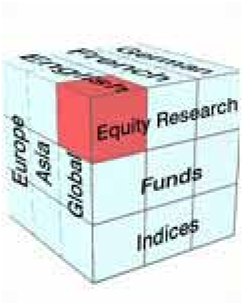
Localization in the context of taxonomies

When you need to localize and categorize

by Christian Donner
13-Mar-2006



Internationalization -- or "I18N", a very geeky abbreviation referring to the number of letters left out -- is commonly defined as a set of practices intended to make software more "localizable" by introducing layers of abstraction in the code and the data of an application. That way, it is easier to later modify the language, currency, date, and number format according to the requirements of a specific locale.

Localization -- or "L10N" -- describes a set of complementary practices used to create the translations and other things specific to a locale, so that information is presented to the user in a format that he can understand.

Today, most programming languages and application frameworks provide ample functionality to accomplish both. Because the technology has been available for many years, we tend to think that Localization is merely a mechanical step on the way to successful completion of an international software project.

Unfortunately, this is not at all true for Web Content Management System implementations, regardless of the vendor and platform. This is because locale information is not isolated, but becomes part of the overall content categorization, or taxonomy. Before you can localize, you must understand the relationships between all the components of the taxonomy, and design your information structures accordingly. Let’s take a look at some of these relationships and dependencies.

I18N as we know it

Internationalization -- the way it is commonly implemented in frameworks, platforms, and ultimately in specific CMS products -- provides only 2 degrees of freedom: *language* and *country*.

The Java platform, for instance, defines locale identifiers that combine a country code and a language code. The locale identifier for the US would be 'us_EN'. 'US' is the country code as defined by ISO standard 639, 'EN' is the language code as defined by ISO 3166. This duality stems from the correct observation that in some countries people speak multiple languages, resulting in multiple locales for a country. For instance, applications localized for Canada generally have to display currency amounts in Canadian dollars, but support both the French and the English language. The corresponding Java locales would be 'ca_EN' and 'ca_FR'.

Why does it not simply suffice to slap the locale code on to your content assets, and be done with it? It actually may, but only in the most simple of all situations. It depends on your localization taxonomy.

Translators and localization experts are essential to any localization project because they bring experience in translating and localizing the content itself – the complexity of which is almost always underestimated. However, what they cannot do for you is provide an information architecture that will make the localized content accessible for the right audience in the right region. At this intersection of Localization and Information Architecture, international enterprises often encounter a void.

Prior to a global CMS implementation, many organizations have not given much consideration to how their markets, languages, and audiences are related, and the project team may not have anyone readily available who can define these relationships. It takes a dedicated effort to analyze, document, and understand them. This effort must be the first thing that you tackle, because everything else depends on it – including the remainder of this article.

Taxonomies

A taxonomy provides conceptual structure and enhances clean navigation design. The value of a CMS increases dramatically with the introduction of a content taxonomy. The taxonomy defines the domains of the content's metadata and the valid combinations of all metadata "dimensions" (what some might call "facets"). *Industry*, for instance, is a common taxonomy dimension. By defining a set of industry codes, or by using standardized SIC codes, you define the range of values that the industry identifier of your content assets can take.

Geography is another dimension that is likely present in most taxonomies. A geographical dimension can be implemented as a set of *country codes*, for instance. Country codes? Wait a minute, you may say, what about the *locale* identifier that we introduced earlier, does it not already contain a country code? Isn't there a potential for conflicting information?

A Content Localization Taxonomy

You are right, of course, there is a problem, and by now you probably start to understand its complexity. It can be valid to have two different *country* codes in a content asset's metadata, because they can mean different things. In fact, it can be valid to have more than two. To find out how many country codes you need, you must develop your own localization taxonomy. Most likely, yours will be a subset of the following "reference" taxonomy.

In the reference taxonomy, every piece of content can be fully described by three geographical dimensions:

- What region is it from?
- What region is it about?
- What region is it for?

In addition, there is a 4th localization dimension that defines the *language*. Note that the dimensions are about *regions*, not *countries*.

Let's consider a hypothetical company that produces financial research and that has a presence in all major markets around the globe. The company maintains an international Web site and visitors must select their region of interest from a list.

It is easy to see that an article in German can originate from the Frankfurt office, discuss a topic that is relevant for the German financial markets, and be shown to German site visitors. All four dimensions would be set to 'Germany', or 'German'. The article in the example below is actually in the English language – our hypothetical company's taxonomy allows this.

By using the standard localization baked into Java-based CMS tools, this would not have been possible (applications and platforms do not define resources for a 'de_EN' locale).

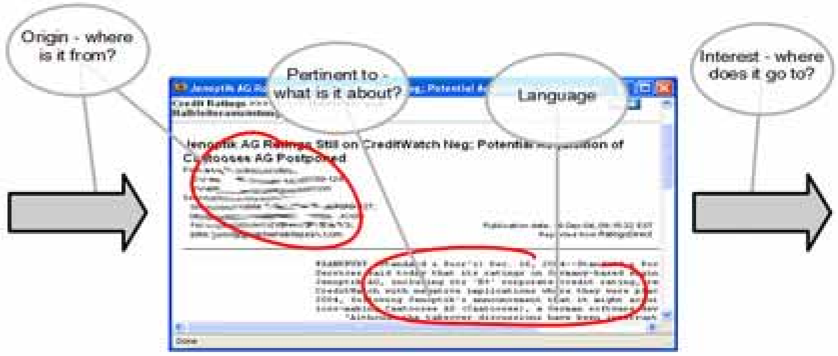

Origin - where is it from?
Pertinent to - what is it about?
Language
Interest - where does it go to?
Jenoptik AG Ratings Still on CreditWatch Neg; Potential Acquisition of Caatoosee AG Postponed

Figure 1 – A generic taxonomy for international content

It is just as conceivable that an article originates from the Tokyo office, discusses the European financial markets, and is intended for a world-wide audience.

What's in a region

This last example is loaded, and I hope that by now you are prepared for the implications. First, the article is about the *region* 'Europe', which is not a *country*. What does that mean? Well, it should not mean that a contributor in this organization has to click on all the countries that make up the European continent on her data entry screen. It means that when creating your localization taxonomy, you need to think about your markets and not about geographic boundaries. Define regions that are meaningful in the context of your content and your audience.

Second, the article is intended for a "world-wide audience," which is not a *country*, either. Once we realize that there is indeed a global financial market that is not simply the sum of all its regional markets, it becomes clear that "Global" must be a separate and independent *region*. Regions can, but don't have to, roll up into other regions. Other parts of your taxonomy can be hierarchical. Maybe you need information at the country level, and your countries roll up into regions.

Advanced localization taxonomies

The requirement that certain content should not be available in all markets, or that content-authoring regions can submit content only to a subset of the available markets, introduces us to the next level of taxonomy design. Maybe there is a *product* dimension in the taxonomy, and our firm does not offer Risk Solution products for the Asian markets, but everywhere else (see figure 2). This selection should therefore not be available on the tagging screen for contributors submitting content that is shown to Asian site visitors. This means that now there is a need for a multitude of regional taxonomies. They are a lot of work to create, but can rarely be avoided when business requirements mandate such regional differences.

Close this window

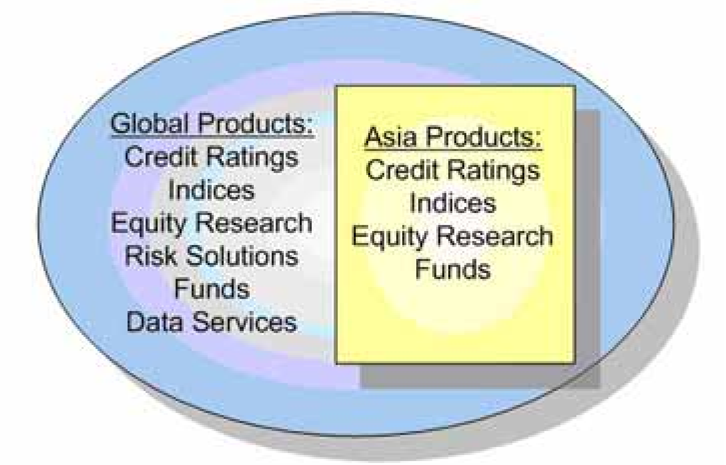


Figure 2 – A local taxonomy

Defining your taxonomy

Start with the generic approach from above and narrow it down to your needs. Is it necessary to know what *region* an article came from? If not, you can drop the *origin* dimension. Does your website use the concept of "region" to target content, not so much to localize content? In other words, when I come to your website and select a region, will that determine what content I see (vs. how content is presented)? If so, you may be able to combine the two remaining dimensions to one. Does your business have regional dependencies that will affect other taxonomy dimensions? You may need to create regional product taxonomies.

Implementation considerations

Multidimensional taxonomies can be complex to visualize, let alone implement. Figure 3 shows a simple taxonomy that has 3 dimensions – *geography*, *product*, and *language*. Each dimension has 3 potential values. The highlighted cube in the example represents Global content about Equity Research in English. In real-life situations, it is likely that not all cubes in a taxonomy are defined; moreover, enterprise taxonomies may contain more than 3 dimensions,

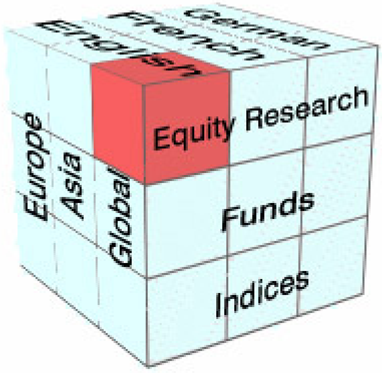


Figure 3 – Dimensions of a taxonomy

Because of the complex dependencies between taxonomy dimensions, and because of the potentially large number of values in a dimension set, it can be challenging to build a user interface within the realms of your global CMS that will let you manage your content effectively. For instance, you may have an article that applies to two regions and discusses a product that is only available in one of the regions. Will you allow the user to label the article with an invalid product/region combination? How will a user select multiple values from a large set?

A multiple-selection drop-down list is probably not an ideal solution here. But drop-down lists is about the extent of what you get out-of-the box from most CMS products. If you need an intelligent user interface that presents only valid combinations of metadata values and that writes the taxonomy metadata to a relational database or XML repository with true many-to-many associations, you have to build a metadata maintenance application and integrate it with your CMS management interface.

Summary

The creation of a localization taxonomy can become a significant piece of an entire CMS implementation project, particularly when your regional offices are in control of their local taxonomies and want to serve local customers in the best way. As you have seen, the concepts available for simple application localization are insufficient for the localization of complex international content. To get it right, you must be prepared for a substantial amount of analysis and the price tag that comes with it.

Next:

Send Feedback

See all CMS Channel feature articles.

Need to select a technology vendor, but confused about your choices? See our vendor-neutral technology reports.

Join the conversation via Technorati.

About the Author

Christian Donner is a Senior Technical Architect at Molecular in Watertown, MA. He helps clients define and build strategic Content Management and Business Intelligence solutions. Molecular, Inc, is a technology consulting firm that has been designing and building Internet-based solutions since 1994.