the results are predictable and easy to explain. However, because the results of the Boolean model are not

ranked by relevance, users have to sift through all the returned resource descriptions in order to find the most useful results.[12]

Tag / Annotate

A tagging or annotation interaction allows a user (either a human or a computational agent) to add information to the resource itself or the resource descriptions. A typical tagging or annotation interaction locates a resource or resource description and lets the

1. Each of the four information retrieval models discussed in the chapter has different combinations of the comparing, ranking, and location activities. Boolean and vector space models compare the description of the information need with the description of the information resource. Vector space and probabilistic models rank the information resource in the order that the resource can satisfy the user's query. Structure-based search locates information using internal or external structure of the information resource.

2. Our discussion of information retrieval models in this chapter does not attempt to address information retrieval at the level of theoretical and technical detail that informs work and research in this field (Manning et al. 2008), (Croft et al. 2009). Instead, our goal is to introduce IR from a more conceptual perspective, highlighting its core topics and problems using the vocabulary and principles of IO as much as possible.

user add their chosen resource property. The resulting changes are stored in the organizing system and can be made available for other interactions (e.g., when additional tags are used to improve the search). An interaction that adds information from users can also enhance the quality of the system and improve its usability.[3]

Interactions Based on Collection Properties

Interactions in this category utilize collection-level properties in order to improve the interaction, for example, to improve the ranking in a search or to enable comparison to collection averages.

Ranked Retrieval with Vector Space or Probabilistic Models

Ranked retrieval sorts the results of a search according to their relevance with respect to the information need expressed in a query. The Vector Space and Probabilistic approaches introduced here use individual resource properties like term occurrence or term frequency in a resource and collection averages of terms and their frequencies to calculate the rank of a resource for a query.[4]

The simplicity of the Boolean model makes it easy to understand and implement, but its binary notion of relevance does not fit our intuition that terms differ in how much they suggest what a document is about. Gerard Salton invented the vector space model

3. A good discussion of the advantages and disadvantages of tagging in the library field can be found in (Furner 2007).

4. (Manning et al. 2008), Ch. 1.

of information retrieval to enable a continuous measure of relevance.[5] In the vector space model, each resource and query in an organizing system is represented as a vector of terms. Resources and queries are compared by comparing the directions of vectors in an n-dimensional space (as many dimensions as terms in the collection), with the assumption is that "closeness in space" means "closeness in meaning."

In contrast to the vector space model, the underlying idea of the probabilistic model is that given a query and a resource or resource description (most often a text), probability theory is used to estimate how likely it is that a resource is relevant to an information need. A probabilistic model returns a list of resources that are ranked by their estimated *probability of relevance* with respect to the information need so that the resource with the highest probability to be relevant is ranked highest. In the vector space model, by comparison, the resource whose term vector is most similar to a query term vector (based on frequency counts) is ranked highest.[6]

Both models utilize an intrinsic resource property called the *term frequency (tf)*. For each term, term frequency (tf) measures how many times the term appears in a resource. It is intuitive that *term frequency* itself has an ability to summarize a resource. If a term such as "automobile" appears frequently in a resource, we can assume that one of the topics discussed in the resource is

5. Salton was generally viewed as the leading researcher in information retrieval for the last part of the 20th century until he died in 1995. The vector model was first described in (Salton, Wong, and Yang 1975).

6. (Manning et al. 2008), p.221.

automobiles and that a query for "automobile" should retrieve this resource. Another problem with the term frequency measure occurs when resource descriptions have different lengths (a very common occurrence in organizing systems). In order to compensate for different resource description lengths that would bias the term frequency count and the calculated relevance towards longer documents, the length of the term vectors are normalized as a percentage of the description length rather than a raw count.

Relying solely on term frequency to determine the relevance of a resource for a query has a drawback: if a term occurs in all resources in a collection it cannot distinguish resources. For example, if every resource discusses automobiles, all resources are potentially relevant for an "automobile" query. Hence, there should be an additional mechanism that penalize a term appearing in too many resources. This is done with *inverse document frequency*, which signals how often a term or property occurs in a collection.

Inverse document frequency (idf) is a collection-level property. The *document frequency* (df) is the number of resources containing a particular term. The inverse document frequency (idf) for a term is defined as $idf_t = \log(N/df_t)$, where N is the total number of documents. The inverse document frequency of a term decreases the more documents contain the term, providing a discriminating factor for the importance of terms in a query. For example, in a collection containing resources about automobiles, an information retrieval interaction can handle a query for "automobile accident" by lowering the importance of "automobile" and increasing the importance of "accident" in the resources that are selected as result set.

As a first step of a search, resource descriptions are compared with the terms in the query. In the vector space model, a metric for calculating similarities between resource description and query vectors combining the term frequency and the inverse document

frequency is used to rank resources according to their relevance with respect to the query.[7]

The probability ranking principle is mathematically and theoretically better motivated than the vector space ranking principle. However, multiple methods have been proposed to estimate the probability of relevance. Well-known probabilistic retrieval methods are Okapi BM25, *language models* (LM) and *divergence from randomness models* (DFR).[8] Although these models vary in their estimations of the probability of relevance for a given resource and differ in their mathematical complexity, intrinsic properties of resources like term frequency and collection-level properties like inverse document frequency and others are used for these calculations.

Synonym Expansion with Latent Semantic Indexing

Latent semantic indexing is a variation of the vector space model where a mathematical technique known as singular value decomposition is used to combine similar term vectors into a smaller number of vectors that describe their "statistical center."[9] This method is based mostly on collection-level properties like co-occurrence of terms in a collection. Based on the terms that occur in all resources in a collection, the method calculates which

7. See (Manning et al. 2008), Ch. 6 for more explanations and references on the vector space model.

8. See (Robertson 2005), (Manning et al. 2008), Ch. 12 for more explanations and references.

9. (Deerwester, Dumais et al. 1990).

terms might be synonyms of each other or otherwise related. Put another way, latent semantic indexing groups terms into topics. Let us say the terms "roses" and "flowers" often occur together in the resources of a particular collection. The latent semantic indexing methodology recognizes statistically that these terms are related, and replaces the representations of the "roses" and "flower" terms with a computed "latent semantic" term that captures the fact that they are related, reducing the dimensionality of resource description (see "Vocabulary Control as Dimensionality Reduction"). Since queries are translated into the same set of components, a query for "roses" will also retrieve resources that mention "flower." This increases the chance of a resource being found relevant to a query even if the query terms do not match the resource description terms exactly; the technique can therefore improve the quality of search.

Latent semantic indexing has been shown to be a practical technique for estimating the substitutability or semantic equivalence of words in larger text segments, which makes it effective in information retrieval, text categorization, and other NLP applications like question answering. In addition, some people view it as a model of the computational processes and representations underlying substantial portions of how knowledge is acquired and used, because latent semantic analysis techniques produces measures of word-word, word-passage, and passage-passage relations that correlate well with human cognitive judgments and phenomena involving association or semantic similarity. These situations include vocabulary tests, rate of vocabulary learning, essay tests, prose recall, and analogical reasoning.[10]

Another approach for increasing the quality of search is to add similar terms or properties to a query from a controlled vocabulary

10. See (Dumais 2003).

or classification system. When a query can be mapped to terms in the controlled vocabulary or classes in the classification, the inherent semantic structure of the vocabulary or classification can suggest additional terms (broader, narrower, synonymous) whose occurrence in resources can signal their relevance for a query.

Structure-Based Retrieval

When the internal structure of a resource is represented in its resource description a search interaction can use the structure to retrieve more specific parts of a resource. This enables parametric or zone searching, where a particular component or resource property can be searched while all other properties are disregarded.[11] For example, a search for "Shakespeare" in the title field in a bibliographic organizing system will only retrieve books with Shakespeare in the title, not as an author. Because all resources use the same structure, this structure is a collection-level property.

A common structure-based retrieval technique is the search in relational databases with Structured Query Language(SQL). With the help of tools to facilitate selection and transformation, particular tables and fields in tables and in many combination or with various constraints can be applied to yield highly precise results.

A format like XML enables structured resource descriptions and is therefore very suitable for search and for structured navigation and retrieval. XPath (see "Structural Relationships within a Resource") describes how individual parts of XML documents can be reached within the internal structure. XML *Query Language*(XQ*uery*), a

11. (Manning et al. 2008), Section 6.1.

structure-based retrieval language for XML, executes queries that can fulfill both topical and structural constraints in XML documents. For example, a query can be expressed for documents containing the word “apple” in text, and where “apple” is also mentioned in a title or subtitle, or in a glossary of terms.

Clustering / Classification

Clustering (“Categories Created by Clustering”) and computational classification (“Key Points in Chapter Eight”) are both interactions that use individual and collection-level resource properties to execute their operation. During clustering (unsupervised learning), all resources are compared and grouped with respect to their similarity to each other. During computational classification (supervised learning), an individual resource or a group of resources is compared to a given classification or *controlled vocabulary* in an organizing system and the resource is assigned to the most similar class or descriptor. Another example for a classification interaction is spam detection. (See “Key Points in Chapter Eight”.) Author identification or characterization algorithms attempt to determine the author of a given work (a classification interaction) or to characterize the type of author that has or should write a work (a clustering interaction).

Interactions Based on Derived Properties

Interactions in this category derive or compute properties or features that are not inherent to the resources themselves or the collection. External data sources, services, and tools are employed to support these interactions. Building interactions with conditionality based on externally derived properties usually

increases the quality of the interactions by increasing the system's context awareness.

Retail Store Activity Tracking

Retail analytics companies use cameras and other sensors to analyze shopper activity in retail stores and generate heatmaps of which areas see the most foot traffic and which items customers interact with most.

(Photo by Flickr user m01229. Creative Commons license. Illustration of heatmap by Ian MacFarland.)

Popularity-Based Retrieval

Google's PageRank (see "Structural Relationships between Resources") is the most well-known popularity measure for websites.[12] The basic idea of PageRank is that a website is as popular as the number of links referencing the website. The actual calculation of a website's PageRank involves more sophisticated mathematics than counting the number of in-links, because the source of links is also important. Links that come from quality websites contribute more to a website's PageRank than other links, and links to qualitatively low websites will hurt a website's PageRank.

An information retrieval model for web pages can now use PageRank to determine the value of a web page with respect to a query. Google and other web search engines use many different ranking features to determine the final rank of a web page for any search, PageRank as a popularity measure is only one of them.

12. (Page et al. 1999).

Other popularity measures can be used to rank resources. For example, frequency of use, buying frequency for retail goods, the number of laundry cycles a particular piece of clothing has gone through, and even whether it is due for a laundry cycle right now.

Citation-Based Retrieval

Citation-based retrieval is a sophisticated and highly effective technique employed within bibliographic information systems. Bibliographic resources are linked to each other by citations, that is, when one publication cites another. When a bibliographic resource is referenced by another resource, those two resources are probably thematically related. The idea of citation-based search is to use a known resource as the information need and retrieve other resources that are related by citation.

Citation-based search can be implemented by directly following citations from the original resource or to find resources that cite the original resource. Another comparison technique is the principle of bibliographic coupling, where the information retrieval system looks for other resources that cite the same resources as the original resource. Citation-based search results can also be ranked, for example, by the number of in-citations a publication has received (the PageRank popularity measure actually derives from this principle).

Translation

During translation, resources are transformed into another language, with varying degrees of success. In contrast to the transformations that are performed in order to merge resources from different organizing systems to prepare them for further interactions, a translation transforms the resource after it has been

retrieved or located. Dictionaries or parallel corpora are external resources that drive a translation.

During a dictionary-based translation, every individual term (sometimes phrases) in a resource description is looked up in a dictionary and replaced with the most likely translation. This is a simple translation, as it cannot take grammatical sentence structures or context into account. Context can have an important impact on the most likely translation: the French word *avocat* should be translated into *lawyer* in most organizing systems, but probably not in a cookbook collection, in which it is the *avocado* fruit.

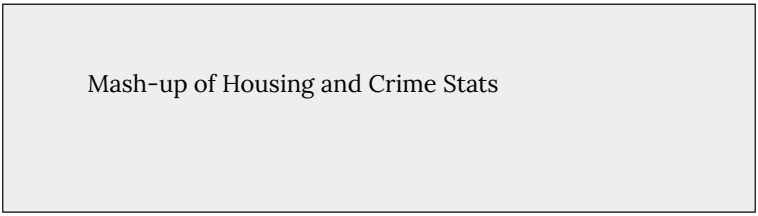
Parallel corpora are a way to overcome many of these challenges. Parallel corpora are the same or similar texts in different languages. The Bible or the protocols of United Nations(UN) meetings are popular examples because they exist in parallel in many different languages. A machine learning algorithm can learn from these corpora to derive which phrases and other grammatical structures can be translated in which contexts. This knowledge can then be applied to further resource translation interactions.

Interactions Based on Combining Resources

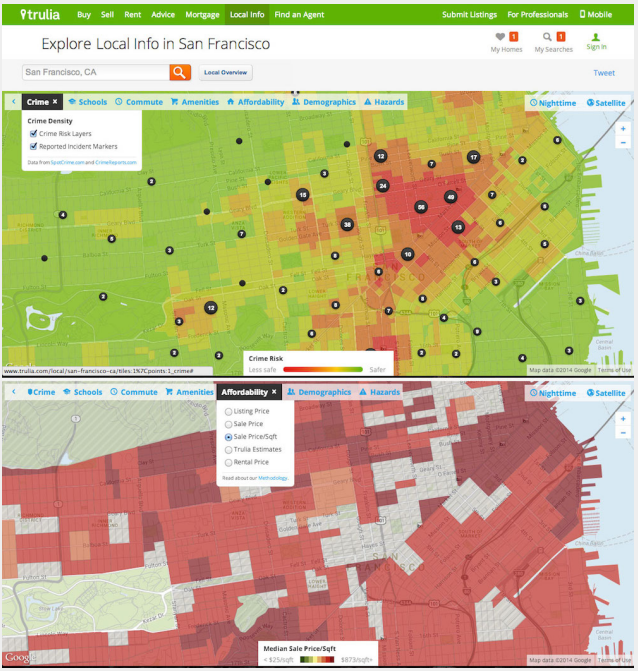
Interactions in this category combine resources mostly from different organizing systems to provide services that a single organizing system could not enable. Sometimes different organizing systems with related resources are created on purpose in order to protect the privacy of personal information or to protect business interests. Releasing organizing systems to the public can have unwanted consequences when clever developers detect the potential of connecting previously unrelated data sources.

Mash-Ups

A mash-up combines data from several resources, which enables an interaction to present new information that arises from the combination.[13] For example, housing advertisements have been combined with crime statistics on maps to graphically identify rentals that are available in relatively safe neighborhoods.



13. (Yee 2008).



The "Local Info" map on real-estate website Trulia mashes up data on crime, schools, housing prices, commute times, and other factors relevant to people searching for a new place to live.

(Screenshots by Ian MacFarland.)

Mash-ups are usually *ad-hoc* combinations at the resource level and therefore do not impact the "mashed-up" organizing systems' internal structures or vocabularies; they can be an efficient instrument for rapid prototyping on the web. On the other hand,

that makes them not very reliable or robust, because a mash-up can fail in its operation as soon as the underlying organizing systems change.

Linked Data Retrieval and Resource Discovery

In "The Semantic Web World", linked data relates resources among different organizing system technologies via standardized and unique identifiers (URIs). This simple approach connects resources from different systems with each other so that a cross-system search is possible.[14] For example, two different online retailers selling a Martha Stewart bedspread can link to a website describing the bedspread on the Martha Stewart website. Both retailers use the same unique identifier for the bedspread, which leads back to the Martha Stewart site.

Resource discovery or linked data retrieval are search interactions that traverse the network (or semantic web graph) via connecting links in order to discover semantically related resources. A search interaction could therefore use the link from one retailer to the Martha Stewart website to discover the other retailer, which might have a cheaper or more convenient offer.

14. (Bizer, Heath, and Berners-Lee 2009).

67. Evaluating Interactions

Managing the quality of an interaction with respect to its intent or goal is a crucial part of every step from design through implementation and especially during operation. Evaluating the quality of interactions at different times in the design process (design concept, prototype, implementation, and operation) reveals both strengths and weaknesses to the designers or operators of the organizing system.

During the design and implementation stages, interactions need to be tested against the original goals of the interaction and the constraints that are imposed by the organizing system, its resources and external conditions. It is very common for processes in interactions to be tweaked or tuned to better comply with the original goals and intentions for the interaction. Evaluation during these stages often attempts to provide a calculable way to measure this compliance and supports the fine-tuning process. It should be an integral part of an iterative design process.

During the later implementation and operation stages, interactions are evaluated with respect to the dynamically changing conditions of the organizing system and its environment. User expectations as well as environmental conditions or constraints can change and need to be checked periodically. A systematic evaluation of interactions ensures that changes that affect an interaction are observed early and can be integrated in order to adjust and even improve the interaction. At these stages, more subjective evaluation aspects like satisfaction, experience, reputation, or "feel" also play a role in fine-tuning the interactions. This subjective part of the evaluation process is as important as the quantitative, objective part. Many factors during the design and implementation processes need to be considered and made to work together. Ongoing quality evaluation and feedback ensures that interactions work as intended.

Evaluation aspects can be distinguished in numerous ways: by the effort and time to perform them (both data collection and analysis); by how quantifiable they are or how comparable they are with measures in other organization systems; by what component of the interaction or organizing system they focus on; or by the discipline, expertise, or methodologies that are used for the evaluation.

A common and important distinction is the difference between efficiency, effectiveness, and satisfaction. An interaction is efficient when it performs its actions in a timely and economical manner, effective when it performs its actions correctly and completely, satisfactory when it performs as expected. Satisfaction is the least quantifiable of the evaluation aspects because it is highly dependent on individual tastes and experiences.

Let us assume that Shopstyle.com develops a new interaction that lets you compare coat lengths from the offerings of their various retailers. Once the interaction is designed, an evaluation takes place in order to determine whether all coats and their lengths are integrated in the interaction and whether the coat lengths are measured and compared correctly. The designers would not only want to know whether the coat lengths are represented correctly but also whether the interaction performs efficiently. When the interaction is ready to be released (usually first in beta or test status), users and retailers will be asked whether the interaction improves their shopping experience, whether the comparison performs as they expected, and what they would change. These evaluation styles work hand in hand in order to improve the interaction.

Efficiency

When evaluating the efficiency of an organizing system, we focus on the time, energy and economic resources needed in order to

achieve the interaction goals of the system. Commonly, the fewer resources are needed for achieving a successful interaction, the more efficient the interaction.

Efficiency measures are usually related to engineering aspects such as the time to perform an action, number of steps to perform an interaction, or amount of computing resources used. Efficiency with respect to the human costs of memory load, attention, and cognitive processing is also important if there is to be a seamless user experience where users can interact with the system in a timely manner.

For a lot of organizing system interactions, however, effectiveness is the more important aspect, particularly for those interactions that we have looked at so far. If search results are not correct, then users will not be satisfied by even the most usable interface. Many interactions are evaluated with respect to their ability to return relevant resources. Why and how this is evaluated is the focus of the remainder of this section.

Effectiveness

Effectiveness evaluates the correct output or results of an interaction. An effective interaction achieves relevant, intended or expected results. The concept of *relevance* and its relationship to effectiveness is pivotal in information retrieval and machine learning interactions. ("Relevance") Effectiveness measures are often developed in the fields that developed the algorithm for the interaction, information retrieval, or machine learning. *Precision* and *recall* are the fundamental measures of *relevance* or effectiveness in information retrieval or machine learning interactions. ("The Recall / Precision Tradeoff")

Relevance

Relevance is widely regarded as the fundamental concept of information retrieval, and by extension, all of information science. Despite being one of the more intuitive concepts in human communication, relevance is notoriously difficult to define and has been the subject of much debate over the past century.

Historically, relevance has been addressed in logic and philosophy since the notion of inference was codified (to infer B from A, A must be relevant to B). Other fields have attempted to deal with relevance as well: sociology, linguistics, and psychology in particular. The subject knowledge view, subject literature view, logical view, system's view, destination's view, pertinence view, pragmatic view and the utility-theoretic interpretation are different perspectives on the question of when something is relevant.[1]

In 1997, Mizzaro surveyed 160 research articles on the topic of relevance and arrived at this definition: "relevance can be seen as a point in a four-dimensional space, the values of each of the four dimensions being: (*i*) Surrogate, document, information; (*ii*) query, request, information need, problem; (*iii*) topic, context, and each combination of them; and (*iv*) the various time instants from the arising of problem until its solution."[2]

This rather abstract definition points to the terminological ambiguity surrounding the concept.

1. Space does not permit significant discussion of these views here, see (Saracevic 1975), and (Schamber et al. 1990).

2. (Mizzaro 1997).

For the purpose of organizing systems, relevance is a concept for evaluating effectiveness that describes whether a stated or implicit information need is satisfied in a particular user context and at a particular time. One of the challenges for the evaluation of relevance in organizing systems is the gap between a user's information need (often not directly stated), and an expression of that information need (a query). This gap might result in ambiguous results in the interaction. For example, suppose somebody speaks the word "Paris" (query) into a smart phone application seeking advice on how to travel to Paris, France. The response includes offers for the Paris Hotel in Las Vegas. Does the result satisfy the information need? What if the searcher receives advice on Paris but has already seen every one of the resources the organizing system offers? What is the correct decision on relevance here?

The key to calculating effectiveness is to be aware of what is being measured. If the information need as expressed in the query is measured, the topical relevance or topicality—a system-side perspective is analyzed. If the information need as in a person's mind is measured, the pertinence, utility, or situational relevance—a subjective, personal perspective is analyzed. This juxtaposition is the point of much research and contention in the field of information retrieval, because topical relevance is objectively measurable, but subjective relevance is the real goal. In order to evaluate relevance in any interaction, an essential prerequisite is deciding which of these notions of relevance to consider.

The Recall / Precision Tradeoff

Precision measures the *accuracy* of a result set, that is, how many of the retrieved resources for a query are relevant. *Recall* measures the completeness of the result set, that is, how many of the relevant resources in a collection were retrieved. Let us assume that a collection contains 20 relevant resources for a query. A retrieval

interaction retrieves 10 resources in a result set, 5 of the retrieved resources are relevant. The precision of this interaction is 50% (5 out of 10 retrieved resources are relevant); the recall is 25% (5 out of 20 relevant resources were retrieved).[3]

It is in the nature of information retrieval interactions that recall and precision trade off with each other. To find all relevant resources in a collection, the interaction has to cast a wide net and will not be very precise. In order to be very precise and return only relevant resources to the searcher, an interaction has to be very discriminating and will probably not find all relevant resources. When a collection is very large and contains many relevant resources for any given query, the priority is usually to increase precision. However, when a collection is small or the information need also requires finding all relevant documents (e.g., in case law, patent searches, or medical diagnosis support), then the priority is put on increasing recall.

The completeness and granularity of the organizing principles in an organizing system have a large impact on the trade-off between *recall* and *precision*. (See *Resources in Organizing Systems*.) When resources are organized in fine-grained category systems and many different resource properties are described, high-precision searches are possible because a desired resource can be searched as precisely as the description or organization of the system allows. However, very specialized description and organization may preclude certain resources from being found; consequently, recall

3. Recall and precision are only the foundation of measures that have been developed in information retrieval to evaluate the effectiveness of search algorithms. See (Baeza-Yates and Ribeiro 2011), (Manning et al. 2008) Ch. 8; (Demartini and Mizzaro 2006).

might be sacrificed. If the organization is superficial—like your sock drawer, for example—you can find all the socks you want (high recall) but you have to sort through a lot of socks to find the right pair (low precision). The trade-off between recall and precision is closely associated with the extent of the organization.

Satisfaction

Satisfaction evaluates the opinion, experience or attitude of a user towards an interaction. Because satisfaction depends on individual user opinions, it is difficult to quantify. Satisfaction measures arise out of the user's experience with the interaction—they are mostly aspects of user interfaces, usability, or subjective and aesthetic impressions.

Usability measures whether the interaction and the user interface designed for it correspond with the user's expectations of how they should function. It particularly focuses on the usefulness of the interaction. Usability analyzes ease-of-use, learnability, and cognitive effort to measure how well users can use an interaction to achieve their task.

Although efficiency, effectiveness, and satisfaction are measured differently and affect different components of the interaction, they are equally important for the success of an interaction. Even if an interaction is fast, it is not very useful if it arrives at incorrect results. Even if an interaction works correctly, user satisfaction is not guaranteed. One of the challenges in designing interactions is that these factors invariably involve tradeoffs. A fast system cannot be as precise as one that prioritizes the use of contextual information. An effective interaction might require a lot of effort from the user, which does not make it very easy to use, so the user satisfaction might decrease. The priorities of the organizing system and its designers will determine which properties to optimize.

Let us continue our Shopstyle coat-length comparison interaction example. When the coat length calculation is performed in an acceptable amount of time and does not consume a lot of the organizing systems resources, the interaction is efficient. When all coat lengths are correctly measured and compared, the interaction is effective. When the interaction is seamlessly integrated into the shopping process, visually supported in the interface, and not cognitively exhausting, is it probably satisfactory for a user, as it provides a useful service (especially for someone with irregular body dimensions). What aspect should Shopstyle prioritize? It will probably weigh the consequences of effectiveness versus efficiency and satisfaction. For a retail- and consumer-oriented organizing system, satisfaction is probably one of the more important aspects, so it is highly likely that efficiency and effectiveness might be sacrificed (in moderation) in favor of satisfaction.

68. Key Points in Chapter Ten

- Where do interactions come from in an organizing system?

  Interactions arise naturally from the affordances of resources or are purposefully designed into organizing systems.

  (See "Introduction")

- What are the most common interactions with resources in organizing systems?

  Accessing and merging resources are fundamental interactions that occur in almost every organizing system.

  (See "Introduction")

- What factors distinguish interactions?

  User requirements, which layer of resource properties is used, and the legal, social and organizational environment can distinguish interactions.

  (See "Determining Interactions")

- What prevents people from making perfectly rational decisions?

  Limited memory and attention capacities prevent people from remembering everything and make them unable to consider more than a few things or choices at once.

  (See "User Requirements")

- Behavioral economics can sometimes produce better

classifications and choices, but what are the possible downsides of its use in design?

The principles of behavioral economics can be used to design organizing systems that manipulate people into taking actions and making choices that they might not intend or that are not in their best interests.

(See the sidebar, Behavioral Economics)

- What activities, with respect to resources, are typically required to enable interactions?

  In order to enable interactions, it is necessary to identify, describe, and sometimes transform the resources in an organizing system.

  (See "Identifying and Describing Resources for Interactions")

- What is a crosswalk?

  Similar to mapping, a straightforward approach to transformation is the use of *crosswalks*, which are equivalence tables that relate resource description elements, semantics, and writing systems from one organizing system to those of another.

  (See "Modes of Transformation")

- How can we distinguish or classify transformations in organizing systems?

  Merging transformations can be distinguished by type (mapping or crosswalk), time (design time or run time) and mode (manual or automatic).

  (See "Granularity and Abstraction")

- What factors distinguish implementations of resource-based interactions?

  Implementations can be distinguished by the source of the algorithm (information retrieval, machine learning, natural language processing), by their complexity (number of actions needed), by whether resources are changed, or by the resource description layers they are based on.

  (See "Implementing Interactions")

- What evaluation criteria distinguish interactions?

  Important aspects for the evaluation of interactions are efficiency (timeliness and cost-effectiveness), effectiveness (accuracy and relevance) and satisfaction (positive attitude of the user).

  (See "Evaluating Interactions")

- What is relevance?

  The concept of *relevance* and its relationship to effectiveness is pivotal in information retrieval and machine learning interactions.

  (See "Relevance")

- What is the recall and precision trade-off?

  The trade-off between recall and precision decides whether a search finds all relevant documents (high recall) or only relevant documents (high precision).

  (See "The Recall / Precision Tradeoff")

- How does granularity of organization affect recall and precision?

  The extent of the organization principles also impacts recall

and precision: more fine-grained organization allows for more precise interactions.

(See "The Recall / Precision Tradeoff")

PART XI

THE ORGANIZING SYSTEM ROADMAP

Robert J. Glushko

69. Introduction (XI)

Foundations for Organizing Systems defined an organizing system as "an intentionally arranged collection of resources and the interactions they support." An organizing system emerges as the result of decisions about **what** is organized, **why** it is organized, **how much** it is organized, **when** it is organized, and **how or by whom** it is organized. These decisions and the tradeoffs they embody are manifested in the four common activities of organizing systems—selecting resources, organizing them, designing and supporting interactions with them, and maintaining them—which we described in Chapter 2. Chapters 4-10 progressively explained each of the parts of the organizing system: resources, resource descriptions, resource categories and collections, and interactions with resources—introducing additional concepts and methods associated with each of these parts.

Along the way we described many types of organizing systems. Sometimes we discussed broad categories of organizing systems, like those for libraries, museums, business information systems, and compositions of web-based services. At other times we described specific instances of organizing systems, like those in the Seed Library, the Flickr photo sharing site, Amazon's drop shipment store, and your home kitchen or closet.

We can now build on the foundation created by Chapters 1-10 to create a "roadmap" that organizes and summarizes the design issues and choices that emerge during an organizing system's lifecycle. These design choices follow patterns that help us understand existing organizing systems better, while also suggesting how to invent new ones by making a different set of design choices.

The roadmap is extensively annotated with references to the preceding chapters where the issues and choices mentioned in the roadmap were introduced and discussed in detail. We will use this

roadmap to analyze a variety of case study examples in *Case Studies*, and to explore the "design neighborhood" around each of them. The design questions from *Foundations for Organizing Systems* serve as a template to give each case study the same structure, which we hope enables instructors, students, and others who read this book to add to this collection of case studies by contributing their own at `DisciplineOfOrganizing.org`.

Navigating This Chapter

We begin with a look at "The Organizing System Lifecycle" which proposes four phases, each of which is discussed in its own section. The first phase, which largely determines the extent and complexity of the resource descriptions needed for organizing and interactions, is "Defining and Scoping the Organizing System Domain". The second phase, which is highly shaped by economic factors and technology constraints or choices, is "Identifying Requirements for an Organizing System". "Designing and Implementing an Organizing System", emphasizes the need for clearly

separating requirements from implementation, a principle we call architectural thinking. The final phase is discussed in "Operating and Maintaining an Organizing System", where we distinguish the maintenance of specific resources and descriptions from the maintenance of the system as a whole.

70. The Organizing System Lifecycle

System lifecycle models exhibit great variety; for our purposes it suffices to use a generic four-phase model that distinguishes a domain definition and scoping phase, a requirements phase, a design and implementation phase, and an operational and maintenance phase. These phases are brief and mostly inseparable for some simple organizing systems, more sequential for others, and more systematic and iterative for complex organizing systems.

Most of the specific decisions that must be made for an organizing system are strongly shaped by the initial decisions about its domain, scope (the breadth or variety of the resources), and scale (the number of resource instances). In organizing systems with limited scope and scale, most of these decisions are made in an informal, unanalyzed, and holistic manner. For example, when we arrange our bookshelves or closets it is not necessary to think explicitly about scoping, requirements, design, implementation, and operational phases. For complex organizing systems, however, especially those in information-intensive domains, it is important to follow a more systematic methodology.

Initial decisions about scope and requirements can create lasting technology and process legacies that impact operational efficiency and flexibility. They can also have profound and unforeseen ramifications for the users of the system and other people affected by the work the system enables. A rigorous, well-documented planning process can help organizers minimize unfair and ineffective outcomes, justify their difficult tradeoffs and decisions, and figure out what went wrong so they can learn from their mistakes.

The consequences of releasing technical systems and tools into the world always include social, business, political, and legal dimensions in addition to technical ones. Some of these implications are due to the context in which the system will operate ("Implementing Interactions"). Others are due to the fact that the work of organizing system designers, architects, and developers is shaped by their experiences, values, beliefs, and circumstances—the often hidden constraints and influences of their social position, education, cultural context, and mental models of the world. Inevitably, the work of information professionals involves "carving up the world at its joints," creating classifications, models, and architectures that support interactions with resources. In practice this often translates to creating artificial "joints" where none truly exist, which will always favor some and injure others. No modeling is ever completely faithful to reality for all people with all experiences (nor is it intended to be), so those people not considered target users for a system, or who have unique circumstances, may end up feeling slighted or ignored and may actually suffer as a result.

71. Defining and Scoping the Organizing System Domain

The most fundamental decision for an organizing system is defining its *domain*, the set or type of resources that are being organized. This is why "What is Being Organized?" ("What Is Being Organized?") was the first of the design decisions we introduced in *Foundations for Organizing Systems*.

We refine how we think about an organizing system domain by breaking it down into five interrelated aspects:

1. the scope and scale of the collection
2. the number and nature of users
3. the time span or lifetime over which the organizing system will operate
4. the physical or technological environment in which the organizing system is situated
5. the relationship of the organizing system to other ones that overlap with it in domain or scope

Addressing these issues is a prerequisite for prioritizing requirements for the organizing system, proposing the principles of its design, and implementing the organizing system.

Scope and Scale of the Collection

The scope of a collection is the dominant factor in the design of

an organizing system, because it largely determines the extent and complexity of the resource descriptions needed by organizing principles and interactions ("Scope, Scale, and Resource Description"). The impact of broad scope arises more from the heterogeneity of the resources in a collection than its absolute scale. It takes more effort to manage a broad and large collection than a narrow and small one; it takes less effort to manage a large collection if it has a narrow scope. A cattle ranch can get by with just one worker for every thousand cows, unlike zoos, which typically have a small number of instances of many types of animals. A zoo needs many more workers because each animal type and sometimes even individual animals can have distinct requirements for their arrangement and care.

Consider a business information system being designed to contain millions of highly structured and similar instances of a small number of related resource types, such as purchase orders and their corresponding invoices.[1] The analysis to determine the appropriate properties and principles for resource description and organization

1. For some kinds of resources with highly regular structure, the distinction between the resource and its description is a bit arbitrary. A transactional document like a payment contains at its core a specification of the amount paid, which we could consider the payment resource. Information about the payer, the payee, the reason for the payment, and other essential information might be viewed as descriptions of the payment resource. In a payment or financial management system, the entire document might be treated as the resource.

is straightforward, and any order or invoice is an equally good instance to study.[2]

Contrast this large but very narrow collection with a small but very broad one that contains a thousand highly variable instances of dozens of different resource types. This heterogeneity makes it difficult to determine if an instance is representative of its resource type, and every resource might need to be analyzed. This variability implies a large and diverse set of resource descriptions where individual resource instances might not be described with much precision because it costs too much to do it manually ("Resource Description by Professionals"). We can extrapolate to understand why organizing systems whose resource collections are both broad and deep, like those of Amazon or eBay, have come to rely on machine learning techniques to identify description properties and

2. The results of this analysis can be represented in a conceptual model or document /database schema that can guide the automated creation of the resource instances and their descriptions ("Abstraction in Resource Description"). Furthermore, these models or schemas can also be used in "model-based" or "model-driven" architectures to generate much of the software that implements the functionality to store the instances and interchange them with other information systems; "imagine if the construction worker could take his blueprint, crank it through a machine, and have the foundation for the building simply appear." Quote comes from (Miller and Mukerji 2003). See also (Kleppe, Warmer, and Bast 2003).

construct resource taxonomies (“Automated and Computational Resource Description”, “Categories Created by Clustering”).[3]

A partial remedy or compromise when the resource instances are highly dissimilar is to define resource types more broadly or abstractly, reducing the overall number of types. We illustrated this approach in “Principles Embodied in the Classification Scheme” when we contrasted how kitchen goods might be categorized broadly in a department store but much more precisely in a wholesale kitchen supply store. The broader categories in the department store blur many of the differences between instances, but in doing so yield a small set of common properties that can be used to describe them. Because these common properties will be at a higher level of abstraction, using them to describe resources will require less expertise and probably less effort (“Scope, Scale, and Resource Description”, “Category Abstraction and Granularity”). However, this comes at a cost: Poets, painters, composers, sculptors, technical writers, and programmers all create resources, but describing all of them with a “creator” property, as the *Dublin Core* requires, loses a great deal of precision.

Challenges caused by the scale of a collection are often related to constraints imposed by the physical or technological environment in which the collection exists that limit how large the collection can be or how it can be organized. (See “Organizing Physical Resources”) Only a few dozen books can fit on a small bookshelf but thousands of books can fit in your two-car garage, which is a typical size because most people and families do not have more than two cars. On the other hand, if you are a Hollywood mogul, superstar athlete, or sultan with a collection of hundreds of cars, a two-car garage

3. See (Chen, Li, Liang, and Zhang 2010), (Pohs 2013).

is orders of magnitudes too small to store your collection.[4] Even collections of digital things can be limited in size by their technological environment, which you might have discovered when you ran out of space for your songs and photos on your portable media player.

Estimating the ultimate size of a collection at the beginning of an organizing system's lifecycle can reduce scaling issues related to storage space for the resources or for their descriptions. Other problems of scale are more fundamental. Larger collections need more people to organize and maintain them, creating communication and coordination problems that grow much faster than the collection, especially when the collection is distributed in different locations.

The best way to prevent problems of scope and scale is through standardization. Standardization of resources can take place if they are created by automated means so that every instance conforms to a schema or model ("Implementing Categories Defined by Properties").[5] Standards for describing bibliographic resources enable libraries to centralize and share much resource description, and using the same standards for resources of diverse types helps address the challenge of broad scope by reducing the need for close monitoring and coordination. Analogous standards for describing

4. See http://autos.ca.msn.com/editors-picks/the-worlds-biggest-car-collectors.

5. Model-driven software generation can be simple—an XFORM specification that creates an input form on a web page. Or it can be complex—a detailed architectural specification in UML sufficient to generate a complete application.

information resources, services, or economic activities business, governmental, or scientific information systems to systematically manage hundreds of millions or even billions of transactional records or pieces of data ("Scope, Scale, and Resource Description").

Number and Nature of Users

An organizing system might have only one user, as when an individual creates and operates an organizing system for a clothes closet, a home bookcase or file cabinet, or for digital files and applications on a personal computer or smart phone. Collections of personal resources are often organized for highly individualized interactions using *ad hoc* categories that are hard to understand for any other user ("Individual Categories"). Personal collections or collections used by only a small number of people typically contain resources that they themselves selected, which makes the most typical interaction with the organizing system searching for a familiar known resource ("User Requirements").

At the other extreme, an organizing system can have national or even global scope and have millions or more users like the Library of Congress classification system, the United Nations Standard Products and Services Code, or the Internet Domain Name System. These organizing systems employ systems of institutional categories ("Institutional Categories") that are designed to support systematically specified and purposeful interactions, often to search for previously unknown resources. In between these extremes are the many kinds of organizing systems created by informal and formal groups, by firms of every size, and by sets of cooperating enterprises like those that carry out supply chains and other information-intensive business processes.

The nature and number of users strongly shapes the contents of an organizing system and the interactions it must be designed to

support. (See "User Requirements") Some generic categories of users that apply in many domains are customers, clients, visitors, operators, and managers. We can adapt the generic interactions supported by most organizing systems ("Determining the Purposes") to satisfy these generic user types. For example, while most organizing systems allow any type of user to browse or search the collection to discover its content, only operators or managers are likely to have access to information about the browsing and searching activities of customers, clients, or visitors.

Once we have identified the organizing system's domain more precisely we can refine these generic user categories, classifying users and interactions with more precision. For example, the customers of university libraries are mostly professors and students, while the customers of online stores are mostly shoppers seeking to find something to purchase. Library customers borrow and return resources, often according to different policies for professors and students, whereas online stores might only allow resources to be returned for refunds or exchanges under limited circumstances.

Just as it is with collection scope, the heterogeneity of the user base is more critical than its absolute size. An airport bookstore typically has a narrowly focused collection and treats its customers as generic travelers browsing imprecisely for something to fill their time in the terminal or on the airplane. In contrast, the local public library will have a much broader collection because it has to meet the needs of a more diverse user base than the airport bookstore, and it will support a range of interactions and services targeted to children learning to read, school students, local businesses, retirees, and other categories of users. A company library will focus its collection on its industry segment, making it narrower in coverage than a local or university research library, but it might provide specialized services for marketing, engineering, research, legal, or other departments of the firm.

Each category of users, and indeed each individual user, brings different experiences, goals, and biases into interactions with the organizing system. As a result, organizing systems in the same domain and with nominally the same scope can differ substantially in the resources they contain and the interactions they support for different categories of users. The library for the Centers for Disease Control and the WebMD website both contain information about diseases and symptoms, but the former is primarily organized to support research in public health and the latter is organized for consumers trying to figure out why they are sick and how to get well. These contrasting purposes and targeted users are manifested in different classification systems and descriptive vocabularies.[6]

The designers of these systems do not necessarily share the same biases as their users, and more importantly, they may not always understand them completely or correctly. This is precisely why good design is iterative: successive cycles of evaluation and revision can shape crude, provisional, and misguided ideas into wildly successful ones. But such nimbleness is not always feasible in highly complex, political, or bureaucratic institutional contexts. Even then, as Bowker and Star conclude, transparency is the best corrective for these sorts of design failures. Designers who recognize that their systems have real consequences for real people should commit to an ongoing process of negotiation that enables those affected by the technology to voice and push back against any detrimental effect it might have on them and their communities. This helps set the stage for effective operation and maintenance of the system ("Properties, Principles and Technology Perspective").

6. Compare www.cdc.gov/philc and www.webmd.com.

Expected Lifetime

The scope and scale of a collection and the size of its user population are often correlated with the expected lifetime of its organizing system. Because small personal organizing systems are often created in response to a specific situation or to accomplish a specific task, they generally have short lifetimes ("Individual Categories").

The expected lifetime of the organizing system is not the same as the expected lifetime of the resources it contains because motivations for maintaining resources differ a great deal. (See "Motivations for Maintaining Resources") As we have just noted, some organizing systems created by individuals are tied to specific short-term tasks, and when the task is completed or changes, the organizing system is no longer needed or must be superseded by a new one. At the other extreme are libraries, museums, archives, and other memory institutions designed to last indefinitely because they exist to preserve valuable and often irreplaceable resources.

However, most business organizing systems contain relatively short-lived resources that arise from and support day-to-day operations, in which case the organizing system has a long expected lifetime with impermanent resources. Finally, just to complete our 2 x 2 matrix, the auction catalog that organizes valuable paintings or other collectibles is a short-lived single-purpose organizing system whose contents are descriptions of resources with long expected lifetimes.

Physical or Technological Environment

An organizing system is often tied to a particular physical or technological environment. A kitchen, closet, card cabinet, airplane

cockpit, handheld computer or smartphone, and any other physical environment in which resources are organized provides affordances to be taken advantage of and constraints that must be accommodated by an organizing system ("Affordance and Capability").[7]

The extent of these physical and technological constraints affects the lifetime of an organizing system because they make it more difficult to adapt to changes in the set of resources being organized or the reasons for their organization. A desk or cabinet with fixed "pigeon holes" or drawers affords less flexible organization than a file cabinet or open shelves. A building with hard-walled offices constrains how people interact and collaborate more than an open floor plan with modular cubicles does. Business processes implemented in a monolithic enterprise software application are tightly coupled; those implemented as a choreography of loosely-

7. Service design, architecture, and user interaction design are the primary disciplines that care about the influence of layout and spatial arrangement on user interaction behavior and satisfaction. One type of physical framework is the "Servicescape" (Bitner 1992), the man-made physical context in which services are delivered. For example, the arrangement of waiting lines in banks, supermarkets, and post offices or the use of centrally-visible "take a number" systems strongly influence the encounters in service systems (Zhou and Soman 2003). Related concepts for describing the use of features and orienting mechanisms in "the built environment" come from the "Wayfinding" (Arthur and Passini 1992) literature in urban planning and architecture.

coupled web services can often transparently substitute one service provider for another.

Relationship to Other Organizing Systems

The same domain or set of resources can have more than one organizing system, and one organizing system can contain multiple others. The organizing system for books in a library arranges books about cooking according to the Library of Congress or Dewey Decimal classifications and bookstores use the BISAC ones, mostly using cuisine as the primary factor ("Bibliographic Classification"). In turn, cookbooks employ an organizing system for their recipes that arranges them by type of dish, main ingredient, or method of preparation. Within a cookbook, recipes might follow an organizing system that standardizes the order of their component parts like the description, ingredients, and preparation steps.

Sometimes these multiple organizing systems can be designed in coordination so they can function as a single hierarchical, or nested, organizing system in which it is possible to emphasize different levels depending on the user's task or application. Most books and many documents have an internal structure with chapters and hierarchical headings that enable readers to understand smaller units of content in the context of larger ones ("Structural Relationships within a Resource"). Similarly, a collection of songs can be treated as an album and organized using that level of abstraction for the item, but each of those songs can also be treated as the unit of organization, especially when they are embodied in separate digital files.

Organizing systems overlap and intersect. People and enterprises routinely interact with many different organizing systems because what they do requires them to use resources in ways that cut across context, device, or application boundaries. Just consider how many

different organizing systems we use as individuals for managing personal information like contacts, appointments, and messages. As company employees we create and organize information in email, document repositories, spreadsheets, and CRM and ERP systems. Now consider this at an institutional scale in the inter-enterprise interactions among the organizing systems of physicians, hospitals, medical labs, insurance companies, government agencies, and other parties involved in healthcare. Consider how many of these are "mash-ups" and composite services that combine information and resources from independently designed systems.

We have come to expect that the boundaries between organizing systems are often arbitrary and that we should be able to merge or combine them when that would create additional value. It is surely impossible to anticipate all of these *ad hoc* or dynamic intersections of organizing systems, but it is surely necessary to recognize their inevitability, especially when the organizing systems contain digital information and are implemented using web architectures.

72. Identifying Requirements for an Organizing System

The two parts of the definition of an organizing system explicitly suggest two categories of requirements, those that specify the intentional arrangement of the resources and those that specify the interactions with the resources. These categories of requirements both depend on resource descriptions, which are implied by but not explicitly called out in the definition of an organizing system.

Because description, arrangement, and interaction are interrelated it is impossible to describe them separately without some redundancy. Nevertheless, in this book we have done that on purpose because taking different perspectives on organizing systems in Chapters 2-10 has enabled us to introduce a broad range of concepts, issues, and methods:

- Every organizing system must enable users to interact with its collection of resources (Chapters 3 and 10);
- The possible interactions depend primarily on the nature and extent of the descriptions associated with the resources (Chapters 4, 5 and 6);
- Intentional arrangement emerges when one or more resource descriptions are used by organizing principles (Chapters 7 and 8);
- Different implementations of the same organizing principle can determine the efficiency or effectiveness of the interactions it enables. (Chapter 10).

If you are creating a personal organizing system or otherwise small-scale one with only a small number of users, you might think there