Cross-Cultural and Cross-Lingual Ontology Engineering

Yuri A. Tijerino

Web Science Lab, Department of Applied Informatics,
School of Policy Studies, Kwansei Gakuin University
2-1 Gakuen, Sanda, Hyogo, Japan 669-1337
ontologist-at-kwansei.ac.jp
http://www.websciencelab.com/

Abstract. Just as internationalization [i18n] and localization [L10n] do not simply consist of translation of interface components, but also of careful cross-cultural and cross-functional considerations to the cultural sensitivities of the intended source and target languages, similar considerations should be given to those issues in the realization of Semantic Web systems. Specifically, this paper describes issues that transcend linguistic and cultural aspects that affect the functional implementation of cross-cultural and cross-lingual semantic web services. The paper describes lexical and semantic aspects of ontology design that need to be considered for ontology-based applications that cross cultural and/or linguistic boundaries. The paper places special emphasis on ontology engineering issues related to intensional and extensional ontological definition. It proposes an engineering framework that grounds intensional definitions tightly with cultural aspects, while grounding extensional definitions more closely with linguistic aspects of ontology engineering, specifically as it pertains to design, mapping and querying.

Keywords: Ontology localization, ontology internationalization, cross-lingual ontology, cross-cultural ontology, ontology engineering.

1 Introduction

According to Guarino [1], ontologies are language dependent, while conceptualizations are language independent. Perhaps the reason for this is that while conceptualizations refer to abstract, simplified views of the world– within a given context–, an ontology requires formal specification of the underlying conceptualization [2]. Since formal specification of conceptualizations must occur in a given language, using a specific representational vocabulary and declarative formalism, it is clear that ontologies must therefore be language dependent. While conceptualizations occur at the abstract level of thought without particular emphasis on formal, declarative definitions. Thus, based on these landmark definitions of ontology, we can surmise that conceptualizations are invariably language independent and that ontologies are unavoidably language dependent. But what

about cultural dependency? Are conceptualizations cultural dependent? If so, are ontologies also culturally dependent? Or, can either or both remain culturally independent? These are some of the issues explored in this paper.

In one of the most common application of ontologies nowadays, the realization of the Semantic Web, cultural and linguistic dependency becomes a crucial issue for practical reasons, as pertaining to internationalization [i18n] and localization [L10n] of Semantic Web systems. Although in the context of the Semantic Web, ontologies are in essence intended to facilitate understanding and interoperability by and within software agents, ultimately and inevitably human users must interact at some level with those agents. As the Web audience increases rapidly and crosses ever more cultural and linguistic borders it has become urgent to define best practices for the engineering of cross-cultural and cross-lingual [C3L] ontologies, or approaches for C3L mappings. In other words, best practices and approaches related to ontology i18n and L10n. This urgency becomes self-evident as we realize that as of June 2010, 72.6% of the almost 2 billion Web users–the ultimate audience for the Semantic web–, speak a language other than English [3]. While Cimiano *et al.* have already identified many of the important issues related to cross-lingual ontology [4], This paper makes an renewed attempt to more broadly describe issues related to ontology i18n and L10n, while taking into consideration relevant cultural as well as linguistic aspects.

In Section 2, the paper describe common practices in software engineering i18n/L10n that are relevant to ontology i18n/L10n. Section 3, discusses the basic premises for intensional versus extensional definitions and how they apply to ontology i18n/L10n. Then, Section 4 introduces 4 basic guidelines for ontology i18n/L10n, while contrasting it to the common practices introduced in Section 2. Section 5 provides final remarks and gives directions for further research in C3L ontology engineering.

2 Common software engineering i18n/L10n best practices

In general, software i18n/L10n consists of the following major considerations in software engineering or re-engineering:

1. *Lexical externalization*: pertains to engineering the software so that no text visible to the users is directly embedded into the code. Instead, text is associated to language variables and placed in language resources files commonly referred to as locale files, which can more easily be translated to other languages or locales as necessary. Lexical externalization also involves using an universal character code set, such as Unicode, that can support input and output of any language.
2. *Grammatical externalization*: deals with avoiding the use of any language-specific grammatical constructs. For example, avoiding concatenations of strings to generate sentences, a common practice that, even if lexical externalization has already taken place, can cause problems during L10n efforts.
3. *UI externalization*: is related to various input and output aspects of the user interface. For one, it is important to allow input of international data

and foreign scripts. In addition, it is important to externalize all styling and formatting onto style sheets, because style and formatting are script-dependent. Other aspects of the UI, such as color schemes and graphics need also be considered for externalization as in some cases they might incorporate cultural aspects that need to be localized.

4. *Functional externalization*: common functional externalizations consists of using system functions to format dates, which may be formatted differently even in the same language. Other common functional externalizations that are language-dependent in nature, include sorting and script comparison.

3 Intensional versus extensional definitions in ontology i18n/L10n

Before we dig deeper into ontology i18n/L10n, it is worth reconsidering the formal descriptions provided by Guarino [1] for intensional definitions versus extensional definitions and how intensional definitions fit more closely with the intended conceptualization of a domain.

On one hand, intensional definitions should capture the intended meaning of concept and relations necessary and sufficient to define a domain. However, the moment those definitions are formalized with a particular language, whether with a formal language or a natural language, the formalized definitions become language-dependent. Nevertheless, some formal languages such as predicate calculus and description logics can do a good job in generalizing the description enough to make the ontology "less" natural language dependent, while still capturing, to great extent, the intensional definition of the conceptualization, because these intensional definitions provide clear and agreeable meanings to the concepts and relations required in order to define the conceptualization formally by unequivocally giving the necessary and sufficient conditions that must be met for the intended meaning to be true. For example, a loose intensional definition of *a live person*[1] is a human being who has been born and is not dead. Being a human being AND being born AND not being dead, are all necessary properties of something referred to as *a live person*. Failure to meet any of those properties would disqualify that something from being *a live person*. Also, these are sufficient conditions since anything that is a human being, has been born and is not dead, is a live person, whether it has additional properties, or not, such as it also has a name and is a male, which do not fall within the necessary and sufficient conditions of being *a live person*.

On the other hand, extensional definitions rely on listing all possible things that are realizations of the conceptualization or its components. For example, we could extensionally define *a live person* by listing ALL human beings who have been born and are not dead. Although, in this case this definition would prove impractical, it can be very useful for conceptualizations of things that have manageable-size listings. For example, we can arrive to an extensional definition

[1] In this paper, labels for conceptualizations are represented in *italics* font.

of the country called *The United States of America* by simply listing all the 50 states, territories and other dependencies that compose that country.

Based on this, it seems that ontologies that use intensional definitions for most of its conceptualizations might be better candidates for ontology i18n/L10n, than ontologies that use more extensional definitions. But what about cultural-dependencies?

Several examples of a cultural dependency in ontology specifications come to mind from work on a recipe ontology we defined a few years ago for translating English recipes to Japanese and vice versa [5]. At first it would seem that a recipe ontology should work for either Japanese and English. After all, in its simplest form, a recipe consists of a list of ingredients, their amount thereof, and a set of instructions on how to combine the ingredients to prepare the intended dish [2]. First, we developed independent ontologies that best fit existing recipe resources in Japanese and in English. When we tried to map equivalent conceptualizations between both ontologies, we run into problems that were caused by language and cultural dependencies of the underlaying conceptualizations of the resources used. An example of a language dependency was in the conceptualizations of some of the units of measurement, not simply the difference between the system and the English systems, but in more subtle measurements like *a cup*, which although very similar conceptually as a unit of measure are not exactly equivalent. As it turns out, a Japanese cup is approximately 0.8 the size of a English cup. Once we realized this and made the necessary adjustments to the ontology to define a cup with more lower-level intensional definitions, which used common units of measurement, the problem could be easily solved. However, this was not the case when we came across some of the conceptualizations about the ingredients themselves, which tend to be more culturally biased. For example, chicken recipes corresponded nicely from the Japanese ontology with the English counterpart, but problems surfaced when attempting to use ingredients that simply did not exist on the counterpart ontology. For instance, turkey is an ingredient commonly found in traditional English-language recipes. But since turkeys are indigenous to the Americas and almost not-existing in Japan[3] it was necessary to make substitutions with other ingredients like *large chicken* or *pheasants*. Although making substitutions of ingredients of such importance in a recipe might not always be practical, they are not uncommon. As an illustration, American expatriates have been known to substitute *the bird*, an American colloquialism for turkey, for chicken or pheasant. Particularly in Japan, where it is not only hard to acquire turkeys, but difficult to find an oven large enough to prepare it. It is important to realize that normally when substitutions such as this are made, they are not random. In most cases, it seems more appropriate, in the intensional sense, to substitute *the bird* with something related, such as chicken or pheasant, rather than with something more unrelated, such as octopus

[2] Notice that this is a loose intensional definition of the conceptualization of recipe, which is culturally independent.

[3] In Japan, a few specialty stores that cater to the expatriate community stock turkeys on a seasonal basis.

or potato chips. Thus, it is reasonable to mistakenly conclude that cultural dependencies in conceptualizations can be resolved through the use of intensional definitions that subsume the target conceptualized things found to be culturally dependent. In this case, poultry, a less culturally-dependent conceptualization, subsumes the conceptualizations of turkey, chicken and pheasant. Alternatively, and more practically we might conclude that cultural dependencies in conceptualizations should be resolved through the use of subsuming extensional definitions with high-degree of overlap with respect to the target conceptualized things found to be culturally dependent. In this case, turkey, chicken and pheasant might have been defined as extensions of conceptualization of poultry in the English-language recipe ontology, while *niwatori* and *kiji*, the respective equivalents of chicken and pheasant in the Japanese-language ontology, are extensions of the conceptualization for *torirui*, the equivalent of poultry.

The ontology purist might argue that this is not a cultural dependency issue in the ontological sense, but a problem of incompleteness, that can be resolved by "fixing" the Japanese-language ontology or its underlying conceptualization, by inclusion of intensional or extensional definitions for turkey. However, this can prove impractical since neither are turkeys common in Japan, nor are most cooking implements such as American-sized ovens, smokers, or deep-fry pots. Thus it is more practical to resort to ingredient substitution, through either of the methods described in the previous paragraph.

Based on this discussion, let's now describe a set of best practices for i18n ontology engineering.

4 Recommendations for ontology i18n/L10n

In a way, externalizations, as described in section 2, can be thought of as special kinds of extensional definitions. Particularly in the case of lexical externalizations, this holds true because the language variables are really data containers for a specific extensional definition of a conceptualization, which is defined by all the possible lexical variations of the conceptualized thing given by the listing of all locale variation associated with the conceptualized thing. For example, the language variable *username* conceptualizes the unique identifier for a user, while the associated locales, such as `(English:‘‘user name’’)`[4], `(Japanese:‘‘yu-za ne-mu’’)`, `(Spanish:‘‘nombre del usuario’’)`, provide the listing for the extensional definition of the conceptualized thing, that is, the *username*. With this in mind, based on cumulative experience in mapping ontologies across languages, taking into consideration the cultural aspects as well as the language aspects, and borrowing from common software engineering i18n/L10n best practices described in 2, the following recommendations come to mind for i18n/L10n in ontology engineering.

[4] Notice that `(English:‘‘user name’’)` represents pseudo-code, thus the `courier` font, for a localized string, where `English` represents the locale and `‘‘user name’’` is the actual localized string.

1. *Lexical definitions are best represented with externalized extensional definitions:*
   Lexical definitions, which occur at the lexical layer [4], usually consists of declaration of the labels associated with the conceptualized thing being described for a particular language. Analogously to lexical externalization in software engineering i18n, it is a good idea to externalize lexical definitions so that they can be localized to other languages without major changes to the ontology itself. In reality, lexical definitions are deeply internalized in most ontologies, and just as in non-internationalized software, it is costly and time consuming to attempt externalizing those definitions. This problem is closely associated to the debate of when it is appropriate to declare something as a concept, an instance or a relation. Noy and MacGuiness [6] refer to this a problem of granularity and leave the decision up to the ontology engineer claiming that it depends on the application of the ontology. Although Nagypal provides a more specific method, based on whether something is *a kind of X* or not, to decide whether to make something a new concept or an instance, this distinction is not always so clear [7]. Since so far most ontologies were designed with no i18n/L10n in mind, these loose methods were practical for most ontology applications, however, in the case of the Semantic Web, which needs to work across multiple languages, a more specific methodology is required.
   What we need is an externalized extensional definitions, which as the name implies, are extensional definitions that can be externalized from the ontology specification. Data Frames, introduced by Embley [8], which have been used successfully for data extraction ontologies for data extraction from data-rich unstructured documents [9], data extraction from the Web [11], formation of database queries from natural language [12], and ontology generation from tables [13] among others.
   Externalization of extensional definitions through data frames, in its simplest form, consists of listing the possible lexical members of a conceptualized thing. In its more complex form, it consists of describing what those lexical members might "look like", by generalizing possible variations, describing position within a document, identifying what other related lexical terms might be close to it, and other relevant lexical properties. Figure 1 shows an example of what an externalization of an extensional definition for the conceptualization of *Price* might look like in Web documents or in human queries in the English language. Figure 2, shows the actual language locale file for English, where those externalizations are stored. Basically, we can now create one of these files per language we want to support with the ontology. More recently, we successfully applied this approach for formation of structured queries from English, Japanese and Chinese to query monolingual web services [14]. This was possible, because we externalized the extensional definitions using the data frame approach described above.
2. *Semantic declarations are best represented as intensional definitions:*
   As opposed to software engineering, where grammatical externalization is recommended to avoid language dependency, in ontology i18n it is recom-

```
Price
   internal representation: integer
   external representation: getExtension(locale,price-external-representation)
   context keywords: getExtension(locale,price-context-keywords)
   ...
   LessThan(p1: Price, p2: Price) returns (Boolean)
   context keywords: getExtension(locale,lessThan-context-keywords)
   ...
```

Fig. 1. Example of an externalized lexical definition for the conceptualization of *Price* in Web documents.

```
%%Locale file for English
price-external-representation = \$[1-9]\d{0,2},?\d{3} | \d?\d [Gg]rand | ...;
price-context-keywords = price|asking|obo|neg(\.|otiable)| ...;
   ...
lessThan-context-keywords = (less than | < | under | ...)\s*{p2} | ...
   ...
```

Fig. 2. Example of the contents of an externalized language locale file for English.

mended that semantic declarations should, in most cases, take the form of intensional definitions. The reasons for this should be clear by now. An ontology is an specification of a conceptualization [2] and since conceptualizations are language-independent [1], it only makes sense that semantic declarations should be closer to the conceptualization layer [4]. In the Semantic Web, in particular, semantic declarations are intended for machine processing, although in some cases they might help humans understand the conceptualizations of the underlying domain. In either case, semantic declarations, which consist of formal declaration of semantics through formal languages such as KIF, Ontolingua or OWL. It is possible to make lexical declarations with these language, too, but as a good ontology i18n practice, these formal languages should be used for intensional descriptions of the entities properties, entity interrelations and non-lexical entities within the conceptualization. If, for some reason, it is necessary to provide extensional semantic declarations, these should be externalized to allow efficient ontology L10n.

3. *Cultural-relevant aspects are best represented as extensional definitions:* Based on the discussion in Section 3, when cultural dependencies in conceptualizations can be identified, they should be defined extensionally. This might not always be possible due to some unavoidable cultural biases that occur during conceptualization, but that can be avoided to some extent by consciously describing the conceptualization with at least more than one culture in mind. In particular, our approach to extended extensional definitions through data frames, which provide templates for such extensional definitions, might provide clues as to what might be considered cultural-relevant aspects of the conceptualization.
   Particularly, there are conceptualizations that, with little care, can be identified as culturally biased. One exemplar conceptualization is that of *address*,

which varies from country to country. Thus, defining an *address* in ontology as being composed of *street-number*, *street-name*, *street-postfix*, *unit-number*, *city*, *county*, *state* and *zip-code*, is culturally-biased to the United States of America. However, conceptualizations for *address* are very different in other countries, for instance, in Japan, where addresses do not have equivalent conceptualizations to most of these American-biased ones. Although, most cultures do share the conceptualization of address, perhaps as benefit of their postal services, it is a mistake to assume that the aggregate conceptualizations for address must, therefore, be the equivalent. The answer is to define address extensionally and externalize the definition so that it can be effectively localized. Again, our framework, based on data frames, provides a firm candidate for these externalizations. Figures 3 and 4, show an example of how to externalize these cultural-dependencies for addresses.

```
Address
   internal representation: array
   external representation: getExtension2Array(locale,address-external-representation)
   context keywords: getExtension(locale,address-context-keywords)
   ...
```

Fig. 3. Example of an externalized cultural-dependent definition for the conceptualization of *Address* for multi-lingual Semantic Web applications.

```
%% Locale file for US-English
address-external-representation = [street city state zip-code]
address-context-keywords = (address | domicile | place ...)
street = regex for US street address goes here!
city = ([a-zA-Z]+|[a-zA-Z]+\s[a-zA-Z]+)$
state = getListFromLocaleFile(US-English,stateList)
Zipcode = ((\d{5}-\d{4})|(\d{5})$
   ...
%% Locale file for Japanese
address-external-representation = [yubinbango, fu-ken-to,shi-ku-gun, cho-mura,chome-ban, banchi]
yubinbango = ((\d{3}-\d{4})$
fu-ken-to = getListFromLocaleFile(Japanese,fuKenToList)
shi-ku-gun = getListFromLocaleFile(Japanese,shiKuGunList)
cho-mura = getListFromLocaleFile(Japanese,choMuraList)
chome-ban = Regex goes here
banchi = Regex goes here
...
```

Fig. 4. Example of an externalized lexical definition for the conceptualization of *Address* for multi-lingual Semantic Web applications.

4. *Functional aspects are best represented as extensional definitions:*

Functional-aspects of ontological design, are also an important aspects that need to be considered and are closely related to cultural aspects of the conceptualization. Functional aspects of ontology design come into play when functional transformations of data of some kind or another needs to be embedded in the ontology itself. For example, a conceptualization for *location* might take multiple culturally-biased forms. In the example above, the conceptualization of *address* was identified as culturally-biased. For the purpose of this discussion, let's assume that *location*, is intensionally defined as the global coordinates on a mapping service. Let's assume further, that we did not have existing services such as **Google Maps**, **Yahoo Maps**, **Bing Maps** or other mapping web service, which currently offer global address to coordinates translation. For this kind of problem, it becomes necessary to externalize this functional aspect so that it can be localized accordingly. Another more realistic example comes from the recipe ontology introduced earlier, where measures and their conversions were also culturally dependent. For example the measure of cup in the Japanese system was different from the measure of cup in the English-language system. Figure 5, shows an example of how this kind of functional definitions can be externalized.

```
Cup
   internal representation: string
   ...
   convert2Locale(c1: cup1, locale) returns (Real)
   ...
```

Fig. 5. Example of an externalized lexical definition for the conceptualization of *Address* for multi-lingual Semantic Web applications.

5 Final Remarks

Section 4, introduced the premises for an ontology i18n/L10n framework, which is based on re-purposing data-extraction ontologies as described by Embley [8–11]. The framework was originally tested with a data-extraction ontology that enabled cross-lingual querying of car ad Web sources either in English in Japanese [15]. The Web sources were originally in English and the ontology was used for parsing the queries, which could be made either in English or Japanese. The framework was then independently refined for querying a Japanese restaurant web service, through a restaurant ontology, for which the extensional specifications for the conceptualizations were externalized into data-frames and populated with Japanese, English and Chinese locales. [14]. Again, the ontology can be used to parse the queries in those languages to generate web service queries to query the Japanese web service. Using this framework, we can further expand the languages supported by the restaurant ontology, by simply creating and populating the necessary locale files.

References

1. N. Guarino, "Formal ontologies and information systems," in *Proceedings of the First International Conference on Formal Ontology in Information Systems (FOIS98)* (N. Guarino, ed.), (Trento, Italy), pp. 3–15, June 1998.
2. T. R. Gruber, "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
3. "Top ten languages used in the web." `http://www.internetworldstats.com/stats7.htm`, June 2010.
4. P. Cimiano, E. Montiel-Ponsoda, P. Buitelaar, M. Espinoza, and A. Gómez-Pérez, "A note on ontology localization," *Appl. Ontol.*, vol. 5, no. 2, pp. 127–137, 2010.
5. M. Kimura, Y. Kitamura, M. Matsuda, and Y. Tijerino, "English-japanese cooking recipe translation system using ontology," *IEICE Technical Report: AI2007-41*, vol. 107, pp. 77–82, January 2008.
6. N. F. Noy and D. L. mcguinness, "Ontology development 101: A guide to creating your first ontology." Online, 2001. url: http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html.
7. G. Nagypal, "Methodology for building sws ontologies in dip." Online, 2005. url: http://www.eecs.iu-bremen.de/archive/bsc-2007/broecheler.pdf.
8. D. Embley, "Programming with data frames for everyday data items," in *Proceedings of the 1980 National Computer Conference*, (Anaheim, California), pp. 301–305, May 1980.
9. D. Embley, D. Campbell, S. Liddle, and R. Smith, "Ontology-based extraction and structuring of information from data-rich unstructured documents," in *Proceedings of the 7th International Conference on Information and Knowledge Management (CIKM'98)*, (Washington D.C.), pp. 52–59, November 1998.
10. D. Embley, D. Campbell, Y. Jiang, Y.-K. Ng, R. Smith, S. Liddle, and D. Quass, "A conceptual-modeling approach to extracting data from the web," in *Proceedings of the 17th International Conference on Conceptual Modeling (ER'98)*, (Singapore), pp. 78–91, November 1998.
11. D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y.-K. Ng, and R. Smith, "Conceptual-model-based data extraction from multiple-record web pages," *Data & Knowledge Engineering*, vol. 31, pp. 227–251, November 1999.
12. D. Embley and R. Kimbrell, "A scheme-driven natural language query translator," in *Proceedings of the 1985 ACM Computer Science Conference*, (New Orleans, Louisiana), pp. 292–297, March 1985.
13. Y. A. Tijerino, D. W. Embley, D. W. Lonsdale, Y. Ding, and G. Nagy, "Towards ontology generation from tables," *World Wide Web*, vol. 8, no. 3, pp. 261–285, 2005.
14. Z. Geng and Y. Tijerino, "Using cross-lingual data extraction ontology for web service interaction – for a restaurant web service," in *2010 Workshop on Cross-Cultural and Cross-Lingual Aspects of the Semantic Web*, (Shanghai, China), November 2010. Submitted.
15. D. Lonsdale, D. Embley, and S. Liddle, "Ontologies for multilingual extraction," in *Proceedings of 1st Workshop on the Multilingual Semantic Web*, (Raleigh, North Carolina, USA), pp. 1–4, April 2010.