# How Enterprise Search Works

*By Stephen E. Arnold*

*This article is excerpted from The Enterprise Search Report, a 350-page evaluation of 28 enterprise search vendors, published by CMS Watch. For more details, see www.cmswatch.com/EntSearch/*

Search is a complex process, and a search system is not one thing. An enterprise search system has modules or, in large-scale systems, separate clusters of servers to perform the different tasks needed to deliver answers. The popularity of consumer search services and the "anybody can use Google" simplicity masks the plumbing underneath search systems. A quick review of the basic components of a typical enterprise search system allows us to highlight the key subsystems, technologies, and functions of the aggregated system. The differences among them often have significant impact on in how a particular search system will match with certain types of requirements. For example, a search system that indexes Web pages in an organization's portal may rely on a software script called a crawler or spider to find content. That system may be "blind" to information in databases or certain types of files unless additional filters, scripts, or subsystems are "plugged into" the search system.

Vendors offer a progression of upgrades to accommodate evolving needs when special content or advanced query processing features are required. This is why the difference between an entry-level search system and the industrial strength search system can be measured in the hundreds of thousands of dollars to a million dollars or more.

## Components of a Search System

Search systems face in two directions: towards the content and toward the searcher. Roughly speaking, a search engine must index content and process queries. To accomplish these tasks, most search systems have four major interdependent components of varying complexity; content acquisition, indexing, query processing (parsing, matching, and post-processing), and formatting results.

If something goes wrong in any one of these subsystems, it can significantly downgrade the performance and effectiveness of the search engine as a whole. Most of the processes in a search system are computationally intensive. In order to improve the overall performance, additional hardware,

memory, storage, or bandwidth may be needed. To speed up response time in the query processing module, the licensee may need to reduce the number of documents acquired and indexed, disable these features until usage declines, limit the number of users, place a ceiling on the number of results displayed per query, or eliminate certain computationally expensive indexing processes such as automated metatagging.

## Content Acquisition

There are several different approaches to content acquisition. These include:

- **The content acquisition subsystem** "obtains" content. Existing content must be identified, copied from its location to a processing folder, then moved to a "to be indexed" folder. This is the "traditional" approach available within most enterprise search systems.
- **The system administrator** configures the servers with content to run a script, identify changed or new content, then copies that content to a new folder, processes the content to some degree, and then "pushes" the processed files to the indexing subsystem. This is an approach supported by such enterprise systems as FAST Search & Transfer, Verity, and others.
- **A script**—usually called a spider—visits on a scheduled basis servers, folders, or files. When a change or a new document is identified, the script copies the file to the index processing subsystem. This is an approach supported by virtually all search systems today, but remains a surprisingly complicated exercise: spiders can experience problems with session variables in URLs, JavaScript, Flash, and forms. Improperly configured, a spider can chase its own tail through a series of infinitely recursive links.

These three options can be mixed and matched by the search system licensee. No single approach is appropriate in most organizations. Hybrid content acquisition techniques are the norm.

## Indexing

In the most basic form, indexing includes the processes associated with identifying the key words in a document. Once the key words are identified, pointers are created to them. Indexing can also perform an extremely wide range of value-added processes in order to create synonyms for words or concepts in a document, identify the specific topics discussed in a document and map the document into an enterprise taxonomy or classification schema. For example, a document may discuss a new technology without referencing a new product initiative, yet still be highly relevant for anyone interested in that initiative. Indexes must also be updated periodically as well.

## Query Processing

Query processing is the search subsystem that processes the user's query, passes the query against the index or indexes, processes the "hits," and retrieves the results. Like indexing, the opportunities for adding functionality to query processing are abundant. Let's look briefly at each of these functions.

**Parsing the query.** Search systems do not understand user queries. The searcher's query—e.g., for content dealing with "dual Xeon IBM"—must be converted by the search system into syntax that the search system uses. The query parser is the component that translates a natural-language sentence such as "What information is available about dual Xeon IBM computers?" into the appropriate, pared-down syntax.

Each search engine uses its own query parser. Some systems such as Thunderstone rely on a proprietary language. Another approach is to break down the user's query and reassemble it as a Boolean query or a SQL instruction. Oracle and Speed of Mind use the latter approach.

One approach is to use a series of templates. Ask Jeeves, the popular search engine, accepts sentences, matches the form of the sentence against a library of templates, and assembles a Boolean query from the template.

Sometimes there is a query, but no one has actively submitted it. For search systems that are anchored to specific business processes, the user's query may already be stored in the system and even set to run in advance of the user's requesting a document. When a customer service agent activates a particular screen or reaches a specific step in a work process, the saved query is run or retrieved from a cache, and the results are available without the user typing anything in a search box. There are many variations on saved searches and workflow integration. These types of querying techniques are popular because they do not require the employee to formulate a query and then browse a list of results looking for a relevant answer. As a result, time is saved, at the possible expense of searcher latitude.

In any event formulating queries remains a somewhat difficult task (which explains why about half of the queries on the Yahoo site are launched by clicking on one of the Yahoo hot links for specific information)—in some cases better left to the enterprise than the individual. Automated queries may not suit general-purpose searching by knowledge workers, but they can often streamline production processes for line workers.

**Passing the query against the index.** Once the engine converts the query into a set of instructions the index "understands," the query is passed against the index. The query processing system captures the matches and creates the results list. In addition to standard matches, the engine may also include "best bets" in the matches. Best bets are hits that have been manually determined relevant to a particular query, regardless of the intrinsic content in or title of the document. This is a labor-intensive (and inherently subjective) process that creates a static list that needs to be frequently reviewed for relevance. However, it can yield good results, especially on e-commerce sites, but your search engine must support this feature.

**Post processing.** In this phase, the engine may perform certain value-adding processes only once the result set has been retrieved from the index. It may merge hits from multiple indexes, sort documents (e.g., according to relevance), or apply some other logic to the results.

This could include, for example, categorizing content on its way out of the system, rather than on its way in. In this way, the load on the document processing and indexing subsystems is shifted to the query processing system in order to improve the speed of document processing—at the expense, perhaps, of retrieval performance.

Alternatively, the search engine may

employ a variety of short cuts to improve usability, such as pulling results from the index of the most recently processed content so the user has some initial results to examine as the more sluggish retrieval of hits from the legacy index is completed.

## Formatting Results

It is here that results are formatted for display to searchers. Typically, hits are streamed out into some sort of template. Modern search systems allow you to employ standard scripting interfaces (e.g., JSP and ASP) so that you can maintain a search results page like any other template on your site, but you will want to check for limitations here. Many, but not all, search vendors will provide results in an XHTML format that allows designers to apply CSS style sheets against the results, such that a document summary can appear in a different format than the "last-modified" information.

An administrator will need to decide which results fields are actually displayed. Best practices suggests a variety of usability conventions to be followed:

- Reiterating the search box with the original query pre-populated.
- Indicating the total number of "hits."
- Displaying document summaries and other metadata (such as size, location, language, date modified, etc.) where appropriate.
- Indicating the relevancy rank (if any) explicitly.

## Streamlining Query Processing

Query processing can be streamlined in many ways. Among the options are:

- Identifying frequently-run queries, running them on a schedule, creating the results list in advance, therefore providing the capability to display results immediately.
- Creating "hard wired" queries. Taxonomy trees, stored searches, and directory-style listings are essentially stored queries. When the searcher clicks on the hyperlink, the system sends the instruction to the index matcher, without the user having to formulate a query.
- Prefetching results. Some systems delivering information to support work processes trigger standing queries when a user reaches a certain point in a work process.

## Retrieving the Document

No search is complete until the searcher has actually retrieved the right documents. Though not a search subsystem per se, retrieving documents may not always be straightforward. Missing information is a concern in an enterprise search environment where it is vital that indexed documents be available to employees authorized to view them.

Enterprise search systems have two basic ways to handle the delivery of a document. First, the enterprise system can

work just like the Web. The index points to the server and location where the indexed document resides. If the network is overloaded, the document may not display or be delayed. The second approach is that the content is stored in a search repository. The documents are retrieved from the repository, not the server or workstation on which the original document resides. The benefit of this approach is that any indexed document is, by definition, stored in the repository. The downside is that synchronizing the documents that change in their source location with the version of the documents in the search repository requires additional system administration work and infrastructure. It will also increase your storage requirements.

*Stephen Arnold (sa@arnoldit.com) has been a search and retrieval technology consultant for more than 3 decades. He heads Arnold Information Technology in Harrod's Creek, KY.*

# Build It So They Can Find It

## The practical uses of building a business taxonomy.

*By Theresa Regli*

*Listen to Theresa speak on this topic at the AIIM ON DEMAND Conference in Philadelphia on May 17-19.*

It's 6:30 p.m. on a weeknight, and as you're headed out of the office you remember there's a few items you've got to pick up at the grocery store: tissues, corn flakes, and eggs. You dash into the store that's on the way—one that you shop at constantly, and thus know all its quirks—and go the paper goods aisle for your tissues, the cereal aisle for your corn flakes, the dairy section for your eggs. You're in and out of the store in less than five minutes.

Taxonomies create massive efficiencies in our everyday lives, and yet we constantly take them for granted. Imagine trying to find tissues, corn flakes, and eggs in a grocery store without the classified aisles that guide you to the right place to find what you—the user, consumer, and/or customer—really need, efficiently and accurately. And yet, without taxonomies, businesses leave their customers to sift through huge collections of products and content without guidance, and their experience is not unlike looking through all the items in a grocery store in a futile attempt to find a box of corn flakes. Grocery stores are organized by major categories, sub-categories, and so forth, until we find the specific product we're after. Finally, after years of sub-optimal CMS implementations—where costly software was implemented without an adequate content architecture to support it—businesses are learning yet another a lesson from the brick-and-mortar product stores. Businesses need to create categorical aisles of their own.

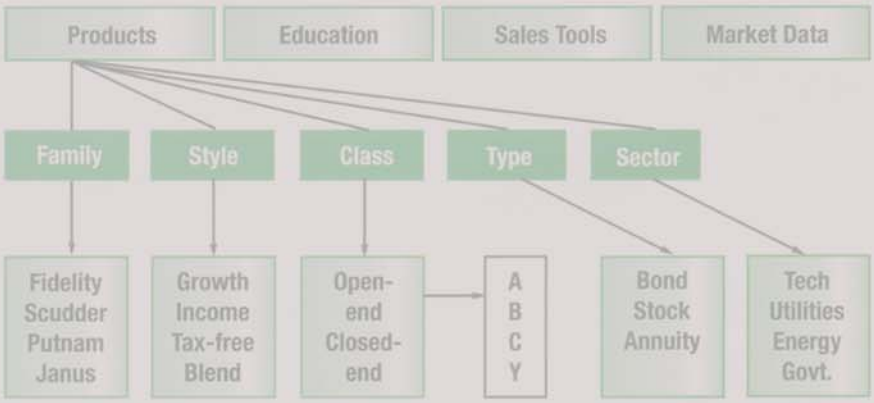
### What Is Taxonomy?

Taxonomy is the science of classification and labeling, or more simply—a law for categorizing information. From the Greek taxis meaning "arrangement" or "division" and nomos meaning "law," a good taxonomy takes into account the elements of a group (taxon) and its subgroups (taxa) that are mutually exclusive and, taken together, include all possibilities.

For the purposes of a content management system implementation, the primary purpose of a taxonomy is to provide a framework for the categorization and tagging of content in the system, enabling the business to present content in very specific ways, for specific sites, and eventually, target that content to specific audiences or individuals.

Common taxonomies include the grocery store scheme cited earlier, the library's Dewey Decimal System, the Periodic Table of Elements, and Carl Linnaeus' classification all of living things that you may remember from your high school biology (kingdom, phylum, class, order, genus, species). Linnaeus' example is particularly relevant, since like any good taxonomy, it progresses from the general to the more specific.

The image [below] represents a sliver of a categorization scheme (taxonomy) for a financial services organization. The financial services industry is one of the most standardized in the way it classifies information, and thus the example below could apply to many financial organizations, since most of them have the same types of products that are classified in similar ways.




Sample taxonomy for a financial service company's product.