Federation and modularization of terminology – DRAFT –

Introduction

As discussed previously in the section "…", it is unrealistic to assume that any single terminology could ever be developed that would satisfactorily cover descriptions of large and diverse groups like plants, fungi, or insects. Consequently, the design for a descriptive data system discussed so far explicitly enables individual researchers to locally define the terminology required for their studies.

However, having a large number of independently developed terminologies prevents data integration even where it would in principle be possible. Furthermore, the development of a satisfactorily terminology for feature-rich groups requires significant effort and time. It is therefore desirable to provide mechanisms that allow projects to share common terminologies. The development of a "standard terminology" has therefore been attempted several times (e. g., for plants the TDWG Descriptors subgroup convened by R. Pankhurst up to ca. 2000, www.plantontology.org, or Prometheus II, McDonald & al., submitted).

Federation models

The simplest case of federating descriptive data systems is that a number of projects voluntarily agree to share a common terminology and other resources. One server might supply the terminology, another server resources like images, and several servers host the descriptions. This model is especially attractive where several institutions form a close collaboration that has a supra-institutional project management. The parts of such a managed federation could either be considered a single project in which only the physical location of data is federated, or separate projects in which different people are responsible for the federated project parts.

The information model supporting managed federation model is relatively simple. In terms of xml documents, the various federated parts are simply included into a combined document. Each description from one of the description servers would be accompanied with terminology obtained from the terminology server and with resource objects from the image server that are used in the terminology or description. The overall management would have to provide mechanisms that the parts of the federation fulfill each others needs. For example, when improvements in the terminology are required, all description services may have to be informed and take appropriate actions in updating their data, and the image resource service must provide services for depositing resources required in new descriptions.

The applicability of managed federation models is, however, limited. While optional centralization is desirable, compulsory centralization is not. As Berners-Lee points out: "Traditional knowledge-representation systems typically have been centralized, requiring everyone to share exactly the same definition of common concepts such as 'parent' or 'vehicle'. But central control is stifling, and increasing the size and scope of such a system rapidly becomes unmanageable." (Berners-Lee & al. 2001). To increase the overall interoperability and the productiveness in creating digital descriptive data, it is desirable that description providers may unilaterally decide to use public terminologies without having to enter into a management agreement with the providers of these terminologies. This situation differs from managed federations in that the provider of terminology does not know about the consumers of the data, but must still adhere to rigid design and versioning principles in providing a terminology that can be used as a "standard". Much of the following discussion addresses this situation. Managed federation projects will, however, also benefit from mechanisms intended to support unmanaged federations of terminology.

Terminology modules

The design of an information model for sharing descriptive terminologies should provide for local (= proprietary) terminological definitions and optionally allow the use of external terminology definitions. This is very similar to the view that an external terminology should provide extension mechanisms for adding local elements – except that the external terminology is not required.

To encourage the widespread adoption of standard terminologies it seems further desirable to provide for the concurrent inclusion of multiple external terminology "modules". Limiting the design of the information model to a single external terminology (in addition to a local terminology module, Fig. 1 a), would impose an all-or-nothing constraint. The competition between different terminological definitions will be much improved if it is possible to link a description project to multiple terminologies, picking the best part of each (Fig. 1 b). Standardization of terminology would then be the result of voluntary choices and agreement on convergence due to evolutionary processes.

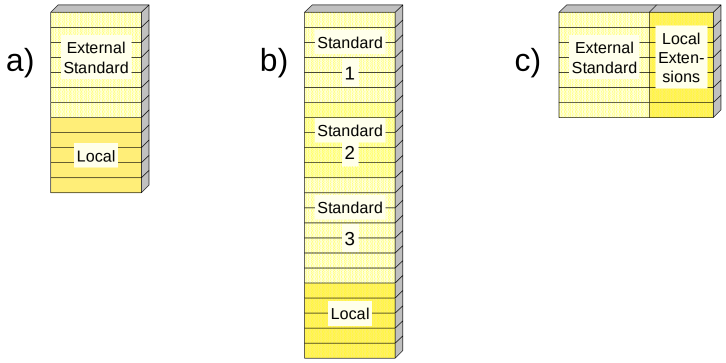

Figure 1. Options for federating, modularizing, and extending descriptive terminologies. a) The project uses a single external standard terminology plus locally defined extensions. b) A modular design integrating multiple standard terminologies and local extensions providing additional terms (characters, etc.). c) The project uses an external standard terminology, each term of which has been locally extended, e. g. to support other languages.

If multiple terminology modules are introduced to the model, the locally defined terminology could either automatically become another module usable by other projects (symmetric design), or it could remain distinct from a terminology module intended for federation. Only terminology modules explicitly designed as a reusable standard would then be available federation. The advantage might be that such projects presumably are more careful regarding publishing, versioning, and evolving or refactoring their terminologies.

Some potential use cases involving federated terminologies are shown in Fig. 2. An important point is that besides accepting external terminology modules, it may also be desirable to define the relations between one terminology (perhaps a local one) and another. Many terminological definitions in independently developed terminologies may be sufficiently identical for the purpose of data integration and comparison.

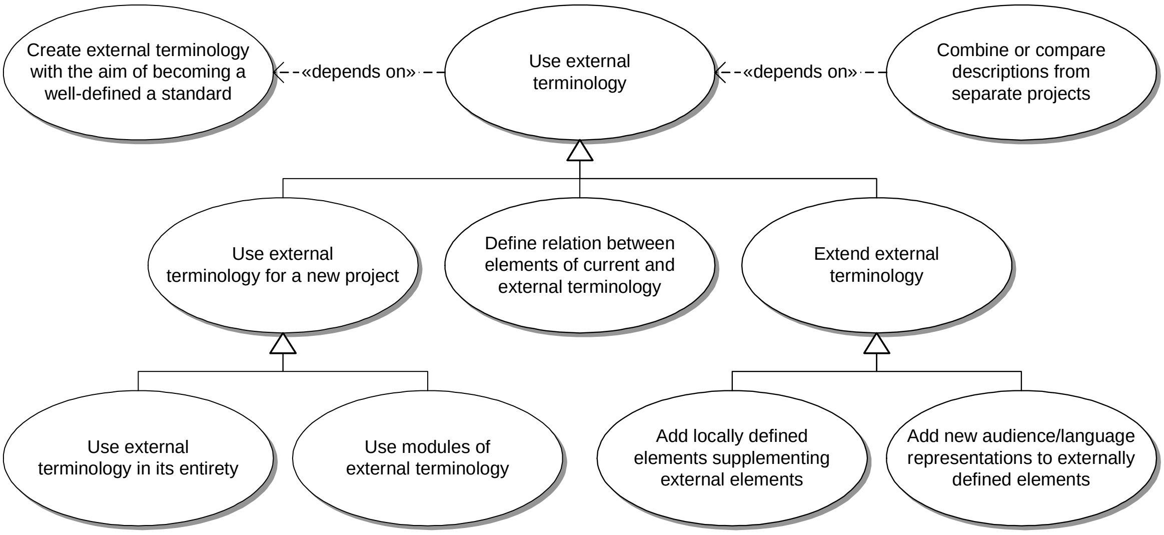
Create external terminology with the aim of becoming a well-defined a standard
«depends on»
Use external terminology
«depends on»
Combine or compare descriptions from separate projects
Use external terminology for a new project
Define relation between elements of current and external terminology
Extend external terminology
Use external terminology in its entirety
Use modules of external terminology
Add locally defined elements supplementing external elements
Add new audience/language representations to externally defined elements

Figure 2. UML use case diagram for some use case that involve the use of external (federated) terminologies. (The «depends on» stereotype is not available in standard UML use case diagrams but has been introduced here.)

Components of terminology

Extending an external terminology with local definitions may occur through the definition of additional terms (Fig. 1 a, b), or through extending terms imported from the external standard terminology (Fig. 1 c). The latter case occurs, e. g., when additional language or audience representations are added. Obviously, extending the terminology objects bears the danger of changing the semantics in a way incompatible with the concept of the original term. A discussion about which components of a terminology object may or may not be changed is thus mandatory.

The components of terminological objects may be classified into *definitional* (or 'essential'), *presentational*, and *assumptional*. Only a small part of a terminology is strictly definitional. Examples are the measurement scale of a character or a frequency value range for a frequency modifier term. Strictly presentational components are the sequence of characters or the wording definitions for natural language reports. An example for an assumption is whether states within a character are assumed to have an inner order (ordinal scale) or not (nominal scale). The assumption is not definitional insofar that the character and its states may be reliably recognized without it. However, it has significant influence on the outcome of statistical or phylogenetic analyses, and different researchers may want to base their analysis on different assumptions.

The major part of terminological definitions, such as labels and definitional text for concepts, characters, states, etc. are, unfortunately, a mixture of definitional and presentational components. This is most obvious in a multilingual situation. The English representation may be seen as an "international" definition and the other languages as presentations for non-English speakers. However, to those speakers, the other language representations are the only means of being informed about the definition and their coding of data will depend on the local representations, not on the "international" definition.

Whereas few problems arise when centralizing strictly definitional parts of the terminology, it is desirable to be able to locally change (extend or even override) presentational and assumptional parts. The major problem is that no method exists to express semantic definitions of terms independent of language. Even though ontology languages like OWL map concepts to language-independent URIs, they still express the ontological concepts only of a specific language. Very few terms in two languages have exactly the same circumscription and can be used interchangeably. For example, the term "bright" may be translated to German: "hell, glänzend, blank, leuchtend, strahlend, klar, durchsichtig, heiter, klug, munter, fröhlich", all of which have circumscription matching only partly with the English term – "strahlend" may also mean radioactive.

As a consequence, the most central part of the definition will always also include presentational aspects. It would be highly undesirable to centralize all labels and definition text and consider them unchangeable. However, whereas purely presentational or assumptional parts may require changes that contradict the original definitions, the mixed definitional/presentational parts may only require extensions by providing additional languages. If the standard terminology provides English labels and definitions, local copies may add German, Chinese, French, Japanese, Spanish, etc. representations, but may not be permitted to change the centralized English representation locally.

Terminology modules and class hierarchy

It is conceivable to create a hierarchy of terminology modules (= sets of terminology elements) that follows the taxonomic hierarchy (see Fig. 3).

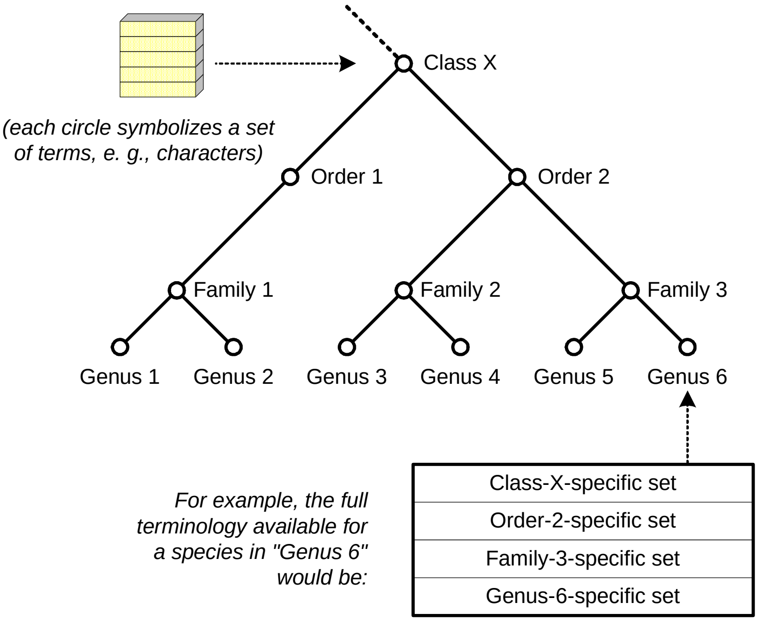

Figure 3. Example for a hierarchy of terminology modules that follows the taxonomic hierarchy. Additional taxonomic ranks may be present above, below, and in between those depicted here. Note that even species-specific terminology would be conceivable, e. g., to distinguish infraspecific taxa.

Although the model is attractive, it has several limitations:

- Phylogenetic classification is an area of active research, and the taxonomic hierarchy in many biological groups is not stable. Changes in a taxonomic hierarchy that defines usable terminologies would be difficult to implement once thousands of researchers would use such a central terminology system on the internet.
- The characters that are desirable at a higher level for the purpose of identification are not necessarily phylogenetically informative. A purely phylogenetic design of the taxonomy-dependent hierarchy is therefore not possible. For example, the vegetative stage of a fern like *Marsilea quadrifolia* L. may easily be confused with a flowering plant. Thus, even if leaf size and shape are too variable to be used for phylogenetic purposes, it is desirable to have them at a very high taxonomic rank, to support vegetative identification without prior knowledge of taxonomy.

- The scientific process of revising taxa is a bottom-up process. The most urgent need for terminology and digital descriptions is present at the level of genus or family. It would be unproductive to postpone using advanced computer-supported description software until the taxonomic tree is stable and the terminology modules for the higher taxonomic levels have been agreed upon.

Despite these limitations, a hierarchy of terminology modules designed for taxonomic groups is desirable and should be supported in the information model. At the moment, however, the taxonomic hierarchy should not be a required element in the organization of the terminology. Instead, it may be used to label and organize terminology modules that are then manually selected and combined in a project. Judicious use can limit the danger that may result from changes in parts of the phylogenetic classification that are poorly understood. For example, it may be desirable to skip a poorly defined order rank and rather duplicate a few characters in multiple family terminologies, or reversely to define the order as the lowest level of the terminology module used. Further, similar to taxonomy-specific standard terminology modules, terminology modules specific to methods or instrumentation could be defined and standardized. The complete terminology for a descriptive project would then be a combination of desired terminology modules (Fig. 4), perhaps with a local extension added.

Class X:	Morphology	Anatomy	Ultrastructure	Physiology
Order 2	Morphology	Anatomy	Ultrastructure	Physiology
Family 3	Morphology	Anatomy	Ultrastructure	Physiology
Genus 6	Morphology	Anatomy	Ultrastructure	Physiology

Figure 4. Combining multiple terminologies ("character definitions") can also be useful to combine characters defined for different methods and add them to the current project as needed.

Models to support multiple distributed terminologies

Three basic approaches to connect local descriptive data with standardized terminologies can be distinguished:

- The **namespace model**, in which the standard terminology resides entirely on the internet and is only referenced in the local terminology. A local cache may be present, but no local changes or extensions are possible (Fig. 5, right side).
- The **template model**, in which a standard terminology is copied to a local terminology and can then be changed. Provided some kind of identifier remains unchanged in the copy, the identity of origin may then be used for data integration. However, without human control the local changes may substantially change the semantics of the terminology up to the point where data integration is no longer sensible (Fig. 5, left side).
- The **declarative model**, in which the terminology is defined locally, but the developer declares that the definition of a given term (character, state, etc.) follows a published standard. This may be achieved by citing a standard identifier or reference, version, plus a specific code for each term from the standard.

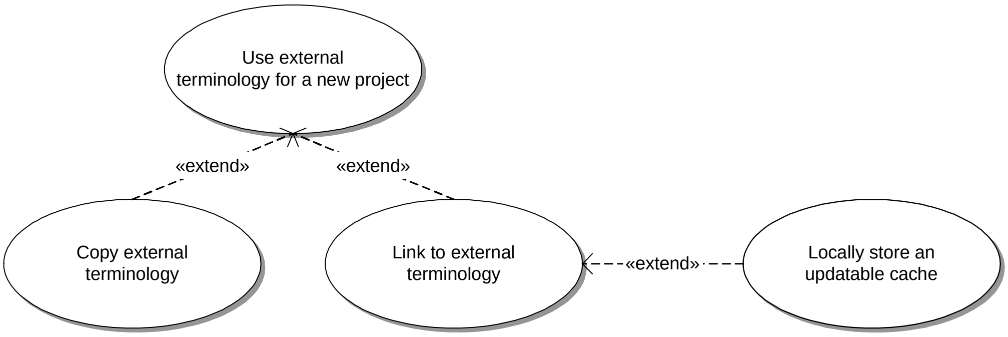
Use external terminology for a new project
«extend»
«extend»
Copy external terminology
Link to external terminology
«extend»
Locally store an updatable cache

Figure 5. External terminology may be copied or linked, the latter optionally with a local cache. Compare also the general use case diagram (Fig. 2).

Namespace model: Standard terminologies could reside on multiple servers on the internet and could be used directly from there (Fig. 6). This is similar to the use of multiple XML namespaces in a single XML document. Given that online internet connections may be expensive, unreliable, or even unavailable (e. g., on a notebook in the field), a mechanism to locally cache external terminologies would be desirable.

Using a namespace model, a standard terminology module would always be included in its entirety. This may be acceptable if each standard is split into small modules (e. g., separate modules for methods/instrumentation) so that the amount of unnecessary terminology that may confuse users is minimal. Alternatively, the information model could provide a local mechanism that allows defining subset views on the standard terminology.

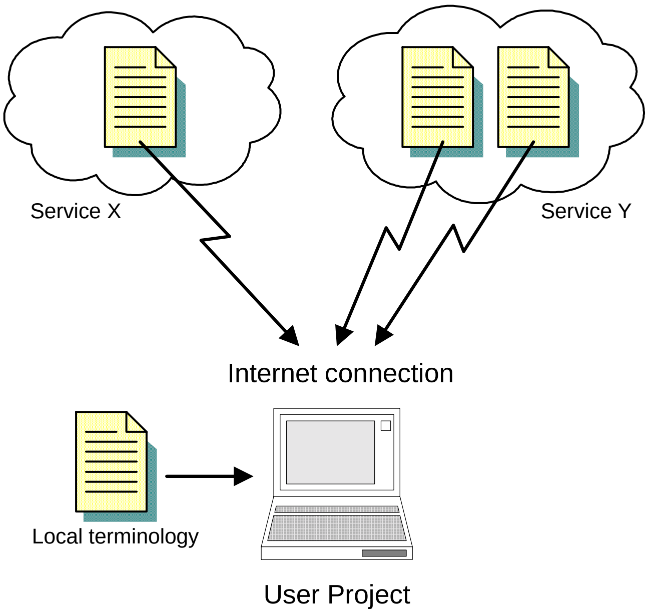

Figure 6. Network namespace model for federated terminologies. Multiple standardized terminologies are stored on the internet and used directly from there. Only terms not defined in a standard are stored in a local terminology.

The disadvantages of the namespace model are:

- The standard terminology modules would have to be available before the work on a project begins. It is difficult to combine this model with locally defined terminologies. If local terminologies overlap with only recently developed standard terminologies, the descriptions have to be ported to make use of the new standard terminology. The model itself provides no mechanism to do this gradually.

- A similar problem may arise, if a new version of a standard is published. A new version that is not fully backwards compatible would not replace an existing standard, but would be added as a new namespace. Changing the referenced standard itself is feasible only in very limited circumstances, since any substantial changes would invalidate the descriptions that use the previous definitions.
- Standards could be published only electronically. This may be acceptable if all programs use a common exchange standard format. As long as multiple formats are used, and given the ongoing importance of printed publications as long as electronic publications are too unstable to guarantee retrieval at a future date, this is however undesirable.

Template model: If one or several standard terminologies are used in a new project, they can be copied from templates that are available from a library of standard terminologies (Fig. 7). To trace the definition of a character back to the standard template it originates from, an explicit mechanism such as a Globally Unique Identifier (GUID) is required to remain unchanged in each term.

Once a template is copied into a local terminology, it can (and usually needs to be) changed. In these cases great care must be taken that the changes do not lead to situations where the human readable definition in character or states contradicts the semantics of the original definition in the standard used as a template. The developers of terminology are ultimately responsible that the terminological concepts perceived by users using the terminology for coding and identification remain sufficiently similar to the concepts defined in the standard terminology template.

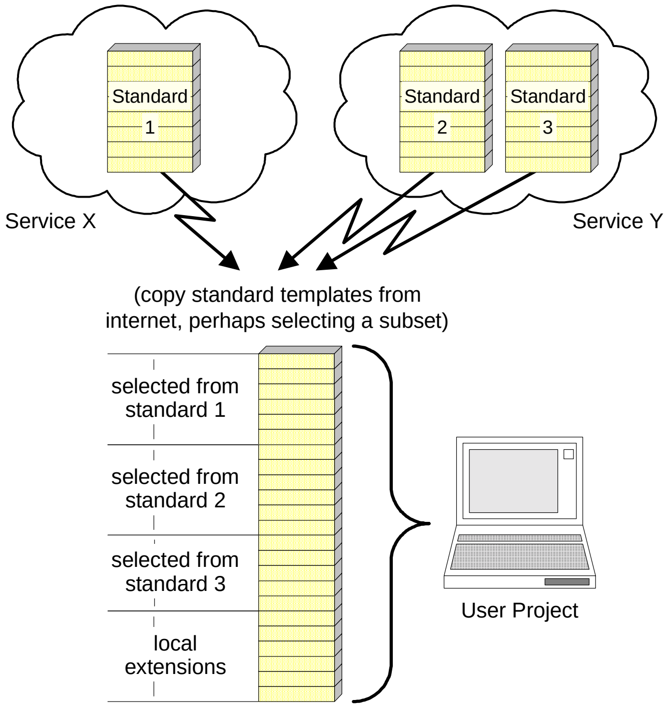

Figure 7. Template model for federated terminologies. Several terminology modules are copied from templates and can then be changed similar to local definitions.

Declarative model: In this model any terminology is primarily developed locally. Wherever possible, the developer adds an explicit declaration that the concept of a local character or state conforms to the concept of a character or state in a standard terminology (Fig. 8). The declaration should consist of an identifier or a reference for the standard, the standard version, and a reference to the individual term. These elements may be combined, so that a single Globally

Unique Identifier (GUID) for each term includes the reference to the standard and version. The standard could be identified through a URL, or through text citing a printed publication. The advantages of this approach are:

- Printed as well as electronic standard development could exist side by side.
- A smooth transition of existing data sets towards increasingly standard-conforming data is possible, since the declarations can be made individually for single terms (character, states) rather than being restricted to entire terminologies.
- The process explicitly supports the process of migrating from existing terminologies to newly developed standard terminologies, or from older to newer standard versions.

Disadvantages are:

- No automatic discovery mechanism for possible relations to standards is anticipated. The machine readable data integration mechanism depends entirely on human comparison of local and standard concepts
- Develop a local terminology for a given group involves significant work and often many revisions to correct for initial errors in the terminology.

These points can be addressed by combining the declarative model with a template model, copying a ready-to-use terminology module, but maintaining the publicly visible declarative reference. If the designers of the terminology detect that they are changing a term in a way that the local and the standard concepts differ, they may remove the declarative reference to indicate this.

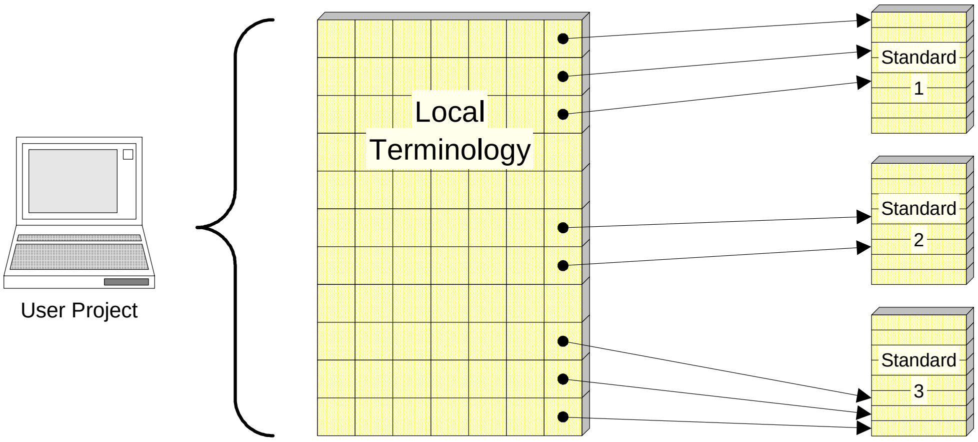

Figure 8. In the declarative model to support federated terminologies each term of the local terminology contains, among other data elements, an optional reference to a standard terminology. This reference is set by the designers of the terminology to declare that the local concept is identical with the concept in the standard. The standard may be available directly in electronic format, or may be published in a printed publication. If the declarative model is combined with a template model, the reference will already be set for those parts copied from a standard template.

Proposal

The combination of declarative and template model allows to migrate individual terms from existing terminologies towards standard conformity, but profit from the work that went into standard terminology (including presentational or assumptional elements) when using external terminology modules as templates. By referring to published and or standardized core terminologies it will be possible to create federated descriptive data collections, where multiple independent sites store descriptions that can be used together. The use of Globally Unique Identifiers (GUIDs) even allows to directly join terminologies, without online access to the standard terminologies to which they ultimately refer to (Fig. 9).

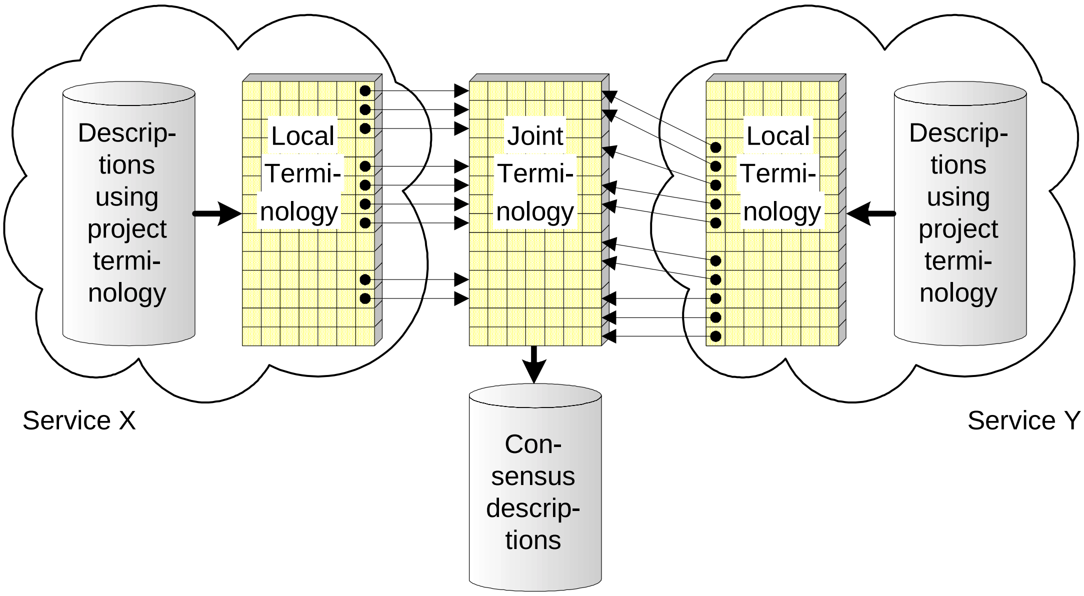

Figure 9. Consensus terminology created by a join of multiple terminologies from multiple sites on the internet. The item descriptions can then be used and queried across database borders. The join shown is an outer join, so that no descriptors are dropped. Only the matching descriptors can be used together. Alternatively, the terminology could be reduced to matching descriptors (inner join).

In the future it is to be hoped that a large library of reusable and tested terminology modules for a wide variety of biological groups and methods can be created. Not all of the terminologies need to be declared a "standard"; this could be an evolutionary process of acceptance or demand of standards. Researchers starting on new groups of organisms could then expand and revise existing definitions rather than start from scratch.

To my knowledge no library exists so far. Even for the DELTA standard only a handful of reusable character definitions can be found on the web, since most "DELTA" data are actually the binary, encoded Intkey data usable only for identification but not as a template for further character development.

Conclusions and Questions for SDD

1. SDD has no mechanism at the moment to define modules within the terminology of a project. Such a module mechanism is not strictly required when following the mixed template/declarative federation model. The terminology modules could simply be packaged as project containing only terminology.

If we decided to introduce terminology modules, we think beyond characters and states (those are left purposely left and thus are simple to split). How do we associate reusable concepts states, modifiers, or statistical measures with modules? In fact, how is this to be done when no separate terminology module mechanism exists and multiple terminology projects are combined? Each would define its own local frequency modifiers that would be synonymous? Clearly, these should be derived from another level of standard, but how?

2. The GUID references needed in the declarative model are not yet present. They should in principle follow the resource proxy model used for objects, class names, references, etc. However, it would only use a related object linking mechanism. The ProxyBaseType would not directly form the basis of the type derivation, since this type makes some assumptions about the external object being viewed with a minimized interface (= as a black box, not knowing about the

exact internal data structures). This is not appropriate when referring to external objects of types defined inside of SDD.

One reason why no placeholder type is yet present in SDD for these purposes is that I am undecided whether it is actually appropriate to place it on objects such as characters or states. I vaguely believe that we are interested in declaring identity of semantics for the purpose of data integration, which would be best defined on the level of the glossary entries. Thus two characters pointing to the same glossary entry, or rather to two glossary entries that are proxy objects of the same external term definition would be considered interoperable. In this scenario only the glossary entries themselves would have to be proxy objects with a GUID! Or do we need to refer to external character and state definitions more directly? What would be achieved by this? I look forward to a discussion on this!