Chapter 2: Identity and Identification

Chapter Authors:
Daniel Drew Turner, Robert J. Glushko, Kimra McPherson, & Jess Hemerly
(ddt, glushko,kimra,jhemerly)@ischool.berkeley.edu

Table of Contents
Chapter 2: Identity and Identification................................................................. 1

2.1 Identity and the Organizing System ........................................................................ 1
  2.1.1 What is a Thing? ................................................................................................. 1
  2.1.2 Identity, Identifiers, and Names .......................................................................... 4
  2.1.3 Tradeoffs in an Organizing System ..................................................................... 5
  2.1.4 Identity Over Time in an Organizing System ................................................... 7
2.2 Identifying “Bibliographic Entities” ......................................................................... 8
2.3 Identifying Information Components ...................................................................... 11
  2.3.1 Information-Intensive Domains .......................................................................... 11
  2.3.2 Defining “Information Component” ..................................................................... 11
  2.3.3 Content, Structure and Presentation .................................................................. 14
  2.3.4 Identifying Equivalent Components ................................................................... 15
2.4 Identifying “Smart Things” ................................................................................... 16
2.5 Naming, Description, and the Vocabulary Problem ................................................ 19
  2.5.1 What’s In a Name? .............................................................................................. 20
  2.5.2 The Problems of Naming .................................................................................. 20
  2.5.3. Controlled Vocabularies: A Solution? ................................................................. 21
  2.5.4 Unique Names and Identifiers ........................................................................... 23
2.6 Persistence, Effectivity, Authenticity and Provenance ............................................ 25
  2.6.1 Persistence and Effectivity ................................................................................ 25
  2.6.2 Authenticity and Provenance .............................................................................. 27

2.1 Identity and the Organizing System

2.1.1 What is a Thing?

If you're reading this book, you're likely in the business of helping people find things or designing systems of organization meant to ultimately help people find things. But what are these “things,” exactly?
It might seem like the question of identity, of what a single “thing” is, shouldn't be a problem. After all, we live in a world of things, and finding, selecting, organizing, using, and referencing them are everyday activities. We are used to interacting with things in organizing systems we've created ourselves, that were created by other people, or that have been created through institutional or social processes. But it’s really not as simple as it first appears. In order to organize these things, we also need to have a sense of how they will be used. How will we look up, select, assemble, reorganize, put away, or otherwise work with these things? Even though we can’t prepare for every possible use, we need to do our best to understand the potential, primary uses as well as the audience, or users.

Although it sometimes feels like pure, unmitigated fact, the basic unit of any organizing system is not necessarily so clear. Recognizing a person or a physical object is an unconscious, perceptual process that lets us make sense of the different “signals” or sensory representations that a thing can produce. Our human perceptual systems do a remarkable job at picking out “things” from their backgrounds and from each other, so we have little difficulty in recognizing something even if we’re seeing it from a novel viewing angle or with different lighting, shading, and so on. But because we add information to things when we name or describe them, and because we can describe the same thing at different levels of abstraction, recognizing something is only the first step toward being able to organize it and other things like it. Depending on context and intent, the same thing can be considered one of many members of a very broad category, or as the unique instance of a category with only one member. For example, we might recognize something as a piece of clothing, as a sock, or as the specific dirty sock with the hole worn in the heel from yesterday’s long hike. The level of abstraction with which we describe the things we’ve recognized determines what things should be treated as similar. Moreover, it directly shapes the principles and general framework of the organizing systems we can use for them.

As tricky as it can be to decide what a thing is when you’re dealing with single physical objects, it is even more challenging when the things are objects or systems composed of other parts, digital, analog, or both. In these cases, we must focus on the whole—a thing—as well as its parts—many things. Our organizing systems might have to handle descriptions of the whole thing, of the separate parts, and of the relationships between the parts and the whole. Sometimes the separate parts can themselves consist of separate parts, creating a complicated information ecosystem of nested sets or collections of things within the overarching framework of a single thing. It can become quite complicated.

For example, a sports team needs to be considered a single thing for some purposes and as a collection of separate players for others. Specifically, an NFL football team needs to be considered a single thing for things like matches through the season and in playoffs, and 53 individual players for other situations, like the NFL draft or play-calling. The team and the team’s roster can be thought of as whole things, and the team’s individual players are also things that make up the whole team. This holds for information composites too; a newspaper can be considered a single thing, but it might also consist of multiple sections, each of which contains separate stories. Each of these is an information “thing” at a different level of abstraction. The number of parts we describe in one of these composite things, the granularity with which we describe those parts, and the way in which those elements can be combined are decisions that are often made explicitly and arbitrarily and depend on the intended use.

Some of the deepest analytic and cognitive challenges arise when we are dealing with an object made entirely of information rather than a representation of the actual information object,
like a book where the physical matter (paper and ink) can be considered a tangible embodiment of what is inherently an abstract information thing (the knowledge inside). Here the organizing system usually emphasizes the representations of a thing, like a bibliographic record, or metadata, rather than the object consisting of bound pages itself. In addition, the organizing system for these representations of things might be designed and operated according to different principles than the organizing system for the objects themselves. For example, it took hundreds of years for librarians and cataloguers to hash out how abstractly to think about a “work” as an intellectual artifact and the nature of its relationship to the specific and tangible “book things” that exist as physical objects in libraries. A “work” is now considered an abstract intellectual artifact that can take different forms, as opposed to the specific, tangible “book” that exists as a physical object on a library shelf. A surrogate or representation of that work, then, is the descriptive information we use to locate that book. Further issues of scale have enabled us to create and store nearly limitless representations of these works, whether through physical books, e-book files, summaries, bibliographic cards, and metadata.

Information system designers and architects face analogous design challenges with respect to these levels of abstraction when they describe “information components” in a technology-independent manner. For example, separating information content from its structure and presentation is essential to repurposing the information for different scenarios, applications, devices, or users. However, content is intrinsically merged or confounded with structural and presentation information whenever it is used in a specific instance and context. From a logical perspective, an order form contains information components for ITEM, CUSTOMER NAME, ADDRESS, and PAYMENT INFORMATION, but the arrangement of these components, their type font and size, and other non-semantic properties can vary a great deal in different order forms and even across a single information system that repurposes these components for letters, mailing labels, and database entries. Identifying the components requires peeling away this structure and presentation to discern the “pure content” of the information.

Designers and information organizers face difficult and system-defining questions about uniqueness, equivalence, reusability, and, specifically, the tradeoff between organizing information and retrieving it. The organizing system’s scope and its level of granularity and abstraction also shape the way we think about the things it contains. Questions of scale also come into play: is the system designed to include new, similar items or will it need to be expanded every time a new type of content enters the picture?

The answers to the design questions around identity, plus the methods used to determine and implement them, form the intellectual foundations for organizing systems as "a theory of description must specify its variables; it must define the entities to be described” (Svenonius, 2000, p. 50). The answers are also at the heart of the global information economy, as it is increasingly driven by automated information exchange between business processes and services because for information to mean the same thing or “flow efficiently from one type of document to another in this chain of related documents, there must be common content components” (Glushko & McGrath, 2005, p 10).

Similar answers are required by the emergence of ubiquitous or pervasive computing, in which information processing capability and connectivity are embedded into physical objects, in devices like smart phones, and in the surrounding environment. Equipped with sensors, RFID, GPS data, and user-contributed metadata, these diverse “smart things” potentially create a jumbled torrent of information about location and other properties that must be sorted into identified streams and then matched or associated with the original thing. This means that
organizing systems for smart things necessarily contain a mix of both the thing and information about or from things, and these constructs don’t always line up precisely across organizing systems or the interfaces and tools we use to access systems.

Identifying the things that belong in a domain, determining which properties are important or relevant to the people or systems operating in that domain, and then specifying the principles by which those properties encapsulate or define the relationships among the things are required for building any organizing system. In organizing systems used by individuals or with small scope, these methods are often ad hoc and unsystematic, and the organizing systems are therefore idiosyncratic and do not scale well. At the other extreme, organizing systems designed for institutional or industry-wide use, especially in information-intensive domains, require systematic design methods to produce robust conceptual models and are often designed for scalability. These methods go by different names in different disciplines, including “data modeling,” “systems analysis,” and “document engineering” (e.g., Kent, 1978/2000; Silverston, 2001; Glushko & McGrath, 2005). What they have in common is that they produce conceptual models of a domain that specify their components or parts and the relationships among these components or parts. These conceptual models are called “schemas” or “domain ontologies” in some modeling approaches, and are typically implemented in models that are optimized for particular technologies or applications.

2.1.2 Identity, Identifiers, and Names

The problem of “what is a thing” has two parts. The first is determining the identity of a thing; this is a property of the thing. The second is differentiating between a single thing and a larger system that may contain it; this is identification of the thing. These problems are closely related. Once you’ve decided what to treat as a thing, you create a name or an identifier so that you can refer to the thing. An identifier is a special kind of name assigned in a controlled manner and governed by rules that define possible values and naming conventions. Identity is determining what a thing is; identification is being able to refer to that thing reliably.

But choosing these names and identifiers—be it for a person, a service, a place, a trend, a work, a document, a concept, etc.—is hardly straightforward. In fact, naming can often be challenging and contentious, made difficult by, among countless other factors, an audience that will need to access, share, and use the names, the limitations of language, institutional politics, and personal and cultural biases.

Even though floor numbers in hotels or office buildings have the straightforward purpose to identify floors from lowest to highest levels, most buildings skip the 13th floor because many people in Western cultures think 13 is an unlucky number. Furthermore, lucky and unlucky numbers differ across cultures; in China the number 4 is dreaded because it sounds like the word for “death,” while 8 is prized because it sounds like the word for “wealth” (Yardley, 2006). While it is tempting to dismiss biases and beliefs about identifiers as harmless superstitions, their implications are ubiquitous and far from benign. For example, the convention to list the co-authors of scientific publications in alphabetic order has been shown to affect reputation and employment by giving undeserved advantages to people whose names start with letters that come early in the alphabet (Efthyvoulou, 2008).

One complication is that a thing can have more than one name or identifier. A book, for example, will usually have a Library of Congress catalog number as well as an ISBN. When the book is in a carton of books being shipped from the publisher to a bookstore or library, that carton will have a bar-coded tracking number assigned by the delivery service, and a manifest or
receipt document created by the publisher whose identifier associates the shipment with the customer. Each of these identifiers is unique with respect to some established scope or context, yet they can coexist, with each unique identifier working for its corresponding system or purpose.

Another problem is that people use many words to name the same thing—consider the regional difference in use of “coke” vs. “pop” vs. “soda” in the United States (Von Schneidemesser, 1996) —or the same word to name different things—a nail can be a part of your finger or a metal object holding your floorboards down. Furnas et. al. (1987) called this issue, which occurs in all human languages the “vocabulary problem” (Furnas et. al, p. 964). It's likely to arise in the design of any organizing system meant for use by more than one person. To mitigate the vocabulary problem, we can impose rules that control the way in which names and labels for things are assigned in the first place, or that map from our natural language to a set of authoritative or standard terms. This is a known as a controlled vocabulary.

Controlled vocabularies can help with some of the problems of information organization and retrieval by specifying the official terms by which some person or thing is known. But even vocabulary control can’t remove all ambiguity. Even within a controlled system, there can be more than one Robert Smith. And controlling the language used for a particular purpose raises other questions, too: Who writes and enforces these rules? What happens when organizing systems that follow different rules get compared, combined, or otherwise brought together in contexts different from those for which they were originally intended?

When designing an organizing system, the ability to retrieve information as well as the ease with which a system can take on new entities must be considered. But designers bring to the table their own cultural and institutional biases, which can skew a system toward one use or audience rather than many. This works for systems of small scope but become troublesome when an intended audience is wide and various. These considerations highlight just a few of the challenges we’ll explore throughout this chapter, leading us to a discussion of some of the tradeoffs involved in information organization and retrieval.

### 2.1.3 Tradeoffs in an Organizing System

Consider this regular chore: taking your clothes out of the dryer. You may either fold and sort your clothes immediately or, if you plan to fold later, throw your clothes into an undifferentiated "clothes container," Before or after folding, most people sort or separate their dried items into different piles or stacks of clothing—shirts are separated from pants, for example—which implies that a particular “clothes thing,” say a thing that displays the characteristics of a shirt, is of the type “shirt.”

But what should we make of socks? To many people, a sock never is a solitary thing; the basic unit of “socks” is a pair, since most of us have two feet and usually require a sock for each foot. Thus, arguably the best way to find socks in pairs is to organize a sock drawer by pairs. But not everyone approaches the sock drawer with the forethought of retrieving pairs, rendering the next level down from the group of things, “socks,” a single sock, not a pair. Thus, tossing all of the single, unpaired socks in a drawer is the simplest way of organizing socks but, unless you regularly wear unmatched socks—or are in a position where you only need to wear one sock—this is the worst way of organizing socks for quickly finding a matching pair.

This seemingly mundane example captures the tradeoff between information organization and information retrieval. An organizing system for clothes that involves sorting,
folding, and putting into drawers or hanging in a closet requires more of your effort as you put things away but less effort when you get dressed or try to find a particular item. In terms of information, the tradeoff between your investment in organization and investment in retrieval persists in nearly every information system. The more resources you put into one side of it, the fewer resources you will need for the other.

There are other tradeoffs, too. For example, your organizing system for clothing also has some level of abstraction. Let’s say you arrange your clothing by type—shirt—or category—blue things. If you treat all your socks the same by tossing them unsorted into a sock drawer, you have defined an abstract concept of “sockness” that ignores many characteristics you might use in a more granular, or fine-grained, system. If you sort dress socks from athletic ones, and further sort by color or pattern, you are creating a hierarchy of sock categories, or equivalence classes, in which each lower level is defined by characteristics that are not required for class membership by the levels above it in the hierarchy. For example, “Socks” includes all socks, but if the next level down is “Formality,” “athletic socks” and “dress socks” would be different categories on the same level.

Extending our simple example to the broader context of information systems, the issue of what to treat as a unit in an organizing system is a complex intellectual one with important practical consequences. We may talk about Shakespeare’s play “Macbeth,” but what is a "Macbeth"? Is it a particular string of words, saved in a computer file or handwritten in a folio? Is it the collection of words printed with some predetermined font and pagination? Are all the printings of these words the same “Macbeth”? In what ways is a first edition the same as a contemporary paperback, and in what ways do they differ? And how should we organize the numerous live and recorded performances of plays and movies that share the “Macbeth” name? What about works based on or inspired by “Macbeth” but do not share the title “Macbeth”?

These questions challenge us to distinguish between the conceptual and physical aspects of information. In the end we are defining a work: “a set or family of documents in which each document embodies essentially the same intellectual or artistic content” (Svenonius, 2000, p. 36). Composing these work sets from documents that share the same content but aren’t necessarily in presented the same way is an essential act in creating an organizing system: It establishes the identities of things and the boundaries between them.

In business systems, design and organizing challenges 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 single, multi-line text object. In order for the address to be complete, it must contain all the requisite components: street number, street name, city, state, country, zip code, etc. And in order for the envelope to reach its destination, that complete address must be presented on the envelope, usually in the center of the front side. The address is meaningless without all of the parts that make it a complete address.

Inside an information system, however, an address may be stored as separate information components for each printed line, or as a set of its components. Different levels of granularity reflect and impose different requirements for organization and storage. Companies that deal internationally will need to include the country name as well and design the system to appropriately handle international postal codes and other nations’ versions of “state” or
“province.” How a business conceives of an address in terms of how a system will handle it depends on what it needs to do with the information.

Because businesses have different requirements, businesses also use different models. This affects not just addresses but also item numbers, customer identifiers, and documents like catalogs, orders, and invoices that contain these information components. 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, some of which may be difficult to detect because of the vocabulary problem.

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). (For more on this, see Chapter 11).

Thus, more granular organization in a system enables easier reuse of information for different contexts and devices and yields greater precision—the proportion of results that are relevant—in retrieval. However, more granular organization implies more “organizing intelligence” in describing information, and if this work must be done by people rather than by automated processes the costs might outweigh the benefits (Svenonius, 2000, p. 27). Less granular organization yields greater recall, or a larger number of documents, but they are not always terribly relevant. Thus, granularity affects the tradeoff between precision and recall.

2.1.4 Identity Over Time in an Organizing System

Problems of "what is the thing?" and "what do we call it?" do not get resolved simply, nor are they resolved once and left unquestioned. Organizing systems continually need to adapt and evolve in response to changes in content and context. As a result, you have to have a way to note how a thing does or does not change over time (its persistence), whether its state and content come into play at a specified point in time (its effectivity), whether the thing is what you think it is (its authenticity), and sometimes who has certified its authenticity over time (its provenance).

The persistence of a document can either be assumed—e.g., we assume the letter of the law remains in effect over time unless amended or interpreted by a court decision—or require confirmation, like when we ask, "Is this document the latest version?" Many things also have effectivity, meaning that they come into being at a particular time. For some types of things, that can be the moment when they’re created, but for others, effectivity can be a time different from the moment of creation. For example, a particular law can be passed in November, and codified in December, but it may not be enforced until January 1. The creator or operator of an organizing system, whether human or machine, can authenticate a newly created thing. A third party can also serve as proof of authenticity. Notary publics are used on a daily basis to verify that a signature on an important document such as a mortgage or other contract is authentic, much as signet rings and sealing wax once proved that no one has tampered with a document since it was sealed.

Today, questions around identity touch nearly every domain, from education to business to design and more. But the way identity is handled in all these areas has its roots in library science and the way professional cataloguers have been identifying items for hundreds of years.
At a more primitive level, the way humans perceive identity and representation is deeply rooted in our psychology. As we’ll see in section 2.2, the variety of entities in the bibliographic domain, their different levels of abstraction, and the subtle relationships among them took substantial intellectual and pragmatic work to sort out and systematize. In section 2.3 we’ll discuss the issues and methods involved in creating models of information components used as the building blocks of information systems. In section 2.4 we’ll discuss the unique modeling challenges that emerge in the domain of “smart things” such as sensors and sensor networks. Section 2.5 then tackles the questions of naming: Once we’ve got something identified, how do we decide what to call it? And in the final section, 2.6, we’ll address issues about verifying a thing’s identity and determining when it came to be.

2.2 Identifying “Bibliographic Entities”

"No entity without identity." – W. V. Quine (1969, p. 23)

As William Denton noted, pondering the question of identity is something relatively recent in the world of librarians and catalogers. Historically, Denton writes, librarians created "bins" of headings and topics to file items and texts without bothering to give each individual item a separate identifier or name (Denton, 2007, p. 36). This meant searchers first had to make an educated guess as to which bin might house their desired information—“Histories”? “Medical and Chemical Philosophy”?—then scour everything in the category in a quest for their desired item. The choices were ad hoc and always local—that is, each cataloger decided the bins and groupings for each catalog.

In the mid-19th century, Antonio Panizzi was hired to catalog the contents of the British Museum. He formed for perhaps the first time a basic concept of identity in organization. As quoted in The Intellectual Foundations of Information Organization, “a reader may know the work he requires; but he cannot be expected to know all the peculiarities of different editions, and this information he has a right to expect from the catalogues” (Svenonius, 2000, p. 11). Panizzi wondered: How do we differentiate similar objects? For example, if a user wants to find Macbeth for an evening's reading, do we provide her with different listings for paperback versus hardback editions, or for a version in English versus a version in French?

Panizzi’s solution? A catalog radically organized by author name with an index of subjects, along with his newly concocted Rules for the Compilation of the Catalogue. This contained 91 prescriptions including how to file author names and titles and what to do with anonymous works. The Rules were meant to codify how to differentiate (identify) and create a surrogate record for (describe) a singular item in his library.

Previously, each library had made up its own rules. But those were systems—to the extent they were at all systematic—of categorization and classification, concepts we’ll discuss at length in chapters 5 and 6. In contrast, Panizzi's two-step organizational system and associated rules focused on the question of identity of "the thing." Instead of drilling down from top-level categories such as “History,” people searching in his library build up from an item. As Seymour Lubetzky and Elaine Svenonius wrote, "Panizzi saw the book as an edition of a particular work that is intimately related to the other editions and translations of the work that the library may have, and thought that it should therefore be integrated with them" (Lubetzky and Svenonius, 2000, p. 424). This integration meant not just grouping all presentations of a work together, but grouping them under a single identity.
Organizing for the modern age of information

In the 1950s, Lubetzky addressed the question of identity when dealing with the Panizzian problem of distinguishing between editions and works. Lubetzky's own idea of "the work" was an abstracted idea of an author's creation: "The works of an author may be issued under different names as a result of a change, translation, transliteration, or even misprint of the author's name, and the editions of a work may be issued under different titles for similar reasons, and could, therefore, be separated in the catalog" (Lubetzky, 1953, p. 113).

In many ways, this was a radical idea and had practical implications for organizing not just books but information in general. Think about how this might extend to multimedia. According to Lubetzky’s principle, an audio book, a video recording of a play, and an electronic book should be listed each as distinct items, yet still linked to the original. This moves the basic unit of categorization away from "the work" to a more specific thing or representation of the work. And keep in mind the practical consequences of this decision:

The question that must then be faced at the outset [of your organization project]... is whether the objective of the catalog should be merely to tell an inquirer whether or not the library has the particular book he is looking for, or whether it should go beyond that and tell him also what other editions and translations—or other representations—of the work the library has so as to help him more effectively determine whether the library has what he needs and to select what might best serve his purposes (Lubetzky, 1969, p. 271).

Indeed, Lubetzky went on to define a book as "a material object or medium used to convey the intellectual work of an author" (Lubetzky, 1969, p. 11). In contrast to Lubetzky’s use of book and work, S. R. Ranganathan later used the terms work and document to express this split between the idealized and the corporeal (Svenonius, 2000). Svenonius then applied the term document to mean the “smallest or basic” entity; a work was the disembodied set or family of documents for which each document “shares essentially the same intellectual or artistic content” (Svenonius, 2000, p. 36), and an edition was “a particular manifestation of the work” (Svenonius, 2000, p. 38).

Svenonius’s hierarchy evolved into a four-step abstraction hierarchy between the abstract work and the physical item in the Functional Requirements for Bibliographical Records (FRBR), published as a standard by the International Federation of Library Associations and Institutions (IFLA), pictured in figure 1.1.

**Figure 1.1 FRBR Four-Step Hierarchy**
The WORK is a purely abstract construct. It is the ideal, in almost a Platonic sense, of an artistic or intellectual creation. It has no single material representation. If we think of Macbeth as a work, we mean what you think of in terms of plot, lessons, and meaning when you think of Macbeth.

The EXPRESSION denotes the multiple artistic realizations of a work and the actual way it is performed. Examples: A high school performance of Macbeth; Kurosawa’s Throne of Blood; and “specific words, sentences, paragraphs” that make up an early edition of Macbeth.

The MANIFESTATION is the physical artifact embodying the work, including digital artifacts. Examples: the high school performance on VHS; streaming movie file of Throne of Blood; the Folger’s Library print version of Macbeth.

The ITEM or COPY is a single document that exemplifies the manifestation. This level of distinction is important if identical manifestations somehow need to be differentiated. Examples: a family member’s copy of the high school performance of Macbeth; the stream you receive from Netflix when you watch Throne of Blood online; and a used copy of the Folger’s Library version in a bookstore down the street. (IFLA, 1998 http://archive.ifla.org/VII/s13/frbr/frbr1.htm#3.2)

Additionally, Denton noted that there are "other arrangements, such as work, version, adaptation; work, text, edition, printing, book; work, document, text; work, derivations, item; or work, edition, subedition, version, document. They can all more or less be mapped to each other. There are ranges of thought about the expression entity—when something is a new expression of an existing work and when it is sufficiently different to be a new work—and what it means for a work to contain other works" (Denton, 2007, p. 49).
2.3 Identifying Information Components

Section 2.2 described the “entity-relationship” models that span the abstraction hierarchy of the work to organize the variety of entities in the bibliographic domain and capture the subtle relationships among them. The broad scope of these models from the specific “item” to the abstract “work” is essential because organizing systems in libraries must organize tangible artifacts while expressing the conceptual structure of the domains of knowledge represented in their collections. Section 2.3 takes a more general view of conceptual models, especially as applied in “information-intensive” domains.

2.3.1 Information-Intensive Domains

“Information-intensive” domains are those in which “the manipulation of symbols” or “information exchanges” (Apte & Mason, 1995; Glushko, 2010, p. 221) are the primary sources of value creation for the human or computational actors in the domain. Libraries and information services are obviously information-intensive domains. Other examples include accounting, financial services, procurement, logistics, supply chain management, insurance underwriting and claims processing, legal and professional services, customer support, computer programming, and energy management. Education, healthcare, and government services are also information-intensive, but they also involve substantial interpersonal and physical dimensions of value creation, so they are not our focus here.

In information-intensive domains, documents, databases, software applications, or other explicit repositories or sources of information are ubiquitous and essential to the creation of value for the user, reader, consumer, or customer. Value is created through the comparison, compilation, coordination or transformation of information in some chain or choreography of processes operating on information flowing from one information source or process to another. Put another way, the processes that create value in information-intensive domains are “glued together” by shared information components that are exchanged in documents, surrogate records, messages, or object descriptions of some kind. For example, the value created by a personal health record emerges when information from doctors, clinics, hospitals, and insurance companies can be combined because they all share the same “patient” component as a logical piece of information.

The value creation processes in information-intensive domains work best when their component parts come from a common controlled vocabulary for components, or when each uses a vocabulary with a granularity and semantic precision compatible with the others (see Chapter 11: Interoperability). This semantic equivalence eases integration of applications and processes to generate more value from combined information than disparate components.

2.3.2 Defining “Information Component”

Information components are the primitive and abstract “things” in information-intensive domains. They are the units of meaning that serve as building blocks of composite descriptions and other information artifacts. Defining information components as things or units or building
blocks doesn’t help us identify them, so we’ll introduce some heuristic definitions. An information component can be:

- Any piece of information that has a unique label or identifier
- Any piece of information that is self-contained and comprehensible on its own

These two criteria for identifying components are often easy to satisfy through observations, interviews, and task analysis because people naturally use many different types of information and talk easily about specific components and the documents that contain them. Some components (e.g., person, location, date, item) and document types (e.g., report, catalog, calendar, receipt) can be identified in almost any domain. Others need to be more specific to meet the more precise semantic requirements of narrower domains, and might be viewed as refined or qualified versions of the generic components and document types, like course grade and semester components in academic transcripts, airport codes and flight numbers in travel itineraries and tickets, and drug names and dosages in prescriptions.

Identifying candidate components is information-intensive domains is inherently an iterative activity. After all, if value is created through information processing and exchange, to understand the domain we need to begin with a person or information source and follow the information exchanges or flow to other people or documents, which in turn will lead to other ones. Because some insights come from analyzing documents and others come from observing and interacting with people, this iterative analysis has been described as interleaved “document archeology and anthropology.”

Decades of practical and theoretical effort in conceptual modeling, relational theory, and database design have resulted in rigorous methods for identifying information components when requirements and business rules for information can be precisely specified (Date, 2003). For example, in the domain of business transactions, required information like item numbers, quantities, prices, payment information, and so on must be encoded as a particular type of data—integer, decimal, Unicode string, etc.—with clearly defined possible values and that follows clear occurrence rules. In situations like these, a group of techniques collectively called normalization produces a set of tightly defined information components that have minimal redundancy and ambiguity. Normalization also suggests the optimal ways to organize information components in aggregate structures to facilitate information reuse and preserve the integrity of content over time (See sidebar).

**<<SIDEBAR: Normalization>>**

A business might record information about customer orders using a “spreadsheet” style of organization in which a row contains cells that record the date, order number, customer name, customer address, item ID, item description, quantity, unit price, and total price. If an order contains multiple products, these would be recorded on additional rows, as would subsequent orders from the same customer.

<table>
<thead>
<tr>
<th>Date</th>
<th>Order#</th>
<th>CustName</th>
<th>CustAddr</th>
<th>Item ID</th>
<th>Item Description</th>
<th>#</th>
<th>Unit Price</th>
<th>Total Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/28/10</td>
<td>1234</td>
<td>John Smith</td>
<td>11 Elm Dr. LA CA 90210</td>
<td>A-19</td>
<td>BBQ Grill</td>
<td>1</td>
<td>199.99</td>
<td>199.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B-52</td>
<td>BBQ Tool Set</td>
<td>1</td>
<td>149.99</td>
<td>149.99</td>
</tr>
<tr>
<td>9/15/10</td>
<td>2346</td>
<td>John Smith</td>
<td>11 Elm Dr. LA CA 90210</td>
<td>C-10</td>
<td>Fire Extinguisher</td>
<td>2</td>
<td>49.95</td>
<td>99.90</td>
</tr>
</tbody>
</table>
All of this information is important to the business, but this way of organizing it has a great deal of redundancy and inefficiency. For example, the customer address recurs in every order, and the customer address field merges street, city, state and zip code into a large unstructured field rather than separating them as atomic components of different types of information with potentially varying uses. Similar redundancy exists for the products and prices. Cancelling an order might result in the business deleting all the information it has about a particular customer or product.

After the process of normalization, this information model will be factored into four separate tables, one for customers, one for customer orders, one for the items contained in each order, and one for item information. This normalized information model encodes all of the information in the “spreadsheet style” model, but eliminates the redundancy and avoids the data integrity problems that are inherent in it. Each information component within a table is now atomic. The tables can be joined together to retrieve business records because of overlapping identifiers called keys.

### Customer Table

<table>
<thead>
<tr>
<th>Customer ID</th>
<th>Name</th>
<th>Street Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>John Smith</td>
<td>11 Elm Drive</td>
<td>Los Angeles</td>
<td>CA</td>
<td>90210</td>
</tr>
<tr>
<td>C2</td>
<td>Jane Doe</td>
<td>12 Ash Road</td>
<td>Los Angeles</td>
<td>CA</td>
<td>90215</td>
</tr>
</tbody>
</table>

### Customer Order Table

<table>
<thead>
<tr>
<th>Order#</th>
<th>Customer ID</th>
<th>Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>C1</td>
<td>8/28/10</td>
</tr>
<tr>
<td>2346</td>
<td>C1</td>
<td>9/15/10</td>
</tr>
<tr>
<td>4578</td>
<td>C2</td>
<td>10/12/10</td>
</tr>
</tbody>
</table>

### Order Items Table

<table>
<thead>
<tr>
<th>Order#</th>
<th>Item ID</th>
<th>Item Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>A-19</td>
<td>1</td>
</tr>
<tr>
<td>1234</td>
<td>B-52</td>
<td>1</td>
</tr>
<tr>
<td>2346</td>
<td>C-10</td>
<td>2</td>
</tr>
<tr>
<td>4578</td>
<td>C-10</td>
<td>1</td>
</tr>
</tbody>
</table>

### Items Table

<table>
<thead>
<tr>
<th>Item ID</th>
<th>Item Description</th>
<th>Unit Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-19</td>
<td>BBQ Grill</td>
<td>199.99</td>
</tr>
<tr>
<td>B-52</td>
<td>BBQ Tool Set</td>
<td>149.99</td>
</tr>
<tr>
<td>C-10</td>
<td>Fire Extinguisher</td>
<td>49.95</td>
</tr>
</tbody>
</table>

<<end SIDEBAR: Normalization>>
Identifying components can seem superficially easy in transactional documents like orders or invoices, forms requiring data entry, or other highly-structured document types like product catalogs where pieces of information are typically labeled and delimited by boxes, lines, white space or other presentation features that encode the distinctions between types of content. For example, the presence of ITEM, CUSTOMER NAME, ADDRESS, and PAYMENT INFORMATION labels on the fields of an online order form suggests these pieces of information are semantically distinct components in a retail application. They follow the “self-contained and comprehensible” heuristic enough to interconnect the order management, payment, and delivery services that work together to carry out the transaction. In addition, these labels might have analogues in variable names in the source code that implements the order form, or as tags in a XML document created by the ordering application; <CustName>John Smith</CustName> and <Item>A-19</Item> in the order document can be easily identified when it is sent to the other services by the order management application.

But the theoretically grounded methods for identifying components like those of relational theory and normalization that work for structured data do not strictly apply when information requirements are more qualitative and less precise. These information requirements are typical of narrative—unstructured—and semi-structured types of documents and information sources like those often found in law, education, and professional services. Narrative documents include technical publications, reports, policies, procedures and other less structured information, where semantic components are rarely labeled explicitly and are often surrounded by text that is more generic. Unlike transactional documents that depend on precise semantics because they are used by computers, narrative documents are used by people, so there is less need to explicitly define the meaning of the information components. Occasional exceptions when components in narrative documents are identified with labels like NOTE and WARNING only prove the rule.

2.3.3 Content, Structure and Presentation

It is possible to apply rigorous component identification methods in less formal fashion by blending them with the traditional techniques of document analysis from the technical writing and publication design disciplines. This approach unifies conceptual modeling across the entire spectrum of document types and is called “document engineering” (Glushko & McGrath, 2005). A conceptual prerequisite for these synthesized techniques for identifying information components is distinguishing three kinds of information that are found in documents and data sets: content, structure, and presentation. Content is the "what does it mean" information; structure is the "where is it" or "how is it organized and assembled" information; presentation is the "how is it displayed or rendered" information.

When people talk about information, they generally do so at the conceptual content level, focusing on meaning or value, and they aren’t very concerned about the structure or presentation of the information. A patient asks, “Do I have a fever?” while a doctor looks uses a thermometer to measure temperature and respond “yes” or “no.” The way the number is displayed on the thermometer is less important than what the number means in terms of fever or no fever.

However, even though we tend to think of information abstractly as just content, whenever we encounter any of it in a document, on a computer, or on some other device, the content is inherently merged with structure and presentation. To put it in terms we used in earlier sections, we encounter a combination of “the thing” and “information about the thing.” The doctor might receive the information about the patient’s temperature in a printed form with
Chapter 2: Identity and Identification

handwritten notes from a nurse, or formatted on a smart phone by an electronic health record application. It is still the same content, but identifying the component requires recognizing or seeing through the structural and presentation conventions being applied to the content in some specific context.

Identifying content components is easier in well-designed documents that 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 can sometimes be 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. For example, document titles are often centered, big and bold, while the type size in section headings decreases with structural depth, and footnotes are in small type and relegated to page bottoms or the end of documents. These layout and typographic conventions are well known to graphic designers but are also fodder for more academic treatment in studies of visual language or semiotics (Williams, 1994; Crow, 2010).

2.3.4 Identifying Equivalent Components

A key activity in creating a conceptual model or domain ontology is identifying which of the candidate information components are semantically equivalent. This does not, however, mean that they must be identical in every respect. In highly structured or transactional domains, document instances and other information sources tend to be homogeneous because they are produced by or for automated processes, and the candidate information components will have the same names or labels and appear predictably in the same structural and presentational contexts. And while these names may not necessarily be consistent across organization—CustomerName versus Name, for example—similarity in identifiers and structure can always work as a starting point. What this means in practice is that it isn’t necessary to analyze very many invoices, orders, or receipts to “harvest” all the potential components they contain because almost every instance across organizations will yield the same or similar components. For example, every order will contain, in some form, ITEM, CUSTOMER NAME, ADDRESS, and PAYMENT INFORMATION. Within a single organization or in a standardized domain, they will have the same name and meaning; across organizations, they may have different names, but the names will be similar enough to recognize that the components share the same meaning. In domains like these a list of the content components and their definitions – a glossary – is a minimal conceptual model that can often be quite useful as a way to convey the semantics of the domain. Different systems within a firm or different firms might not “carve up” a domain into identical components, but the consistency within each of them usually makes it straightforward to determine what and how things differ.

In contrast, when we are modeling 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, even within the same firm, because they are produced by and for people. The information conveyed in the documents is conceptual or thematic rather than transactional, so instead of precise rules about datatypes and possible values, textual content is more often just words, sentences, and paragraphs. What this all means is that it will be necessary to analyze a larger set of instances to understand the content variation in each type of document, which in turn makes it more challenging to model the domain being described. Furthermore, because of the weaker rules governing the content, the component harvest might
include homonyms—components with the same name but with different meaning—or synonyms—components with the same meaning but with different names or without names at all.

For example, consider the kinds of information components that would be contained in a travel brochure. The brochure would surely contain semantic “nuggets” like names for people and places, dates, prices and other components that also be found in transactional documents. But in the context of a travel brochure, these components are probably surrounded by more narrative text, like creative marketing copy describing a destination and its amenities. There aren’t likely to be rules that specify the exact usage, number and arrangement of this narrative text, and they aren’t likely to be explicitly labeled or tagged in either the presentation or source forms. Different brochures might use PRICES and FARES synonymously, or they might both use PRICES but with very different meanings (e.g., one is per person and doesn’t include any fees and taxes, and another is per room and includes all fees and taxes).

Identifying which components should be treated as equivalents, just like identifying them in the first place, is often an iterative process. The requirements for semantic precision depend on the size and complexity of the domain being modeled. But sometimes the shape and boundaries of the domain aren’t clear until some of the information components in it are identified and defined. This especially true when a domain is of great scale and includes a wide variety of participants with different roles and perspectives. And because of the challenges posed by synonymy and homonymy, a simple glossary is unlikely to be a sufficient conceptual model of the domain. In such cases, it will be necessary to identify additional attributes and relationships for the components to capture the distinctions among the components that matter. This process of going beyond simple identity to more fully describe instances of things and components is the subject of Chapter 3: Describing Instances.

2.4 Identifying “Smart Things”

As mentioned in section 2.1, the addition of information processing capabilities to objects and devices is once again blurring the distinctions between “things” and “information about things.” Objects now generate information about themselves and their identities—where they are, where they’ve been, what they’re doing, and more. And they aren’t simply dispatching this information to people; machines are able to communicate with other machines, building a complex network of intelligent objects that create and share information about themselves and their surroundings.

In a talk at SIGGRAPH, science fiction writer Bruce Sterling (2004) first coined the term “Spime” to describe “objects that can track themselves through space and time” (http://www.boingboing.net/images/blobjects.htm, para. 45). While we are far from realizing Sterling’s vision, we already see early implementations, largely based on radio frequency identification (RFID) tags (Dodson, 2003, para. 1-4). RFID tags, developed by the military, are what allow us to track Federal Express packages, and enable companies to precisely locate objects within “flexible production and distribution systems” (Crandall, 2005, para. 33-34).

Smart things create and share data about their histories and locations with computers and other objects establishing a network of networks, an invisible web of data that operates according to human design but without human intervention. For example, self-regulating appliances can communicate with other self-regulating appliances across a specialized network, sharing information about their processes in order to minimize energy consumption. Companies can also
closely monitor their supply chains and the lifecycles of products. Smart things will be—and, in many cases, already are—equipped with unique identifiers that act as recorders and transmitters for objects’ life spans. And these smart things make up what is being called the Internet of Things.

The Internet of Things is a concept from ubiquitous computing in which objects are technologically equipped with unique identifiers, often radio-frequency identification (RFID) tags. These identifiers work with sensors, cameras, GPS, and other sensing technologies to create a network of objects able to communicate with each other. The data created and attached to locations and objects through social networking sites also contributes to information that makes up the Internet of Things. Computers and other objects transmit, read, and interpret this information, building and making sense of the web of data.

This information gives us clues about more than just the objects themselves. Take, for example, the SenseCam, developed for Gordon Bell’s MyLifeBits project, a project inspired by Vannevar Bush’s 1945 vision of the Memex.1 The SenseCam is a lifelogging camera unlike any other. Worn around the user’s neck, the camera lacks a viewfinder and captures images without the user asking it to. It has a wide angle lens to capture as much of the user’s own field of vision as possible; light- and color-sensitivity sensors; a body heat detector; a thermometer; and an accelerometer. The user may set thresholds that trigger the photo take a camera—a defined increase in body temperature, for example—or can use the timer so that the camera fires at designated intervals. This smart camera, equipped with advanced computational capabilities, pushes the Memex vision of personal information management, where every memory becomes a smart thing. It’s a tool for identifying and remembering one’s own identity, or a lifestream.

There are a few tradeoffs involved in developing smart things. Where, exactly, do you put “the smart?” Low-level sensor data capture can easily result in entirely too much data being transmitted and requiring substantial processing to identify significant events in the torrent. On the other hand, giving every sensor the computing capability to refine its data might make smart devices too expensive to deploy. Pattern analysis can help escape this dilemma, and, as we’ll see later in this section, enable predictive modeling to make optimal use of the data.

Of course, smart data from devices attached to or used frequently by a single person may also make it easy to piece together bits of information that help to clearly identify that person. Think about a world where everyone had a SenseCam and MyLifeBits was the norm. Embedding objects with the ability to track and trace their locations and histories raises serious concerns about privacy. The Internet of Things (section 2.4.1), “a world of information overlays that is no longer virtual but wedded to objects, places, and positions, and no longer fully simulative since it facilitates an active trafficking between model and reality” (Crandall, 2005, para 54), raises serious concerns about ownership and privacy.

2.4.2 Context Awareness

Context awareness, also a concept from ubiquitous computing, specifically focuses on computers’ ability to sense changes in their surrounding environments and react according to these changes. If devices can exploit emerging technologies to infer the current state activity of

---

1 The Memex is a personal organization system that allowed a its user to store and easily retrieve photographs, books, documents, and anything else that was part of their personal information ecosystem. (see Vannevar Bush, “As We May Think,” Atlantic Monthly, 1945)
the user and the characteristics of the environment, they can intelligently manage both the information content and the means of information distribution. The parameters, or thresholds, are designed by programmers, but once the context aware object are deployed into their environment, they act and react independent of programmers’ intervention.

Context aware objects are “spimey,” able to track their location as we move around with them, and are not simply figments of science fiction writers’ imaginations. Take, for example, mobile devices like “smartphones,” which are becoming smarter with the addition of geolocative technologies. With location services turned on, other computers, like Google’s servers, can recognize the phone’s location and provide contextual search results based on location. Geolocative game Foursquare uses this information to provide users nearby businesses from which they can “check-in” as part of the badge-earning game.

The locative information beamed from a mobile device is usually tied to the phone’s owner and dependent on his or her whereabouts. But tracking a person and tracking an object like, say, a head of lettuce are quite different because the information necessary to identify a head of lettuce is much less complex than that needed to identify a person. A single head of lettuce passes through a supply chain only once, while a person can be tracked repeatedly, in a number of different ways. The lettuce can only go where it’s sent, with a much less variable path than people, who are generally able to go where they choose, with special exceptions.

Environmental factors such as humidity, temperature, travel time and other objectively measured phenomena affect the way the quality of the lettuce—how it feels, looks, and tastes—when it completes its passage through the supply chain. While people can be affected by similar phenomena, we’re not affected as predictably as a head of lettuce. For example, some people enjoy hot weather, while others hate it. All heads of lettuce, however, will suffer when exposed to heat for too long.

Objective information about location and environment may be the same, but its impact on an object or organism is different depending on information provided by the object itself. Thus, “identifying” becomes a different problem for every type of smart thing, especially since mobility and activity result in location uncertainty. However, patterns help to make movement more predictable. In designing smart things and devices for people, it is helpful to create a smart model in order to predict the kinds of patterns and locations relevant to the data collected or monitored. These allow designers to develop a set of dimensions and principles that will act as smart guides for the development of smart things. Modeling helps to enable automation, security, or energy efficiency, and baseline models can be used to detect anomalies. As for location, exact locations are unnecessary; use of a “symbolic space” to represent each “sensing zone”—e.g., rooms in a house—and an individual’s movement history as a string of symbols—e.g., abcddeghia—woks sufficiently as a model of prediction.

<<SIDEBAR: Smart and Social Things>>

Many technologists, scientists, and artists conceive of a special world where objects do not simply communicate with each other. Instead, armed with sensors, circuit boards, and locative tracking capabilities, objects can communicate with humans through the same social media that people call Web 2.0, or the social web. Blogjects—objects that blog—and Tweetjects—objects that post messages to Twitter—are neologisms for objects that are socialized by plugging them into the social web. Thanks to the combination of ubiquitous computing technologies and open
application programming interfaces (API) for social media, our plants can tweet when we forget to water them and pigeons can blog and map information about air pollution.

Now imagine carrying a device equipped to specifically send information to Foursquare. DoGood, for example, envisions a foldable coffee cup that automatically checks you in to Foursquare when you enter an establishment (Vasquez, 2010, para. 2). The cup also comes with a solar clock and Bluetooth to work with your mobile device. While it’s only a concept, this is the kind of object designers looking to leverage the Internet of Things are dreaming about. But its communication doesn’t occur only within an invisible web of data. This particular object represents a subset of objects in the Internet of Thing where objects connect to us through the social web: socialized objects. By social web I mean self-broadcasting tools like blogs, Twitter, and Flickr; social location games; social networking sites; and even simple Google Maps interfaces.

University of Southern California professor Julian Bleecker (2006) coined the term “Blogjects” to describe objects that blog (p. 2). These objects blog not by writing editorial commentary about their experiences, but through the work of designers who use APIs and customized programs to harness the information captured by sensors and RFID. This information then appears on blogs in the form of human-readable maps, charts, and text. Bleecker’s early example of a Blogject is Beatriz da Costa’s Pigeon Blog. Da Costa, a Los Angeles–based artist working at the intersection of life sciences, politics, and technology, armed urban pigeons with pollution sensors and locative tracking devices, released them, and created a web interface—in this case Pigeon Blog—to display their flight patterns on Google Maps alongside the pollution levels in the air as they flew. “Whereas once the pigeon was an urban varmint whose value as a participant in the larger social collective was practically nil or worse, the Pigeon that Blogs now attains first-class citizen status” (Bleecker, 2006, p. 5).

Tweetjects share the characteristics of Blogjects. Where Blogjects are objects that join the conversation through information posted to blogs, Tweetjects join the conversation on Twitter. IBM’s Andy Stanford-Clark has been credited with coining the term when he wired his house with sensors, enabling appliances to send information about things happening in the house to the house’s Twitter account, @andy_house (MacManus, 2009, para. 4). On a more product-based level, Botanicalls sells a kit consisting of a soil sensor that sends information about the water level in the soil through an Arduino circuit board, converting thresholds to messages like, “Please water me, I’m thirsty!” Botanicalls also enables plants to notify owners of their needs via text or a phone call. (Botanicalls, n.d.) The kit is entirely open source and hackable, allowing users to customize the thresholds based on the kind of plant and the messages the plant tweets.

<<end SIDEBAR: Smart and Social Things>>

2.5 Naming, Description, and the Vocabulary Problem

Identifying the thing, document, information component, or data we need isn’t always enough. We often need to give that thing a name, some kind of identifier that will help us understand and talk about what it is. But naming often isn’t as simple as just assigning a sequence of characters. In this section, we’ll discuss why we name, some of the problems with naming, and the principles that help us name things in useful ways.
2.5.1 What’s In a Name?

If you know any parents of young children, you may have heard of the stress that often accompanies choosing a name for a baby. Not only are you bringing a child into the world, you also have to pick the collection of letters, syllables, and sounds that will be your child’s persistent identifier. Books list traditional meanings of various names, charts rank names by popularity in different eras, and dozens of websites tout themselves as the place to find a special and unique name. But what is a name, really? It’s just one means we use to indicate who a person is. As long as we can tell who you are, your name could be anything.

Well, maybe not anything. Different states and even countries have rules about characters or words that can be used in names. In Germany, for example, the government regulates the names parents can give to their children; there’s even a book, the International Handbook of Forenames, to guide them (Kulish, 2009). In Portugal, the Ministry of Justice publishes lists of prohibited names (Cornell, 2006). In California in the 1990s, a high school student made waves by changing his name to “Trout Fishing in America” (Associated Press, 1994). Meanwhile, in 2007, Swedish tax officials rejected a family’s attempt to name their daughter Metallica (BBC, 2007).

Regardless, though, the name on a birth certificate is just one of many possible ways we can be identified. We have nicknames, online usernames shaped by character limits, names we use professionally and names we use with friends, or an online alias that we use for privacy or personal branding. At the bank we’re identified by strings of numbers that correspond to our account. The United States federal government knows who we are because of another string of digits known as a Social Security Number. We can also change our names, whether a woman takes on her husband’s surname after marriage or, like the Trout Fishing young man in California, we just find something that better suits us than the name given by our parents.

2.5.2 The Problems of Naming

Naming is challenging, and not just when it comes to people. Giving names to anything, from a business to a concept to an action, can be a difficult process. Often, we get stuck merely choosing what words to use. The sections that follow detail some of the major challenges in determining how to name.

The Vocabulary Problem

If people had a common vocabulary for things in the world, choosing names would be simple and obvious. Unfortunately, such a common vocabulary doesn’t exist. Instead of seamlessly coming up with names that everyone can immediately understand, we often get our wires crossed. And this isn’t merely a conceptual problem. It’s one we deal with every day.

Think back to the last time you tried to order a drink at a coffee shop. Were you at Starbucks? Then perhaps you ordered a “tall.” If you were somewhere else, maybe you ordered a “small” or a “single.” And if you tried to use some other coffee shop’s preferred term at the counter, perhaps the barista didn’t understand your order.

Our inability to naturally find agreed-upon, common terms for the same thing or concept was rigorously explained in a seminal paper by George Furnas and his collaborators, who conducted a number of experiments attempting to figure out whether people would ever independently choose the same name for some thing or function. The short answer: Probably not.
Left to our own devices, we come up with a shockingly large number of terms for a single common thing.

In one of Furnas’s experiments, a thousand pairs of people were asked to “write the name you would give to a program that tells about interesting activities occurring in some major metropolitan area.” Less than 12 pairs agreed on a name. In another, 48 typists were asked to come up with names for text editing operations; the probability that they used the same verb to describe a given operation was less than one in 14.

Furnas called this phenomenon “the vocabulary problem,” concluding that no single word could ever be considered the “best” name. “People use a surprisingly great variety of words to refer to the same thing,” Furnas wrote. “If everyone always agreed on what to call things, the user’s word would be the designer’s word would be the system’s word. . . . Unfortunately, people often disagree on the words they use for things” (Furnas, 1987, p. 964).

There is not a single cause for the vocabulary problem, which is why it’s a persistent issue in information organization. Our individual and cultural experiences help to the context in which we come up with names. For example, the idea of calling the smallest beverage size a “tall” seems counterintuitive for many, but becomes natural to customers who regularly frequent Starbucks. Additionally, languages are riddled with synonyms and homonyms so what one word means to one person may indicate something different for another. At Starbucks, “tall” and “small” are essentially synonyms, but outside of Starbucks, something that is “tall” is rarely, if ever, also considered “small.”

The Semantic Gap

The different ways we can assign something a name further complicates matters. We’ve already discussed all of the different names one particular thing can have, depending on the purpose of the name and who’s doing the naming. When two different processes assign different names, we call the conceptual space between the two the semantic gap (Dorai and Venkatesh, 2001).

This problem often arises when mechanical processes like computer programs or sensors, encode some information in a format optimized for efficient capture, storage, or other technical parameters. Those names — IMG20268.jpg on a digital camera — might be great for your computer but are not useful for human description. We may prefer names that more precisely capture the content of the picture, like “joesunsetbeach.doc” or “goldengatebridge.gif.” And when we’re working with machine-encoded materials, like a clip of music or a snippet of code, a human-language text rendering of the content simply looks like nonsense.

2.5.3. Controlled Vocabularies: A Solution?

Between the vocabulary problem and the semantic gap, different people are likely to come up with words for things that differ from what other people might choose or from what the automatic names assigned by machines. How, then, do we ever find names that allow us to communicate?

One major way information professionals have set about trying to solve the vocabulary problem and bridge the semantic gap for a given domain or task is by establishing a controlled vocabulary. A controlled vocabulary isn’t simply a set of allowed words; it also includes their definitions and the rules by which they should be used and combined. Different domains can create specific controlled vocabularies for their own purposes, but the important thing is that the vocabulary be used consistently throughout that domain.
Svenonius (2000) calls vocabulary control “the *sine qua non* of information organization” (p. 89). Creating a controlled vocabulary reduces synonymy and homonymy, leaving behind a set of words with precisely defined meanings and rules governing their use. “The imposition of vocabulary control creates an artificial language out of a natural language” (p. 89), leaving behind an official, normalized set of terms and their uses.

A controlled vocabulary can be thought of as a dictionary that covers all of the possible terms that can be used in a particular domain. Throughout this book, you’ll see examples of controlledvocabularies, from bibliographic languages that determine the ways books are shelved in a library to business languages that define the set of information components that can be used in transactional documents. Often, though, merely defining a set of words that are acceptable in a vocabulary is only the first step. Information professionals also have to decide what to do with the other terms that don’t make the cut for the controlled vocabulary but that still exist in the world.

If you’re only assigning names to your own things, you can use whatever terms make the most sense to you, even if that means naming your dog “Cat” or vice versa. But if you’re designing a system where other people will have to figure out how to find some piece of information, you must account for the variety of words people might use. And while you may think that certain terms are more obviously “good” than others, Furnas’s study concludes that, “there is no one good access term for most objects. The idea of an ‘obvious,’ ‘self-evident,’ or ‘natural’ term is a myth!” (Furnas, 1987, p. 967).

Furnas suggests “unlimited aliasing” to deal with this: “the designer must make available many, many alternate verbal access routes” (p. 968) to each word or function that a user should be able to find. While he leaves the technical details of achieving this goal up to individual system designers, the general principle is to start with a list of words users might use for a given thing or function and then map them to the controlled term for that thing.

Determining the official name under which an author’s works will be collected or the terms by which a particular subject will be known is essential to good information organization. In library science, the process of creating and maintaining this set of official names and terms is known as **authority control**. Librarians first decide on which term will be the official one used by following some set of standards; for example, when evaluating what name to use as the official one for an author, librarians typically look for the name that’s used most commonly across that author’s body of work. Then they create references—typically “see” and “see also”—to map the variations to the official record. This mapping is “the means by which the language of the user and that of a retrieval system are brought into sync” (Svenonius, 2000, p. 93) and allows an information-seeker to understand the relationship between, say, Samuel Clemens and Mark Twain. Often, official authority files are maintained for different domains: a gazetteer can tell us whether we should be referring to Bombay or Mumbai, while the Library of Congress maintains a list of standard, accepted names for authors, subjects, and titles. In some cases, authority files might be created or maintained by a community, as in the case of Metaweb’s Freebase, an open source collection of data gathered from around the web and mapped or grouped under official titles and categories.

In some cases, it may be impossible to create a controlled vocabulary for a specific information need; in other cases, it may be undesirable. Controlled vocabularies take time and manpower to create and administer, which means using more resources up front. And even if a controlled vocabulary exists, it won’t necessarily square with what your average information-seeker types into a search engine because natural language is an unpredictable thing. But
methods exist for connecting a searcher’s words to the official and artificial version: “synonym rings” can pull together similar words and treat them as the same in a query (e.g., making “car” and “automobile” equivalent); search engines can provide feedback to refine a query or correct a misspelling (“did you mean California?”) and steer searchers closer to what is more likely to be the correct term.

2.5.4 Unique Names and Identifiers

Controlled vocabularies can help with some of the problems of information organization and retrieval by specifying the official terms by which some person or thing is known. But vocabulary control can’t remove all ambiguity. As we mentioned earlier, special names, called identifiers, go one step further in helping us to identify a precise person, place, or thing. The word “identifier” often comes with an adjective attached, typically “unique,” “persistent,” or “global.” In our increasingly digital world, the term “identifier” defines the sequence of characters that tells a computer how to recognize a particular thing. But these sequences of characters cannot be completely random; identifiers are a convention “that requires some person or organization to assert the relationship between the string and the thing” (Coyle, 2006, p. 428) by setting a standard or defining a rule.

Unique Identifiers

Say you have a dinner party with your aunt, your best friend, your mentor, someone named "Mary Josephson," and a professor. How many places at the table should you set? Anywhere from two to six; it's possible all those identifiers refer to a single person, or to five different people. Or say you attempt to e-mail a birthday greeting to your aunt Susan, and because of the way your e-mail program suggested the correct address when you started typing, you accidentally send it to your boss Susan. If things were truly unique — if we only knew one Susan — then identification would be much simpler. But one of the tricky things about names in our everyday world is that they do not generally map one-to-one to objects. This can make it challenging to determine the uniqueness of a given name.

This kind of confusion happens in all sorts of domains. One image on the Internet may end up posted multiple places with multiple names. On the other hand, a single document distributed to multiple people for editing could have numerous differences in each person’s own version, despite sharing a name. And a single person often has dozens of different identifiers — name, initials, driver’s license number, social security number, student ID number, even credit card number — some of which may be unique and some of which may not.

So what does it mean for an identifier to be unique? Once again, it’s all about context. How identifiers are assigned can depend on the scope and granularity of the system. If you have a large group of people, you cannot depend solely on first names, as you'll likely have more than one "Steve" or "Dave" in any given crowd. If you are setting up a routing system for letters, you can't just map that one addressed to “Springfield” will end up in Missouri rather than Massachusetts or Maine.

The good thing, though, is that naming conventions can be developed to fit the scope of a particular system—say, adding a state abbreviation code to letters that travel via the United States Postal Service or using middle initials to distinguish people in a given company. William
Kent noted that one seven-digit telephone number may exist in different area codes, but "the boundaries of the scopes, and the default rules, are well defined" (Kent, 1978/2000, p. 55) by the structure of the phone system. For example, many people have to dial the digit 9 first to get an outside line from their work phone, but not from their home or cell phone.

As always, it is important to understand that any indicating mechanism will be an artificial convention, one you must define when creating an organizing system. As with granularity, or scope, there is no natural, foolproof answer.

**Persistent Identifiers**

How long must an identifier last? There’s one conventional, if unsatisfying, answer: “As long as it’s needed” (Coyle, 2006, p. 429). In some cases, the time frame is relatively short. If you’ve purchased an airline ticket recently, you’ve probably been assigned a confirmation number to locate your flight record. You only need to keep that number around until you’ve completed your travel, at which point your information might be removed from the airline’s system and the confirmation number reused for another passenger. But other time frames are much longer, even indefinite. Think about the Internet. How long will a given URL need to work to access a website on the Internet?

When you evaluate what sorts of identifiers to use in a given system, you have to have a realistic sense of how long you’ll need to be able to use those identifiers. How many identifiers will a proposed format allow? Is it possible that they’ll run out or default to a value that the system wasn’t designed to handle? The latter question was, in one sense, the cause of the Y2K panic: Computer programmers had thought that identifying the calendar year by its last two digits would be sufficient while saving memory space, but when the year 2000 (or "00") loomed, people worried that all these systems would fail.

A given identifier itself may not be persistent: For example, the online music site Pitchfork initially operated at PitchforkMedia.com. Eventually, its founders acquired the Pitchfork.com site from a farming-related company and began posting its content there instead. And the meaning of an identifier is also subject to change. Initially, International Standard Book Numbers (ISBN) applied only to books. Now, with an ever-expanding range of products being sold at bookstores that need a convenient way to track their inventory, everything from teddy bears to breath mints could have an assigned ISBN.

Maintaining the persistence of an identifier can require significant management and regulation. Consider Washington Mutual, the bank that was acquired by Chase in the fall of 2008. Plenty of customers were accustomed to going to wamu.com URLs for their online banking needs; a few months after that acquisition, those URLs no longer worked because Washington Mutual no longer existed. New owner Chase chose to solve the problem by redirecting some URLs to chase.com with a reminder to customers to update their bookmarks. Taking redirects one step further, the Persistent URL project of the Online Computer Library Center attempts to create permanent, stable URLs that can always be accessed at the same location even if the underlying service changes its address.

**Global Identifiers**

Global is the third and final adjective often associated with identifiers in the information community. A “globally unique” identifier comes with some assurance that it will never be in conflict with another identifier in any system in the world (Coyle, 2006). This sounds like a tall
order, but technological methods exist to create globally unique identifiers. The Universally Unique Identifier standard, for example, generates a random 16-byte number that will not be duplicated for hundreds if not thousands of years. Globally unique identifiers are often used in software development, but have many other applications as well, Tacking on the country and area code to a phone number makes it a globally unique identifier.

2.6 Persistence, Effectivity, Authenticity and Provenance

2.6.1 Persistence and Effectivity

Even if you have reached an agreement as to the meaning of “a thing” in your organizing system, you still face the question of the identity of the thing over time, or its persistence. This is not a concern that can be overlooked; one of the major reasons we organize in the first place is to be able to find things over time, which is why Norman Paskin (2010) has called persistence "interoperability with the future” (p. 1586).

Once we identify a thing, we need to make sure it remains functionally the same thing over time and in a different context. This applies to physical items as well as digital ones. Because Leonardo da Vinci used poor materials when painting his original *The Last Supper*, the painting began to fade within his own lifetime. That, plus other damages to the painting’s location, meant the painting was almost destroyed by the late 20th century. Subsequent restorations saved the image, but some critics now wonder if da Vinci can even be considered the painter because so little of his actual paint remains.

Jonathan Furner (2008) used another example:

The paradox of the ship of Theseus might be familiar in this context. Every day that Theseus’s ship is in the harbor, a single plank gets replaced, until after a few years the ship is completely rebuilt: not a single original plank remains. Is it still the ship of Theseus? And suppose, meanwhile, the shipbuilders have been building a new ship out of the replaced planks? Is that the ship of Theseus? (p. 6)

Most serious workflow processes deal with version control, which is the practice of saving past work so you can “step back” or “revert” to an earlier version of the document or code. This is like when you are working on a text document and use the “Save As…” function to create a new file instead of the “Save” function, which overwrites your currently open file. When you design a system, it’s a good practice to be sure that when an item is altered or updated, it does not by default replace the item of the same name.

Even with items that aren’t being intentionally revised for content, the question of “is this the same thing?” can arise. "There is the 'one book' containing the ideas expressed by an author, which is the same book regardless of which language it is translated into, or how it is edited, abridged, condensed, revised, etc.,” writes William Kent (1978/2000, p. 12). But editions may change the content, there are versions of books and documents as well as software. This book you are currently reading was created by many different authors with several editors, meaning it could potentially appear in some cataloging systems under many names yet still be the same thing.

Persistence challenges are compounded by the fact that whenever things persist for a long time, they might change in important ways but still bear much in common with a previous version. In San Francisco, what was once called Army Street is now Cesar Chavez Street.
route negotiating the streets of San Francisco has not changed, but how you describe the route has. Before the 2010 earthquake in Chile, there was a city near the country’s coast called Concepcion. After the earthquake, there’s still a city called Concepcion, but now it’s located 10 feet further west (Mason, 2010).

The complement of persistence is **effectivity**. Effectivity refers to the idea that there is some date when some unit of information is put into service and at which time old versions of that information no longer apply. Put another way, effectivity encodes the knowledge of when something will or did happen, such as knowing that the cost of a USPS First Class stamp will go up on the first day of June. Think of an effectivity date as the opposite of the “Sell By” date on a milk carton. That date indicates when a product goes bad, whereas an item’s effectivity date is when it “goes good” and you’ll need to dispose of (or archive) the other stuff.

We can also think of effectivity in terms of tax or legal code. Many laws are passed with the stipulation that they will not go into effect until a certain date. If this date is January 1, they can be said to have an effectivity date, or effectivity, of January 1. Once we reach January 1, any old versions of the given code should be archived only to retain record of how they looked at a given time in the past. If you are facing an IRS audit, the tax code that you should be poring over isn’t the version that’s current the day you receive notice – it’s the version that was in effect the day you filed the tax return in question. If you are looking up “tax code” in some system, the system has to store effectivity metadata—in this case, the dates the particular tax code was in effect—and you or the system has to know how to read it. There are times when we want the “current version” and times when we want to access or reconstruct a previous version, like reading today’s newspaper versus looking at yesterday’s.

But effectivity in the tax code is simple compared to that relating to documents in complex systems, like commercial aircraft. Because of their long lifetimes—the Boeing 737 has been flying since the 1960s—and continual upgrading of parts like engines and computers, each airplane has its own operating and maintenance manual that reflects the changes made to the plane over time. Every change to the plane requires an update to the repair manual, making the old version obsolete. And while an aircraft mechanic might refer to “the 737 maintenance manual,” each 737 aircraft actually has its own unique manual.

A good organizing system will encompass both persistence and effectivity. The proposed Digital Object Identifier (DOI®) system (www.doi.org) is an example of an attempt to build for persistence. This project posits to assign a "name" (we can also call this an identifier, see Section 2.5.4) to entities for use on digital networks. Paskin, at the time the Director of the International DOI Foundation, wrote, "As an example, URLs do not identify content but a file location: using them as a substitute for such identifiers is not sustainable for reliable automation. The content may be removed (‘404 not found’), or changed (not being the same as the user anticipated, or the user being unaware of such change)” (Paskin, 2010, p. 1586-87). Properly administered, this would offer a persistent way to locate or point to an item, even if its situation changes over time.

However, DOI has its issues too. It’s a highly political, publisher-controlled system, not a universal solution to persistence. When Tim Berners-Lee wrote “Cool URIs Don't Change” in 1998, he advised organizations to do away with organizational identifiers in URIs/URLs such as author name, or section, or topic, or group, or subject when creating an information organization and retrieval system composed of digital documents. He noted that authors could get fired, sections of a document could be reorganized, and so on.

Berners-Lee’s plan also has drawbacks. First, it was suited mostly for digital information objects, which take up far less storage space than larger items like books. The proposed system
relied on being able to search the entire contents of each item, much as you do when searching web pages online. This is simply not feasible with the items on library shelves. Imagine all the books in a library in one big bin. Even if you know which translation of *War and Peace* you need, how do you even begin to find it? And what if you want to browse all works by Tolstoy or perhaps look at all works on a subject? Berners-Lee knew this would be a long-term problem: “Every term in the language is a potential clustering subject, and each person can have a different idea of what it means. Because the relationships between subjects are web-like rather than tree-like, even for people who agree on a web may pick a different tree representation” (1998, para. 51).

### 2.6.2 Authenticity and Provenance

It's critical to any organizing system, especially one concerned with archiving, to retain the notion of "the same document," and critical to this is the concept of **authenticity**. Authenticity means that we can determine that the document is what its identifier purports it to be. This directly relates to the concept of **provenance**, meaning knowledge about the chain of authenticity over time—where it’s been and who’s had it.

It's easy to think of real-world examples were authenticity of an item is of great importance: a signed legal contract, an original work of art, a historical artifact, even a person’s signature. Evidence can be internal, based on the "logical relations that hold between objects and in determining the ontological nature of the object" (Alison, et al., 2005, p. 364) as well as based on external attributes such as examination of the material, or certified chain of ownership, and so on. Many professional careers are based on figuring out if an item is, indeed, the real thing. This can be an issue with everything from people themselves — as we previously discussed, people can change their names (officially or unofficially) in different contexts — to digital information. And while we have ways of positively identifying a person, digital authenticity is more complex; after all, digital items can be reproduced at almost no cost, exist in multiple locations, carry different names on identical documents or identical names on different documents, and bring about other complications that do not arise with physical items.

On the question of authenticating whether a digital item is what you think it is—as opposed to whether it is "the original" or not—Allison et al. (2005) proposed thinking of “salient features” of digital objects, casting them by terms of purpose and usage (p. 370). When creating a system of identifiers, you can consider what attributes of the digital object will remain constant despite differing experiences of the object, and what attributes of the object are necessary for that digital object to be the same object in different representations. In this case, authenticating is a matter of knowing the salient features, extracting them, and comparing to your reference item. If a system of determining authenticity (whether it is based on analyzing bit streams, character streams, or normalized serialization) is in place, we can still only assume the constancy of identity as far back as this system reaches in the chain of maintenance of the document.

**References**


---

2 See Renear and Dubin (2003), who express reservations about all of these


