Comments on STEMMA, by Tom Wetmore, July, 2012

Here are some comments stemming from reading the STEMMA documents. I found STEMMA to be a very well thought out and rather complete data model for genealogical and other family history data. I could not help but constantly compare it to my own DeadEnds model that has the same goals as STEMMA. Much of what I have to say is in the form of making comparisons between the two. Below are a bunch of unconnected comments.

Top Level Objects

There are many similarities between STEMMA and DeadEnds, starting with their choices of top level objects. STEMMA has five top level objects: Person, Place, Event, Resource and Citation. Similar to STEMMA, DeadEnds has Person, Place and Event, but DeadEnds combines STEMMA's Resource and Citation with the single Source object. Not unexpectedly I find the DeadEnds approach the correct one.

Keys

STEMMA uses keys to identify each top level object, and can also use keys to identify internal substructures within objects. In the STEMMA examples it seems important for keys to be able to convey some meaning to the user, so they can be constructed by the user. STEMMA has the notion of a dataset, which is a closed set of records, and keys need only be consistent within datasets. By contrast DeadEnds keys are UUIDs that are automatically generated upon the addition of each record. This means that every DeadEnds record created by any user using any compliant program until the end of time will have a unique key. DeadEnds also supports the idea of an additional user-assigned key that can be given to any top level object a user has an affinity for, for example, key ancestors in a pedigree, or key persons in an historical study. The idea of of using unique, "readable" keys for for datasets with thousands (hundreds?) of objects seems untenable.

Datasets

Because STEMMA keys do not have to be unique, STEMMA uses the dataset concept. Only within a dataset must keys be unique. In consequence datasets are required to have unique names, at least within a single transmission. There is no such limitation in DeadEnds, so the dataset concept is not necessary in archive files.

Relationships between Persons

The only high level relationships between STEMMA persons are done through "progentive parent" references. All others are done within Narratives, or via implications supported by roles in events (though this isn't mentioned, I believe, in the STEMMA documents). DeadEnds allows arbitrary one-to-one, typed relationships between any two persons.

Roles in Events

Both STEMMA and DeadEnds include multi-role Events as top level entities. In STEMMA the pointers go from Persons to Events; in DeadEnds the pointers go from Events to Persons. There is very little difference between these approaches as any memory model used when loading up data from either model would construct the reverse pointers if necessary for the software's algorithms.

Eventlets and Vitals

In addition to top level Events, DeadEnds allows vital events to be placed directly within person entities. To distinguish event structures from Event objects, DeadEnds calls these structures Vitals. STEMMA added the same idea in its latest version, calling the structure an Eventlet. The two concepts are identical.

Hierarchical Events and Dates Based on Events

STEMMA includes the interesting concepts of hierarchical Events and assigning dates to events based on the dates of other Events. These are novel features that as far as I know are wholly unique to STEMMA. I don't know whether the utility of these features justify their complexity, but they are certainly thought provoking. STEMMA also allows Events to occur over extended time frames, which is also a feature DeadEnds Events.

Conclusion-Based Emphasis

STEMMA seems intended for conclusion-based use. Each STEMMA Person seems intended to represent a different real person. As research progresses a user enters Resources, Citations and Events that refer to Persons or that allow new Person objects to refer to them. If a Person mentioned in the evidence (what the citations cite) is deemed new to the dataset, a new Person is created with pointers/references to the other objects. If the Person is deemed to already be in the database, then new reference structures are added to an existing Person.

The DeadEnds model also supports this conclusion-based approach, but it also supports the records-based or research-based approach. In that approach the user may create new Person objects as evidence is sifted, regardless whether the user believes the dataset may contain other Person objects that refer to the same assumed real person. Person entities used in this way are often called Personas. With this approach there may be many Personas within a dataset that may eventually be deemed to represent the same real person. In DeadEnds, when used in this records-based or research-based mode, trees of person records are constructed, where each tree-building operation represents a clear conclusion made by the user (now better termed a researcher) that a particular record or particular set of records (here record means a conventional item of evidence such as a birth certificate or census record) refer to the same real person. In this view each "person tree" represents a single real person.

To my mind this is one of the only significant differences between STEMMA and DeadEnds.

Emphasis on Hierarchical Representation via Nested Persons

The STEMMA document puts some emphasis on the idea that a purely lineage linked file might allow Person objects to be nested within one another in order to show ancestral relationships. I don't believe that anyone in their right mind would choose such a representation (and ancestor collapse would render it impossible) so I don't think the STEMMA document even needs to mention it. This is isn't an advantage that STEMMA has over other formats use the approach.

References Hold Non-intrinsic Values of Objects

STEMMA and DeadEnds both stress the idea that references are not simple pointers, but rather are sub-structures that may contain properties of the "pointing object" that exist only with respect to the context of the "pointed to" object. Thus an age or a name property might be placed in a reference between a Person and an Event, rather than in either of the objects.

Places

STEMMA and DeadEnds treat Places as hierarchical top level objects. DeadEnds allows Places to simultaneously belong to more than one hierarchy. I'm not sure about STEMMA in this regard. STEMMA allows Places to have date properties that can specify when that Place was valid. STEMMA Places span the range of concepts from individual houses up to countries. At present DeadEnds spans from small populated places up to countries, even continents, and thinks of addresses as something different. The STEMMA approach may be better here.

One thing I find anti common sense about the STEMMA model is that the Place object may contain Event references and Eventlets. Events certainly need to know where they occurred, by why does a Place need a list of the things that happened there?

Narratives and Rich Text Notes

STEMMA uses Narratives to provide rich, natural language based information, presumably intended primary for presentation in output documents, but useful for many other purposes. The textual body of a Narrative is plain UTF-8 text with markup consisting of the keys to top level objects or to substructures within them. Interestingly there is no markup allowed for concepts such as lists, titles, emphasis, and so on. The STEMMA documentation stresses that certain relationships that cannot implemented in any other way (say a relationship between two cousins where there is no information on how they connect) would be done by referring to the two persons via their keys in a Narrative. This I find rather unsatisfying, thinking that I would have to find important information that is not otherwise represented in the model, by reading the narratives. DeadEnds allows any one-to-one relationship to exist at the Person object level.

DeadEnds allows rich-text notes (probably to be replaced by a different markup technique), but does not allow markup based on object keys. However, this STEMMA idea is such a great one that I will likely copy it and add it to DeadEnds.

The Narrative is a great extension to the Note concept, in my opinion, one of the best ideas in the STEMMA model, especially with the object key markup it allows. I believe it should be extended with the type of “style-less” markup that was intended by the original release of HTML (before the style hackers started adding all the purely style-based tags to it.)

Resources and Citations versus Sources

STEMMA uses Resources and Citations as top level objects, whereas DeadEnds uses only Sources. I believe the DeadEnds approach is simpler. In DeadEnds citations do not occur in the model at all – they are strings created by extracting and formatting information from a hierarchy of Source objects. In DeadEnds each object, in fact each attribute of each object, may contain a source reference structure that points to a Source object. And each Source object may also have source references that point to Source objects further up one or more source hierarchies. I haven’t found any examples where a single hierarchy of Source objects, with the bottom objects in the hierarchy referred to by other objects via source references, is not the best way to go.

Person Names

My gut says the STEMMA approach to name substructures is too complex. DeadEnds treats names pretty much as pure strings with no internal structure. The only exception to this is that a substring of the overall string (Western cultures would use the surname) can be singled out for use as a top level sorting key, with the rest of the string being used as the secondary sorting key. I admit this might not be good enough.

Multiple Birth and Death Events

STEMMA treats birth and death Events as special, and allows only one each per Person. I don’t see any important reasons for this restriction.

Sex Values

What is wrong with male or female or unknown, or M, F or U? Using 0 or 1 is strange.