Toward a Model for Language Identification
Defining an ontology of language-related categories*

Peter G. Constable,
SIL Non-Roman Script Initiative (NRSI)

1. Introduction

Global trends in international commerce and multilingual computing and the expanding horizon of the Internet have contributed to increasingly diverse needs within information technologies (IT) for systems of language identification.[1] On the one hand, users and implementers are working with data in a rapidly increasingly number of the world's languages. On the other hand, implementers need to be able to make distinctions within single languages for a variety of language-related parameters, such as regional variations in spelling or vocabulary.

It is not immediately clear, however, what an adequate solution to the overall needs should be like. In fact, the likelihood of finding a good solution can at times seem remote. Let us consider this in greater depth.

1.1 Current problems

To deal with the diverse language identification needs, people are looking to the ISO 639 family of standards, which provide over 400 different language identifiers. For those working with hundreds or thousands of less well-known languages, however, this number falls well short of what is needed. Similarly, these standards do not provide mechanisms that accommodate intra-language distinctions involving parameters such as script.

Some protocols have some ability to overcome the limitations in ISO 639 by making reference to the derivative standard provided in RFC 3066, which allows for the creation of tags that add additional qualifiers to the ISO 639 codes, or for the registration of entirely original identifiers. There are potential concerns with introducing a greatly expanded set of tags under the terms of

---

* The ideas presented here have germinated in discussions with many different people—more than I can now recall—who have, thus, contributed directly or indirectly in some way. I would like to mention at least those that I do recall: Carl Brown, David Dalby, Asmus Freytag, Håvard Hjulstad, Richard Ishida, Rick McGowan, Sue-Ellen Wright, and members of the Locales and IETF-Languages mailing lists. Particular thanks are due to Ken Whistler, who first prompted me to work on language identification issues, and to Gary Simons, with whom I have collaborated in most of this work and without whose input this work would not have been possible. Various ones also provided useful comments on drafts, for which I am also thankful. Any remaining shortcomings are, of course, my own.

[1] In this paper, "language identification" is used to mean the use of metadata attributes on information objects to indicate the language in which the content of the information object is expressed or, in the case of resources for linguistic processes such as spell checking, the language to which the resource applies.

RFC 3066, however, since it could quickly lead to considerable confusion, for reasons I will describe momentarily.

As I have interacted with numerous people on the topic of language identification in IT, I have encountered a variety of opinions about what we have and what is needed. Some—a minority, I believe—feel that needs can be met by introducing new tags as needed under the terms of RFC 3066, though most would probably welcome an expanded ISO standard. But many lack confidence in the standards process when it comes to language identification. In part, this is due to a fear that what would get standardised would not be what is actually needed, and that we could be worse off than before. Another factor (one that also contributes to the previous point) is that some in industry consider the existing ISO 639 set of identifiers to be poorly done and full of inconsistencies.[2] A further factor contributing to a lack of confidence is a concern that the standards process may attempt to provide solutions before the problems are really understood. Ultimately, there is a lack of confidence because the whole area of languages and language identification is seen to be inherently confusing and resistant to any consistent, analytical treatment.

The lack of confidence, then, is due in part to the problem area being confusing. The confusion in turn is due to various factors. One is that language as a social phenomenon presents itself to us as a complex network rather than as a set of discrete and well-defined entities. Another cause of confusion is the problem areas in the existing ISO 639 standards, mentioned above.

A further cause of confusion is problematic connections that have been made between "language" and "locale" in many implementations (see §3.1 below for further discussion). So, for example, one might create a "language" tag combining the language *Italian* and the country *Switzerland* and apply that in situations in which the only distinctions to be made are actually non-linguistic: number and date formats or the like.

Confusion is also arising as IT workers identify certain needs that break current mechanisms and implementations. As such situations are considered individually, it may not be difficult to imagine possible solutions for each, yet it is more difficult to anticipate how all of these might interact with one another, or with other unanticipated mechanisms that might be motivated by future needs. If a limited number of currently identified needs present a measure of uncertainty, the degree of uncertainty about a greatly expanded tagging system is rather greater.[3]

---

[2] There certainly are problem areas in the existing ISO 639 identifiers, most of which have been described in Constable and Simons 2002. It should be noted that, to a considerable extent, people in the IT sector find the existing ISO 639 identifiers, and especially those in ISO 639-2, to be problematic because they are evaluating them in terms of their usefulness for IT purposes in general. These standards were not developed with such a broad range of applications in mind, however. Rather, they were created for use in specific application areas in terminology and bibliographic use. Some might argue, therefore, that the perception of problems is due to misapplication of the standards to uses for which they were not intended. Nevertheless, the ISO committees responsible for these standards have recognised that problems exist, and that steps need to be taken in order to meet needs in a much broader range of applications.

[3] Some may object at this point that I have not actually illustrated that trying to create new tags will involve any confusion but have only made reference to hypothetical possibilities. I believe the possibilities are more than hypothetical, but it is difficult to demonstrate that here without the benefit of material covered in the following sections. Hopefully a suggestive illustration at this point will suffice.

I propose below that individual languages should be identified using only atomic identifiers, and that country codes should be used only for certain derivative types of category, such as orthographies. Now, certain tags have been registered under the terms of RFC 3066 that use country codes to identify individual signed languages; e.g. "sgn-US" to denote American Sign Language (ASL). ASL is written by a portion of the speaker community, but there is nothing close to a consensus within the community on writing ASL. This could potentially lead in the future to more than one writing system being in use, which would lead to a need

Certainly, such causes of confusion need to be addressed if overall solutions are to be found. As a result of the analysis of ISO 639 described in Constable and Simons 2002, problem areas in those standards are beginning to be understood and evaluated by the ISO committees responsible for the standards. Yet, other sources of confusion also need to be dealt with. We need to consider whether that is even possible.

1.2 The need for a model

Ultimately, I believe, most of the potential for confusion in language identification is due to this: the overall problem area has not been understood well enough to identify appropriate principles on which to base adequate solutions. "Language" tags have often been created or proposed without having first made clear exactly what specifically is being distinguished. Tags are devised as seem to fit some need, but this is done in a somewhat *ad-hoc* manner without any guiding principles regarding the *semantics* and *morphology* of tags.

For example, a distinction between Simplified and Traditional Chinese writing is usually expressed in terms of country codes for the People's Republic of China on the one hand and Taiwan on the other. Yet country is actually an orthogonal factor that may be completely irrelevant, and some users have needed to distinguish the writing systems while keeping country unspecified. There is even potential that a user might, for other reasons, need to include a reference to one or the other country that contradicts the conventional associations with regard to writing.

Restating the problem another way, there has never been any careful analysis regarding questions such as the following:

- For what *kinds* of language-related entities are distinctions needed for IT purposes?
- What various types of qualifiers are relevant for making distinctions between different language-related categories?
- For what kinds of distinctions is any given type of qualifier relevant?
- What kinds of interactions exist between qualifiers, and are there any constraints on how qualifiers should appropriately be combined?

In short, until now, "language" identification (and "locale" identification) has proceeded in the absence of a model that describes what kinds of problems they are intended to solve and that provides an analysis of the problem area as a whole. As mentioned above, solutions have been created before the problem was understood.

I would suggest, then, that any prevailing confusion regarding language identification is due primarily to this lack of a model of language-related categories. Furthermore, in spite of pessimistic views regarding the inherent intractability of the language problem, I suggest that an adequate model is possible, and that such a model can provide the clarity that is needed to find

---

for script qualifiers, something like "sgn-US-swri", perhaps. As suggested below, however, country codes are appropriate for distinguishing orthographies, and orthographies are narrower categories than writing systems, hence I propose that country codes may follow but should never precede script codes. Using country codes to identify individual languages breaks an otherwise consistent pattern, making parsing of tags more difficult. In addition, as ASL is written, orthographic variations may arise between different countries in which it is used. Imagine, then, 30 years in the future, that this could give rise to possibilities like "sgn-US-swri-CA".

Generalising, if we consider possible tagging needs for hundreds or thousands of new situations, each with its own quirks, the potential for confusion is certainly real if qualifiers are combined with no principles to guide how this is to be done.

solutions to current and future needs in the area of language identification. Indeed, such a guiding model is essential before any significant progress can be made in finding solutions to those needs.

1.3 In pursuit of a model

This paper is intended to explore what an adequate model of "language" identification should look like. In particular, it aims to describe the ontology for which "language" identifiers are needed; that is, the different *kinds* of language-related entities in the real world that are relevant for IT purposes, and the relationships between them. In view of this ontology, I will also attempt to derive implications for an adequate system of "language" identifiers to be used in IT applications.

By now, it should be somewhat apparent that, in the view presented here, we are dealing with multiple types of categories, all of which are related to language *per se* but some of which are also somehow different. In other words, not all of the distinctions for which we use "language" identifiers are between languages. Thus, in making reference to "language" identification, what is really meant is identification with regard to various types of language-related categories.

Lack of an adequate ontological model is not the only problem to be addressed in relation to systems of "language" identification. Constable and Simons (2000) discussed five problem areas, some of which have no relation to an ontological model. Among the key problems identified, however, were issues of operational definition. These issues would be addressed by an adequate ontological model. In that work, we suggested that a system of language identification should allow for different operational definitions of "language" as may be needed for different purposes. That need may be eliminated or diminished at least to some extent by identifying and providing operational definitions for category types other than "language".

It should be understood that this paper is intended as a starting point for discussion and development, not as a finished proposal. It is expected that others will find many ways in which refinements can be made in the model, and comments to that effect are welcomed.

There will inevitably be scenarios that can be raised to suggest significant flaws in the model. Given the continuous variability in linguistic phenomena in the world, this is not surprising. I would hope, however, that in on-going discussion it will be possible to distinguish between hypothetical possibilities and realistic, potential IT needs. We should also be willing to forego expectations that an adequate model will directly reflect every nuance of linguistic phenomena and accept instead a model that makes simplifications that may involve compromises to make it workable but yet is still adequate for IT needs.

2. Summary of applications

Before considering a model in any detail, it is helpful to review the general types of application for which "language" identifiers are used. This is but a partial classification of application types and not a comprehensive assessment of usage scenarios. Various considerations will be overlooked until later sections.

Application areas can probably be divided into two general types:

- cataloguing and retrieval of content, and
- resources for localisation and language enabling of software.

2.1 Cataloguing and retrieval of content

Cataloguing and retrieval of content according to linguistic properties is often important for multilingual repositories. Significant contexts for this include bibliographic use (libraries), and content on the Internet. More generally, it can apply to any repository containing multilingual content, including private repositories such as a linguist's database of comparative-linguistic data.

In some cases, a particular repository may have a given information object in one language only. In a university library, for example, the English holdings may include a book of poems by Longfellow, while the French holdings may include a philosophical treatise by Pascal. In such situations, we do not necessarily expect any overlap between results returned by queries that specify one language or another. In other cases, a repository may contain the same information repeated in multiple languages. So, for example, a company Web site may provide user support information in several languages. In these situations, queries that specify different languages are expected to return analogous results. The former set of cases is typical for libraries and similar contexts; the latter cases are typical of the localisation and translation sector within IT industries.

The level of granularity and detail used in cataloguing and retrieval can vary, the only requirements being what is considered acceptable for users in a given context. Thus, in a general-purpose library in North America, it would probably be considered acceptable to catalogue items under certain broad categories such as "Semitic Languages". On the other hand, for a university departmental library specialising in Ancient Near Eastern Studies, users would expect and require a finer level of differentiation.

Also, the levels of granularity and detail used in cataloguing do not necessarily have to match that used in retrieval. For instance, if a user queries for information in a Scandinavian language and the results include an object that has been catalogued as being expressed specifically in Icelandic, that will likely fit within the user's expectations. Note that the opposite is not generally true, however: if a user requests items in Icelandic and the results returned include items in a variety of Scandinavian languages, that will not, in general, be satisfactory.[4]

For cataloguing purposes, broad categories are generally appropriate only in libraries and similar contexts. They are not likely to be useful for cataloguing in localisation/translation scenarios. Generally, content is translated with fairly specific target audiences in mind, which usually implies relatively specific rather than vague linguistic properties. In deployment of content, for instance in creating a multilingual Web site, it will be necessary to allow for users requesting content with fairly specific language parameters, even if some users express requests in broad terms. This requires cataloguing in terms of narrow rather than broad categories. So, for example, if we create content in the Naskapi language and want users to be able to retrieve that content by asking specifically for that language, then we must tag that content specifically as "Naskapi" rather than in terms of a broader category such as "Algonquian". If it were catalogued in terms of the broader category, then it could only be retrieved in terms of that broad category. Yet, it is highly unlikely that requests for such content will only ever be requested in terms of broad language categories.

---

[4] This matches the behaviour specified for the Accept-Language request-header field of the HTTP protocol (RFC 2616) and the notion of language-range as defined in RFC 3066. Note that, in some situations, results that are broader than what is specified in the request may be considered acceptable as a fallback.

2.2 Resources for localisation and language enabling of software

A second general class of application types pertains to development of software for multiple language markets. This involves two types of task: user-interface localisation, and enabling for multilingual data.

Localisation of user-interface elements involves translation of user-interface strings so that they are expressed in a language suited to a particular target market. This application sub-type is essentially like the cataloguing/retrieval application type described above, and does not introduce any new factors to be considered. User interface strings must be translated for a particular target audience, and will generally be constrained in terms of linguistic parameters in ways that are suited to the target audience. They would then be catalogued in terms of comparably narrow language categories and retrieved in terms of equally-narrow or possibly-broader categories.

The more interesting application sub-type to consider pertains to enabling of software for multilingual data. This involves development of resources used in various types of linguistic data processing. There are many types of processes that this might include, such as the following: voice recognition (speech-to-semantics matching or speech-to-text conversion), speech synthesis, spell checking, grammar checking, semantic interpretation, morphological analysis, sorting, optical character recognition, and language-variety recognition (automated language detection).

It should be noted that resources for most linguistic data processing must be tailored for relatively specific language varieties. For example, it would not make sense to create a spelling checker for "Romance Languages"; indeed, spelling checkers usually involve a level of granularity finer than that of particular languages. There are some limits to this, however. For example, a speech-to-semantics matching voice recognition system for menu navigation in a telephone support system would likely be developed to accommodate a number of accents for a given language rather than being designed with a very specific local pronunciation in mind.

2.3 Other possible applications for "language" identifiers

The application types described above probably cover the most important range of IT applications for "language" identifiers, though they are not necessarily only situations to which they might be applied. For instance, "language" identifiers might be used for subject indexing—that is, for indicating a language that content is *about* rather than the language in which the content is expressed. Applications such as subject indexing can have rather different requirements from the application types mentioned in the previous sections, however. Thus, a language identification system designed for those application types should not be constrained by needs in applications such as subject indexing, even though they may happen to get used for such applications.

Also, there may be situations in which a user wants to record a detailed list of linguistic attributes regarding certain content that go beyond typical needs for IT purposes. For instance, a linguist might want to record numerous details about language data he or she is collecting, including parameters such as the social status of the speaker, speech context, genre of text, etc. This could potentially include an *n*-dimensional set of orthogonal parameters, none of which are relevant for the application areas described above. It would be inappropriate to require that a system of identifiers intended to meet the needs of those application types be extended to include parameters such as these. If someone has a need to apply such metadata values, this would need to be done using distinct metadata attributes.

In addition, some may wish to use "language" identifiers to identify "locales", on the assumption that a particular language generally implies a particular culture. This may be appropriate in some situations, but there is growing opinion in industry that in general this is too limiting (see §3.1 below for further discussion).

3. An ontological model of core language-related categories

The fundamental purpose of "language" identifiers is to indicate distinctions related to linguistic properties, and specifically distinctions that are relevant for IT purposes. There is a wide variety of distinctions pertaining to several distinct linguistic parameters that have been suggested as potentially relevant for "language" identification: languages, language families, dialects, country variants, other regional-based variants, script variants, style variants, modality variants, time-based variants, typographic variants, etc. Many different orthogonal parameters could be used in metadata attributes, and the potential combinations and permutations are daunting. I propose, however, that in actual practice many of the potential distinctions are not needed for realistic usage scenarios. The main suggestion to be made in this paper, rather, is that a small set of category types with well-defined relationships among them can provide an adequate model on which to base distinctions that are needed for IT purposes.

The core of the model I am proposing includes four types of categories:

- individual languages,
- writing systems,
- orthographies, and
- domain-specific data sets.

In this proposed model, writing systems apply specific writing conventions to individual languages, and orthographies apply specific spelling conventions to particular writing systems. Then, in certain usage contexts, additional qualifications are imposed on specific orthographies.

It turns out that different kinds of qualifiers apply to each level in the model. This has implications for the morphology of identifiers. Also, a key aspect to the model is that each of these category types sub-classifies the previous type. This also has implications for the morphology of identifiers: it implies that there is a logical ordering for the different kinds of qualifiers. Having a well-defined morphology for identifiers is a desirable result since it allows for easier parsing of code elements that make up complex identifiers, and it means that the meanings of complex identifiers will be more transparent.

These are not the only category types that will be considered here or that are likely to be needed for IT purposes, but they are the most significant ones and, I believe, should cover most usage scenarios.

In the sections that follow, I will describe each of these category types in detail and discuss the kinds of usage scenario for which they are relevant. Following that, I will explore other category types that also merit some consideration.

3.1 Individual language

There are many possible definitions for the notion *language*, none of which has sole claim to the status of *the perfect definition*. This is so because languages are not, in fact, discrete entities, but rather a collective, complex network of inter-related varieties. Individual languages we identify are, strictly speaking, abstractions that we infer because it is convenient and practical to do so. But there are no set rules dictating that we should map those projected entities onto the actual linguistic network in any particular way.

As discussed in Constable and Simons (2000), different operational definitions of language may be considered appropriate for different purposes. For instance, a member of a minority cultural group who has nationalist aspirations may want to focus on commonality within that group so as

to present a single and stronger cultural identity with which to confront a dominant, majority culture, and so may claim a single language with distinct dialects. On the other hand, a linguist developing literacy programs for the same cultural group may encounter significant barriers to communication within the group, and may need to recognise those "dialects" as distinct languages for practical purposes in order to ensure the programs are effective. Which person has the "right" view? Most linguists would probably lean in favour of the many-languages view, but it could be countered that theirs is not the only view on social phenomena to be considered.

Fortunately, what is *not* necessary for us to do is arrive at the *perfect* definition of language. All that is needed is *some* operational definition that meets IT needs as best as possible.[5] The only question, then, is how to do so.

The first problem in finding an appropriate definition for *individual language* has to do with the boundaries between categories. In creating an enumerated list of languages, we are, to borrow the metaphor, *tiling the plane* of linguistic varieties. "Tiling" suggests clearly demarcated boundaries. Yet, as mentioned above, languages are not discrete entities. They are continua with fuzzy rather than sharp boundaries.

One of the important results of cognitive sciences has been the observation that the traditional assumptions about categories being defined in terms of sets of properties common to all members are invalid. We tend to expect that categories have discrete boundaries and that it will always be clear (given sufficient analysis) whether or not a given individual belongs to any particular category. But this is neither an accurate reflection of human cognition nor of the world around us. In very many situations, the cognitive categories we define in our minds are based not on properties that are common of all members but rather on *prototypes*, best-case examples.[6]

For instance, if you took a large collection of colour chips and began to ask people to tell you for each one whether it was red, yellow, green or blue, you would find considerable variation in the responses. People simply do not have clear and consistent conceptual boundaries between red and yellow, or yellow and green, etc. On the other hand, if you were to ask people to look through all the chips and identify *true red* or *true blue*, you would get surprisingly consistent results.[7]

This is directly analogous to the matter of distinguishing languages. When we "tile the plane" of languages, we are not actually explicitly identifying all of the boundaries. Rather, we are identifying centers, focal points within the network of variations, which serve to represent each given category, and to represent the contrasts between one category and another. What is most valid about the "tiling" metaphor is not the boundaries, but rather the aim to have complete coverage without any significant overlap or duplication. Thus, we want to obtain an enumeration of languages that is comprehensive in the sense of covering all varieties yet without duplication, and we want to do so by focusing on clear distinctions between centers rather than fuzzy distinctions at the boundaries. The question that remains is how to decide how many distinct centers to enumerate and what they are.

The question of how many is directly affected by the problem of finding the appropriate level within a hierarchy of related varieties. In a continuum of linguistic variation, any two varieties

---

[5] As Rick McGowan (in a message on the Langtag discussion list) put it, we have several sets of codes for languages "in various degrees of unworkability," and we just need one that is "the least unworkable."

[6] See Lakoff (1987) for a good overview of prototype semantics and issues of semantic categorisation.

[7] This result was first demonstrated Berlin and Kay (1969). See Lakoff (1987) for a good overview.

have some measure of similarity,[8] and we can infer categories based on differing degrees of similarity. A broad category will include a larger number of speakers, probably with a greater geographic distribution, but would have a larger degree of internal variation. Conversely, a narrow category will include fewer speakers, probably with a smaller geographic distribution, but also with a greater degree of internal homogeneity. For example, at one extreme we could consider a category that includes all Romance varieties, or at the other extreme just the varieties spoken in some Barcelona suburb.

Clearly, we do not want a definition of *individual language* that corresponds to language families or to very local dialects. There is some question, though, of what point between these extremes is appropriate. This question is a very real one for us, as can be seen by comparing the existing set of identifiers in ISO 639 with the enumeration of languages in the *Ethnologue* (Grimes 2000). As discussed in Constable and Simons 2002, one of the issues with ISO 639 has to do with the degree of grouping or splitting of varieties. It was found in a number of cases that the *MARC Language Codes List*, which was a source for ISO 639-2, groups varieties that are listed as distinct languages in the *Ethnologue*.

For instance, consider the ISO 639 entry for [sq] / [sqi] / [alb] "Albanian".[9] MARC has a single code for Albanian, while *Ethnologue* lists four distinct Albanian languages: Gheg Albanian, Tosk Albanian, Arbëreshë Albanian and Arvanitika Albanian. MARC also makes specific reference to Calabrian Albanian (and groups it together with "Albanian"), which *Ethnologue* identifies as a dialect of Arbëreshë Albanian. So, we have possible candidates for an *individual language* category at three different levels of granularity: all Albanian varieties; four varieties, including Arbëreshë; and more than four varieties at a finer level of granularity, including Calabrian. In a comparison of ISO 639 / MARC and *Ethnologue*, comparable situations are to be found for Kurdish, Dinka, Chinese and numerous other cases.

I would suggest that the level of differentiation found in the *Ethnologue* is generally a good fit for the needs of IT applications. The cataloguing of languages reflected in the *Ethnologue* was undertaken with a very practical purpose in mind: to identify varieties that represent potential needs for language development.[10] The operational definition of language used in the *Ethnologue* is based on a primary criterion of mutual non-intelligibility.[11, 12] Thus, given two speakers

---

[8] Similarity between varieties may be due to genetic relatedness, or it may be related to other factors such as language contact or bilingualism. Actual sources of similarity in any given case do not affect the conclusions that are drawn here, however.

[9] As a notational convention, references to ISO 639 codes in isolation will be given in square brackets. Hypothetical codes or multi-part tags constructed using ISO 639 codes will be given within quotation marks. When code elements from both Part 1 and Part 2 of ISO 639 are cited, they will be separated by a slash; for example, [af] / [afr]. In those cases in which ISO 639-2 contains distinct B and T codes, both will be cited with the T code occurring first; for instance, [sq] / [sqi] / [alb].

[10] In some situations listed in the *Ethnologue*, the need for separate language development may be potential rather than actual, yet varieties can still be counted as distinct languages. For instance, speakers of Lombard generally have a high level of bilingualism in Standard Italian, with the result that there may not be any perceived need within the Lombard community for language development activities such as literacy or development of separate literature. Lombard is certainly distinct from Italian, however, and if social circumstances were different, or if circumstances happen to change in the future, then development for Lombard might be considered appropriate.

[11] It is important to note that mutual *non*-intelligibility is used rather than mutual intelligibility. The reason for this is that levels of intelligibility between two related varieties are not generally symmetrical. So, for instance, speakers in community A may not adequately understand speakers in community B, but if those in community B understand those in community A at an adequate functional level, then B does not require separate language development.

representing two varieties, if neither can understand the other at a functional level, then they are considered to speak different languages. If literature were to be developed for these varieties, each would need to have its own in order to ensure successful communication. Where there is mutual non-intelligibility, therefore, there is a potential need for separate language development activities. But it is developed language varieties that are most likely to be significant for IT purposes.

The implication of this, then, is that we need an enumeration of individual languages that is in some cases more granular than the "individual language" codes currently found in ISO 639. This has already been acknowledged by many, however, and so is not a new result. Some may ask, though, whether slightly broader categories might not also be needed: if ISO 639 already has a category that includes all Albanian varieties, somebody must have wanted to use it. I would not argue against that. Instead, the solution being proposed is that we need categories at both levels. What we need to avoid, though, is to call categories at both levels "individual languages". Thus, where ISO 639 currently refers to Albanian as an "individual language", I propose that we should call it something different, a "related-language cluster", perhaps. (Categories that are broader than individual language will be discussed further in §0.)

I propose, then, that the notion of *individual language* that best suits overall IT needs is one that identifies varieties that are potential candidates for separate language development (or, of course, that have already undergone separate language development).

I would also suggest that the inventory listed in the *Ethnologue* is the best available comprehensive list of this sort, although questions of how to actually begin populating code lists does take us beyond the scope of this paper, which is to find an appropriate model for the overall structure of those lists. It is relevant, though, in one particular respect. In Constable and Simons 2000, it was suggested that different operational definitions of *language* may be appropriate for different purposes, and that it may be appropriate to allow for an identifier space that is partitioned so as to allow for subsets based on different operational definitions, possible controlled by different agencies. It is unclear to me at this time whether that might still be useful, or whether providing a more carefully-constructed framework as proposed here eliminates that need. This question is left for further investigation.

It should be noted that I have not differentiated between spoken languages and written languages. Spoken languages are distinct language varieties that have not yet undergone language development. In the case of developed languages, we sometimes recognise distinctions between written speech and spoken speech, but for practical purposes these can usually be treated as sub-language variations. Where there is a considerable amount of difference, then for practical purposes we should treat these in our model as separate, individual languages.

So, for example, the differences between written English and colloquial speech are not great enough to warrant counting these as separate languages within our model. There may be certain usage scenarios in which such distinctions may matter, but these are the exception and so should be handled in terms of some other type of category. On the other hand, the Standard German that is written in Switzerland is quite distinct from Schwyzerdütsch, which is generally spoken but not written. The fact that it is not commonly written is more an accident of history than it is due to any lack of linguistic distinctness. On linguistic grounds, it is a potential candidate for separate

---

[12] For reasons described in note 10, the *Ethnologue* endeavours to distinguish between learned intelligibility (intelligibility that results from exposure to the other variety) and inherent intelligibility. Because of learned intelligibility there may not be a current need in a given situation for separate language development. But circumstances can change such that levels of learned intelligibility drop, at which point the only intelligibility obtained is that which is due to inherent similarities between two varieties.

language development, and so should be counted as a separate individual language. (This corresponds to the treatment given by the *Ethnologue*.)

In some situations, treating spoken, colloquial varieties as distinct individual languages may not correspond with popular opinion. There is an issue here between popular notions of "language" and "dialect" and technical usage of these terms. For non-linguists, "dialects" are very often lower prestige varieties, often perceived as being defective in some way. There are, of course, correlations between levels of prestige and varieties that historically have or have not been developed into written forms with literary traditions. But history does not control the future: just because a variety has not been developed up to now does not mean that it will never be. More to the point, popular opinions do not necessarily determine what is of practical importance for IT purposes.

I propose, then, that an adequate model does not need to distinguish between categories for spoken versus written languages, and that spoken varieties that are distinct enough to have potential need for separate language development should be reckoned as distinct individual languages.

Other issues relating to language modalities also get raised from time to time: what to do with signed languages or Braille. These do not require any special treatment: signed languages are distinct individual languages, and so should be handled in the same manner as languages like Japanese or Navajo. As for Braille, it is a script that can be used for textual representation of various languages. It is relevant for distinguishing writing systems, which are discussed in the next section.

There is still a question of how to deal with modality of language data itself: do we need identifiers that can distinguish text data from audio-visual data? I think that we do not. In most usage scenarios, this is not needed since this can immediately be determined by inspecting the data itself. It clearly is not needed for software localisation and language-enabling, for example, since resources are always retrieved in terms of individual identities rather than common attributes. For instance, when presenting an error message (be it aural or textual) for a particular condition, you don't ask for any resources in (say) French, or even any *text* resource (or voice resource) in French. No, you request a specific resource, and individual resource identifiers subsume issues of modality.

The only situations in which modality may be relevant is cataloguing and retrieval involving repositories with multi-modality content and in which users need to be able to specify queries that return content of one modality only. Even in these situations, text modality can be implied if identifiers indicate a particular writing system or orthography. If modality distinctions must be indicated in cataloguing and retrieval using explicit metadata attributes, I propose that this should be done using a distinct attribute. This will avoid complicating the proposed model to deal with a single issue that is not likely to be a common concern.

As mentioned above, the other category types in the proposed model sub-classify individual languages, and so involve derivative categories. As a result, identifiers for individual languages are primary, and identifiers for other category types will be constructed from these primary identifiers by adding additional, qualifying code elements. Because of the central role of individual-language identifiers, I propose that there should be a comprehensive set of identifiers for individual languages that are atomic—not constructed from combinations of other code elements.[13] At the very least, individual-language identifiers should not be constructed using the

[13] As a result of a resolution passed in August 2001, ISO/TC 37/SC 2 is preparing to undertake a new work project to extend the ISO 639 family of standards to meet broader application needs, and there appears to be a

kinds of qualifiers that apply to derivative category types, such as script or country. This is logical since these kinds of qualification are orthogonal to the identity of individual languages.

Since the proposed model includes other category types beside that of individual language, there is a question as to the kinds of usage scenarios in which an individual-language identifier is relevant as opposed to an identifier for a derivative category type. It turns out that in most situations what is actually relevant is a derivative category type. For instance, people very often want to use "language" identifiers to distinguish what are actually orthographies.

There are still situations in which this basic category type is relevant, however. This can be the case where derivative notions do not apply; for instance, writing systems and orthographies are not relevant for voice data. Individual language may also be the appropriate category type if finer-level distinctions do not matter to the user. For example, if a user wants to retrieve text content in a given language but is not concerned about issues of script or spelling, they would want to specify a query just in terms of an individual language.

3.2 Writing system

The second type of language-related category in the proposed model is a *writing system*. As proposed for this model, a writing system is a particular implementation of one or more scripts to form a complete system for writing a particular individual language. There are two factors that are relevant in distinguishing between writing systems, both of which always apply: an individual language, and a particular set of characters used to represent that language in writing. Also closely associated with the set of characters are the rendering behaviours of those characters; that is, the kinds of shape transformations that take place when characters are actually displayed: selection of contextual shapes, ligature formation, positioning of diacritic marks, etc.

So, for example, the standard written forms for English and Indonesian represent distinct writing systems since different individual languages are involved. Also, the standard written form of English and English written in Braille are distinct writing systems since two different character sets are involved.

It is not common to find significantly different rendering behaviours within a single script, though it is possible. For example, there are different conventions for Latin script with regard to the relative positioning of diacritics in multiple-diacritic combinations. The most common convention is for diacritics to stack vertically away from the base character. Certain writing systems position diacritics side-by-side, however, as in the case of the standard writing system for Vietnamese. I am not actually aware of any language that has two different writing systems that differ only in terms of rendering behaviours. As a hypothetical example, if Vietnamese happened to be written within some communities using vertically-stacked diacritic combinations, that would constitute a distinct writing system using the definition being proposed here.

A writing system determines case mappings. It also determines a complete set of characters, both those used to form words as well as digits and commonly-used punctuation. There may be a fuzzy boundary to this, though. For example, most mathematical operator symbols should lie outside the category for the commonly-used English writing system, but it is not clear to me whether a few symbols such as "=" or "+" should be included. In practical terms, it may not actually matter whether the borderline cases in specifying character inventories are strictly specified.

---

consensus among the task force assigned to provide recommendations regarding this new work project that the ISO standards should include a comprehensive set of atomic identifiers.

It is important to note what is *not* involved in the notion of *writing system*. One key thing is spelling. Thus, the standard written forms US English and UK English represent the same writing system, even though spellings differ. They involve the same individual language and use the same set of characters.[14] Differences in spelling are dealt with in terms of a derivative category type, *orthography*, discussed in the next section.

Capitalisation conventions are also beyond the scope of writing systems. Cases mappings do apply to a writing system, but rules regarding where capitals are used are typically going to be checked along with spelling. Thus, capitalisation conventions are also considered to pertain to the derivative category type *orthography*.

Also, while a writing system includes an inventory of punctuation symbols, at least some aspects of punctuation usage—perhaps most—pertain to derivative category types. For example, details regarding the usage of colons or dashes may depend on writing styles or publication domains. Processes that check such details of punctuation usage are likely going to be integrated into style or grammar checkers that apply to specific orthographies, and thus pertain to narrower category types than writing system.

Overall, a writing system is intended to determine a written form that is readable for a given user and that includes general conventions for writing behaviours (such as relative positioning of Latin diacritics) that are considered appropriate by given communities, but little more. It is assumed that a fluent reader can deal with different spelling conventions, provided that gross sound-symbol relationships are maintained.[15]

Obviously, a writing system implies a text modality. As a result, it is never necessary to indicate explicitly a text modality when it is known that a writing system is involved.

A writing system also implies a particular individual language. This category type, therefore, sub-classifies the individual-language category type. Thus, a writing system identifier can be constructed using an individual-language identifier with an additional qualifier designating the set of characters involved.

It is typically the case when there is more than one writing system for a particular language that different scripts are involved. For instance, Turkish has been written using Arabic and Latin scripts. As a result, in most cases writing system identifiers can be constructed by combining an individual-language identifier and a script identifier. The obvious candidate to be used for script identifiers is the proposed ISO 15924 standard, which is in preparation.[16]

There are some exceptions, however. A very familiar case is Chinese languages, which are written in two different variants of the Chinese script, Simplified and Traditional Chinese. One option for constructing tags to deal with Traditional and Simplified Chinese would be to use a language code and an ISO 15924 script code, and then add an additional qualifier to distinguish

---

[14] Note that country-specific currency symbols are not relevant here since any English text might make reference to any forms of currency. It would serve no purpose to require that a string such as "¥500" embedded within English text be tagged as a different writing system.

[15] If there were some community that wrote English using the letters p, b, t, d, k to represent vowel sounds and letters a, e, i, o, u to represent consonant sounds, then this would represent a distinct writing system since it would be unreadable to readers of the common English writing system. Fortunately, this is a very unlikely scenario.

[16] As of late 2001, this proposed standard was entering the final balloting stage.

the two script sub-types; for instance, “zh-Hani-trad” versus “zh-Hani-simp”.[17] Another option would involve adding codes to ISO 15924 for the two script sub-types. Thus, we would end up with identifiers such as “zh-Hant” and “zh-Hans”. Because of the importance of Simplified and Traditional Chinese writing, I would argue for the latter solution.[18]

It should be noted that what is *not* a particularly good solution for handling Traditional and Simplified Chinese is to use country codes to distinguish them: the issue of country is orthogonal to that of these two script sub-types, and the practice of using country codes “TW” and “CN” has already created problems for some users. Once the script codes are available for creating writing system identifiers, the current practice ought to be deprecated.[19] Generalising, I propose that country codes should never be used to distinguish merely writing systems. Country codes may be relevant for derivative category types, but not at this level in the overall model.

There are other situations involving multiple writing systems for a single language that involve only one script: most systems of phonetic transcription use Latin script. Thus, the standard form of written English and IPA transcriptions of English represent two distinct writing systems for one language using the same script. This situation could perhaps be handled in a manner similar to that proposed above for Chinese; namely, to introduce a script code such as “Ipaa”. The only problem with this is that there are other systems of phonetic transcription beside IPA that rely on Latin letters; for example, the Americanist tradition of transcription. There are three options for dealing with this:

- Give each phonetic tradition its own ISO 15924 script code.
- Use the script code for Latin together with additional qualifying codes to differentiate the variants. For instance, “en-Latn-ipa” versus “en-Latn-amerphon”.
- A variant of the previous option: introduce a script code such as “Lphn” to denote all Latin-based phonetic systems and use additional qualifying codes to differentiate the variants. For instance, “en-Lphn-ipa” versus “en-Lphn-american”.
- Introduce a script code such as “Lphn” to denote all Latin-based phonetic systems, and reckon them within the model to be different spellings of a single writing system. In other words, push this distinction into a different category type within the model.

I am personally inclined at this time to favour one of the latter two options, but do not yet feel ready to reach a particular conclusion. I leave this issue for further consideration.

Let us briefly consider usage scenarios in which a writing-system identifier might be appropriate. Clearly, it is appropriate only for text data. Also, it is appropriate in situations in which written form matters but spelling and any other further issues do not. That could happen in cases in which no spelling variations exist or are considered important.

---

[17] As discussed in Constable and Simons 2000, the ISO 639-1 code [zh] actually designates a collection of related languages rather than an individual language. That detail does not affect the points being made here, however.

[18] This would impact the operational definition for *script* assumed in ISO 15924. Since the set of scripts is somewhat limited, however, and since the Chinese case is exceptional, I think this could be done without introducing any great potential for confusion in that standard. Moreover, the draft standard already provides a precedent for variations on the narrow definition of script, in that the draft includes a code “Jpan” that denotes a combination of Chinese characters plus Hiragana plus Katakana.

[19] The use of country codes “TW” and “CN” may still be appropriate for identification of other category types, such as locales and that is not inappropriate. But if all that is being indicated is the writing system, then script codes provide a much better solution.

This would apply in some contexts to phonetic transcription since the transcription is a record of a speech event, and two speech events involving the same sequence of words are not necessarily going to involve exactly the same pronunciations. On the other hand, phonetic transcriptions that are used in a published dictionary represent idealised pronunciations, and so spellings are relevant in those situations. A writing-system identification is appropriate for the former situations, but it would be incomplete for the latter.

Perhaps the usage scenario for which a writing-system identifier would be most useful is in retrieval of content: the content may be catalogued with orthographic or other finer distinctions, but the user may not be concerned about spelling differences when they specify a query: they may be happy with any result set, provided it is in a particular language and uses a particular script. In such situations, queries should be specified by making reference specifically to a writing system.

3.3 Orthography

The *orthography* category type goes beyond *writing system* in that it specifies particular spelling conventions in addition to particular languages using particular writing conventions. Thus, US English and UK English represent a single language and single writing system but two different orthographies.

Orthographies also include conventions for hyphenation, abbreviations and contractions. As mentioned in the previous section, case mappings are considered to apply at the level of writing system, but general case-usage conventions are considered part of orthography. Some less common case-usage conventions that are specific to certain contexts may lie beyond an orthography specification, however. For example, the capitalisation of titles in bibliographic references may depend upon the publication in which they occur. Such specific rules are handled by derivative category types.

Commonly-used aspects of punctuation (e.g. periods, commas, apostrophes and quotation marks) are inherited from a writing system, and less commonly-used aspects of punctuation, such as different conventions regarding the use of colons and dashes that can involve additional factors such as writing styles or publication domain, are handled by derivative category types.

Orthographies always imply particular writing systems, and so the orthography category type sub-classifies the writing-system category type. Thus, orthography identifiers should involve a writing-system identifier plus additional qualifiers that distinguish between orthographic variants of a writing system. Thus, at this point we have a logical ordering or identifier code elements: individual language identifiers, followed by script or other writing-system qualifiers, followed by orthography qualifiers.

Orthographic conventions are generally the result of a standardisation effort, and typically these are governed by government-sponsored language academies in individual countries, such as the Thai Royal Academy (ราชบัณฑิตยสถาน). At least for well-developed languages, it would not normally be the case to find competing orthography conventions in simultaneous use within a single country. As a result, a generally-useful way to construct an orthography identifier would be to combine a writing-system identifier with an additional qualifier designating a particular country. Of course, the obvious candidate for country identifiers is the ISO 3166 standard. Country codes are useful, then, for identifying orthographies, but not individual languages or writing systems, and a country code should never be more tightly bound to a language code than is a script code.

In some situations, orthographic conventions may be common between multiple countries. For example, a small country may officially adopt the orthography conventions of a larger country that uses the same language. In such situations, resources such as spelling checkers could be

duplicated and tagged for the different countries, though this creates some inefficiencies. A better solution would perhaps be to have software algorithms that track which resources to use in each country context. I leave this issue for further consideration.

Orthographic variations sometimes do occur within a single country, such as when a language academy proscribes a spelling revision. This was recently done in various German-speaking countries, for example. Where spellings change over time, an additional qualifier is needed to distinguish between the different conventions. It is recommended that the year in which a set of orthographic conventions were adopted be used as a qualifier.

In rarer occasions, competing spelling conventions might be used simultaneously by different communities within a given country. This can happen, for example, in emerging literacy situations before widespread standardisation has occurred. Thus, some domain qualifier other than a country code or perhaps in addition to a country code may be needed. What might be needed, then, is a registry of identifiers for orthographic domains other than just countries. There is some potential overlap here with the following category type, which makes reference to general domains of usage, which could lead to potential ambiguity between category types. As will be discussed in §3.9, that is probably not a problem, though this matter is not fully clear to me at this time. Nevertheless, if there are competing orthographies in simultaneous use within a given country that come from different sources, then probably the best kind of qualifiers to use to distinguish these are code elements that identify the agencies that created the two orthographies.

As a completely hypothetical example, suppose linguists working with the Colegio de México and others from El Instituto Lingüístico de Verano, México independently create orthographies for Sierra Popoluca, and neither has been conventionalised throughout the language community. The two orthographies might be identified as “sierpopo-ilvmx” and “sierpopo-colmx”.

Whether qualifiers identifying originating agencies or some other kinds of domain qualifiers are used, it would be essential that the denotation of the identifiers in documented.

The orthography category type is relevant to many usage scenarios. Obviously, software resources for spell checking and hyphenation need to be identified in terms of this category type. Word processors will often want to tag text to indicate orthography so that appropriate proofing tools can be applied. Orthography distinctions are also often important in cataloguing and retrieval of text data, though not necessarily more so than writing system distinctions. Of course, orthography is not relevant to voice data.

3.4 Domain-specific data set

There are many usage scenarios in which it is necessary to specify more than just orthography. For example, in resource localisation, it is often necessary to specify particular vocabulary to be used. It should be noted, though, that whenever it is necessary to specify particular vocabulary, the vocabulary generally gets represented in terms of some particular orthography. For instance, a user-interface string cannot appear with multiple spellings at one time! This suggests the need for a fourth category type that sub-classifies. For now, I have proposed the name *domain-specific data set* since the usage scenarios that first suggested the need to me involve particular data sets (localised resources) for use in specific target domains.

Many current implementations of “language” identifiers that combine language and country sub-elements are being used to distinguish categories of this type. This would be the case, for example, with localised software text resources that are tailored (say) for US English, UK English, Canadian French, French French, etc. that differ not only in spellings but also in vocabulary and perhaps even nuances of grammar and style.

Another example would be in terminology, for instance in standard vocabularies for things like food commodities that get used in international commerce. So, for instance, when dealing with international trade in potatoes, it is important to know that “patata” is used in Spain but “papa” is used in Mexico.

This category type is defined, then, in terms of linguistic distinctions that involve more than just orthography and that relate to fairly specific domains of usage. It sub-classifies orthography, and so identifiers for this category type would combine an orthography identifier with additional qualifiers.

In most current implementations, the domains of usage that are used are countries. This has probably been so because RFC 3066 makes specific reference to language-country combinations. But in many cases the relevant domains of use do not correspond to countries. They may correspond to multiple countries; so, for example, there has been a recognised need in the localisation industry to identify Spanish text resources targeted for all of Spanish-speaking Latin America. Usage domains may also be smaller than a country; for instance, an Australian corporation may wish to specify particular orthographic, grammar, style and vocabulary conventions for use in company documents. Usage domains can be completely independent of geography, as might be the case if the previous example were applied to a multinational corporation or an international agency such as ISO.[20]

What might be needed for identification purposes, then, is a registry of identifiers for domains of usage other than just countries.[21] Such a list would include many different types of domains: geographic regions, businesses, government agencies, professional societies—any kind of entity that may exercise some type of jurisdiction with regard to language use. Of course, whatever the source or form of such identifiers, it is essential that their denotation is documented.

The primary usage scenarios for this category type may be in identification of software resources for linguistic processes (such as grammar and style checkers or thesauri), and in localisation. It may be less relevant for cataloguing and retrieval of content, though it may be appropriate for those applications in certain contexts. (We will return to this possibility in §3.6.)

Software resources for grammar checking are generally implemented together with spell checkers, and will certainly be sensitive to orthographic conventions. In some cases, they may be distinguished at the level of orthography. But often grammar checkers also incorporate elements of style checking. If a grammar checker is specific to a particular written style, then it should be identified in terms of this category type. If a grammar checker accommodates multiple different styles, it should probably be identified in terms of orthographies, although style-specific settings would relate to this category type.

There can be multiple sort orders associated with a language. Sort orders always imply particular writing systems, however, and they often imply particular orthographies.[22] Thus, the *domain-specific data set* category type may be able to accommodate sort orders. The main possibility that could keep this from working would be a need for factors such as vocabulary to vary independently of sort orders. This would entail multiple dimensions of variation that require

---

[20] Indeed, the thing that first suggested to me the importance of domains other than countries was a request made to ISO/TC 37/SC 2/WG 1 to have codes added to ISO 639 for “ISO English” and “ISO French”.

[21] ISO/IEC JTC 1/SC 32 is currently working on two proposed standards that may be relevant in this regard: ISO/IEC 18022, “IT-Enablement for Widely-Used Coded Domains”, and ISO/IEC 18038, “Identification and Mapping of Various Categories of Jurisdictional Domains”. I have not been able to locate drafts of either of these proposed standards, however, and so do not know whether they are, in fact, relevant or not.

[22] Sort orders do not *always* imply particular orthographies, as can be seen in the case of English.

independent qualifiers in metadata attributes. It seems to me, though, that specific sort orders are required in particular domains of usage, just as a usage domain may determine a need for particular vocabulary. Therefore, I would expect that a given domain determines a collection of linguistic parameters, including both vocabulary and sort orders (in addition to orthography, etc.) If that is the case, then a single domain qualifier within an identifier tag should be all that is needed. I leave this matter to further investigation.

3.5 Super-ordinate categories

In discussions regarding needs in language identification, people have often suggested the need for broad categories denoting collections of languages, language families in particular.

Of course, ISO 639-2 currently includes a number of codes for language collections, though they have a specific property of excluding any individual languages that have their own identifiers within the standard. Thus, the ISO 639-2 collections are not simply categories at a coarser level of granularity than individual languages.

It would be possible to define many different types of super-ordinate categories based on various factors: genetic relatedness at various levels of reconstruction (e.g. Romance languages, or Indo-European languages); common linguistic properties (e.g. tonal languages), geographic distribution (e.g. South American languages), etc. There are, in fact, no limits to the number of potential super-ordinate category types that are conceivable. It is not self-evident whether there is a need to distinguish between these category types or, more to the point, whether any such categories are actually needed for IT purposes. Therefore, before proposing any such category types, we should consider potential usage scenarios in IT applications.

It is reasonably obvious that super-ordinate categories are not generally useful for localisation or for software-enabling resources: one cannot create a spelling checker for “Romance languages”, a translator will not index translation memory resources as “Philippine languages”, and a localiser will not prepare Web sub-sites for “Germanic languages” or “African languages”.

Super-ordinate categories may be useful as subject indicators, e.g. to catalogue books about Mon-Khmer languages. As mentioned in §2.3, however, requirements for a system of identifiers for language-related categories should perhaps not been determined by the needs of subject indexing.

The main application area in which super-ordinate categories are most likely to be relevant is in cataloguing and retrieval of content, and mainly only for retrieval at that. As discussed in §2.1, if information objects are catalogued in terms of broad categories, it hinders flexibility in retrieval; for instance, if items are catalogued in terms of language families such as Germanic and Romance, it is not possible to request items in a specific individual language such as Dutch. On the other hand, cataloguing in terms of individual languages can provide flexibility to request items in specific languages only, or in multiple languages.

Thus, the main use for super-ordinate categories is in retrieval of content. Such categories should only be used for cataloguing if there are not adequate trained personnel resources to identify every individual language or if it is known that there will not be a user need to retrieve content in terms of queries that specify individual languages. For example, if a library held a total of ten items written in various Algonquian languages and a user was looking for items specifically in Naskapi, it would not be difficult for the user to inspect all ten items by hand, whereas it might be expensive for the library to catalogue those ten items in terms of the exact languages used in each.

There is some possible usefulness, then, in super-ordinate categories. Let us consider for a moment the usefulness of the kinds of collections currently used in ISO 639-2: that is, collections that specifically exclude individual languages that have their own identifiers. We might think of

these as fallback categories, since they are only intended to be used when there is no better alternative. It seems to me that collective categories with this "fallback" characteristic offer no benefits and actually create problems.

If we have super-ordinate categories that are fully inclusive in addition to individual-language categories—for example, a category for Russian and also a category for *all* Slavic varieties including Russian—then we have flexibility to do whatever is needed. If a cataloguer for a given repository wants to specify "Russian" for items in Russian but use the generic category "Slavic" for all other Slavic languages, that should not present any difficulties either in cataloguing or in retrieval. The only possible difficulty would be if someone wanted to retrieve items in any Slavic language *other than* Russian, but that is a rather unlikely scenario. Additionally, this particular scenario already is not supported by the existing ISO 639-2 code elements since the code [sla] "Slavic (Other)" excludes not only Russian, but also Czech, Ukrainian, and several other languages. Moreover, it seems that a better way would be to perform logical set-arithmetic operations in specifying the query: "all Slavic languages and not Russian". This kind of query would be readily supported by fully-inclusive super-ordinate categories by not very well by "fallback" collections.

"Fallback" collections, then, provide less flexibility than fully-inclusive collective categories. Moreover, as pointed out in Constable and Simons 2000, they create problems since, any time a new identifier is introduced for an individual language within the collection, the denotation of the collective category changes with the result that existing data can become incorrectly tagged. For example, consider the impact on data by adding a new code to ISO 639-2 for a language like Northern Yi. Currently, if existing Northern Yi data is tagged with an ISO code, [sit] "Sino-Tibetan (Other)" would be the code of choice. Now, suppose after some time a new tag is added for Northern Yi. Because [sit] represents a "fallback" category, the range of languages that it covers sit has suddenly changed since it no longer includes Northern Yi. The result is that the existing data in now *incorrectly* tagged. If the code [sit] denoted a fully-inclusive category, adding a new code for Northern Yi would not have changed the denotation. The addition would have resulted in the data being sub-optimally tagged, but not incorrectly tagged. As it is, with the use of "fallback" collections, the data becomes broken and, to make matters worse, there is no way to know which items have been adversely affected.

If super-ordinate categories are to be used, then, they should be fully-inclusive categories for whatever level of granularity they correspond to.[23]

The remaining question, then, is what types of super-ordinate categories should be included in standards for language identification. As mentioned above, many different types of category are possible. Clearly, some constraints should be imposed. Thus, we should not allow for any ad hoc collection that might be requested. In addition, we should specify the allowed types as this provides operational definitions for the categories that serve to tell us what specific categories of each type are needed in a comprehensive collection, and also serve to make it clear what each individual category actually denotes.

---

[23] It might be recommended to ISO, therefore, that the denotation of existing collective language codes be redefined as fully-inclusive collections. There are additional factors to motivate this. As discussed in §3.7 of Constable and Simons 2002, a careful analysis of ISO 639 indicates that certain existing collective categories, such as [bat] "Baltic (Other)" and [no/nor] "Norwegian" contain only one or even zero modern languages. There is little question that existing uses of the code [no] assume a denotation that includes both [nn/nno] "Nynorsk" and [nb/nob] "Bokmål" rather than denoting the empty set.

I propose that only three types of super-ordinate category should be accommodated in standards for "language" identification. These correspond to the three types already found in ISO 639, as identified in Constable and Simons 2002:

- collections based on genetic classifications,
- collections based on geographic region, and
- collections of closely-related languages that may be referred to by a common name.

Collections based on genetic classification would include categories such as "Indo-Aryan languages" or "Quechuan languages". Collections based on geographic region would include categories such as "North American Indian languages" or "Australian languages".

The third of these categories is slightly ad hoc: it includes collections such as [zh] / [zho] / [chi] "Chinese" and [apa] "Apache languages". The problem with this is that the name that is taken to be common is a particular label used by outsiders and doesn't necessarily correspond to actual cultural identity perceived by the speakers of the languages in question. I include the category type mainly because it is already in use in ISO 639. Also, in the particular case of [zh] "Chinese" there may be some reasonable motivation in terms of practical use since it corresponds to all languages of the Chinese genetic classification that do get referred to by outsiders as "Chinese" and, more significantly, have any likely potential of being written using Chinese characters. Since the names involved are in common use by outsiders, some implementers may also find these categories useful in the event that they want to provide some minimal support for these linguistic varieties but do not have a need to provide individual treatment for the individual languages involved.[24]

In addition to the three types of super-ordinate category mentioned above, certain special-purpose collections may also be useful. These include two that are already included in ISO 639-2: [art] [sgn] "Sign Languages" and "Artificial (Other)".[25] The potential benefits of other special collections in relation to actual IT usage scenarios should be considered on a case-by-case basis.

Let us briefly consider the form of identifies for super-ordinate category types. These should not be constructed from combinations of code elements for other category types but rather should be atomic. It is essential that the type and denotation of each is explicitly documented so that the intended meaning of the identifier is clear to users. The denotation of collective categories should be documented by explicitly listing all of the individual languages contained within the given collection. In situations in which users might easily be misled, it may also be necessary to list individual languages that are *not* included in a collection.

The only remaining questions regarding super-ordinate categories are how many are needed and what exactly they are. I leave this matter open for further consideration.

---

[24] Since it is one of the goals of SIL International to promote development of lesser-known languages, I must add that I would personally encourage implementers to not use common-name collections for such reasons since it puts these languages at a disadvantage. This does not apply in the same way to Chinese languages due to the exceptional nature of Chinese writing: to a significant extent, text written in one Chinese language using Chinese characters can be understood by speakers of other Chinese languages because of the nature of the Chinese characters. Other groups of languages such as Apache do not enjoy the same benefit.

[25] Of course, as suggested above I would recommend that the denotation of [art] "Artificial (Other)" be changed so that it includes all artificial languages.

3.6 Dialects and other sub-language variants

In discussing potential future needs with regard to language identification, problems of dialects, style variants and other forms of language-internal variation are occasionally mentioned. They can seem overwhelming, but I believe they can be handled as part of a structured framework.

To begin, it is important to note some important similarities as well as important differences between problems of identifying individual languages and those of identifying dialects. It was claimed earlier that it is possible to enumerate a comprehensive list of individual languages based on a particular operational definition—to "tile the plane" of languages. In contrast, this is in principle impossible for dialects. The reason has to do with operational definitions.

In the case of the notion *individual language*, we were able to arrive at an operational definition that was suitable for IT purposes, expressed in terms of potential need for development as determined by barriers to communication. The key thing to note is that the metric of incommunicability is sufficiently consistent to create a *single* partition. The boundaries may be fuzzy, but all that matters is that we have a *one-dimensional* set of distinctions.

Dialects are another matter altogether. Dialect variations are differences with respect to any linguistic parameter or combination of parameters that are noticeable but that do not completely impede functional communication. The variations can be of many different types: differences in pronunciation, morphology, syntax, vocabulary, the semantic range covered by lexical items, and collocations, just to name some variables. Moreover, any one of these types of variation may actually entail several different parameters. The critical point is that these parameters can be independent of one another. In other words, we are dealing with an *n*-dimensional set of factors that can be involved in dialect distinctions.

Furthermore, many of these axes of variation can involve continuous rather than discrete variations. In other words, there are fuzzy boarders. In this way, dialects are actually like languages. But while there are ways to measure comprehension—a primary measure on which we can distinguish between languages—there may not be obvious ways to measure the degree of perceived significance in relation to some parameter of linguistic variation, since we may not have any way to know what impact any given measured amounts of variation has on how humans perceive the differences.

Dialect distinctions are also subject to individual perceptions. For instance, people in community A may notice certain distinct qualities in the speech of people in community B, but those in C may not notice the same distinctions, and those in B may not notice the opposite qualities in the speech of those in community A. Each may be sensitive to a different set of distinct qualities in each other's speech. So, not only are there multiple, independent dimensions of variation with no necessarily obvious way to determine significant graduations in variation along any particular dimension, but there is also no one point of reference.

In short, there simply is no operational definition for *dialect* that will make it possible for us to enumerate a comprehensive list of dialects—to "tile the plane" of dialect variations.

It should be noted that this is not at all a disappointing loss. If it *were* possible to create a comprehensive list of dialects, their number would be at least on the order of 20-40 thousand, and probably considerably higher. This would completely overwhelm users and IT implementers. But even if it did not, the vast majority of these would remain completely unused: there simply is not an IT need for that many distinctions. For instance, dialect distinctions are often mentioned in relation to voice recognition, but it is unlikely that any company would ever come up with a business plan that justifies creating dozens of distinct implementations for English, let alone multiple implementations for some lesser-known language with low economic viability, such as

Kensiw (spoken by a nomadic negrito people group found in the vicinity of the Thai-Malay border).

While there are multiple reasons why it is impossible to list all dialects of a language, it will be noted that in some situations there *can* be sub-language distinctions that are conventionally recognised through much or all of the overall language community. So, for example, Cockney is a conventionally recognised dialect of English. There are also regional English accents that are at least somewhat widely recognisable: "Brooklyn", "Scottish", "Texan", etc.[26] Thus, coherent dialect identities can and do occur; but in general there is no way that we know of to predict what identities will emerge from among all of the network of variations within a given language. They just happen.

Not all conventionally-recognised sub-language distinctions correspond to what is usually considered *dialect*. For example, Thai has several distinct variations that are usually referred to as *registers* and not *dialects.* For this reason, it is preferable to talk in more general terms by referring to *sub-language variants* (i.e. variants within the scope of a single independent-language category) rather then *dialects*.

As mentioned above, any or all of the parameters of variation that differentiate to given varieties can have fuzzy boundaries. This points to an important way in which dialects and other sub-language variants are *like* languages: when we *do* recognise them as identifiable entities, we do not define them in terms of their boundaries. Rather, we define them in terms of the qualities of the prototypical centers—the representative cases. Thus, just like colour terms, anyone who can recognise a Cockney accent thinks of Cockney in terms of some prototypical example, and these prototypes are probably quite consistent from person to person.

So, there are conventionally-recognised sub-language variants that we conceptualise in terms of prototypes, but these conventional identities emerge on an *ad hoc* basis, and we cannot enumerate a comprehensive set. The implication of this is that, if we need identifiers for sub-language variants, we do not need to attempt to create an extensive list, but rather need to provide a mechanism whereby categories and their associated identifiers can be enumerated as needs arise.

This brings us to a significant question: is there, in fact, any actual IT need to have identifiers for a sub-language-variant category type? To answer this we need to consider potential usage scenarios. The usage scenarios are somewhat different for voice data than for text data, so I will consider each of these separately.

First, let us consider voice data.[27] Since text is not involved, there are no potential implications for the *writing system*, *orthography* and *domain-specific data set* category types. As mentioned in §3.1, bare individual-language identifiers will often be used for voice data. As also mentioned in this section, however, dialect issues are often mentioned in connection with voice data.

It should be noted that, of all the possible dimensions of dialect variation, the main one that we particularly need to consider here is pronunciation.[28] The need for indicating pronunciation distinctions in most content cataloguing and retrieval scenarios is unlikely: people looking for information will generally be far more concerned with the information content than with

---

[26] Of course, a category like "Scottish accent" still has quite a bit of variation, so we still have the problem of a hierarchy of granularity as we did with languages, but nothing to suggest where to find an appropriate intermediate level. This just adds to the problems involved in defining dialects.

[27] These considerations also apply to visual recordings of signed languages.

[28] For instance, as far as I know, nobody has thus far suggested the development of grammar or style checkers for voice data.

pronunciation. Granted, big differences in pronunciation may matter to a user, but they probably will not know how much their comprehension is hindered until they have actually heard the content. Also, we should keep in mind that pronunciation has to do with more than dialect: Danish voice content may be spoken by someone with a cleft palette or a lisp, or by a native of Vietnam who has only learned Danish late in life. If users need to be aided in knowing whether pronunciation will be a concern to them, it may be more useful to tell them about the speaker than to label the content in terms of a pronunciation-dialect identity.

Linguists may want to index data according to pronunciation differences, but as mentioned early on, researchers in the humanities may want to index data according to many different, orthogonal properties that should not be mixed into identifiers for language-related categories to be used for general IT purposes.

Overall, then, it is not clear to me whether there is a real need to identify pronunciation variants for cataloguing and retrieval of voice content. I will concede the possibility, and take that into consideration below, but suggest that this issue requires further investigation.

I have heard suggestions that pronunciation may be important in relation to voice recognition systems. This is possible, though I think only to a limited degree. Given the mobility of most people today, it is probably preferable for voice recognition systems in industrial usage contexts to be accent-neutral. It is, perhaps, possible that, in some very specific contexts where it is necessary to ensure a high degree of accuracy, pronunciation distinctions will matter, though this seems unlikely to me. Pronunciation-dialect distinctions are probably more likely to matter for voice synthesis applications.[29] That possibility should, perhaps, be considered in designing a system for identification of language-related categories.

Pronunciation-dialect distinctions may be important for localisation scenarios. I would think that in most situations implementers would get as much mileage as possible out of prestige dialects or accents that are considered culturally neutral. There may be limits to have far this can be used, though. For example, in creating voice resources for user feedback in a software system, a Mid-west US pronunciation may be acceptable to most North American users, though users in other regions of the world may prefer certain British accents. Providing this level of usability tailoring is expensive, however, so it is not clear to what extent there is an industry need. The possibility of needing to distinguish pronunciation-dialect distinctions for localisation data is probably realistic, though, and so should be considered.

There may also be localisation needs to specify particular vocabulary for voice data. Such requirements are determined by usage domains, and so we have a category that is a voice-data counterpart to the *domain-specific data set* category. The difference is that in this case we do not need to distinguish between writing systems and orthographies. Thus, an identifier for such a category needs to specify a particular individual language and also a specific domain, but it does not ever require writing-system or orthography qualifiers. For reasons to be discussed in §3.9, identifiers for voice-data domain-specific data sets may often be the same as those needed for their text-data counterparts. As a result, it may be possible in practical terms to treat the two as single category type.

Let us turn now to consider usage scenarios in relation to text. In discussions regarding needs for "language" identification, the scenario I usually here referred to is the need to tailor vocabulary in localisation contexts. As pointed out in the previous section, though, this will imply specific

---

[29] I suppose it is possible that some day there may be needs in relation to voice synthesis to indicate pronunciation variations that go beyond dialect: cleft palette, lisp, native of Burkina Faso, etc. Presumably, though, such factors would be treated as independent parameters.

spellings, and is determined by particular domains of usage. Thus, the most common needs for distinguishing sub-language variations in text data can be handled in terms of the *domain-specific data set* category type.

Again, I consider all of the linguistic-variation parameters that may be of interest to researchers in the humanities to be beyond the scope of our purposes here.

We also need to consider text content other than localisation resources that may be characterised by distinctive sub-language variations. Consider, for example, the dialog in *Huckleberry Finn*, by Mark Twain. In the preliminary material, Twain provides an explanatory note in which he indicates that various characters use a number of dialects: "the Missouri negro dialect; the extremest form of the backwoods South-Western dialect; the ordinary 'Pike-County' dialect; and four modified varieties of this last." Twain proceeds, then, to write the dialog as well as the narrative (the story is told from the perspective of the title character) using spellings, vocabulary and grammar that are suggestive of these different dialects. For instance,

> "Say—who is you? Whar is you? Dog my cats ef I didn' hear sumf'n. Well, I knows what I's gwyne to do. I's gwyne to set down here and listen tell I hears it agin."
>
> ——*Huckleberry Finn, Chapter 2*

Since most written communication is in standard written varieties, there is not likely a common need for cataloguing and retrieval of sub-language variants of this sort. It is possible, though, that some needs of this sort do exist. What is more interesting about this example, though, is the potential implication with regard to resources for linguistic processing and language enabling of software. For instance, if there were enough user need for the written English variety shown above, a separate spell checker would be needed, as well as a separate grammar/style checker.

If we think of what category types and identifiers would be needed to accommodate a sub-language written variant such as this, there are two options open to us. First, we could add to our model a category type for sub-language variant, and then say that the *writing system* and *orthography* category types could sub-classify this as well as or instead of the *individual language* category type. (The *sub-language variant* type would, of course, sub-classify the *individual language* type.) Thus, in terms of the language above, we would have the following:

- one individual language, English;
- two sub-language written variants: standard (the unmarked case), and "Missouri negro dialect";
- two writing systems: the common English writing system, and the (presumed) common "Missouri negro" writing system;
- an additional orthography: the (presumed) common "Missouri negro" orthography; and
- an additional domain-specific data set (e.g. to accommodate grammar checking) for "Missouri negro".

Strictly speaking, this is probably the most logical relationship between category types, but there are some inefficiencies: because of the logical relationships, we have a distinct writing system category, even though there are not any actual differences. And, of course, we have added a new category type to our model.

The other possibility is that we simply reckon this written variant to be an instance of a *domain-specific data set* category. That category type applies when a given domain of usage requires a particular orthography, vocabulary, grammar and style, which seems to be applicable here. It implies the existence of a distinct orthography, but that distinction would be ignored: systems

would not have separate identifiers for the *orthography* and the *domain-specific data set* categories. In terms of the example above, we have the following:

- one individual language, English;
- one writing system, the common English writing system;
- an additional orthography: the (presumed) common "Missouri negro" orthography—but this is ignored; and
- an additional domain-specific data set for "Missouri negro".

This is admittedly bending our definitions. The justification that would be made for this is that it avoids introducing a need for an additional category type (though that may be independently motivated), as well as the resulting additional derivative categories, for the benefit of text data of a non-standard variety that is not likely to be encountered all that much. In other words, we would be "fudging" to keep things simple.

If the former approach were taken for text data, then qualifiers for sub-language variants would be defined as needs were identified, and then any qualifiers for distinguishing writing systems, orthographies or domain-specific data sets would be added following that. In the case of the example above, we might potentially end up with an identifier like "en-mingro-Latn". [30] If the latter approach were taken for text data, then nothing new would be needed: the same mechanisms used for constructing identifiers for *domain-specific data set* categories—using some set of domain identifiers—would be used here. In the case of the example above, we might potentially end up with an identifier like "en-Latn-mingro".

I present the latter option with a view to finding a way to avoid adding categories and a whole new category type for a kind of text data that is not likely to be very common. It is not clear to me at this time which approach is to be preferred, however. It may be that this simplification of the model would only result in confusing users and implementers.

It should be noted that there are hypothetical situations that could eliminate the latter option as a possibility. Specifically, if there were multiple writing systems or orthographies specifically for a sub-language variant, then we would want the sub-language variant qualifiers to precede other qualifiers that distinguish between writing systems and orthographies. This is very improbable, though: any linguistic variety that had undergone that level of language development would certainly already have been recognised as an independent language.

Returning briefly to voice data, we probably do need to accommodate some pronunciation variants. It was also noted above, though, that there is probably a need for a voice-data counterpart to the *domain-specific data set* category type for text. This again raises the question as to whether these sub-language variants could simply be treated within the framework as though they were instances of the *domain-specific data set* category type, which is independently motivated. I leave such questions for further investigation.

3.7 Historical language varieties

If there is any set of issues that is likely to exceed the limits of the proposed model, it is those issues that pertain with identification of historical language varieties. Earlier, I described

[30] For reasons described in §3.9, this is not the form that would most like be used. This example identifier and the one that follows are presented mainly to show the implications for the way in which tags are constructed. The critical point is that the two ways of constructing tags assume two different view as to how different kinds of language-related distinctions are handled within a model.

language at a given point of time as a network of inter-related varieties with many dimensions of continuous variations between them. As we look from a given language back through time, we see the same kinds of continua of variation occurring. The problem for historical linguists and palaeographers is in how to infer distinct entities within those continua. Synchronically, we can look at issues such as barriers to communication and language development to guide us. Perhaps similar approaches can be used diachronically, but that it is not immediately obvious that that is the case.

As described earlier, synchronically, languages do not have discrete boundaries. If we can infer discrete entities onto historical varieties, the same will be true. Therefore, just as synchronically entities have to be defined in terms of representative centers, the same would have to be true diachronically. We also saw that, synchronically, sub-language variations can be multi-dimensional, with the result that there is no single principle by which we can determine all sub-language variants. The same is true diachronically. So, if diachronic "sub-language" entities are to be distinguished, their identities are determined on an *ad hoc* basis as particular varieties stand out. If the historical linguists and palaeographers can find a way to identify "individual language" distinctions, then, perhaps historical languages can be handled in more or less the same way as modern languages.

There are some additional complications, though: not only do palaeographers have to deal with continua of *language* variations, they also have to deal with variations in orthography and even in scripts. These are *not* issues that apply to modern languages.

It is not at all clear to me how these issues should be resolved. Hopefully, the historical linguists and palaeographers will be able to analyse their needs and arrive at a reasonably simple and workable solution.[31] And, hopefully, the mechanisms proposed here for modern languages etc. will provide what is needed for historical varieties.

3.8 Language-related categories and locales

Earlier in this paper, I made reference to potential problems due to confusing the notion of "locale" with "language". Having considered various language-related category types, we can now return to consider this in a little more detail.

A locale is a set of culturally-determined characteristics of information and how it is presented to the user. It primarily has to do with user interface parameters; for instance, the language to use for menu strings, or the format to use when presenting numbers. A locale can also include parameters that affect more than user interface, however; for instance, the way that postal addresses should be structured when printing mailing labels, or the format that should be used for dates that are automatically generated and inserted into a word-processor document.

In the past, it has been thought that two factors were typically needed to identify a culture: language and country. In other words, locales were theoretically conceived as being two-dimensional. But then, further simplifications were considered possible since not all languages are necessarily spoken in all locations. If one considered the cultures of primary commercial significance, the number of language-country combinations needed was quite constrained.[32] So, for example, only English was considered relevant for the US, just English and French for Canada, etc.

---

[31] I also hope that this current work will stimulate their investigation into their problems.

[32] It should be noted, though, that even a small number of cultural contexts can entail a lot of work for implementers.

Also, in some earlier software implementations, a locale setting was global: it determined not only user-interface language and things like date formats, but also constrained what languages could be used in data, what spelling checkers were used, etc.

Early assumptions led to situations in which "locale" and "language"—actually, the various language-related categories we have discussed—were confused. Since in some systems a locale parameter was the only thing that was available to determine cultural settings of any kind, then whenever a particular language or writing system or orthography had to be specified, a complete locale identity was used.

It is very likely the case, since language-country combinations were the mechanism available for culture-related parameters, that language-country combinations were used to distinguish Simplified and Traditional Chinese, even though this writing system is independent of country. It may have been adequate for early implementations that targeted very limited markets to assume a correlation between the two writing systems and particular countries, but in hindsight we can now say that that was not a valid assumption.

Also, because of the global use of locales in software systems, it meant that entire locales needed to be duplicated if just one parameter changed, including those that are unrelated to language. This issue can be seen in the Win32 infrastructure. As described in Constable and Simons 2000, Win32 "language identifiers" (referred to in Win32 documentation as LANGIDs) are, in fact, locale identifiers, and certain "language" categories have been created in order to support non-linguistic distinctions. So, for example, there are distinct LANGIDs for Italian Italian and Swiss Italian. Yet, as far as I can tell, there are no linguistic distinctions between these—the main differences seem to be in relation to date and number formats and also currency symbol.

As people become increasingly mobile and as industry and commerce become more global in nature, the old assumptions in the design of infrastructures for locale no longer work. Many in industry have begun to recognise fundamental problems, such as the following:

- Locales are multi-dimensional rather than two dimensional.
- Settings required for different culture-dependent variables may be determined by multiple factors, including inherent qualities of the data, the user's cultural background, the user's current context, and the user's personal preferences.
- These factors may affect different variables differently, with the result that values for one dimension (like country) do not necessarily determine values for any others (although in some cases certain combinations may have a high probability of co-occurrence).
- In some situations, only certain culture-dependent variables may be relevant; for instance, when specifying an orthography for text being entered in a word-processor, one should not be required to specify a particular choice for postal address or date formats at the same time.

Let us turn now to consider how these issues apply to the current topic. The main point to be made is that we need to be more careful in distinguishing between different *types* of category, and to make sure that we apply the correct category type in a given application situation. For one thing, we should not use a "locale" identifier to identify individual languages, writing systems, orthographies or domain-specific data sets *unless* that identifier happens to be what is appropriate for that type of category, and vice versa. Also, we should make sure before we use a "locale" identifier that that is really what we need—often, it isn't.

In addition, if software implementations are designed around identifiers that are language-country combinations, it should be recognised that they are not adequate for many current needs, let alone future needs. It must be possible to specify linguistic properties for language data independent of

non-linguistic culture-dependent variables. This may require a redesign of existing infrastructures for "locale" identification.

Finally, if changes are made in how "locales" are handled in software implementations, and in particular in how "locale" identification is done, the full range of needs in relation to identification of language-related categories must be borne in mind. A direct implication of this is that no new system for "locale" identification should be undertaken until problems of identification for language-related categories are first solved (unless a new system is created using identifiers that do not contain any language-related code elements whatsoever).

3.9 Summary of language-related category types

We have discussed several different types of category as part of the proposed model. There are the four main category types, each of which sub-classifies the one before it: *individual language*, *writing system*, *orthography*, and *domain-specific data set*. We discussed a fifth possible category type that may be needed, *sub-language variant*, which would fit between the *individual language* and *writing system* category types. Then we also discussed three super-ordinate category types: collections based on genetic classification, collections based on region, and collections for closely-related languages with a shared name.

The relationships that exist between the main four (possibly five) category types imply a certain morphology for identifiers:[33]

> *individual language ID*
>
> *individual language ID + sub-language variant qualifier*
>
> *individual language ID + sub-language variant qualifier + writing system qualifier*
>
> *individual language ID + sub-language variant qualifier + writing system qualifier + orthography qualifier(s)*
>
> *individual language ID + sub-language variant qualifier + writing system qualifier + orthography qualifier(s) + other domain qualifiers*

It was also noted that identifiers for individual languages and super-ordinate categories should be atomic, and that particular kinds of values are appropriate for the various position classes.

4. Default values and implicit tagging

We have looked at a proposed ontological model of language-related categories and considered implications for morphology of identifiers. There are some further considerations in applying the proposed model that merit some discussion.

The proposed morphology of identifiers begins with an individual language identifier and adds additional qualifiers that are appropriate for each derivative category type. It might seem that what is being proposed is that any time an orthography (say) is being identified that the identifier must include writing-system-qualifier and orthography-qualifier components. That would suggest a need for much richer identifiers than are used in existing implementations.

---

[33] This illustration assumes the inclusion of the fifth category type, *sub-language variant*, in the model. Also note that this illustration is not intended to suggest any specific details of actual syntax, such as what characters might be used to delimit code elements from which complex tags are constructed.

It is important to point out that that is not at all the intent here. The proposed morphology describes the structure of a fully-qualified identifier. Often, however, qualifiers may not be needed, even to describe category types at the lower levels of the model.

The reason for this is that certain defaults are applicable. For instance, while language and script are logically independent, in actual practice only certain combinations do occur, and in most situations there is an unmarked case. So, for example, the vast majority of English text data the average person is likely to encounter uses the common English writing system. Some English text data may be in Braille, some may be in phonetic transcription, some may be in some form of shorthand or any script you might want to imagine, but by a large margin most of it is written using the characters of good old ASCII. It is the unmarked case.

In general, where there are conventions that are considered the norm, they can be treated as default, unmarked cases that do not need explicit indication in an identifier. As a result, identifiers can have implicit semantics, and can also be used for more than one category.

For example, consider the ISO 639-1 code [en]. This represents an individual language, English. But English has an unmarked writing system, as already mentioned. Thus, [en] can also be used to denote that writing system. If we need to identify the English Braille writing system, that is an unmarked case and so requires an explicit qualifier; hence "en-Brai". Similarly, if we need to identify English in IPA transcription, an explicitly qualified identifier would be needed.

Continuing the example, if we consider the next level of orthography, there is no unmarked case for English: no single set of orthographic conventions is considered the norm. As a result, all English orthographies require explicitly qualifiers. Note, however, that the writing system in the common orthographies is unmarked. Thus, an orthography qualifier can be added without needing any explicit writing-system qualifier; hence, "en-CA" or "en-UK", etc. On the other hand, if we were discussing orthographic conventions where Braille were involved,[34] both writing-system and orthography qualifiers must be explicit; thus, "en-Brai-US" or "en-Brai-AU".

Taking the example further to domain-specific data sets, if we were localising resources using vocabulary specific to the UK on the one hand and the US on the other (but not more specific domains), in each case the respective country's orthographic conventions would be involved, and so the orthographic qualifiers (the country codes) also provide the distinctions needed for the domain-specific data sets. Thus, "en-UK" and "en-US" can be used to identify the standard orthographies for the UK and the US, or they can be used to identify vocabularies (domain-specific data sets) that are generally applicable for the UK and the US.

It was mentioned in §3.4 that a language can have multiple sort orders and that these would generally assume certain orthographies and be used in certain domains, and so belong at the level of the *domain-specific data-set* category type. In the case of English, though, there is an unmarked sort order that is implied by the unmarked writing system. Thus, assuming the unmarked writing system, English sort order never has to be explicitly indicated unless a non-standard sort order is used.

Default values can have interesting implications for domain-specific data sets. Continuing with the previous example, suppose that we had localised English voice-data resources that used vocabulary appropriate for the UK and the US countries. Thus, we are dealing with voice-data counterparts to the *domain-specific data set* category type. As mentioned in §3.6, writing-system and orthographic distinctions are irrelevant for voice data, the implication of which is that we can add a domain qualifier without ever needing writing-system or orthography qualifiers. So, in this

[34] I do not actually have any specific knowledge regarding English Braille orthographies. I am assuming there are some orthographic difference in this example.

example, we end up with “en-UK” and “en-US”. But because of unmarked defaults in relation to text, these turn out to be exactly the same identifiers used for the text-data domain-specific data-set categories.

Generalising, when dealing with voice and text data sets that are localised for a particular country domain and the text using unmarked writing systems and orthographies for each country, the identifiers needed for the voice data and text data would be the same. This would apply in the most likely scenarios in which localised voice and text data would be used.

Default values and the ability to assume implicit semantics are important for two reasons: the all for considerable economy in identifiers, and they also mean that most current uses of language and language-country identifiers conform to the framework being proposed. They do introduce some ambiguity, so that it may not be clear in a given instance of usage what level of distinction is intended to be conveyed. Also, given a category of a particular type, it is less obvious exactly what form an identifier should take. Use of implicit default semantics depends upon the ready availability of information for all written languages regarding defaults with respect to writing systems and orthographies. Otherwise, a user or developer may not know whether an individual-language identifier can also be used to indicate a writing system or even an orthography, or not.

It should be noted that, even if the implicit tagging of default semantics does introduce potential difficulties, the proposed model should benefit us overall since it gives us a framework with which to better understand distinct kinds of language-related categories we need to identify and to be discerning in regard to the nature of the actual content and resources that we tag. Without a model, we would not know to ask whether or not an identifier for a given individual language can be taken to imply more.

5. Special application scenarios

Before drawing to a close there are a few additional considerations that need to be given to certain particular application scenarios.

5.1 Transliteration

As discussed in §3.2, transcriptions can be treated as variant writing systems (or orthographies, if no one convention for spelling is assumed) of a given language. Transliteration systems present slightly different considerations. A transliteration, in the sense assumed here, is a reversible mapping of characters in one script onto characters of another script.[35] As a result, a transliteration system is language-independent. The identity of a transliteration system involves only the source script, the conversion (target) script, and the particular choice of conventions (since more than one transliteration system may exist for a given pair of scripts).

Given transliterated data, since the conversion script can be immediately determined by inspecting the characters in the data, it is, perhaps, not necessary to indicate the conversion script in an identifier. On the other hand, both scripts need to be indicated when identifying a transliteration process. Since the identity of the conversion conventions is needed independently, though, that identity may imply the scripts involved. For instance, “ISO 9:1995” implies transliteration from Cyrillic script into Latin script.

[35] This is the sense of the term *transliteration* used by ISO/TC 46. See http://www.elot.gr/tc46sc2/purpose.html for further details.

The best form for identifiers for transliteration conventions, then, probably combines three pieces of information: the source agency, an unambiguous name or designation used by that agency for that transliteration convention, and a version.

What is key to note is that an individual-language identifier is not relevant for the identity of a transliteration process. On the other hand, for a given instance of transliterated data, a particular language is involved, and the distinction to be made can be treated as a writing-system distinction. It may be appropriate, in this case, to identify that writing system by combining an individual-language identifier with an identifier for the transliteration convention of the form described above.

This discussion has assumed a definition of transliteration that is language-independent. If a script-to-script conversion is particular to a given language, then converted data can be treated as being in an alternate writing system, but a software resource for the conversion *process* represents a different type of category. I leave this detail for further consideration.

5.2 Typographic variations

For a given script, there may be variations in how certain characters or character sequences are written. In many cases, these variations are at the discretion of the individual scribe. Occasionally, though, there are preferences that apply across a culture.

For instance, both Serbian and Russian are written with Cyrillic script, but they use different shapes for italic forms of certain characters.

Typographic variations that apply throughout a language community (or, at least, the portion of the community that use a given writing system) can be handled as part of what is specified at the level of writing system. Since typographic variations are an aspect of script-rendering behaviours, this conforms to the definition given in §3.2.

These conventions may span multiple languages, however, and it would be inefficient in terms of both font resources and development process to duplicate support for these conventions for each writing system involved. Therefore, another useful category type to include in a complete model would be a category type for collections of writing systems that share typographic conventions. This category type could be called a *writing-system typographic-variation group*, or a *writing-system group* for short. Note, though, that this category type is relevant only for font and font-technology developers.

I envision, then, that software would tag text data using writing-system (or derivative-category) identifiers and that, in the rendering process, software would map these identifiers to writing-system-group identifiers. Within a font, then, the latter would translate into setting various attributes ("features") that trigger certain glyph transformations.[36]

This has assumed typographic variations that apply throughout the portion a language-community that uses a particular writing system. There is a question as to what to do if different variations are used across large portions of a single language community. For example, there are alternate shapes used for the character U+014A LATIN CAPITAL LETTER ENG. For some languages, one of these may be preferred by the entire community, but it is possible that a language community might be split, half using one shape, and half using another. The question is how this can best be handled. Following the definitions, this constitutes a change in rendering behaviours, implying a change in writing system. It may be expensive to use that mechanism for this one

[36] I believe that the "Language System Tags" defined by Microsoft as part of the OpenType specification are intended to be used this way, though this is not entirely clear to me.

detail, however. A completely different option might be to use variation-selector control characters, but this also has its own drawbacks. I leave this for further investigation.

5.3 Other considerations

As we look through the many components that make up complete software systems, we will find additional processes or resources that may have their own unique requirements with regard to identification.

Consider keyboard input methods, for example. Asian input method editors are designed for particular writing systems, but most keyboard input methods can be useful for multiple writing systems. For example, a "US English" keyboard layout can be used for writing systems of many languages besides English. This may suggest a different kind of collection of writing systems, but the situation is not as simple as this since keyboard layouts typically have a non-linguistic culture-dependent variable associated with them: a currency symbol. I leave issues regarding category types, identifiers for keyboard layouts and mappings between layouts and writing systems open for further investigation.

There are also situations in which special-case identifiers are needed. For example, the language of an information object may be unknown, or perhaps even unknowable. Also, language identity may be irrelevant for some data. This could apply, for instance, to mathematical formulas or even to some proper names. Also, in some application programming interfaces, there may be a need to specify "default" as a language or writing system (etc.) identity. ISO 639-2 currently includes one identifier of this sort, [und] "Undetermined". Others may also be needed as part of a complete system of identifiers.

Finally, automated language detection algorithms introduce another interesting consideration: confidence levels regarding accuracy of the language-related category identifiers applied to information objects. This implies a possible need for meta-metadata to indicate whether the language-related category attribute was generated by hand or by an automated process, and what confidence level can be assumed. I leave such issues open for further investigation.

6. Summary

We have looked in detail at a proposed model of ontological categories related to language that are relevant for IT purposes. We have seen that several types of category need to be recognised, and that these categories stand in certain relationships to one another. These relationships have certain implications with regard to the morphology of tags.

As mentioned at the outset, this proposal is tentative, and is offered as a starting point for discussion. Indeed, there have been a number of issues left open for further consideration. It is hoped that stakeholders in industry will undertake a critical analysis of this proposal to see where it works and where refinements are needed. Ultimately, it is hoped that this endeavour will lead to a complete and adequate system of identifiers that meet a very broad range of IT needs and that is embraced throughout the IT industry.

As mentioned, a number of issues remain for further investigation:

- Is there a need for a mechanism whereby an identifier space for individual languages can be partitioned to allow for individual-language categories based on different operational definitions and, perhaps, controlled by different agencies? (See §3.1 for discussion.)
- Should different traditions of phonetic transcription based on Latin script be treated as distinct writing systems or as distinct orthographies? (See §3.2 for discussion.)

- How should sets of orthographic conventions that are common to several countries be identified? (See §3.3 for discussion.)
- Can sort orders (when not implied by a writing system or orthography) be handled in terms of domain-specific data sets, or do they represent an additional, orthogonal parameter? (See §3.4 for discussion.)
- Exactly what super-ordinate categories of each type are needed? (See §3.5 for discussion.)
- How real is the need to distinguish pronunciation variants of languages, and to what extent can or should sub-language variants be handled in terms of the *domain-specific data set* category type? (See §3.6 for discussion.)
- How should historical language data be handled? (See §3.7 for discussion.)
- There are open issues in relation to locales and the relation to language-related categories, though these are probably outside the scope of the issues under consideration here (albeit, still closely related). (See §3.8 for discussion.)
- How should language-specific transliteration/transcription conversion processes be identified? (See §5.1 for discussion.)
- How should minor typographic variants used by major sub-communities of a single language community be handled? (See §5.2 for discussion.)

Of course, there are very likely other open issues that have yet to be identified.

7. References

Alvestrand, H. 2001. *Tags for the identification of languages.* Internet Engineering Task Force (Request For Comments (RFC) 3066. Also designated as Best Current Practice (BCP) 47.) Internet Engineering Task Force. Available online at http://www.ietf.org/rfc/rfc3066.txt.

Berlin, Brent, and Paul Kay. 1969. *Basic color terms: Their universality and evolution.* Berkeley: University of California Press.

Constable, Peter, and Gary Simons. 2000. *Language identification and IT: Addressing problems of linguistic diversity on a global scale.* (SIL Electronic Working Papers, 2000-001.) Dallas: SIL International. Revised version of paper presented at the 17th International Unicode Conference, San Jose, CA. Available online at http://www.sil.org/silewp/2000/001/SILEWP2000-001.html.

Constable, Peter, and Gary Simons. 2002. *An analysis of ISO 639: Preparing the way for advancements in language identification standards.* Dallas: SIL International. Presented at the 20th International Unicode Conference, Washington, D.C. Available online at http://www.ethnologue.com/iso639/An_analysis_of_ISO_639.pdf.

Fielding, R.; J. Gettys; J. Mogul; H. Frystyk; L. Masinter; P. Leach; and T. Berners-Lee. 1999. *Hypertext transfer protocol – HTTP/1.1.* (Internet Engineering Task Force Request For Comments (RFC) 2616.) Available online at http://www.ietf.org/rfc/rfc2616.txt.

Grimes, Barbara F. 2000. *Ethnologue*. 14th edition. 2 volumes. Dallas: SIL International. Web edition available online at http://www.ethnologue.com.

International Organization for Standardization. 1995. *ISO 9:1995, Information and documentation—Transliteration of Cyrillic characters into Latin characters—Slavic and non-Slavic languages.* Geneva: International Organization for Standardization.

International Organization for Standardization. 1998. *ISO 639:1998(E/F), Code for the representation of names of languages.* Geneva: International Organization for Standardization.

International Organization for Standardization. 1998. *ISO 639-2:1998(E/F), Codes for the representation of names of languages—part 2: alpha-3 code.* Geneva: International Organization for Standardization. Available online at http://lcweb.loc.gov/standards/iso639-2/langhome.html.

International Organization for Standardization. 1997. *ISO 3166-1:1997, Codes for the representation of names of countries and their subdivisions - Part 1: Country codes.* Geneva: International Organization for Standardization. The ISO 3166-1 code lists are available online at http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/index.html.

Lakoff, George. 1987. *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago: University of Chicago Press.

Library of Congress Network Development and MARC Standards Office. 2000. *MARC Language Codes List*. Available online at http://www.loc.gov/marc/languages/.