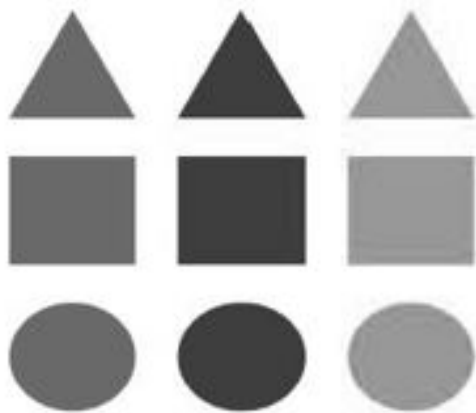


THE DISCIPLINE OF **ORGANIZING**

CORE CONCEPTS EDITION



Edited by **ROBERT J. GLUSHKO**

This version of the 4th Edition (2016) is available under Creative Commons license
CC BY-NC - <https://creativecommons.org/licenses/by-nc/4.0/>

Chapter 10
Interactions with Resources

Vivien Petras
Robert J. Glushko
Ian MacFarland
Karen Joy Nomorosa
J.J.M. Ekaterin
Hyunwoo Park
Robyn Perry
Sean Marimpietri

10.1. Introduction 395
10.2. Determining Interactions 400
10.3. Reorganizing Resources for Interactions 406
10.4. Implementing Interactions 412
10.5. Evaluating Interactions 421
10.6. Key Points in Chapter Ten 425

10.1 Introduction

Picture a dim room in the basement of a Detroit police station, lined with metal shelves: the shelves contain boxes and boxes of cold case files, evidence meticulously logged and categorized for no one to look at, documenting murders that will never be solved. Or the library of a small-town historical society in New Jersey: struggling with budget cuts, the board of directors has been forced to close its doors, locking its treasures inside, carefully curated and preserved but inaccessible to the public. Or a valuable data store encoded in an orphaned storage format: business records in a legacy database system that will not run on modern computers, census data on proprietary magnetic tape reels from the 1970s, your unfinished novel on a series of eight-inch floppy disks. You know the data is there, but you cannot interact with it.

An organizing system without interactions is a sad one indeed.

Interactions are the answer to two of the fundamental questions we posed back in **Chapter 1**: why and when are the resources organized?

The question of “why?” has been in the background (and often the foreground) of every chapter in this book thus far; whenever we select a resource for inclusion in an organizing system, describe it, or arrange it according to an organizing principle, we have an interaction in mind. We include a resource in our system because our users will need it; we assign a resource to one or more categories to help our users find it, understand it, and connect it with other resources in a meaningful way.

In this chapter we will pivot from design for interactions to the design of interactions—and to do this we must pause to consider the question of “when?” In §2.5, we contrasted organization done “on the way in” with that done “on the way out,” but this distinction is not always a particularly relevant one. Consider a bookshelf: if you do not organize its resources on the way in (i.e., when you put a book on the shelf), you cannot really organize them on the way out; you just have a disorganized bookshelf. When the time comes to retrieve a book, you’ll have to employ a brute-force linear search algorithm—reading every spine until you find the one you want, and it will not make the remaining books on the shelf any more organized.

But digital resources and networked organizing systems are an entirely different story. In fact, we argue that they blur the traditional boundary between the academic disciplines of “information organization” and “information retrieval”; with the World Wide Web, ubiquitous digital information, and effectively unlimited processing, storage, and communication capability driven by cloud computing architecture and Moore’s law, billions of people create and browse websites, blog, tag, tweet, and upload and download content of all media types without thinking “I am organizing now” or “I am retrieving now.” When people use their smartphones to search the web or run applications, location information transmitted from their phone is used to filter and reorganize the information they retrieve. Arranging results to make them fit the user’s location is a kind of computational curation, but because it takes place quickly and automatically we hardly notice it. Likewise, almost every application that once seemed predominantly about information retrieval is now increasingly combined with activities and functions that most would consider to be information organization.

Thus we come to the question of when a system’s resources are organized: we may apply the techniques of computational information retrieval to a set of resources that simply are not organized the way we need them to be in order to support our desired interaction. Maybe the system was designed poorly or for a different purpose than the one we are pursuing; maybe we are attempting to collect or aggregate resources from multiple organizing systems, each of which

has its own separate purposes and design flaws. Regardless of the reasons, what we are essentially doing is reorganizing these resources on the fly, or “on the way out,” following many of the same principles and procedures we’ve covered in the preceding eight chapters of this book.

Most Common Museum Interaction



Because museums often contain extremely rare or valuable resources that do not circulate, their most popular items are mobbed by visitors. The crowding often makes it impossible to get a good look at the rare item. This ironic situation is typified by the crowd control cordon that creates a 20-foot barrier around La Gioconda (aka “The Mona Lisa”) at Musée du Louvre in Paris.

(Photo by R. Glushko.)

The fundamental interaction of any organizing system is accessing resources or resource descriptions, whether physically or digitally. Sometimes we must combine or merge resources or resource descriptions to access them effectively; this poses numerous strategy, design, and implementation challenges, as producers often use different identifiers, description or cataloging formats, and practices for similar resources. Different service providers use different technol-

ogies, have different information policies, and follow different processes developed in their separate organizing systems.

Some organizing systems have the power to determine the description standards that others must use. Walmart, the largest retailer in the United States, has devised an organizing system for its supply chain that supports access and movement of physical goods with maximal efficiency and effectiveness. This system saves the corporation money on inventory management and distribution, but to maximize savings, Walmart requires its suppliers to employ the same data model, follow company-set standards, and adopt new technologies such as bar codes and RFID tags that support the highly efficient interactions it requires.

Other organizing systems must adapt to whatever their counterparts develop. Online retailer Shopstyle.com presents a typical ecommerce interface, allowing shoppers to browse a multitude of fashion and beauty products organized into familiar categories. But behind the scenes, Shopstyle is aggregating the catalogs of more than 250 online stores and providing a seamless access interaction for all their merchandise. It does not actually sell anything: it directs shoppers to those third-party stores to make their purchases. Rather than moving physical resources like Walmart, Shopstyle’s most important interactions involve moving and combining digital resource descriptions.

Browsing Merchandise Catalogs

Shopstyle.com provides a transparent interface to the catalogs of hundreds of other online clothing retailers, aggregating their listings to allow users to browse them all from a single page.

(Screenshot by Ian MacFarland.)

Still others choose to abide by what a standard-setting body decides, or participate in laborious, democratic processes to align their organizing practices and interactions. Libraries and museums are the classic examples of this. The most important interaction in a library, of course, is borrowing: checking out a book to use it off the premises, and checking it back in when you're done. Patrons search descriptions in a catalog to find books on a certain topic, by a certain author, or with a certain title, and access them by fetching them from the stacks or asking a librarian to retrieve them. As institutions that serve the public interest, libraries adhere to standards and democratic processes to ensure consistent and familiar user experiences for patrons, but also to enable powerful search interactions such as union catalogs, where resource descriptions from multiple libraries are merged before they are offered for search. Union catalogs allow patrons to find out with a single search whether a resource is available from any library that is accessible to them.

Museums serve the public interest as well, and employ standards and democratic procedures for similar reasons as libraries, but their visitors generally look at their resources rather than borrowing them. Museums enable people to discover or experience resources by exhibiting artifacts in creative contexts, and when they implement this interaction digitally, as in a website, they vastly increase the opportunity for public access. Virtual collections are accessible to remote patrons who are unable to visit the physical museum, and they allow access to resources that are not currently on view.

The digitization of museum resources also allows visitors to experience them from a perspective that might not be possible in a physical museum. For example, in Google's Art Project, users can zoom in to view fine details of digitized paintings. Museums are starting to leverage technology and the popularity of Web 2.0 features such as tagging and social networking to attract new audiences.

Implemented in 2004, the MuseumFinland project aims to provide a portal for publishing heterogeneous museum collections on the Semantic Web. Institutions such as the Getty Information Institute and the International Committee for Documentation of the International Council of Museums have worked on standards that ensure worldwide consistency in how museums manage information about their collections.

How can these differences be handled in order to provide seamless interactions within and across organizing systems? Which requirements have to be met in order to provide the interactions that are desired? How are different interaction types implemented? Finally, how can the quality of interactions be evaluated with respect to their requirements? These are the main questions for interactions that we will try to answer.

Navigating This Chapter

This chapter concentrates on the processes that develop interactions based on leveraging the resources of organizing systems to provide valuable services to their users (human or computational agents). It will discuss the determination of the appropriate interactions (§10.2), the organization of resources for interactions (§10.3), the implementation of interactions (§10.4), and their evaluation and adaptation (§10.5). Although the fundamental questions pertain to all types of organizing systems, this chapter focuses on systems that use computers to satisfy their goals.

10.2 Determining Interactions

Creating a strategy for successfully implementing interactions involves an intricate balance between the resources, the organizing system that arranges and manages them, its producers, and its intended users or consumers. The design of interactions is driven by user requirements and their impact on the choices made in the implementation process. It is constrained by resource and technical system properties and by social and legal requirements. Determining the scope and scale of interactions requires a careful analysis of these individual factors, their combination, and the consequences thereof.

Stop and Think: Constraint vs Flexibility

Think of an information organization project you were involved in. Can you recall ways in which you were constrained in representing an idea by the organizing system the project was implemented with? In what ways was the project negatively affected by the implementation? In what ways might the constraint have had a positive effect?

It is useful to distinguish decisions that involve choices, where multiple feasible alternatives exist, from decisions that involve constraints, where design choices have been eliminated or rendered infeasible by previous ones. The goal when creating an organizing system is to make design decisions that preserve subsequent choices or that create constraints that impose design decisions that would have been preferred anyway.

10.2.1 User Requirements

Users (human or computational agents) search or navigate resources in organizing systems not just to identify them, but also to obtain and further use the selected resources (e.g., read, cluster, annotate, buy, copy, distribute, adapt, etc.). How resources are used and by whom affects how much of the resource or its

description is exposed, across which channels it is offered, and the *precision* and *accuracy* of the interaction.

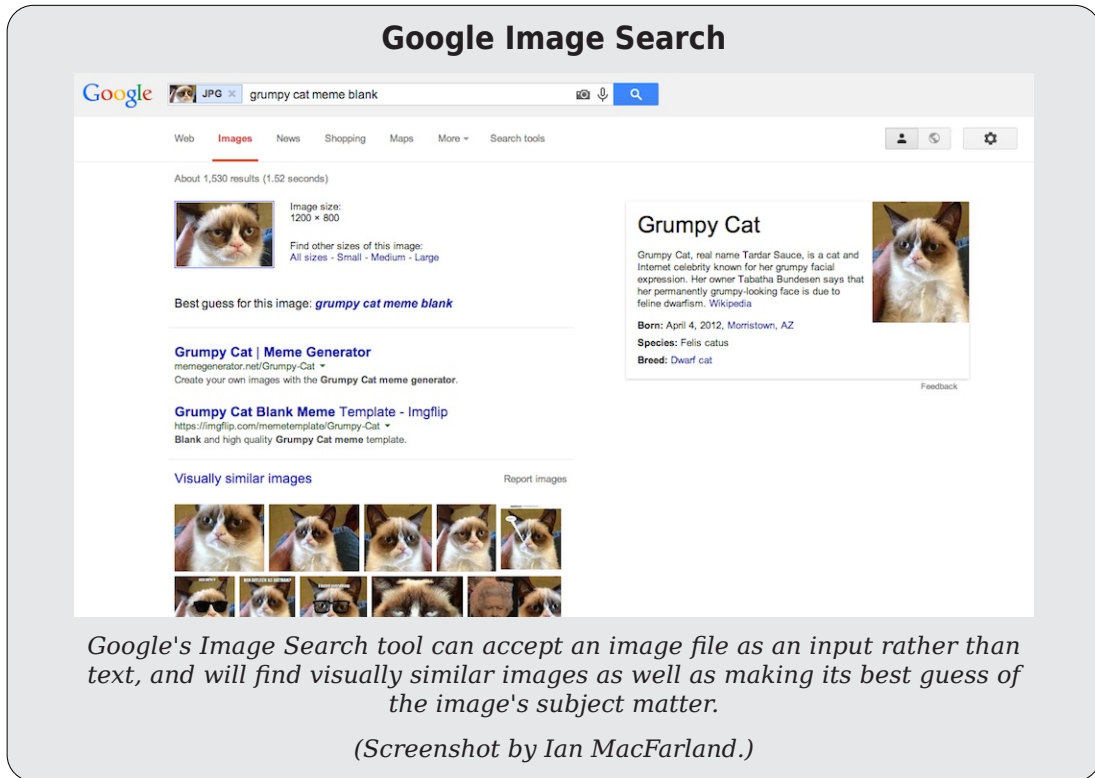
An organizing system should enable interactions that allow users to achieve their goals. The more abstract and intermediated the interaction between a user and an organizing system becomes, the more precisely the requirements must be expressed. User requirements can be stated or implied, depending on the sophistication and functional capabilities of the system.

In a closet, which is a personal organizing system for physical resources, the person searching with an intent to find a particular shirt might think, “Where is my yellow Hawaiian shirt?” but does not need to communicate the search criteria to anyone else in an explicit way. In a business or institutional organizing system, however, the user needs to describe the desired resource and interact with the system to select from candidate resources. This interaction might involve a human intermediary like a salesperson or reference librarian, or a computational one like a search engine.

A user’s information need usually determines the kind and content of resources required. User information needs are most often expressed in search queries (whatever is typed into a search box) or manifest themselves in the selection of one or more of the system categories that are offered for browsing. Queries can be as simple as a few keywords or very complex and specialized, employing different search fields or operators; they may even be expressed in a query language by expert users. Techniques such as spelling correction, query expansion, and suggestion assist users in formulating queries. Techniques like breadcrumb navigation and faceted filtering assist users in browsing an organizing system’s category system. Some systems allow the query to be expressed in natural language and then transform it into a description that is easier for the system to process. Queries for non-textual information like photos or videos are typically expressed as text, but some systems compute descriptions from non-textual queries such as images or audio files. For example, a user can hum a tune or draw or drag an image into an image query box.

Information needs of computational agents are determined by rules and criteria set by the creators of the agents (i.e., the function or goal of the agent). When a computational agent interacts with another computational agent or service by using its API, in the ideal case its output precisely satisfies those information needs.

Designers of organizing systems must recognize that people are not perfectly capable and rational decision makers. Limited memory and attention capacities prevent people from remembering everything and make them unable to consider more than a few things or choices at once. As a result of these fundamental limitations, people consciously and unconsciously reduce the cognitive effort they make when faced with decisions.



Google's Image Search tool can accept an image file as an input rather than text, and will find visually similar images as well as making its best guess of the image's subject matter.

(Screenshot by Ian MacFarland.)

One important way in which this affects how people behave demonstrates what Barry Schwartz calls *The Paradox of Choice*. You might think that people would prefer many options rather than just a few because that would better enable them to select a resource that best meets their requirements. In fact, because considering more choices requires more mental effort, this can cause stress and indecision and might cause people to give up. For example, when there were 24 different types of jam offered at an upscale market, more people stopped to taste than when only 6 choices were offered, but a greater percentage of people who were presented a smaller number of options actually made a purchase.

We see the same phenomenon when we compare libraries and bookstores. A rational book seeker should prefer the detailed classification system used in libraries over the very coarse BISAC system used in bookstores. However, many people say that the detailed system makes them work too hard, leading to calls that new libraries adopt the bookstore organizing system. (See §8.3.3)

People can avoid making choices if a system proposes or pre-selects an option for them that becomes a default choice if they do nothing. Often people will make a cursory assessment about how well the option satisfies a requirement and if it is good enough they will not consider any other alternatives.

Behavioral Economics

Classical economics assumes that humans are perfectly rational goal-oriented actors who act to achieve maximal satisfaction or utility. In contrast, behavioral economics recognizes the cognitive and emotional constraints on human behavior and assumes that people are biased and flawed decision makers.

Daniel Kahneman and Amos Tversky systematized the psychological foundations for behavioral economics, building on the work of Herbert Simon, who first proposed to understand people as “boundedly rational.” Kahneman and Tversky identified the systematic biases that prevent people from making optimal decisions and the heuristics they use to save cognitive effort. Kahneman contrasts classical and behavioral economics as follows:

Psychological theories of intuitive thinking cannot match the elegance and precision of formal normative models of belief and choice, but this is just another way of saying that rational models are psychologically unrealistic.

— (Kahneman 2003, p 1449)

Sunstein and Thaler popularized the application of behavioral economics as “libertarian paternalism,” with the goal of encouraging the design of organizing systems and policies that maintain or increase freedom of choice but which at the same time influence people to make choices that they would judge as good ones. This perspective is nicely captured by the title of their best-selling book, *Nudge*. Many government agencies and businesses in the US and elsewhere are building “nudging” principles into policies and products in the areas of social services, healthcare, and financial services because of the complexity of their offerings.

Behavioral economics complements the discipline of organizing by offering insights into the thinking and behavior of typical users that can lead to classifications and choices that make them more effective and satisfied. However, the principles of behavioral economics can be used to design organizing systems that manipulate people into taking actions and making choices that they might not intend or that are not in their best interests. (See *Dark Patterns* (page 99).)

The study of the limits to human rationality in decision-making is the centerpiece of the discipline known as **Behavioral Economics** (page 403).

Organizing systems should plan for interactions based on non-purposeful user behavior. A user who does not have a particular resource need in mind might interact with an organizing system to see what it contains or to be entertained or educated. Imagine a user going to a museum to avoid the heat outside. Their re-

quirement is to be out of the heat and—possibly—to see interesting things. A visitor to a zoo might go there to view a specific animal, but most of the time, visitors follow a more or less random path among the zoo’s resources. Similarly, web surfing is random, non-information-need-driven behavior. This type of requirement cannot be satisfied by providing search capabilities alone; other interaction types (e.g., browsing, suggestions) must be provided as well.

Lastly, not all users are human beings, typing in search queries or browsing through catalogs. An organizing system should plan for interaction scenarios where computational agents access the system via APIs (application programming interfaces), which require heavily standardized access procedures and resource descriptions in order to enable interactions.

10.2.2 Socio-Political and Organizational Constraints

An important constraint for interaction design choices is the access policies imposed by the producers of organizing systems, as already described in §3.4.3 *Access Policies* (page 115). If resources or their descriptions are restricted, interactions may not be able to use certain properties and therefore cannot be supported.

Inter-organizational or socio-political constraints are imposed when certain parties in an interaction, or even producers of an organizing system, can exert power over other parties and therefore control the nature of the interaction (or even the nature of the resource descriptions). We can distinguish different types of constraints:

Information and economic power asymmetry

Some organizations are able to impose their requirements for interactions and their resource description formats upon their clients or customers. For example, Google and Apple each have the power to control the extent of interoperability attainable in products, services, or applications that utilize their numerous platforms through mandated APIs and the process by which third-party applications are approved. The asymmetry between these dominant players and the myriad of smaller entities providing peripheral support, services, or components can result in *de facto* standards that may pose significant burden for small businesses and reduce overall competition.

Standards

Industry-wide or community standards can be essential in enabling interoperability between systems, applications, and devices. A standard interface describes the data formats and protocols to which systems should conform. Failure to adhere to standards complicates the merging of resources from different organizing systems. Challenges to standardization include organizational inertia; closed policies, processes, or development groups; intellectual property; credentialing; lack of specifications; competing standards;

high implementation costs; lack of conformance metrics; lack of clarity or awareness; and abuse of standards as trade barriers.

Public policy

Beyond businesses and standards-setting organizations, the government sector wields substantial influence over the implementation and success of possible interactions in organizing systems. As institutions with large and inalienable constituents, governments and governmental entities have similar influences as large businesses due to their size and substantial impact over society at large. Different forms of government around the world, ranging from centrally planned autocracy to loosely organized nation-states, can have far-reaching consequences in terms of how resource description policies are designed. Laws and regulations regarding data privacy prevent organizing systems from recording certain user data, therefore prohibiting interactions based on this information.

Even within the same firm or organization, constraints on interaction design may result from contradictory policies for organizing systems or even require the implementation of separate, disjoint systems that cannot be integrated without additional investment. Siloed business functions may be resistant to the merging of resources or resource descriptions in order to gain competitive advantage or command resources over other business functions.

Often characterized by different kinds of value contribution, different policies, processes, and practices, organizational units must clearly define and prioritize different interaction goals, align and coordinate processes, and build collaboration capabilities to achieve a high level of interoperability within the organizing system or between different organizing systems in the organization.

In addition to information exchange, organizational interoperability also aims to provide services that are widely available, easily identifiable, and accessible across the enterprise.

Nevertheless, inter-organizational constraints are inherently less deterministic than intra-organizational ones, because it is possible that a decision-maker with broad authority can decide that some interaction is important enough to warrant the change of institutional policies, formats, or even category systems. (See §7.2.3 Institutional Categories (page 274).)

Stop and Think: Standards

It is easy to take standards for granted, but without them our lives would run less smoothly because many products and services would not work very well or even be dangerous to use. If you search for the phrase “ISO Standard” along with almost anything, there is a good chance that you will find something. Try “currency,” “food,” “sunglasses,” “tea,” “water,” “wine,” and then a few of your own.

Regulatory Constraints: Right to be Forgotten

A controversial idea known as the “right to be forgotten” gained the force of law in the European Union in May 2014 after the EU's highest court ruled that people could ask search engines such as Google, which dominates the European market, to remove certain kinds of personal information from their search results.

The ruling had its foundations in the EU's 1995 Data Protection Directive, a data retention policy crafted in a time before the dominance of the Internet and search engines. While many privacy advocates hailed it as a victory, others in the technology and media firms have decried it as censorship. Either way, it has highlighted the need for the European Commission to update and modernize its data policy; a proposal has been before the European Parliament since 2012, and plans for its adoption were underway as of summer 2014. (Source: EC **fact sheet** on the “right to be forgotten” ruling.)

10.3 Reorganizing Resources for Interactions

Once the scope and range of interactions is defined according to requirements and constraints, the resources and the technology of the organizing system have to be arranged to enable the implementation of the desired interactions.

Commonly, interactions are determined at the beginning of a development process of the organizing system. It follows that most required resource descriptions (which properties of a resource are documented in an organizing system) need to be clarified at the beginning of the development process as well; that is, resource descriptions are determined based on the desired interactions that an organizing system should support. Most of these processes have been described in detail in **Chapter 5**, **Chapter 6** and **Chapter 9**.

Resources from different organizing systems are often aggregated to be accessed within one larger organizing system (warehouses, portals, search engines, union catalogs, cross-brand retailers), which requires resources and resource descriptions to be transformed in order to adapt to the new organizing system with its extended interaction requirements. Elsewhere, legacy systems often need to be updated to accommodate new standards, technologies, and interactions (e.g, mobile interfaces for digital libraries). That means that the necessary resources and resource descriptions for an interaction need to be identified, and, if necessary, changes have to be made in the description of the resources. Sometimes, resources are merged or transformed in order to perform new interactions.

10.3.1 Identifying and Describing Resources for Interactions

Individual and collection resource descriptions need to be carefully considered in order to record the necessary information for the designed interactions. (See [Chapter 9](#).) The type of interaction determines whether new properties need to be derived or computed with the help of external factors and whether these properties will be represented permanently in the organizing system (e.g., an extended topical description added due to a user comment) or created on the fly whenever a transaction is executed (e.g., a frequency count).

Determining which resources or resource descriptions will be used in an interaction is simple when all resources are included (e.g., in a simple search interaction over all resources in a data warehouse). Sometimes resources need to be identified according to more selective criteria such as resources exhibiting a certain property (e.g., all restaurants in your neighborhood with four stars on Yelp in an advanced search interaction).

10.3.2 Transforming Resources for Interactions

When an organizing system and its interactions are designed with resources or resource descriptions from legacy systems with outdated formats or from multiple organizing systems or when the new organizing systems has a different purpose and requires different resource properties, resources and their descriptions need to be transformed. The processing and transformation steps required to produce the expected modification can be applied at different layers:

Infrastructure or notation transformation

When resources are aggregated, the organizing systems must have a common basic infrastructure to communicate with one another and speak the same language. This means that participating systems must have a common set of communication protocols and an agreed upon way of representing information in digital formats, i.e., a notation ([§9.3.1](#)), such as the Unicode encoding scheme.

Writing system transformation

During a writing system transformation ([Chapter 9](#)), the syntax or vocabulary—also called the data exchange format—of the resource description will be changed to conform to another model, e.g., when library records are mapped from the MARC21 standard to the Dublin Core format in order to be aggregated, or when information in a business information system is transformed into an EDI or XML format so that it can be sent to another firm. Sometimes customized vocabularies are used to represent certain types of properties. These vocabularies were probably introduced to reduce errors or ambiguity or abbreviate common organizational resource properties. These customized vocabularies need to be explained and agreed upon by organizations combining resources to prevent interoperability problems.

Semantic transformation

Agreeing on a category or classification system (Chapter 7 & Chapter 8) is crucial so that organizing systems agree semantically—that is, so that resource properties and descriptions share not only technology but also meaning. For example, because the US Census has often changed its system of race categories, it is difficult to compare data from different censuses without some semantic transformation to align the categories.

Resource or resource description transformation

Resources or resource descriptions are often directly transformed, as when they are converted to another file format. In computer-based interactions like search engines, text resources are often pre-processed to remove some of the ambiguity inherent in natural language. These steps, collectively called *text processing*, include decoding, filtering, normalization, stopword elimination, and stemming. (See the sidebar, *Text Processing* (page 409))

10.3.2.1 Transforming Resources from Multiple or Legacy Organizing Systems

The traditional approach to enabling heterogeneous organizing systems to be accessed together has been to fully integrate them, which has allowed the “unrestricted sharing of data and business processes among any connected applications and data sources” in the organization. This can be a strategic approach to improving the management of resources, resource descriptions, and organizing systems as a whole, especially when organizations have disparate systems and redundant information spread across different groups and departments. However, it can also be a costly approach, as integration points may be numerous, with vastly different technologies needed to get one system to integrate with another. Maintenance also becomes an issue, as changes in one system may entail changes in all systems integrating with it.

Planning the transformation of resources from different organizing systems to be merged in an aggregation is called *data mapping* or alignment. In this process, aspects of the description layers (most often writing system or semantics) are compared and matched between two or more organizing systems. The relationship between each component may be unidirectional or bidirectional. In addition, resource properties and values that are semantically equivalent might have different names (the *vocabulary problem* of §4.4.2.1). The purpose of mapping may vary from allowing simple exchanges of resource descriptions, to enabling access to longitudinal data, to facilitating standardized reporting. The preservation of version histories of resource description elements and relations in both systems is vital for verifying the validity of the data map.

Similar to mapping, a straightforward approach to transformation is the use of *crosswalks*, which are equivalence tables that relate resource description

Text Processing

Decoding

A digital resource is first a sequence of bits. Decoding transforms those bits into characters according to the encoding scheme used, extracting the text from its stored form. (See §9.3.1 Notations (page 380).)

Filtering

If a text is encapsulated by formatting or non-semantic markup, these characters are removed because this information is rarely used as the basis of further interactions.

Tokenization

Segments the stream of characters (in an encoding scheme, a space is also a character) into textual components, usually words. In English, a simple rule-based system can separate words using spaces. However, punctuation makes things more complicated. For example, periods at the end of sentences should be removed, but periods in numbers should not. Other languages introduce other problems for tokenization; in Chinese, a space does not mark the divisions between individual concepts.

Normalization

Normalization removes superficial differences in character sequences, for example, by transforming all capitalized characters into lower-case. More complicated normalization operations include the removal of accents, hyphens, or diacritics and merging different forms of acronyms (e.g., U.N. and UN are both normalized to UN).

Stopword elimination

Stopwords are those words in a language that occur very frequently and are not very semantically expressive. Stopwords are usually articles, pronouns, prepositions, or conjunctions. Since they occur in every text, they can be removed because they cannot distinguish them. Of course, in some cases, removing stopwords might remove semantically important phrases (e.g., “To be or not to be”).

Stemming

These processing steps normalize inflectional and derivational variations in terms, e.g., by removing the “-ed” from verbs in the past tense. This homogenization can be done by following rules (*stemming*) or by using dictionaries (*lemmatization*). Rule-based stemming algorithms are easy to implement, but can result in wrongly normalized word groups, for example when “university” and “universe” are both stemmed to “univers.”

elements, semantics, and writing systems from one organizing system to those of another. Crosswalks not only enable systems with different resource descrip-

tions to interchange information in real-time, but are also used by third-party systems, such as harvesters and search engines to generate union catalogs and perform queries on multiple systems as if they were one consolidated system.

As the number of organizing systems increases, crosswalks and mappings become increasingly impractical if each pair of organizing systems requires a separate crosswalk. A more efficient approach would be the use of one vocabulary or format as a switching mechanism (also called a pivot or *hub language*) for all other vocabularies to map towards. Another possibility, which is often used in asymmetric power relationships between organizing systems, is to force all systems to adhere to the format that is used by the most powerful party.

10.3.2.2 Modes of Transformation

The conceptual relationships between different descriptions can be mapped out manually when creating simple maps. This, however, becomes more difficult as maps become more complex, due to the number of properties being mapped or when there are more structural or granularity issues to consider.

10.3.2.3 Granularity and Abstraction

Within writing system and semantic transformations, issues of *granularity* and level of abstraction (§5.3.1 **Determining the Scope and Focus** (page 191) and §7.4.1 **Category Abstraction and Granularity** (page 298)) pose the most challenges to cross-organizing system interoperability. *Granularity* refers to the level of detail or precision for a specific information resource property. For instance, the postal address of a particular location might be represented as several different data items, including the number, street name, city, state, country and postal code (a high-granularity model). It might also be represented in one single line including all of the information above (a low-granularity model). While it is easy to create the complete address by aggregating the different *information components* from the high-granularity model, it is not as easy to decompose the low-granularity model into more specific information components.

This does not mean, however, that a high-granularity model is always the best choice, especially if the context of use does not require it, as there are corresponding tradeoffs in terms of efficiency and speed in assembling and processing the resource information. (See the sidebar, **AccuWeather Request Granularity** (page 411))

The level of abstraction is the degree to which a resource description is abstracted from the concrete use case in order to fit a wider range of resources. For example, many countries have an address field called *state*, but in some countries, a similar regional division is called *province*. In order to accommodate both concepts, we can abstract from the original concrete concepts and establish a more abstract description of *administrative region*. Granularity and abstraction differ-

ences can occur at every resource property layer when resources need to be transformed; therefore, they need to be recognized and analyzed at every layer.

AccuWeather Request Granularity

Requests for AccuWeather data have exploded in the last years, due to automated requests from mobile devices to keep weather apps updated. The company has dealt with this challenge by truncating the GPS coordinates sent by the mobile device when it requests weather data (a transformation to lower granularity). If the request with the truncated coordinates is identical to one recently made, a cached version of the content is served, resulting in 300 million to 500 million fewer requests a day.

10.3.2.4 Accuracy of Transformations

Automatic mapping tools can only be as accurate as the specifications and criteria that are included in the mapping guidelines. Intellectual checks and tests performed by humans are almost always necessary to validate the *accuracy* of the transformation. Because description systems vary in expressive power and complexity, challenges to transformations may arise from differences in semantic definitions, rules regarding whether an element is required or requires multiple values, hierarchical or value constraints, and controlled vocabularies. As a result of these complexities, absolute transformations that ensure exact mappings will result in a loss of precision if the source description system is substantially richer than the target system.

In practice, relative crosswalks where all elements in a source description are mapped to at least one target, regardless of semantic equivalence, are often implemented. This lowers the quality and *accuracy* of the mapping and can result in “down translation” or “dumbing down” of the system for resource description. As a result of mapping compromises due to different granularity or abstraction levels, transformations from different organizing systems usually result in less granular or specific resource descriptions. Consequently, whereas some interactions are now enabled (e.g., cross-organizing system search), others that were once possible can no longer be supported. For example, conflating geographical and person subject fields from one system (e.g., geographical subject = Alberta, person subject = Virginia) to a joint subject field (e.g. subject = Alberta, Virginia) to transform to the resource description of another system does not allow for searches that distinguish between these specific categories anymore.

Stop and Think: Dumbing Down

Can you think of an example where resource description elements from one system are available for interaction in another due to a transformation, where the target system does not retain all the details of the descriptions in the source?

10.4 Implementing Interactions

The next sections describe some common interactions in digital organizing systems. One way to distinguish among them is to consider the source of the algorithms that are used in order to perform them. We can mostly distinguish *information retrieval* in-

teractions (e.g., search and browse), *machine learning* interactions (e.g., cluster, classify, extract) or *natural language processing* interactions (e.g., named entity recognition, summarization, sentiment analysis, anaphoric resolution). Another way to distinguish among interactions is to note whether resources are changed during the interaction (e.g., annotate, tag, rate, comment) or unchanged (search, cluster). Yet another way would be to distinguish interactions based on their absolute and relative complexity, i.e., on the progression of actions or steps that are needed to complete the interaction. Here, we will distinguish interactions based on the different resource description layers they act upon.

Chapter 3, *Activities in Organizing Systems*, introduced the concept of *affordance* or behavioral repertoire—the inherent actionable properties that determine what can be done with resources. We will now look at affordances (and constraints) that resource properties pose for interaction design. The interactions that an individual resource can support depend on the nature and extent of its inherent and described properties and internal structure. However, the interactions that can be designed into an organizing system can be extended by utilizing collection properties, derived properties, and any combination thereof. These three types of resource properties can be thought of as creating layers because they build on each other.

The further an organizing system moves up the layers, the more functional capabilities are enabled and more interactions can be designed. The degree of possible interactions is determined by the extent of the properties that are organized, described, and created in an organizing system. This marks a correlation between the extent of organization and the range of possible interactions: *The more extensive the organization and the number of identifiable resource properties, the larger the universe of “affordable” interactions.*

Interactions can be distinguished by four layers:

Interactions based on properties of individual resources

Resource properties have been described extensively in **Chapter 4** and **Chapter 5**. Any information or property that describes the resource itself can be used to design an interaction. If a property is not described in an organizing system or does not pertain to certain resources, an interaction that needs this information cannot be implemented. For example, a retail site like Shopstyle cannot offer to reliably search by color of clothing if this property is not contained in the resource description.

Interactions based on collection properties

Collection-based properties are created when resources are aggregated. (See **Chapter 1**.) An interaction that compares individual resources to a collection average (e.g., average age of publications in a library or average price of goods in a retail store) can only be implemented if the collection average is calculated.

Interactions based on derived or computed properties

Derived or computed properties are not inherent in the resources or collections but need to be computed with the help of external information or tools. The popularity of a digital resource can be computed based on the frequency of its use, for example. This computed property could then be used to design an access interaction that searches resources based on their popularity. An important use case for derived properties is the analysis of non-textual resources like images or audio files. For these content-based interactions, intrinsic properties of the resources like color distributions are computationally derived and stored as resource properties. A search can then be performed on color distributions.

Interactions based on combining resources

Combining resources and their individual, collection or derived properties can be used to design interactions based on joint properties that a single organizing system and its resources do not contain. This can lead to interactions that individual organizing systems with their particular purposes and resource descriptions cannot offer.

Whether a desired interaction can be implemented depends on the layers of resource properties that have been incorporated into the organizing system. How an interaction is implemented (especially in digital organizing systems) depends also on the algorithms and technologies available to access the resources or resource descriptions.

In our examples, we write primarily about textual resources or resource descriptions. Information retrieval of physical goods (e.g., finding a favorite cookie brand in the supermarket) or non-textual multimedia digital resources (e.g., finding images of the UC Berkeley logo) involves similar interactions, but with different algorithms and different resource properties.

10.4.1 Interactions Based on Instance Properties

Interactions in this category depend only on the properties of individual resource instances. Often, using resource properties on this lower layer coincides with basic action combinations in the interaction.

10.4.1.1 Boolean Retrieval

In a Boolean search, a query is specified by stating the information need and using operators from Boolean logic (AND, OR, NOT) to combine the components. The query is compared to individual resource properties (most often terms), where the result of the comparison is either TRUE or FALSE. The TRUE results are returned as a result of the query, and all other results are ignored. A Boolean search does not compare or rank resources so every returned resource is considered equally relevant. The advantage of the Boolean search is that the results are predictable and easy to explain. However, because the results of the Boolean model are not ranked by relevance, users have to sift through all the returned resource descriptions in order to find the most useful results.

10.4.1.2 Tag / Annotate

A tagging or annotation interaction allows a user (either a human or a computational agent) to add information to the resource itself or the resource descriptions. A typical tagging or annotation interaction locates a resource or resource description and lets the user add their chosen resource property. The resulting changes are stored in the organizing system and can be made available for other interactions (e.g., when additional tags are used to improve the search). An interaction that adds information from users can also enhance the quality of the system and improve its usability.

10.4.2 Interactions Based on Collection Properties

Interactions in this category utilize collection-level properties in order to improve the interaction, for example, to improve the ranking in a search or to enable comparison to collection averages.

10.4.2.1 Ranked Retrieval with Vector Space or Probabilistic Models

Ranked retrieval sorts the results of a search according to their relevance with respect to the information need expressed in a query. The Vector Space and Probabilistic approaches introduced here use individual resource properties like term occurrence or term frequency in a resource and collection averages of terms and their frequencies to calculate the rank of a resource for a query.

The simplicity of the Boolean model makes it easy to understand and implement, but its binary notion of relevance does not fit our intuition that terms dif-

fer in how much they suggest what a document is about. Gerard Salton invented the vector space model of information retrieval to enable a continuous measure of relevance. In the vector space model, each resource and query in an organizing system is represented as a vector of terms. Resources and queries are compared by comparing the directions of vectors in an n -dimensional space (as many dimensions as terms in the collection), with the assumption is that “closeness in space” means “closeness in meaning.”

In contrast to the vector space model, the underlying idea of the probabilistic model is that given a query and a resource or resource description (most often a text), probability theory is used to estimate how likely it is that a resource is relevant to an information need. A probabilistic model returns a list of resources that are ranked by their estimated *probability of relevance* with respect to the information need so that the resource with the highest probability to be relevant is ranked highest. In the vector space model, by comparison, the resource whose term vector is most similar to a query term vector (based on frequency counts) is ranked highest.

Both models utilize an intrinsic resource property called the *term frequency (tf)*. For each term, term frequency (tf) measures how many times the term appears in a resource. It is intuitive that *term frequency* itself has an ability to summarize a resource. If a term such as “automobile” appears frequently in a resource, we can assume that one of the topics discussed in the resource is automobiles and that a query for “automobile” should retrieve this resource. Another problem with the term frequency measure occurs when resource descriptions have different lengths (a very common occurrence in organizing systems). In order to compensate for different resource description lengths that would bias the term frequency count and the calculated relevance towards longer documents, the length of the term vectors are normalized as a percentage of the description length rather than a raw count.

Relying solely on term frequency to determine the relevance of a resource for a query has a drawback: if a term occurs in all resources in a collection it cannot distinguish resources. For example, if every resource discusses automobiles, all resources are potentially relevant for an “automobile” query. Hence, there should be an additional mechanism that penalize a term appearing in too many resources. This is done with *inverse document frequency*, which signals how often a term or property occurs in a collection.

Inverse document frequency (idf) is a collection-level property. The *document frequency (df)* is the number of resources containing a particular term. The inverse document frequency (idf) for a term is defined as $idf_t = \log(N/df_t)$, where N is the total number of documents. The inverse document frequency of a term decreases the more documents contain the term, providing a discriminating factor for the importance of terms in a query. For example, in a collection contain-

ing resources about automobiles, an information retrieval interaction can handle a query for “automobile accident” by lowering the importance of “automobile” and increasing the importance of “accident” in the resources that are selected as result set.

As a first step of a search, resource descriptions are compared with the terms in the query. In the vector space model, a metric for calculating similarities between resource description and query vectors combining the term frequency and the inverse document frequency is used to rank resources according to their relevance with respect to the query.

The probability ranking principle is mathematically and theoretically better motivated than the vector space ranking principle. However, multiple methods have been proposed to estimate the probability of relevance. Well-known probabilistic retrieval methods are Okapi BM25, *language models (LM)* and *divergence from randomness models (DFR)*. Although these models vary in their estimations of the probability of relevance for a given resource and differ in their mathematical complexity, intrinsic properties of resources like term frequency and collection-level properties like inverse document frequency and others are used for these calculations.

10.4.2.2 Synonym Expansion with Latent Semantic Indexing

Latent semantic indexing is a variation of the vector space model where a mathematical technique known as singular value decomposition is used to combine similar term vectors into a smaller number of vectors that describe their “statistical center.” This method is based mostly on collection-level properties like co-occurrence of terms in a collection. Based on the terms that occur in all resources in a collection, the method calculates which terms might be synonyms of each other or otherwise related. Put another way, latent semantic indexing groups terms into topics. Let us say the terms “roses” and “flowers” often occur together in the resources of a particular collection. The latent semantic indexing methodology recognizes statistically that these terms are related, and replaces the representations of the “roses” and “flower” terms with a computed “latent semantic” term that captures the fact that they are related, reducing the dimensionality of resource description (see §5.3.4.4 **Vocabulary Control as Dimensionality Reduction** (page 209)). Since queries are translated into the same set of components, a query for “roses” will also retrieve resources that mention “flower.” This increases the chance of a resource being found relevant to a query even if the query terms do not match the resource description terms exactly; the technique can therefore improve the quality of search.

Another approach for increasing the quality of search is to add similar terms or properties to a query from a controlled vocabulary or classification system. When a query can be mapped to terms in the controlled vocabulary or classes in

the classification, the inherent semantic structure of the vocabulary or classification can suggest additional terms (broader, narrower, synonymous) whose occurrence in resources can signal their relevance for a query.

10.4.2.3 Structure-Based Retrieval

When the internal structure of a resource is represented in its resource description a search interaction can use the structure to retrieve more specific parts of a resource. This enables parametric or zone searching, where a particular component or resource property can be searched while all other properties are disregarded. For example, a search for “Shakespeare” in the title field in a bibliographic organizing system will only retrieve books with Shakespeare in the title, not as an author. Because all resources use the same structure, this structure is a collection-level property.

10.4.2.4 Clustering / Classification

Clustering (§7.5.3.3) and computational classification (§8.7) are both interactions that use individual and collection-level resource properties to execute their operation. During clustering (unsupervised learning), all resources are compared and grouped with respect to their similarity to each other. During computational classification (supervised learning), an individual resource or a group of resources is compared to a given classification or *controlled vocabulary* in an organizing system and the resource is assigned to the most similar class or descriptor. Another example for a classification interaction is spam detection. (See §8.7.) Author identification or characterization algorithms attempt to determine the author of a given work (a classification interaction) or to characterize the type of author that has or should write a work (a clustering interaction).

10.4.3 Interactions Based on Derived Properties

Interactions in this category derive or compute properties or features that are not inherent to the resources themselves or the collection. External data sources, services, and tools are employed to support these interactions. Building interactions with conditionality based on externally derived properties usually increases the quality of the interactions by increasing the system’s context awareness.

10.4.3.1 Popularity-Based Retrieval

Google’s PageRank (see §6.5.3) is the most well-known popularity measure for websites. The basic idea of PageRank is that a website is as popular as the number of links referencing the website. The actual calculation of a website’s PageRank involves more sophisticated mathematics than counting the number of in-

Retail Store Activity Tracking



Retail analytics companies use cameras and other sensors to analyze shopper activity in retail stores and generate heatmaps of which areas see the most foot traffic and which items customers interact with most.

(Photo by Flickr user m01229. Creative Commons license. Illustration of heatmap by Ian MacFarland.)

links, because the source of links is also important. Links that come from quality websites contribute more to a website's PageRank than other links, and links to qualitatively low websites will hurt a website's PageRank.

An information retrieval model for web pages can now use PageRank to determine the value of a web page with respect to a query. Google and other web search engines use many different ranking features to determine the final rank of a web page for any search, PageRank as a popularity measure is only one of them.

Other popularity measures can be used to rank resources. For example, frequency of use, buying frequency for retail goods, the number of laundry cycles a particular piece of clothing has gone through, and even whether it is due for a laundry cycle right now.

10.4.3.2 Citation-Based Retrieval

Citation-based retrieval is a sophisticated and highly effective technique employed within bibliographic information systems. Bibliographic resources are linked to each other by citations, that is, when one publication

cites another. When a bibliographic resource is referenced by another resource, those two resources are probably thematically related. The idea of citation-based search is to use a known resource as the information need and retrieve other resources that are related by citation.

Citation-based search can be implemented by directly following citations from the original resource or to find resources that cite the original resource. Another comparison technique is the principle of bibliographic coupling, where the information retrieval system looks for other resources that cite the same resources as the original resource. Citation-based search results can also be ranked,

for example, by the number of in-citations a publication has received (the PageRank popularity measure actually derives from this principle).

10.4.3.3 Translation

During translation, resources are transformed into another language, with varying degrees of success. In contrast to the transformations that are performed in order to merge resources from different organizing systems to prepare them for further interactions, a translation transforms the resource after it has been retrieved or located. Dictionaries or parallel corpora are external resources that drive a translation.

During a dictionary-based translation, every individual term (sometimes phrases) in a resource description is looked up in a dictionary and replaced with the most likely translation. This is a simple translation, as it cannot take grammatical sentence structures or context into account. Context can have an important impact on the most likely translation: the French word *avocat* should be translated into *lawyer* in most organizing systems, but probably not in a cookbook collection, in which it is the *avocado* fruit.

Parallel corpora are a way to overcome many of these challenges. Parallel corpora are the same or similar texts in different languages. The Bible or the protocols of United Nations (UN) meetings are popular examples because they exist in parallel in many different languages. A machine learning algorithm can learn from these corpora to derive which phrases and other grammatical structures can be translated in which contexts. This knowledge can then be applied to further resource translation interactions.

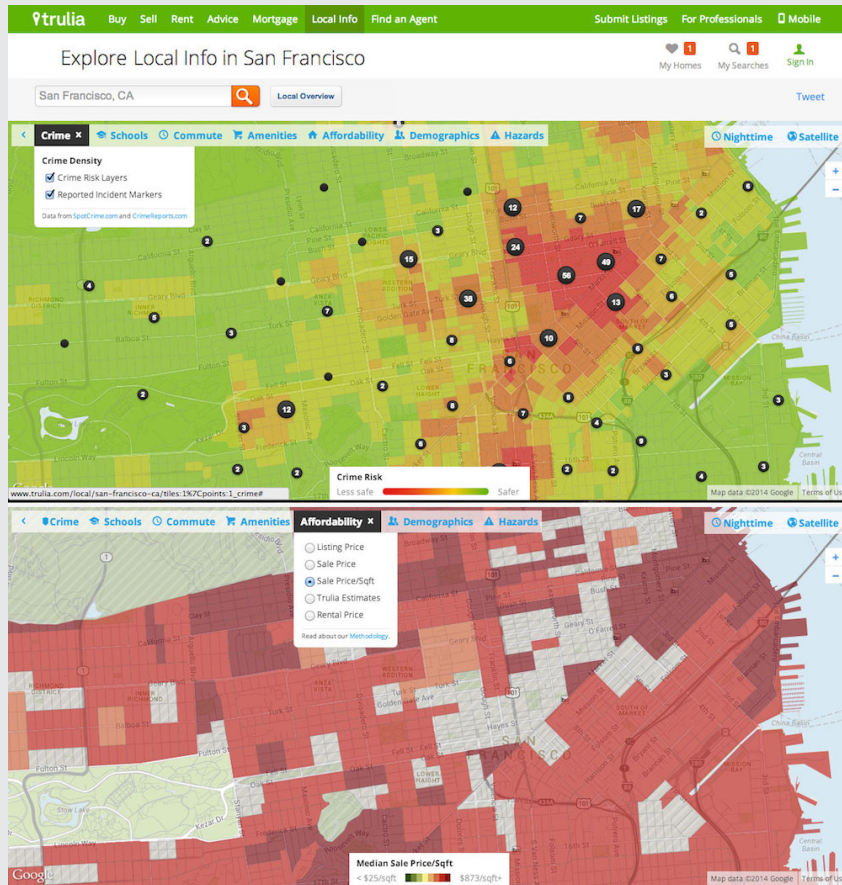
10.4.4 Interactions Based on Combining Resources

Interactions in this category combine resources mostly from different organizing systems to provide services that a single organizing system could not enable. Sometimes different organizing systems with related resources are created on purpose in order to protect the privacy of personal information or to protect business interests. Releasing organizing systems to the public can have unwanted consequences when clever developers detect the potential of connecting previously unrelated data sources.

10.4.4.1 Mash-Ups

A mash-up combines data from several resources, which enables an interaction to present new information that arises from the combination. For example, housing advertisements have been combined with crime statistics on maps to graphically identify rentals that are available in relatively safe neighborhoods.

Mash-up of Housing and Crime Stats



The “Local Info” map on real-estate website Trulia mashes up data on crime, schools, housing prices, commute times, and other factors relevant to people searching for a new place to live.

(Screenshots by Ian MacFarland.)

Mash-ups are usually *ad-hoc* combinations at the resource level and therefore do not impact the “mashed-up” organizing systems’ internal structures or vocabularies; they can be an efficient instrument for rapid prototyping on the web. On the other hand, that makes them not very reliable or robust, because a mash-up can fail in its operation as soon as the underlying organizing systems change.

10.4.4.2 Linked Data Retrieval and Resource Discovery

In §9.4.3 *The Semantic Web World* (page 392), linked data relates resources among different organizing system technologies via standardized and unique identifiers (URIs). This simple approach connects resources from different systems with each other so that a cross-system search is possible. For example, two different online retailers selling a Martha Stewart bedspread can link to a website describing the bedspread on the Martha Stewart website. Both retailers use the same unique identifier for the bedspread, which leads back to the Martha Stewart site.

Resource discovery or linked data retrieval are search interactions that traverse the network (or semantic web graph) via connecting links in order to discover semantically related resources. A search interaction could therefore use the link from one retailer to the Martha Stewart website to discover the other retailer, which might have a cheaper or more convenient offer.

10.5 Evaluating Interactions

Managing the quality of an interaction with respect to its intent or goal is a crucial part of every step from design through implementation and especially during operation. Evaluating the quality of interactions at different times in the design process (design concept, prototype, implementation, and operation) reveals both strengths and weaknesses to the designers or operators of the organizing system.

During the design and implementation stages, interactions need to be tested against the original goals of the interaction and the constraints that are imposed by the organizing system, its resources and external conditions. It is very common for processes in interactions to be tweaked or tuned to better comply with the original goals and intentions for the interaction. Evaluation during these stages often attempts to provide a calculable way to measure this compliance and supports the fine-tuning process. It should be an integral part of an iterative design process.

During the later implementation and operation stages, interactions are evaluated with respect to the dynamically changing conditions of the organizing system and its environment. User expectations as well as environmental conditions or constraints can change and need to be checked periodically. A systematic evaluation of interactions ensures that changes that affect an interaction are observed early and can be integrated in order to adjust and even improve the interaction. At these stages, more subjective evaluation aspects like satisfaction, experience, reputation, or “feel” also play a role in fine-tuning the interactions. This subjective part of the evaluation process is as important as the quantitative, objective part. Many factors during the design and implementation pro-

cesses need to be considered and made to work together. Ongoing quality evaluation and feedback ensures that interactions work as intended.

Evaluation aspects can be distinguished in numerous ways: by the effort and time to perform them (both data collection and analysis); by how quantifiable they are or how comparable they are with measures in other organization systems; by what component of the interaction or organizing system they focus on; or by the discipline, expertise, or methodologies that are used for the evaluation.

A common and important distinction is the difference between efficiency, effectiveness, and satisfaction. An interaction is efficient when it performs its actions in a timely and economical manner, effective when it performs its actions correctly and completely, satisfactory when it performs as expected. Satisfaction is the least quantifiable of the evaluation aspects because it is highly dependent on individual tastes and experiences.

Let us assume that Shopstyle.com develops a new interaction that lets you compare coat lengths from the offerings of their various retailers. Once the interaction is designed, an evaluation takes place in order to determine whether all coats and their lengths are integrated in the interaction and whether the coat lengths are measured and compared correctly. The designers would not only want to know whether the coat lengths are represented correctly but also whether the interaction performs efficiently. When the interaction is ready to be released (usually first in beta or test status), users and retailers will be asked whether the interaction improves their shopping experience, whether the comparison performs as they expected, and what they would change. These evaluation styles work hand in hand in order to improve the interaction.

10.5.1 Efficiency

When evaluating the efficiency of an organizing system, we focus on the time, energy and economic resources needed in order to achieve the interaction goals of the system. Commonly, the fewer resources are needed for achieving a successful interaction, the more efficient the interaction.

Efficiency measures are usually related to engineering aspects such as the time to perform an action, number of steps to perform an interaction, or amount of computing resources used. Efficiency with respect to the human costs of memory load, attention, and cognitive processing is also important if there is to be a seamless user experience where users can interact with the system in a timely manner.

For a lot of organizing system interactions, however, effectiveness is the more important aspect, particularly for those interactions that we have looked at so far. If search results are not correct, then users will not be satisfied by even the

most usable interface. Many interactions are evaluated with respect to their ability to return relevant resources. Why and how this is evaluated is the focus of the remainder of this section.

10.5.2 Effectiveness

Effectiveness evaluates the correct output or results of an interaction. An effective interaction achieves relevant, intended or expected results. The concept of *relevance* and its relationship to effectiveness is pivotal in information retrieval and machine learning interactions. (§10.5.2.1) Effectiveness measures are often developed in the fields that developed the algorithm for the interaction, information retrieval, or machine learning. *Precision* and *recall* are the fundamental measures of *relevance* or effectiveness in information retrieval or machine learning interactions. (§10.5.2.2)

10.5.2.1 Relevance

Relevance is widely regarded as the fundamental concept of information retrieval, and by extension, all of information science. Despite being one of the more intuitive concepts in human communication, relevance is notoriously difficult to define and has been the subject of much debate over the past century.

For the purpose of organizing systems, relevance is a concept for evaluating effectiveness that describes whether a stated or implicit information need is satisfied in a particular user context and at a particular time. One of the challenges for the evaluation of relevance in organizing systems is the gap between a user's information need (often not directly stated), and an expression of that information need (a query). This gap might result in ambiguous results in the interaction. For example, suppose somebody speaks the word "Paris" (query) into a smart phone application seeking advice on how to travel to Paris, France. The response includes offers for the Paris Hotel in Las Vegas. Does the result satisfy the information need? What if the searcher receives advice on Paris but has already seen every one of the resources the organizing system offers? What is the correct decision on relevance here?

The key to calculating effectiveness is to be aware of what is being measured. If the information need as expressed in the query is measured, the topical relevance or topicality—a system-side perspective is analyzed. If the information need as in a person's mind is measured, the pertinence, utility, or situational relevance—a subjective, personal perspective is analyzed. This juxtaposition is the point of much research and contention in the field of information retrieval, because topical relevance is objectively measurable, but subjective relevance is the real goal. In order to evaluate relevance in any interaction, an essential prerequisite is deciding which of these notions of relevance to consider.

10.5.2.2 The Recall / Precision Tradeoff

Precision measures the *accuracy* of a result set, that is, how many of the retrieved resources for a query are relevant. *Recall* measures the completeness of the result set, that is, how many of the relevant resources in a collection were retrieved. Let us assume that a collection contains 20 relevant resources for a query. A retrieval interaction retrieves 10 resources in a result set, 5 of the retrieved resources are relevant. The precision of this interaction is 50% (5 out of 10 retrieved resources are relevant); the recall is 25% (5 out of 20 relevant resources were retrieved).

It is in the nature of information retrieval interactions that recall and precision trade off with each other. To find all relevant resources in a collection, the interaction has to cast a wide net and will not be very precise. In order to be very precise and return only relevant resources to the searcher, an interaction has to be very discriminating and will probably not find all relevant resources. When a collection is very large and contains many relevant resources for any given query, the priority is usually to increase precision. However, when a collection is small or the information need also requires finding all relevant documents (e.g., in case law, patent searches, or medical diagnosis support), then the priority is put on increasing recall.

The completeness and granularity of the organizing principles in an organizing system have a large impact on the trade-off between *recall* and *precision*. (See [Chapter 4](#).) When resources are organized in fine-grained category systems and many different resource properties are described, high-precision searches are possible because a desired resource can be searched as precisely as the description or organization of the system allows. However, very specialized description and organization may preclude certain resources from being found; consequently, recall might be sacrificed. If the organization is superficial—like your sock drawer, for example—you can find all the socks you want (high recall) but you have to sort through a lot of socks to find the right pair (low precision). The trade-off between recall and precision is closely associated with the extent of the organization.

10.5.3 Satisfaction

Satisfaction evaluates the opinion, experience or attitude of a user towards an interaction. Because satisfaction depends on individual user opinions, it is difficult to quantify. Satisfaction measures arise out of the user's experience with the interaction—they are mostly aspects of user interfaces, usability, or subjective and aesthetic impressions.

Usability measures whether the interaction and the user interface designed for it correspond with the user's expectations of how they should function. It particularly focuses on the usefulness of the interaction. Usability analyzes ease-of-

use, learnability, and cognitive effort to measure how well users can use an interaction to achieve their task.

Although efficiency, effectiveness, and satisfaction are measured differently and affect different components of the interaction, they are equally important for the success of an interaction. Even if an interaction is fast, it is not very useful if it arrives at incorrect results. Even if an interaction works correctly, user satisfaction is not guaranteed. One of the challenges in designing interactions is that these factors invariably involve tradeoffs. A fast system cannot be as precise as one that prioritizes the use of contextual information. An effective interaction might require a lot of effort from the user, which does not make it very easy to use, so the user satisfaction might decrease. The priorities of the organizing system and its designers will determine which properties to optimize.

Let us continue our Shopstyle coat-length comparison interaction example. When the coat length calculation is performed in an acceptable amount of time and does not consume a lot of the organizing systems resources, the interaction is efficient. When all coat lengths are correctly measured and compared, the interaction is effective. When the interaction is seamlessly integrated into the shopping process, visually supported in the interface, and not cognitively exhausting, is it probably satisfactory for a user, as it provides a useful service (especially for someone with irregular body dimensions). What aspect should Shopstyle prioritize? It will probably weigh the consequences of effectiveness versus efficiency and satisfaction. For a retail- and consumer-oriented organizing system, satisfaction is probably one of the more important aspects, so it is highly likely that efficiency and effectiveness might be sacrificed (in moderation) in favor of satisfaction.

10.6 Key Points in Chapter Ten

- Interactions arise naturally from the affordances of resources or are purposefully designed into organizing systems.
(See §10.1 Introduction (page 395))
- Accessing and merging resources are fundamental interactions that occur in almost every organizing system.
(See §10.1 Introduction (page 395))
- User requirements, which layer of resource properties is used, and the legal, social and organizational environment can distinguish interactions.
(See §10.2 Determining Interactions (page 400))
- Limited memory and attention capacities prevent people from remembering everything and make them unable to consider more than a few things or choices at once.

(See §10.2.1 User Requirements (page 400))

- In order to enable interactions, it is necessary to identify, describe, and sometimes transform the resources in an organizing system.

(See §10.3.1 Identifying and Describing Resources for Interactions (page 407))

- Merging transformations can be distinguished by type (mapping or cross-walk), time (design time or run time) and mode (manual or automatic).

(See §10.3.2.3 Granularity and Abstraction (page 410))

- Implementations can be distinguished by the source of the algorithm (information retrieval, machine learning, natural language processing), by their complexity (number of actions needed), by whether resources are changed, or by the resource description layers they are based on.

(See §10.4 Implementing Interactions (page 412))

- Important aspects for the evaluation of interactions are efficiency (timeliness and cost-effectiveness), effectiveness (accuracy and relevance) and satisfaction (positive attitude of the user).

(See §10.5 Evaluating Interactions (page 421))

- The concept of *relevance* and its relationship to effectiveness is pivotal in information retrieval and machine learning interactions.

(See §10.5.2.1 Relevance (page 423))

- The trade-off between recall and precision decides whether a search finds all relevant documents (high recall) or only relevant documents (high precision).

(See §10.5.2.2 The Recall / Precision Tradeoff (page 424))

- The extent of the organization principles also impacts recall and precision: more fine-grained organization allows for more precise interactions.

(See §10.5.2.2 The Recall / Precision Tradeoff (page 424))