# Data Classification Using a Digital Taxonomy

## As the volume of digital content grows, so does the need for efficient and accurate data retrieval

By David Athey

As an organization's vast collection of data continues to grow, it becomes increasingly difficult for users to find the information they need. You need only to look at the success of Google to see the importance of search engine technology. Unfortunately, traditional search engines that rely primarily on keyword matching often return unintended results.

This makes finding the information that you're really after time consuming and inefficient. To make data search and discovery more productive, organizations are turning to taxonomy-based data classification.

Taxonomy classification is a means of creating order out of large collections of data. At its most basic level, taxonomy is simply a collection of terms or subjects. The strength of the model, however, comes from the taxonomy's ability to also define a term's relationship to other terms. This provides the means to derive the terms' context based on the relationships in the taxonomy. If a single term has several different meanings, it will have these additional associations defined in the taxonomy.

In most implementations the model is flexible, allowing for relationships to be expressed in much greater detail than is available from a strict hierarchical model. This allows for the definition of "related" and "equivalent" terms; something that is more difficult in typical hierarchical trees. Among other benefits, this makes it possible to implement an Amazon-like "recommendation" engine to find related items that are defined in similar topic areas within the same taxonomy.

When a taxonomy classification is being used, data that is added to a system is classified using the terms that have already been defined in the taxonomy. When data is associated with one or more terms, the data inherits the properties and relationships of those terms. This reduces the work involved with classifying new data. Also, as the taxonomy definition is improved and updated, the new data associations will be effective for the existing data without the need to go back and manually reclassify it.

## Finding Your Data: No Problem!

Removing keyword ambiguity should be a goal of all search implementations. With conventional search engines, a keyword search for the term "star" could return results both on Astronomy as well as Hollywood actors. With a taxonomy-based search, the multiple contexts would be known and presented to the user allowing for further refinement of search results based only on the desired subject matter, or "facets." Irrelevant data is filtered out, leaving behind only the results that are applicable to the selected topic area.

The same system that removes ambiguity also allows for the benefit of data discovery. Looking for data using a taxonomy navigation tool is similar to browsing the book aisles of a library. You may not know what you're looking for, but you'll know it when you see it. Users are able to "browse" the data that is associated with nearby terms in the taxonomy, allowing them to find information they might not have discovered in a search using known keywords.

Given that the same term could be relevant to many different subject areas, there are potentially many paths to the same data, allowing for expansive data discovery. Imagine searching for a brand of merlot red wine and then being presented a selection of foods that go best with that variety. That is the power of a taxonomy classification based search!

## Enter ColdFusion and XML

XML is the emerging standard for defining taxonomies. Many of the currently available tools for creating a taxonomy specification provide XML export functionality. This is good news for ColdFusion developers, who already have a collection of functions available for working with XML.

In an XML definition, each term in the taxonomy is an element with its own collection of attributes and sub-elements. A standard definition will include tag markup for each type of relationship that can be represented. For most terms this will include “narrower term” and “broader term” tags, indicating the term’s hierarchical position in a given context. In more advanced systems, XML elements would also be added to represent the nonhierarchical relationships. A sample XML specification is shown in Listing 1.

Although XML is widely used as the language for taxonomy definition, an authoritative-format standard for this definition is still pending. Given this, it is best to make the implementation as flexible as possible, allowing for future attributes and term relationship types to be added easily with little to no refactoring. Ideally, an accepted Document Type Definition (DTD) will be created that allows for the validation of the XML. Until then, it is possible to implement custom validation that uses XMLSearch() with an XPath expression to validate the required XML elements. Organizations may also want to consider creating their own DTD to be used for the validation.

Given a plain XML definition, it becomes trivial to use ColdFusion’s XMLParse() to load the definition and create the XML object in memory. Once the XML object is obtained, an XPath expression can be used with ColdFusion’s XMLSearch() to extract the relationships. Depending on the criteria specified in the XPath expression, it is possible to process a single specified term or the entire taxonomy at once. (See ***CFDJ***, Vol. 4, issue 4 for an excellent article on parsing XML.)

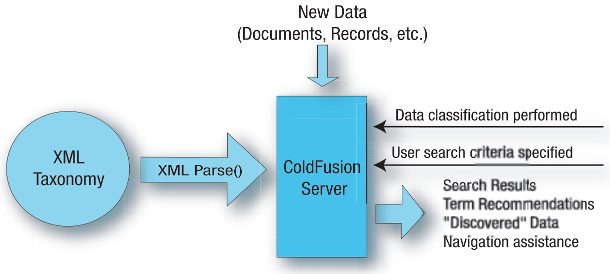

**Figure 1: Process flow for a taxonomy search**

## Can ColdFusion Handle It?

Organizations that are most likely to implement a taxonomy classification system are those with high volumes of digital data. Given the large amount of data, it becomes important to keep performance in mind when designing the system. The two main factors that effect scalability are the number of terms in the taxonomy and the amount of data associated with those terms.

A taxonomy with 20,000 or more terms is generally considered large. Parsing the XML and storing the terms into memory are potentially intense processes. With ColdFusion, however, tests using a 100,000 term taxonomy on a mid-powered server resulted in load times of only a few seconds. This included the XMLParse() call to create the XML object, the XMLSearch() call to retrieve the terms, and multiple assignment calls to create an associative array of the terms along with the defined relationships. An additional step to perform the validation added only a marginal increase to the total processing time.

Even though a taxonomy will typically be limited to several thousand terms, there may be millions of data records associated with those terms. Once finalized, the size of the taxonomy definition tends to stay more or less fixed, unlike the data count of an active system, which will see continued growth. Even though reading the data associations from memory is fast, the memory consumption could become excessive as the system ages. It is usually safe to load the entire taxonomy into memory, because even a large classification will make only a modest dent in memory consumption. This is not true with the actual system data in which the typical design tradeoff between speed and memory must be considered.

To avoid scalability problems, you can rely on a simple CFQUERY database call to retrieve the associations given to a specific term. For further improvements, commonly referenced terms and their respective data associations can be cached for fast lookup. See Figure 1 for a basic process flow starting with the initial XML import, and concluding with the user obtaining results based on the specified criteria.

## Future Development

The use of a taxonomy classification system for digital data is still relatively new. Over the past five years there has been much progress. However, work is still needed before there is a widely accepted vocabulary and common understanding of the framework and concepts.

One of the biggest challenges for an organization that wants to implement a taxonomy classification is the time and effort involved in creating the definition specification. Currently there are some commercially available definitions, but these are offered only in a limited number of business areas. Organizations that already have an institutional thesaurus are well positioned to use taxonomy-based classification. A thesaurus is often a precursor to a taxonomy, and the terms and vocabulary used to create a thesaurus are easily transferable. The National Information Standards Organization (NISO) has published guidelines for the construction of a Monolingual Thesauri, which is available in the ANSI Z39 specification. This is a good starting point for those exploring the possibility of implementing such a system.

Another implementation challenge is ensuring that data is classified correctly. There are auto-classification tools available that attempt to derive data context by using natural-language algorithms. These tools attempt to “understand” the content of the given data by evaluating not just the keywords, but also the circumstance. Once attained, the tools will assign the data to the proper term in the taxonomy. The accuracy of these tools won't match human classification but could be acceptable especially if the data is already tagged using some form of metadata.

## Gaining Steam...

The idea of using a taxonomy to organize and classify data is not new. In fact, the term "taxonomy" comes from biology in reference to the classification of living things. Applying this idea to vast stores of digital content, however, is a practice that has only recently gained steam. As digital content repositories grow, finding your target data quickly and accurately will seem more like finding the proverbial needle in a haystack. A taxonomy classification system is an excellent complement to a traditional keyword search and will help users efficiently find the data they need. Listing 1  A sample taxonomy XML specification for classifying wines by type and region .

### About the Author

*David Athey is a senior developer with PaperThin Inc. based in Quincy, MA. He is an Advanced Certified ColdFusion Developer with expertise in Enterprise Content Management and Web-based publishing.*

*dathey@paperthin.com*

```
<taxonomy version="1.0">
  <name>An XML Taxonomy Specification</name>

  <!-- Define a topic area -->
  <facet>
    <name>Wine Varieties</name>
  </facet>

  <term>
    <name>Red Wine</name>

    <!-- Additional term properties -->
    <annotation type="sn">Red wine is good for cooking</annotation>

    <!-- Narrower term designations -->
    <nt>Pinot Noir</nt>
    <nt>Merlot</nt>
    <nt>Cabernet Sauvignon</nt>
  </term>

  <!-- Define a topic area -->
  <facet>
    <name>Geographic Regions</name>
  </facet>

  <term>
    <name>Italy</name>

    <!-- Additional term properties -->
    <npt type="synonym">Italian</npt>

    <!-- Narrower term designations -->
    <nt>Tuscany</nt>
    <nt>Piedmont</nt>
    <nt>Adige</nt>
  </term>
</taxonomy>
```

**Download the Code...**
Go to www.coldfusionjournal.com