A Data Model for Micro-History Data

Tony Proctor, 19 Apr 2017

1 Introduction

These pages describe a generalised data model and source format for genealogy, family history, and micro-history: STEMMA®. This serves several different purposes, including:

- Definitive source format (see Glossary) for historical data.
- Import/export or exchange format (including over the Internet).
- Database load-format.
- Long-term preservation, and general backup format.
- Historical-information "document" format (see STEMMA Mark-up).

See FAQ page for specific questions.

This R&D project came about because I didn't want my family history data locked in to any single commercial product. Plus I didn't believe that existing data models were able to represent the depth and variety of my data completely and accurately because of its micro-history approach. STEMMA incorporates biological lineage (as in family trees), structured events, geography, source assimilation & analysis, and rich-text narrative to go beyond family history to address the micro-history of arbitrary people, places, animals, groups, and events.

Genealogy and family history — irrespective of whether you consider them to be the same or different — are part of the bigger circle of micro-history, alongside One-Name Studies, One-Place Studies, personal historians (as in APH), military history, house histories, etc. (see What is Genealogy?) Trying to compartmentalise these pursuits can be artificial when we're looking at real history, so why shouldn't there be a single, consistent approach to their data representation?

The STEMMA data model may be a source of inspiration to other people or groups with the same goals, primarily because it was developed independently and with minimal influence from other models and products.

This documentation constitutes V4.1 of STEMMA, and supersedes previous specifications.

2 Goals

Some of the primary goals of the STEMMA project were:

- Define a data model and source format to represent my existing family history data (including micro-history data) accurately and without having to bend any rules.

® STEMMA is a registered trademark of Tony Proctor.

- To make the complete data (including history, evidence, reasoning, and transcription) searchable in a structured way, not just of plain text.
- To be able to store copious amount of rich-text narrative in a structured way, including reference notes, semantic mark-up, transcription anomalies, and hyperlinks — not a simple *Notes* feature.
- To clearly indicate the source of any data, and to separate objective information from subjective inference and conclusions.
- To allow the data to be crafted by hand in the absence of a compliant software product.
- To make the representation as globally applicable possible (i.e. locale-independent and culturally neutral).
- To allow datasets to be validated for conformity to a well-defined schema using standard tools.
- To make the model easily extensible without having multiple versions of the main data schema.
- To make extensive use of modern data standards.

Rather than getting mired in arguments over whether the data should be lineage-linked, event-linked, or evidence-linked, STEMMA strives to be able to represent all aspects of the data and leave the flavour of analysis or presentation to the software modules that manipulate it. This means focusing primarily on a complete and accurate representation of the data that was found, and providing comprehensive support for representing related inferences and conclusions. It deliberately does not mandate any specific research process, or strive for compatibility with any existing software product or data model.

A *source format* is just a plain-text machine-readable definitive version of data that can be used for multiple different purposes. The term is analogous to *source code* in a programming context, which is a definitive representation of computer instructions that can be compiled for different machines, and in any locale. Most people will think of *file formats* when seeing this term, although *serialisation format* is the more correct technical term. A serialisation format includes other contexts where data is represented in a series of bytes, such as for transmission over a communications network.

A *data model* underpins the design of any source format. It dictates what entities need to be represented, and what relationships exist between those entities, but without specifying a particular syntax to use in a physical source format.

A run-time *object model* defines the structure of indexed data held in memory, and the software interfaces for accessing them. A standard model would allow run-time interoperability between products of different types (e.g. analysis or reporting) or from different vendors. It would also be required for connecting to Datasets published over the Internet, or in the ‘cloud’.

A data model hints at an object model but doesn’t actually define one. A run-time object model has to be optimised for data lookup and access, whereas a data model has to define a normalised copy of the data that is self-consistent

and has no duplication. If the data model were ever used to form the basis of a commercial product then a subsequent project could be to define an associated run-time object model.

Neither a standard data model nor a standard object model would mandate a particular database type or format, including in-memory architectures. That would be a choice for commercial product designers. In other words, a database model is describes a derivative data form rather than a definitive one.

In order to separate objective information from subjective inference and conclusions, the STEMMA data model has two notional sub-models: informational and conclusional (see Our Days of Future Passed — Part III).

3 Overview

STEMMA stands for “Source Text for Event and Ménage MApping”.

STEMMA was primarily a data model for long-term storage and data exchange — although it has since become much more. It therefore strived to hold data entities and their connections rather than any presentation information. There is deliberately no concept of font control, colouring, or date and numeric formatting. These are the responsibility of the software tools utilising a STEMMA Document.

The STEMMA source format is portrayed here using XML (eXtensible Markup Language) but this should be viewed as just one physical representation since the data model itself could be represented in a variety of ways (e.g. JSON). XML is a textual serialisation format designed originally by W3C to represent structured data in a generalised way, mainly to facilitate communication over the Internet. XML is a well-established standard that provides for automatic validation according to schema definition files. It also supports namespaces for versioning of the main schema and for formal extensions to the main schema.

STEMMA is designed to represent generalised micro-history data, including family history, as opposed to merely biological lineage (literally *genealogical* data). Genealogy is often associated with biological lineage, particularly in the form of family trees, whereas micro-history is concerned with more types of subject entity — such as persons, places, animals, and groups — all events in the history of those subjects and arbitrary connections between them. Family history is therefore a form of Micro-history. This is a far more ambitious goal and STEMMA strives to find a balance between a strong formal approach and the flexibility to accommodate the unexpected and human contributions (i.e. narrative work, reasoning, and general notes).

STEMMA gives Persons, Places, Animals, Groups, and Events equal billing as top-level entities, thus providing social, geographical, and temporal dimensionality to the data. In addition, there are top-level entities representing narrative work, sources, citations, source analysis, and attachments, which contribute an informational dimension. Each of these entities has a Key by

which they can be referenced from other entities, and each supports a strong narrative structure. This narrative is more than just a set of brief comments or plain-text notes since it can arbitrarily reference other entities by their Key. The embedding of references within narrative also allows a STEMMA viewing tool to easily generate hyperlinks to the referenced items, thus supporting drill-down in order to identify supporting evidence and reasoning, or to find the source of the evidence.

Most of these top-level elements are also hierarchical. A hierarchy for Persons or for Animals is expected (i.e. lineage), and some products already represent a Place-hierarchy where each place is connected to a broader parent Place. STEMMA goes further by giving Events a hierarchical structure, thus allowing longer-term Events to embrace shorter-term Events and so give them a more granular structure to help with the representation of history. Furthermore, it allows Groups to be hierarchical, and Citations to be hierarchical such that, say, two documents from the same collection can share common data along a chain.

Other noteworthy structural features of STEMMA include:

- Ability to represent not just biological lineage, or extended family (as in adoptive parents), but arbitrary connections between people who may not be part of the family at all. See The Lineage Trap.
- Powerful multi-language narrative feature allowing rich-text to be added to all entities, or used for complete research articles. Narrative subsections can be given attributes such as Surety and Sensitivity, and can distinguish information from inference. It can also embed computer-readable links to other entities, including other narrative, using semantic mark-up.
- Separate syntax to handle all top-level entity definitions (e.g. Event) independently of references to them (e.g. EventLnk). This allows both the definitions and the references to contribute information in a natural contextual way.
- Inheritance mechanism allowing entity data to be factored out to increase sharing and reduce duplication.
- Defined methods for declaring extended vocabulary, properties, and for formally extending the core schema.

Although STEMMA was initially conceived as supporting import/export or long-term storage of data, this quickly became a secondary feature. A result of its deep level of representation meant that no database-orientated product could adequately index it.[1] However, indexing it into memory, on-the-fly, meant that (a) full and efficient indexing was possible, (b) that no import/export was necessary as the definitive source format could be exchanged, and (c) that no special consideration was needed for long-term storage or backup of database content. The article Do Genealogists Really

[1] A simple but demonstrable example is the transcription of *Jos*^{h} (with a superscript 'h'), which is shorthand for *Joseph* but is invariably transcribed as *Josh*, and so is incorrect.

Need a Database? explains how reliance on a conventional database is folly, and introduces performance degradation, risk of corruption, incompatibility between different database vendors or proprietary schemas, and forces the need to invent other representations for import/export, etc. The Introduction section mentions ‘historical-data document format’, meaning that a STEMMA Document can be shared and transmitted just as a word-processor document or spreadsheet might — you don’t need a database for viewing those!

4 Locale-independence

This section is concerned with the transportability of the data and of avoiding ambiguity caused by the loading of the data on a machine with different regional settings. This is distinct from providing foreign-language translations of text and of localised formatting of dates, numbers, etc. The format of data values within each STEMMA Dataset should be considered a computer-readable format even though it is text-based. The complete STEMMA Document is not directly concerned with presentation of data to the end-user, and so does not use your preferences for numeric formatting, date formatting, citation styles, etc.

The character set should be global which nowadays means UTF-8, and this is also the default with XML. Although an XML-like header could explicitly nominate a non-default character set name, this would put an onus for supporting all possible translations on the receiving software and could limit portability between different operating systems.

A small issue with UTF-8 is that some editors don’t acknowledge it. For most people reading this, the default character set in their computer account will probably be a Latin-1 set (e.g. ISO 8859/1 or Windows Latin-1). Unless the editor was smart and recognised XML, or BOM sequences in the start of the data, then it may render some characters incorrectly. It is possible to avoid 8-bit character codes in XML, and so avoid the potential ambiguity, by restricting codes to 7-bit ASCII and using either entity references (e.g. &) or character entities (e.g. €) for all other cases. The impact of this would be small since only developers would be looking at the raw representation.

Data values should be in a locale-independent format, similar to literals in the source code of a programming language. For this reason, it is sometimes called using a ‘programming locale'. This effectively means using a period in any decimal numbers (not a comma), all-numeric ISO 8601 format for (Gregorian-)dates (e.g. yyyy-mm-dd), and non-localised 1/0 for Booleans (e.g. for option selections). These conventions are good practice for the designer of any shared data, and are not specific to XML or to micro-history data. Again, just as with programming languages, this ensures transportability and that the data will be loaded (or compiled) identically by any compliant product in any locale.

It should be noted that tag values for types, subtypes, modes, and other taxonomies (e.g. Union, Birth, Marriage, etc) should be considered part of the data syntax, just as with element names and attribute names. They should

never be presented directly in the UI of a software product but rather should undergo a mapping to a meaningful term for the locale of current end-user.

The term *culturally neutral* is used in this specification and refers to the ability of STEMMA to represent data from different cultures. The specification must avoid assumptions about the structure of personal names, the types of possible unions (e.g. marriage), religious ceremonies, family units, inheritance of names, etc.

Although Time Zones (TZ) and Daylight Saving Time (DST) are usually applied to local clock times, they can also apply to local calendar dates. The importance for genealogy is going to be slim at best but the area should be clarified. ISO 8601 does not include any TZ designators — values are either 'local time' or relative to UTC (Coordinated Universal Time). Local date/times should be interpreted in the context of the data location rather than the current location of the user but this would only be significant when creating a timeline across TZ boundaries.

Issues of date format, such as numeric form, long text form, abbreviated text form, and month/day ordering, are issues for the user interface and are, therefore, controlled by end-user's regional settings. The stored dates must be independent of that.

It should be stressed that this section is concerned with computer-readable data. If, for example, a document image shows a written date, a transcribed version of that date can still be held as text in the data. However, if that written date can be interpreted then the format of the computer-readable value, including any error margins, is rigidly prescribed here.

Dates expressed in different Calendars — particularly ones that cannot be precisely converted to Gregorian dates — are usually a bit more challenging. Examples of other calendars include Julian, Hebrew, Islamic, Hindu, Persian, French Republican, Mayan, and Chinese. The STEMMA research notes under Dates and Calendars describe the requirement for a standard representation of dates from all calendars. The same goals as influenced the Gregorian ISO 8601, e.g. unambiguous computer-readable form that is locale-independent, resulted in the encoding used for STEMMA Dates.

5 Notation

The following notation will be used in each section in order to indicate the model structure, including its cardinality, ordinality, and optionality.

- **[Optional]**

An optional collection of elements or sequence of attributes.

- **{ Grouping }**

A mandatory collection of elements. The braces are implied if square brackets have not been employed but they are sometimes necessary where a collection of elements can be repeated.

- **| Choice |**
  An exclusive choice of elements. For instance: { choice1 | choice2 | choice3 }.

- **|| Choice ||**
  An inclusive choice of elements. For instance: { choice1 || choice2 || choice3 }. In other words: one or more of the choices.

- **... Repeated**
  Ellipsis following an element or a collection of elements indicates that it can be repeated a further 0-or-more times. Hence, “[term]...” represents 0-to-many instances while “{term}...” represents 1-to-many instances.

- **SUBSTRUCTURE**
  Indicates a sub-structure defined elsewhere in the document, e.g. NARRATIVE_TEXT. Giving common element collections and common attribute sequences their own name allows the structure to be factorised to improve both readability and consistency.

Here are a few examples for illustration:

{ a b c }	Mandatory collection of items a, b, and c. Any order.
{ a b } [c]	Mandatory collection of items a and b, optionally followed by c
{ a | b } [c | d]	Mandatory choice of a or b, followed by optional choice of c or d
{ a | b } ...	Repeated sequence of 1-or-more items, each of which is a or b
[a | b] ...	Repeated sequence of 0-or-more items, each of which is a or b

Identities
Note that [a]... (mandatory 0-or-more sequence) and [a ...] (optional 1-or-more sequence) are therefore equivalent. Also, [a || b] includes all combinations (a, b, ab, and none), and so is equivalent to [a] [b].

Although this notation uses XML syntax to show element values and attributes, it is designed to show the entity relationships and can be converted directly to a traditional ER diagram if necessary. Hence, there is no specific ordering implied between consecutive elements (at the same hierarchical level) or attributes in the notation. However, the order of the actual elements in a STEMMA Document should be treated as meaningful and preserved during import/export. With attributes — which function to characterise an entity or element — there is no associated ordering that needs preservation.

The text in all the STEMMA documentation capitalises the names of STEMMA entities and elements to distinguish them from the common usage of those English terms, e.g. Person as distinct from person, or Place as distinct from

place. It also hyphenates technical polynyms that may also be confused with words in common use, e.g. event-type, citation-element, person-property.

Some special element data and attribute value names are defined. These may be used as-is or used to qualify another data name by forming a suffix to it:

- **iso-date**= subset of ISO 8601 Gregorian date standard (i.e. yyyy[-mm[-dd]]).
- **iso-datetime**= full ISO 8601 Gregorian date and time, i.e. yyyy-mm-ddThh:mm:ssZ.
- **narrative-text**= text incorporating semantic and descriptive mark-up (including nested <Text>), and <Ucf> elements.
- **[*-]orig-text=** Original text, possibly with alternate meanings or spellings identified. Text incorporating descriptive mark-up, and optional <Alt>, <Ucf> elements.
- **std-date**= STEMMA date referenced in specific calendar, analogous to **iso-date**. Calendar can be specified by optional prefix, as suggested under Dates and Calendars research notes, or implied by the syntactic context. Also accepts "?" as per **value**.
- **std-fulldate**= full STEMMA date reference in specific calendar. That is, with a granularity of only one day. Also accepts "?" as per **value**.
- **ucf-text**= text with possible uncertain characters. Text incorporating optional <Ucf> elements.
- **[*-]value**= representation of a value of a particular data-type (e.g. in mark-up, or a property, or a date). The special value "?" explicitly indicates *not known* or *not provided* — obviously important for deaths.

6 Document Structure

The top-level structure of the STEMMA Document is as follows:

```
<?xml version="1.0"?>
<Datasets xmlns='http://stemma.parallaxview.co/2017-04'>
        <Header>
                <Created> iso-datetime </Created>
                <Product>
                        <Id> product-id </Id>
                        <Name> product-name </Name>
                        <Version> product-version </Version>
                </Product>
                { <Language> code </Language> |
                  <Locale> code </Locale> }
                [ TEXT_SEG ] ...
        </Header>
        { <Dataset Name='name' xmlns:prefix='URI' ... >
                [ <Content>
                        <Created> iso-datetime </Created>
                        <Author> content-author-name </Author>
```

```
                <Version> content-version </Version>
                [ <Copyright> copyright-notice </Copyright> ]
                [ <LastModified> iso-datetime </LastModified>
                <ModifiedBy> name </ModifiedBy> ]
                { <Language> code </Language> |
                  <Locale> code </Locale> }
                [ <Counters>
                        { <Counter [Tag='tag']> integer </Counter>} ...
                </Counters> ]
                [ TEXT_SEG ] ...
            </Content> ]
            [ EXTENDED_PROPERTIES ]
            [ IMPORTS ]
            DATASET_BODY
        </Dataset> } ...
</Datasets>

IMPORTS=

<Import>
        <entity-type Key='key' [Abstract='boolean']/> ...
</Import>
```

The URI for the default namespace is versioned in order to accommodate potential schema updates. The xmlns attribute of the <Dataset> element provides namespaces for custom types and other tag values. A discussion of extending partially controlled vocabularies may be found at Extended Vocabularies.

The <Dataset> element is an envelope for a self-consistent set of data. It has its own name, change history, and linkages. Although a STEMMA Document may contain a single Dataset, multiple ones can be concatenated in order to separate different family branches, or to isolate common Places, Citations, or Events. Under those circumstances, all Key names are local to their respective Dataset. Key references cannot span Datasets in the stored format but can do when the associated data is loaded into memory. The IMPORTS element controls explicit name imports to a Dataset.

The Dataset Header contains the name of the original author, the current version string (no prescribed format), and the date and author of the current revision. The associated <Text> elements (discussed below) can be used to maintain any change history beyond that.

An 'iso-datetime', as specified here, will have the format: <yyyy-mm-dd>T<hh:mm><tz>, i.e. numeric date followed by literal-T, followed by numeric time. If present then 'tz' is a time-zone offset from UTC, i.e. ±hh:mm. For instance, 2011-12-25T14:00Z or 2011-12-25T14:00+00:00. See Dates.

The Language or Locale designations deserve special mention here. The <Language> element takes an ISO 639-2 three-letter code for a default

language, e.g. “eng” for English. The <Locale> element provides a more detailed specification since that involves both an ISO 639-1 two-letter language code plus an ISO 3166-1 two-letter territory code, e.g. “en_GB” for British English. This is a subset of the POSIX locale format and was chosen for its simplicity as opposed to the IETF’s Language Tags (BCP 47) which are very similar but far more comprehensive, even though the latter are acknowledged by XML (see xml:lang attribute). It should be noted that these specifications do not make the Dataset specific to that language or locale (see Locale-independence). What it provides is a default language for the interpretation of narrative text. Each narrative <Text> element can provide an explicit Language or Locale but this Header element provides the default. An example situation where a Locale may be more useful that a mere Language is when the text contains an ambiguous date such as: 09-06-1956. Knowing the Locale would help clarify whether it was June-9 or September-6. If both are specified in the same element then Locale should take precedence because it is more specific.

The EXTENDED_PROPERTIES element defines the custom Properties that may be attached to Persons, Places, Animals, Groups, and Events.

The <Counters> element contains an indefinite number of integer counters that can be used for assisted generation of key values. Key values can be generated algorithmically from names and dates for subject entities such as Persons and Places, but it may be a little more difficult for Events, Citations, Resources, etc. Having a number of persisted counters should help support a simple sequential allocation scheme, or something more involved. Values begin at one, and increment by one after the current value has been utilised. The optional tag value can be used by an application to arbitrate on which counter to use.

Dataset Loading

Although the following section is not technically part of the STEMMA specification, it is included to help clarify why STEMMA does not have an explicit *Include* statement, and how this works given that it does not use a relation database or any other type of database engine (see Do Genealogists Really Need a Database?).

What a STEMMA Dataset does have is an *Import* statement, where the <Import> entity-type may be Person, Place, Animal, Group, Event, Citation, Resource, Source, Matrix, or Narrative. Imported Keys may be used as though they are defined in the current Dataset. It is an error if an imported Key is already defined. The end-user may require the hosting product to load full definitions from their respective Datasets — either in the current Document or elsewhere. That product may also cache the Keys defined in each Dataset in order to quickly identify which ones to load.

When modifying a Dataset then the Imports act like a 'forward' declaration for validation purposes. However, when viewing the Dataset contents (in a tree, timeline, map, or whatever) then they currently cause the other necessary Datasets to be loaded. There's a subtle difference between the two

modes, depending on whether a 'view' or 'edit' operation is being performed on a given Dataset, and this is similar to the difference between the 'compilation' and 'edit' modes for a software source module. This distinction is only the way that the current software has evolved, though, and it could quite easily do a multi-Dataset load every time.

When a Dataset has been modified then it is persisted back to the original file, and an associated directory (not a folder-type "directory") updated that records what entities are defined in each Dataset of every local Document. It is this directory that the loading process interrogates. In other words, the <Import> element says which entities are required but it doesn't indicate where from. There is deliberately no concept of an explicit "Include dataset-name" statement. When loading a dataset, a memory-resident table of unresolved entities is iteratively scanned to resolve all outstanding references: after the first load then it will include the entries from its Imports. The first entry in the table may require a second Dataset to be loaded, which in turn might resolve many outstanding entries from the table, but also add a few new ones for itself, effectively resulting in the loading of a "dataset tree".

6.1 Dataset Structure

The body of a Dataset has the formal structure:

DATASET_BODY=

[PERSON | ANIMAL | PLACE | GROUP | EVENT | CITATION | RESOURCE | SOURCE | MATRIX | NARRATIVE_TEXT] ...

The main entities in the body of a Dataset are defined by the following elements:

- <Person> — Properties and history of a particular person.
- <Animal> — Properties and history of a particular animal.
- <Place> — Properties and history of a particular place.
- <Group> — Properties and history of a particular group.
- <Event> — Description of an event that occurred at a particular place & time.
- <Citation> — Source reference information.
- <Resource> — Digital and/or physical artefact, such as files, images, objects.
- <Source> — Source description, assimilation, and analysis.
- <Matrix> — Cross-source correlation and analysis.
- <Narrative> — Larger items of narrative text, such as biographies, essays, and reports.

Each of these entities has an alphanumeric Key associated with it by which it can be referenced from other elements, e.g. **<Person Key='Tony123'>**. See Symbolic Names for more details.

Those top-level entities are primarily linked (i.e. ignoring narrative references, inheritance, and hierarchies) as follows:

Person → Event (for Birth & Death), Group, Source
Animal → Event (for Birth & Death), Group, Source
Place → Event (for Creation & Demise), Source
Group → Event (for Creation & Demise), Source
Event → Source
Source → Citation, Resource
Matrix → Source

These direct links from one entity to another must be unique, i.e. no duplicates. A reference to a Person, Place, Animal, Group, or Event can be identified purely by its target Key. Links to Resources and Citations are slightly different because those linked entities can be parameterised. However, the expanded URL of a Resource, and the source-type URI plus the effective parameter set (see Inheritance and Parameters) of a Citation, will identify corresponding unique links.

STEMMA distinguishes a "link" from one entity to another from an embedded "reference" to some real-world subject or notion. The latter may occur when generating a reference from an entity Key or when marking-up a textual reference and equating it with a given entity (see Semantic Mark-up). The following elements are defined for linking entities in this fashion:

PERSON_LNK=

<PersonLnk [Key='key']>
[PERSON_PROPERTY] ...
[TEXT_SEG] ...
</PersonLnk>

ANIMAL_LNK=

<AnimalLnk [Key='key']>
[ANIMAL_PROPERTY] ...
[TEXT_SEG] ...
</AnimalLnk>

PLACE_LNK=

<PlaceLnk [Key='key']>
[PLACE_PROPERTY] ...
[TEXT_SEG] ...
</PlaceLnk>

GROUP_LNK=

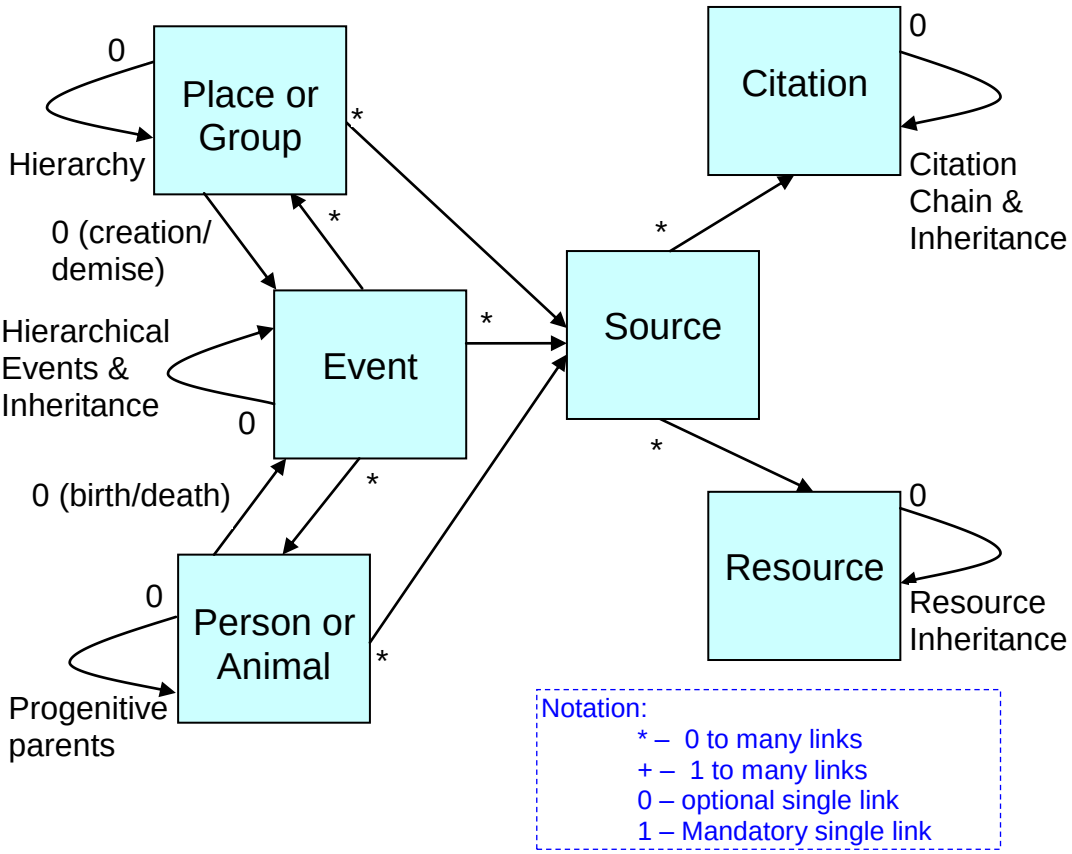
<GroupLnk [Key='key']>
[GROUP_PROPERTY] ...

```
        [ TEXT_SEG ] ...
</GroupLnk>

EVENT_LNK=

<EventLnk Key='key'>
        [ TEXT_SEG ]
</EventLnk>

RESOURCE_LNK=

<ResourceLnk Key='key'>
        [ PARAM_VALUE  ] ...
        [ TEXT_SEG ] ...
</ResourceLnk>

CITATION_LNK=

<CitationLnk Key='key'>
        [ PARAM_VALUE ] ...
        [ PARENT_CITATION_LNK ]
        [ TEXT_SEG ] ...
</CitationLnk>

SOURCE_LNK=

<SourceLnk Key='key'>
        [ PLACE_PROPERTY ... | ANIMAL_PROPERTY ...
        | PLACE_PROPERTY ... | GROUP_PROPERTY ...  ]
        [ TEXT_SEG ] ...
</SourceLnk>
```

For SOURCE_LNK, the type of Property value allowed depends on what entity type the element is embedded within. See also EV_SOURCE_LNK.

Sub-models

STEMMA has two notional sub-models: *conclusional* — representing conclusions for presentation and sharing — and *informational* — representing the assimilation and analysis of information used to form the conclusions. The basic interrelation between these top-level entities in STEMMA's conclusional sub-model is as follows:



See Source for a mention of the complementary informational sub-model.

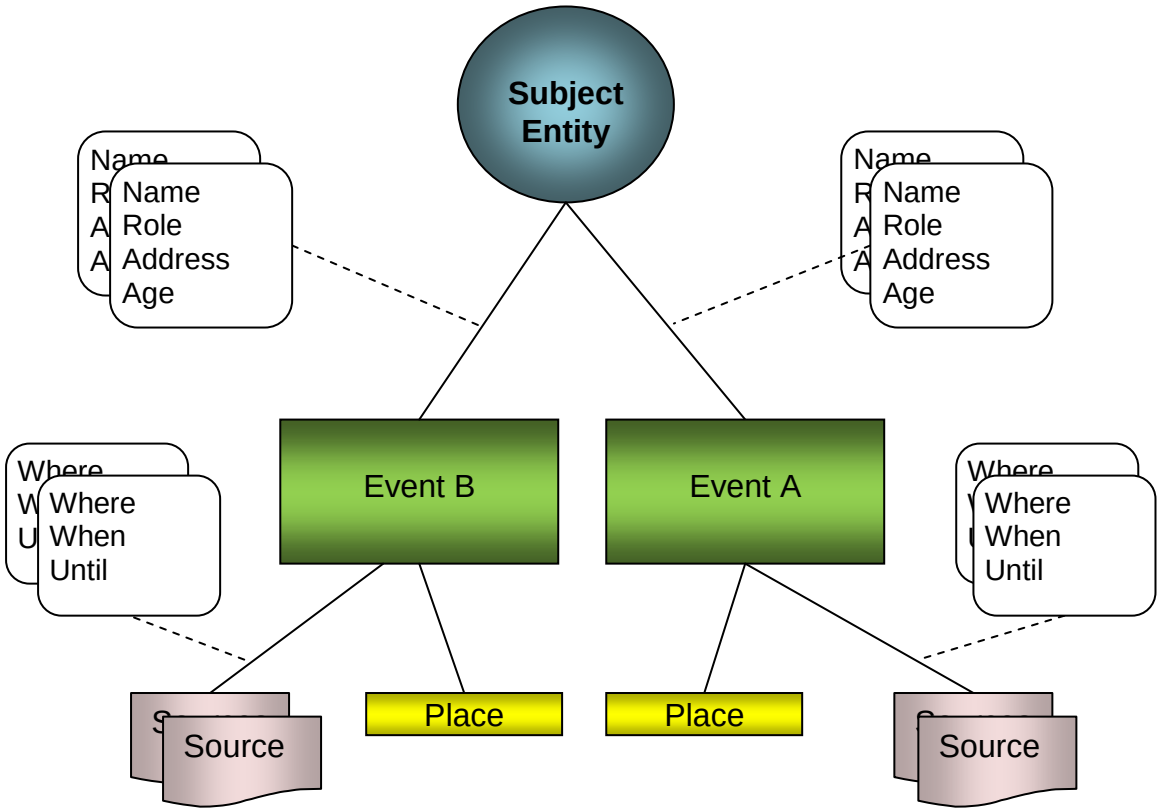
An Event is basically a representation of a date (or range of dates) for which source information exists, and provides a contextual container identifying the *where, when, and who* of that source information. The subject references within the sources are represented by the Event entity being connected to multiple subject entities, such as Persons. Source entities may reference supporting Citations, and Resources such as images, photographs, documents, etc.

Each element can contain narrative sub-elements, as defined below. The narrative text can freely embed references to the other top-level entities and this facility for ad hoc linkage allows arbitrary historical connections to be recorded.

Several entities may link to others of the same type in order to define a hierarchical structure: Place/Group Hierarchies, Person/Animal lineage, Citation chains, and hierarchical Events. An additional form of same-type linkage is used to define an inheritance mechanism for Events, Citations, and Resources.

A second diagram is necessary in order to explain how STEMMA associates Properties with its subject entities. These constitute extracted and summarised information from a supporting source, and are usually associated

with an Event. This because most such items are time-dependent (see Time-dependent Attributes) and an Event represents something that happened in a given place at a given time. It is possible to declare static properties, within the respective subject entity, but these are rarer. In both cases, they are assembled in a corresponding <SourceLnk> element that identifies the supporting source. Properties for the Event itself, such as *where* and *when*, are provided separately in the <SourceLnk> element.



The entities Person, Animal, Place, Group, and Event can also be associated with entities in one-or-more external systems using instances of the following element:

EXTERNAL_ID=

<ExtID> prefix:id </ExtID>

The value prefix identifies the external system. It is a prefix associated with a namespace URI, as described in Extended Vocabularies. For instance, **<ExtID>fs:KWVC-NG4<ExtID>** linking to a entry for a *John Williams* on FamilySearch.

An alternative use of this feature is to export an associated identifier for use in some external system. For instance, if exporting STEMMA data for use in a

Web-based system then ExtID could be used to define corresponding identifiers — different to STEMMA's Key values — for use on that Web site.

6.2 Symbolic Names

This section defines the format of the symbolic names used within STEMMA. This includes the Key attribute associated with various entities (both top-level and enclosed ones), the Dataset Name attribute, and the names of any Source/Citation/Resource Parameters.

They are composed of ASCII alphanumeric characters (i.e. A-Z, a-z, 0-9), underscore, and hyphen, but must start with a letter or underscore. Symbolic names are treated in a case-sensitive way.

Q: What length limits should apply? Any limits should only be for the purpose of a sanity constraint. Q: Should any other characters be allowed?

Entity Key names are local to the respective Dataset. If references need to be analysed across multiple Datasets (e.g. when comparing them) then their Key names should be decorated with the parent Dataset name in order to make them unique, e.g. TonysTree:Tony123. If multiple STEMMA Documents are loaded concurrently then a similar decoration must be applied using the external Document names, e.g. "Example":TonysTree:Tony123. This effectively defines a hierarchical namespace and the syntax of the decorations is designed to follow the precedent set by the URN standard.

All Key names are considered to be the same scope level within their Dataset. That is, even though some entities may have named sub-elements, their names are not considered to be subordinate to that of their parent entity. Although this was considered, it would introduce an unjustifiable complexity since some cases involve three levels and their access is not constrained by those same parents. A reference to Key.Key2 might otherwise require decomposition using separate attributes to facilitate XML query languages such as XPath, XSL Pattern, etc.

Parameter names are local to their respective Source/Citation/Resource entity. When they are substituted into the allowed text values, they employ the ${name} style of substitution syntax.

There are a number of vocabularies implemented in the attribute values and element data used by STEMMA, and these may be controlled (predefined) or partially controlled (extensible). These include type names, role names, mode names, and Property names. As section Extended Vocabularies explains, these values constitute Fragment identifiers and may be qualified with their corresponding namespace URI. They are therefore case-sensitive and limited to the 'unreserved character' set defined in RFC3986 (URI: Generic Syntax). This is similar to the aforementioned character set but additionally including period and tilde.

In the STEMMA examples provided, a single leading character is used in Key names to reflect their declared type. This is purely a convention and not prescribed by the specification. The following characters are currently used:

a	Animal entity.
c	Citation entity.
d	Detail link, for drill-down/drill-up (see below).
e	Event entity.
g	Group entity.
l	SourceLet
m	Matrix entity.
n	Narrative element.
p	Person entity (and sometimes a Contact entity).
r	Resource entity.
s	Source entity.
t	Text element.
w	Place entity (for “where”).

For detail links (explained under Source), a second character usually indicates the nature of the target as follows:

a	ProtoAnimal.
c	Commentary.
d	ProtoDate.
e	ProtoEvent.
g	ProtoGroup.
p	ProtoPerson.
s	Source fragment.
w	ProtoPlace.

Various free-form identifying tags are used on the <Counter> element (Document Structure); the <ts>, <ms>, and <voice> ‘id’ and ‘scheme’ attributes (Descriptive Mark-up); and various elements in <ContactDetails> (Contacts). There are currently no restrictions on their content other than the characters being printable, including space, and that any leading “prefix:” should be associated with a corresponding namespace. They therefore follow the same rules as external identifiers in the <ExtID> element (see EXTERNAL_ID).

6.3 Narrative Structure

There are many uses for narrative text but the two most important categories are for transcriptions and for generating new narrative work (e.g. essays, reports, inference, etc.). These have markedly different characteristics that STEMMA tries to streamline:

- Transcription (including transcribed extracts) — requires support for textual anomalies (uncertain characters, marginalia, footnotes, interlinear/intralinear notes), audio anomalies (noises, gestures, pauses), indications of alternative spellings/pronunciation/meanings, indications of different contributors, different styles or emphasis, and

semantic mark-up for references to persons, places, groups, animals, events, and dates. The latter semantic mark-up also needs to clearly distinguish objective information (e.g. that a reference is to a person) from subjective information (e.g. a conclusion as to whom that person is).

- Narrative work — requires support for layout and presentation. Descriptive mark-up captures the content and structure in a way that provides visualisation software with the ultimate control over its rendering It needs to be able to generate references to known persons, places, and dates that result in a similar mark-up to that for transcriptions. The difference here is that a textual reference is being generated from the ID of a Person entity, say, as opposed to marking an existing textual reference and possibly linking it to a Person with a given ID. Also needs to be capable of generating reference-note citations and general discursive notes.

The basic structure of a narrative block is as follows:

```
NARRATIVE_TEXT=

<Narrative Key='key'>
        [ <Title> narrative-title </Title> ]
        TEXT_SEG ...
</Narrative>

TEXT_SEG=

<Text [Key='key'] [TEXT_TYPE] ...  [DATA_ATTRIBUTE] ... >
        [ <Title> text-title </Title> ]
        narrative-text
</Text>
|
<FromText Key='key'/>
```

A single <Narrative> entity is divided into separate rich-text <Text> segments, each of which has independent properties controlled by the attributes listed below. When <Text> elements are nested, their stacked properties are merged, and then un-stacked when the nesting finishes. Elements with separate opening and closing tags must be closed before the corresponding </Text>.

The Text segments can reference the Keys of arbitrary main entities using the semantic mark-up defined below.

The <Text> element has an optional Key attribute that allows it to be referenced or utilised from elsewhere. The <FromText> element also allows the content of a named <Text> element to be re-used as though it were physically present in the current narrative container.

```
<Text>
```

```
<Text Key='tDemiseJessamine-en' Language='eng'>
    <Title> Demise of Jessamine Cottages </Title>
    <PlaceRef Key='wJessamine' Mode='Hierarchy'/>, were
    demolished in <DateRef Value='1956'/>.
</Text>
<Text Key='tDemiseJessamine-fr' Language='fra'>
    <Title> Disparition de Cottages Jessamine</Title>
    <PlaceRef Key='wJessamine' Mode='Hierarchy'/>, ont été
    démolies en <DateRef Value='1956'/>.
</Text>
</Text>
```

This text could be referenced from another Text section using the appropriate Key name in a <NoteRef> mark-up element. It might generate the following English text on the screen when loaded by an appropriate viewer:

> Jessamine Cottages, Nottingham, were demolished in 1956

The nature of the text in a Text segment may be characterised using one-or-more of the following attributes:

TEXT_TYPE=

Language='code' | Locale='code'

Sets Language or Locale value for the text segment. These are discussed in the section on the overall Document format.

Class=Header | H1 | H2 | H3 | Caption | Legend | Endnote | Footnote | Tablenote

Characterises the content of the text segment; software can apply appropriate styles based on this class. *Header* marks the section as containing header information for the main body. Although this may be used for transcriptions (e.g. of a letterhead), it is primarily designed for the header containing authorship, title, etc., in a narrative work (including essays and reports). *H1–H3* are header levels for section headings. *Caption* may be used to set image (see ResourceRef) and table captions. *Footnote*, *Endnote*, and *Tablenote* are for text to be displayed at the bottom of a page, document, or table. See http://stemma.parallaxview.co/text-class.

Translation='code'

Flags a <Text> section as a translation of a cited source from its original language, as specified by the given ISO 639-2 three-letter language code, to the current language of the enclosing <Text> element.

DATA_ATTRIBUTE=

Sensitivity='level'

The values for ‘level’ are: Public (default), Family, Private, and Sensitive. Access is granted based on which 'family' the requestor belongs to. Hence, *public* is OK to everyone, *family* is OK to an appropriate family member, *private* OK to ad hoc people selected by the owner, and *sensitive* to no one but the owner. These values are part of the controlled vocabulary associated with the http://stemma.parallaxview.co/sensitivity namespace.

Surety=’value%’
Some percentage of how certain the data is. Default=100%. The surety of a datum or inference is more specific than the confidence in a data source or the information derived from it. See Probabilities.

Inference=’boolean’
Indicates the text or datum is inferred from other data. Inference, as used here, implies the *reasoning* (or *proof argument*) as well as any *conclusion*. The default is ‘0’ (i.e. False) implying *information* rather than *reasoning* or *conclusion*. In conjunction with Surety, a value of ‘1’ (i.e. true) can therefore represent conjecture, an educated guess, or even pure speculation.

DetLnk=’key’
Part of a details-link that connects conclusions to their explanation, evidence, and original information. See Source for more information.

For instance:
<Text Sensitivity=’Private’ Surety=’20%’>…some sensitive comment that I'm not sure about…</Text>

6.3.1 Recording Evidence

A number of features are required to correctly record source information in a transcription. This section illustrates how STEMMA deals with them.

- Positional anomalies, meaning text added outside the main body or flow. This may have been added by the author or by-hand after it was printed or published. The following types are supported by the <Anom> semantic mark-up element:
  - Footnotes and Endnotes. Text added at the end of a page or document.
  - Tablenote. Text added at end of table.
  - Maginalia. Text added in a margin.
  - Interlinear notes. Text added between lines.
  - Intralinear notes. Text inserted within a line, usually marked with a caret. Identification of sublinear and supralinear variants.
- Audio anomalies, including non-verbal gestures or movements, noises, and significant pauses.
- Marking sections from different contributors, or suspected different contributors. This includes different (written) hands and different voices. See ‘id’ attribute of <ms>, <ts>, and <voice> elements.

- Uncertain characters. Sequences of characters may be unreadable or uncertain (i.e. there are several distinct possibilities). Recording this correctly is essential for accurate searching. See <Ucf> element below.
- Struck-out characters. Characters crossed out in the original. See the <s> element in Descriptive Mark-up.
- Uncertain interpretation. Adding a suggested meaning or spelling correction to a word or phrase that is readable but is unusual or not recognised. Similarly with unusual pronunciation in audio transcription. Supported via the <Alt> mark-up.
- Specific emphasis, such as bold, italic, or underline. See the respective elements in Descriptive Mark-up.
- Stylistic variations from a given contributor, including different colours, different fonts, different intonation. See 'scheme' attribute of <ms>, <ts>, and <voice> elements.
- Numbering of pages, columns, paragraphs, and lines. See <page>, <col>, <p>, and <line> mark-up.
- Linking textual transcription to locations in an image (see 'x' and 'y' attributes on various elements), and audio transcription to locations in a recording (see <time> element).

Some of these terms and concepts may be found in Editorial Methods for Journals, volume 1, and The Conventions of Textual Treatment, chapter five. For other attempts at audio transcription, see http://clu.uni.no/icame/manuals/WSC/MARKCONV.HTM and https://www.univie.ac.at/voice/documents/VOICE_mark-up_conventions_v2-1.pdf.

Traditional editorial notations for uncertain characters are not well-suited to digital text as they do not facilitate efficient and accurate searching within the limits of what is known. TEI has elements such as <choose> and <unclear>, and a comprehensive formalised notation may be found at: http://igenie.org under Transcriptions. Although less comprehensive, perhaps the most compact is the UCF (Uncertain Character Format) devised by FreeUKGEN. This is based on the regex pattern-matching language although it must be remembered that this exists within target strings rather than search strings. Regex, in turn, is an extension of tradition wildcard characters[2].This UCF is the basis of the notation used within STEMMA and the following table is from the FreeBMD pages:

_ (Underscore)	A single uncertain character. It could be anything but is definitely one character. It can be repeated for each uncertain character.

[2] Wildcard characters represent variable sequences. There are several schemes but most allocate a single character to represent 0-or-more unknown characters (e.g. '*') and another to represent exactly one unknown character (e.g. '?'). These may be combined so that, for instance, '?*' represents 1-or-more unknown characters. Note that since '*?' ≡ '?*' and '**' ≡ '*' then any contiguous sequence of '*' and '?' can be simplified to just [?...][*], i.e. 0-or-more '?' followed by an optional '*'.

* (Asterisk)	Several adjacent uncertain characters. A single * is used when there are 1 or more adjacent uncertain characters. It is not used immediately before or after a _ or another *. Note: If it is clear there is a space, then * * is used to represent 2 words, neither of which can be read.
[*abc*]	A single character that could be any one of the contained characters and only those characters. There must be at least two characters between the brackets. For example, [79] would mean either a 7 or a 9, whereas [C_] would mean a C or possibly some other character.
{*min*,*max*}	Repeat count - the preceding character occurs somewhere between *min* and *max* times. *max* may be omitted, meaning there is no upper limit. So _{1,} would be equivalent to *, and _{0,1} means that it is unclear if there is any character.

UCF also defines a '?' character that is used to represent the situation where all of the characters have been read but you remain uncertain of the word, e.g. "RACHARD?" This is not used within STEMMA because it is ambiguous with '?' representing an absent value, and the equivalent feature is supported by <Alt> mark-up.

Some examples:

[lt]	Can't tell if it's an l or a t.
___	Three unreadable characters.
[x_]	I think the character is an 'x'
_{2,3}	Two or three unreadable characters.
*	Unknown number of unreadable characters.
_{0,1}	Not sure if there's a letter or an ink blob.

Early STEMMA designs considered using an ANSI escape sequence to bracket a set of UCF characters. For instance, **<APC>_12[68]<ST>** where APC=0x9F and ST=0x9C. This was partly to avoid unconditionally reserving a whole set of characters but also to allow them in attribute values as well as element data. The current version accommodates them in a <Ucf> element:

<Ucf> ucf-sequence </Ucf>

6.3.2 Semantic Mark-up

The following elements contribute to support for semantic mark-up in narrative text. A number of these are used for marking-up transcribed information (e.g. a reference to a person), and these are mirrored in the support for Properties and the Source entity. A number also support the generation of marked-up references in new works created by the author, including narrative essays and narrative reports. The difference being that they generate a marked-up reference as opposed to marking-up a prior reference.

PERSON_REF=

<PersonRef [Key='key' | DetKey='key'] [Mode='mode']>
name-orig-text
[TEXT_SEG] …
</PersonRef>

If the name text is provided then this marks-up that text as a person reference. This may be a personal name but not always (e.g. "My grandfather"). If the Key attribute is also specified then it links that person reference to an actual conclusion Person (or Contact) entity. If a DetKey is specified them it allows the reference to be analysed in a Source entity. The Mode attribute is ignored in these instances.

If the name text is not provided then this element generates a reference to the Person entity identified by the mandatory Key attribute. The Mode attribute controls the generated reference using the following controlled vocabulary from the http://stemma.parallaxview.co/person-name-mode and http://stemma.parallaxview.co/name-mode namespaces:

Mode	Operation
Title (default)	The Title of the Person, if any. If the Person has no title defined, or the referenced entity is a Contact, then Mode defaults to SemiFormal.
Informal, SemiFormal, Formal, Listing	The first instance of a canonical name with a corresponding name-mode.

Fall-back sequence: Listing, {Formal, SemiFormal, Informal}, Title, Key — curly brackets indicating a circular choice.

ANIMAL_REF=

<AnimalRef [Key='key' | DetKey='key'] [Mode='mode']>
name-orig-text
[TEXT_SEG] …
</AnimalRef>

If the name text is provided then this marks-up that text as an animal reference. This may be a name but not always (e.g. "My dog"). If the Key attribute is also specified then it links that animal reference to an actual conclusion Animal entity. If a DetKey is specified them it allows the reference to be analysed in a Source entity. The Mode attribute is ignored in these instances.

If the name text is not provided then this element generates a reference to the Animal entity identified by the mandatory Key attribute. The Mode attribute controls the generated reference using the following controlled vocabulary from the http://stemma.parallaxview.co/animal-name-mode and http://stemma.parallaxview.co/name-mode namespaces:

Mode	Operation
Title (default)	The Title of the Animal, if any. If the Animal has no title defined then Mode defaults to SemiFormal.
Informal, SemiFormal, Formal, Listing	The first instance of a canonical name with a corresponding name-mode.

Fall-back sequence: Listing, {Formal, SemiFormal, Informal}, Title, Key.

PLACE_REF=

<PlaceRef [Key='key' | DetKey='key'] [Mode='mode']>
place-orig-text
[TEXT_SEG] ...
</PlaceRef>

If the place text is provided then this marks-up that text as a place reference. If the Key attribute is also specified then it links that place reference to an actual conclusion Place entity. If a DetKey is specified them it allows the reference to be analysed in a Source entity. The Mode attribute is ignored in these instances.

If the place text is not provided then this element generates a reference to the Place entity identified by the mandatory Key attribute. The Mode attribute controls the generated reference using the following controlled vocabulary from the http://stemma.parallaxview.co/place-name-mode and http://stemma.parallaxview.co/name-mode namespaces:

Mode	Operation
Title (default)	The Title of the Place, if any. If the Place has no title defined then Mode defaults to SemiFormal.
Informal, SemiFormal, Formal, Listing	The first instance of a canonical name of a corresponding name-mode.
Hierarchical	Generates a Place-hierarchy-path. The ordering (coarse-to-fine or vice versa) and separating characters must be configurable in the software generating the output.

Fall-back sequence: Hierarchical, Listing, {Formal, SemiFormal, Informal}, Title, Key.

GROUP_REF=

<GroupRef [Key='key' | DetKey='key'] [Mode='mode']>
group-orig-text
[TEXT_SEG] ...

</GroupRef>

If the group text is provided then this marks-up that text as a group reference. If the Key attribute is also specified then it links that group reference to an actual conclusion Group entity. If a DetKey is specified them it allows the reference to be analysed in a Source entity. The Mode attribute is ignored in these instances.

If the group text is not provided then this element generates a reference to the Group entity identified by the mandatory Key attribute. The Mode attribute controls the generated reference using the following controlled vocabulary from the http://stemma.parallaxview.co/group-name-mode and http://stemma.parallaxview.co/name-mode namespaces:

Mode	Operation
Title (default)	The Title of the Group, if any. If the Group has no title defined then Mode defaults to SemiFormal.
Informal, SemiFormal, Formal, Listing	The first instance of a canonical name of a corresponding name-mode.

Fall-back sequence: Listing, {Formal, SemiFormal, Informal}, Title, Key.

EVENT_REF=

<EventRef [Key='key' | DetKey='key'] [Mode='mode']>
event-orig-text
[TEXT_SEG] …
</EventRef>

If the event text is provided then this marks-up that text as an event reference. If the Key attribute is also specified then it links that event reference to an actual conclusion Event entity. If a DetKey is specified them it allows the reference to be analysed in a Source entity. The Mode attribute is ignored in these instances.

If the event text is not provided then this element generates a reference to the Event entity identified by the mandatory Key attribute. The Mode attribute controls the generated reference using the following controlled vocabulary from the http://stemma.parallaxview.co/event-mode namespace:

Mode	Operation
Title (default)	The Title of the Event, if any

Fall-back sequence: Title, Key.

RESOURCE_REF=

<ResourceRef Key='key' [Mode='mode'] [Align='L|R|C']>
[PARAM_VALUE] …
[TEXT_SEG] …
</ResourceRef>

Generates a reference to the Resource identified by the specified Key and Parameters. Default alignment is L (left). Any enclosed Class='Caption' text segments are used as an image caption, or if none then any similar segments in the Resource entity itself. The Mode attribute controls the generated reference using the following controlled vocabulary from the http://stemma.parallaxview.co/resource-mode namespace:

Mode	Operation
Title (default)	The Title of the Resource, if any.
Small, Medium, Large	Generates an icon or image of the corresponding logical size, if the output media is capable. The physical size for each is controlled by the software generating the output.
SynchImage	Identifies an image (not displayed) to which transcription entities can be positionally related. See x/y attributes in Descriptive Mark-up.
SynchAudio	Identifies a recording to which transcription entities can be synchronised. See <time> in Descriptive Mark-up.

Fall-back sequence: {Large, Medium, Small}, Title, Key.

CITATION_REF=

<CitationRef Key='key' [Mode='mode']>
[PARAM_VALUE] …
[PARENT_CITATION_LNK]
[TEXT_SEG] …
</CitationRef>

Generates a reference to the Citation identified by the specified Key and Parameters. The Mode attribute controls the generated reference using the following controlled vocabulary from the http://stemma.parallaxview.co/citation-mode namespace.

Pre-formed (preferred) citation strings may be defined by enclosed Class='Footnote | Endnote | Tablenote' text segments, as appropriate. When a citation string is generated from the accumulated parameters, a shortened form may be used if this is a subsequent reference (in the same Narrative element) to the same source, and with the same values for Parameters having WhereIn='0'.

If no hand-crafted form is provided then the corresponding <DisplayFormat> for the Citation entity (and any chain parents) is used. Otherwise, an external citation-template system is assumed to be available.

The actual style of the reference (e.g. EE, or CMOS) should be a configurable option of the viewing software.

Mode	Operation
Footnote (Default) Endnote Tablenote	Inserts a reference to a corresponding footnote, endnote, or tablenote using a superscript or bracketed character. On an interactive device, the character may be a live link.
Implied	Associates a Citation entity with the current sentence, but without it contributing to its text. This is useful where a NoteRef may be conflating two source references into a single sentence.
Inline	Inline reference, with no trailing punctuation. This mode is particular useful for creating complex citations, or adding a source reference to discursive notes.
Parenthetical	Parenthesised in-text reference, as supported by CMOS.
Title	Generates a reference to the expanded citation-title (if any) or the associated key name.

This selection of citation modes is designed to provide features from which various citation forms can be constructed; both traditional ones and ones more applicable to interactive displays. They are not presented as a stylistic gallery.

NOTE_REF=

<NoteRef [Mode='mode']>
[orig-text]
[TEXT_SEG] ...
</NoteRef>

Creates a general note or annotation for a specific piece of text. The note text is formed from the collected <Text> elements.

If no original text is embraced then a live link cannot be supported and that mode is converted to a footnote reference instead.

Again, the output is controlled by the Mode parameter. The following values are part of the controlled vocabulary associated with the http://stemma.parallaxview.co/note-mode namespace.

Mode	Operation
Link (Default)	On an interactive device, this inserts a live link to the respective text. On a non-interactive device, such as printer output, then it generates a normal footnote reference as described below.

Inline	The note text is inserted inline and distinguished from the surrounding text. This usually means employing editorial brackets, [...]
Footnote Endnote Tablenote	Inserts a reference to a corresponding footnote, endnote, or tablenote using a superscript character or symbol. On an interactive device, the superscript may be a live link.

ANOM_REF=

<Anom [Mode='mode'] [Posn='posn'] [Ref='ref-mode'] [Dur='[[hh:]mm:]ss'] [Descr='description']>
narrative-text
[TEXT_SEG] ...
</Anom>

Represents a transcription anomaly at the current position. For Marginalia, Interlinear, and Intralinear, the narrative-text represents the equivalent <Text> value to place outside the main body of text. For Noise and Gesture, narrative-text is the spanned text, and 'Descr' must describe the associated anomaly. No prescribed taxonomy is provided since the identification, interpretation, and description may be subjective. Any other collected <Text> contributions are deemed to be explanatory text.

Again, the output is controlled by the Mode parameter. The following values are part of the controlled vocabulary associated with the http://stemma.parallaxview.co/anomaly-mode namespace.

Mode	Operation
Footnote Endnote Tablenote	A reference to a corresponding footnote, endnote, or tablenote. Ref must identify marker character.
Marginalia	A reference to text added in a margin. Posn='L|R|T|B', Ref='--' (line) or empty. Defaults: Post='R', Ref=''.
Interlinear	Text added between lines.
Intralinear (Textual default)	Text added within a line. Posn='T|B' (above/below), Ref='^|--' (caret/line). Defaults: Posn='T', Ref='^'.
Noise (Audio default)	Untranscribable noise uttered by individual, e.g. sneeze, cough, sniff, yawn, whistle, laugh, swallow. Described via Descr.
Pause	Significant pause during recording of individual. Dur attribute indicates the number of hours, minutes, and seconds.
Gesture	Non-verbal action by individual, e.g. nod, applause, smile, head-shake, squint, frown. Described via Descr.

ALT_REF=

<Alt [Value='alt-value'] [Mode='note-mode'] [Sic='read-as']>
orig-text
[TEXT_SEG] ...
</Alt>

Marks a reference with an alternative spelling, wording, or meaning. If the Value attribute is specified then it provides an alternative piece of text. At least one of the Value attribute, explanatory text, or the Sic attribute must be specified. The note-mode is identical to that documented for the NoteRef mark-up.

When an explanation is 'Inline' then it may explicitly include the Latin word *sic* ("thus"). However, in other cases, it may still be preferred to have *sic* follow inline, possibly with the alternative. The attribute Sic='' will generate the suffix "[*sic*]", while Sic=', read: ' (or other separating text) would generate "[*sic*, read: alternative]". For instance:

<Alt Sic=''>jurney</Alt>
generates "jurney [*sic*]"
<Alt Value='journey' Mode='Inline'>jurney</Alt>
generates "jurney [journey]"
<Alt Value='journey' Sic=', should be: ' Mode='Inline'>jurney</Alt>
generates "jurney [*sic*, should be: journey]"

LINK_REF=

<Link URL='url'>
hyperlink-narrative-text
[TEXT_SEG] ...
</Link>

Inserts a reference to an external location via a hyperlink. Embraced text is highlighted in the configured way which will usually be underlined and coloured blue. If the element embraces a ResourceRef then it allows an image to be hyperlinked.

DATE_REF=

<DateRef [DetKey='key'] [Value='std-date'] [Mode='mode']>
date-orig-text
[DATE_ENTITY]
[TEXT_SEG] ...
</DateRef>

If the original date text is provided then this marks-up that text as a date reference. If the DATE_ENTITY sub-structure or the Value attribute is also

specified then it links that date reference to an actual conclusion date. The Mode attribute is ignored in this instance. If a DetKey is specified then it allows the reference to be analysed in a Source entity.

If the date text is not provided then this element generates a formatted reference to the date defined by the DATE_ENTITY sub-structure or the Value attribute. The Mode attribute controls the formatting and must take one of the values: Short, Medium (default), Long, or Full. These values are part of the controlled vocabulary associated with the http://stemma.parallaxview.co/date-mode namespace.

The DATE_ENTITY and the Value attribute are mutually exclusive. The formatting associated with these Mode values should be a configurable property of the software manipulating the data, and should honour the regional settings of the end-user. The interpretation of these values is well-established for the Gregorian calendar. For other calendars, the support of a Date Authority may be required.

<Subs> narrative-text </Subs>

Allows the substitution of parameter values using their normal ${name} syntax. By default, this substitution syntax is not recognised in <Text> elements.

<Mark DetKey='key'> narrative-text </Mark>

Marks the embraced text as accessible via the given detail-key within a Source entity. This allows it to be discussed, analysed, and built into a stepwise proof argument.

<MarkRef DetLnk='key'/>

Reproduces a copy of the identified text that was previously marked with a <Mark> element. The choice of quotation style (inline or separate paragraph) is controlled outside of the element. For instance:

```
<p>The report stated that "<MarkRef DetLnk='dsKey'/>".</p>
```

or

```
<p>The report stated that:</p>
<p indent='1'><MarkRef DetLnk='dsKey'/></p>
```

Intrinsic Methods

A number of intrinsic methods may be used in the Value and Key attributes of the semantic mark-up elements, depending on the contextual validity. These will eventually be incorporated in the specification for a run-time object model that can interrogate and manipulate STEMMA data programmatically.

Each entity has a static constructor that takes the associated key as an argument. All entity objects have a default implicit Key() method that returns the entity key.

The following Event methods are defined:

```
std-date = Event (event-key).Start()
std-date = Event (event-key).End()
place-object = Event (event-key).Place()
```

Hence, Event(key).Place() can be used where the Key value of a Place entity is required.

Methods of the following general form are available for all subject-entities (for Person, Animal, Place, and Group), and Contact, are defined:

```
person-name = Person(person-key).Name(name-mode)
```

The name-mode term must be from the http://stemma.parallaxview.co/name-mode namespace.

Use of these methods retains the context of their arguments in order to support drill-down operations. For instance,

<PersonRef Key='pJohnSmith'/> was born at <PlaceRef Key='w12BackRoad'/> on <DateRef Mode='Long' Value='Event(eBirthJohnSmith).Start()'/>.

This might generate the following output:

John B. Smith was born at 12 Back Road, Smartville on 21 February 1920.

When clicking on such a derived date or place then the software should remember which Event they were derived from.

6.3.3 Descriptive Mark-up

This section includes those mark-up elements related to structure and content rather than to semantics. While the structure obviously impacts presentation, STEMMA is not a presentation format and so many aspects are left to the control of such formats (e.g. HTML+CSS) when data has been transformed for presentation.

HTML was originally created with a rather relaxed syntax describing presentational mark-up, and was designed for rendering content from Web pages, etc. Through its various versions, there has been an attempt to separate mark-up related to structure and content from that related to its presentation. For instance, the <del> tag for representing deleted text is now recommended over the older <s> tag for representing strikethrough text (the even-earlier <strike> tag was discontinued after HTML V4). Similarly, the and tags for representing emphasis and strong text are now

recommended over explicit italic and bold control using the and tags. After HTML V4, and in parallel with HTML V5, an XML variant called XHTML was developed, and its XML foundation meant that its syntax was stricter and its meaning was extensible through namespaces; it also dropped many of the presentational tags and a reliance on Cascading Style Sheets (CSS) was presumed as a way of achieving a more flexible rendering.

This is less than adequate for STEMMA: although its earlier versions intended to use the XHTML set of tags, it quickly became clear that there are instances where rendering must be explicit, such as for pre-formatted citations, and where prior formatting or emphasis needs to be described accurately in transcriptions. The latter is not simply a presentational matter as the mark-up needs to describe the text as-it-was, which in turn impacts on the semantics and significance of that text. In this respect, STEMMA is trying to describe a document in the same way that TEI (http://www.tei-c.org/) would, except that TEI is too comprehensive and the resulting overlap with STEMMA would cause a clash of approach that does not align with its micro-history goal. For instance, the cite, personography, and placeography tags are inappropriate in STEMMA. A gentle introduction to the way TEI handles transcription may be found at: Transcription Guidelines.

The current version of STEMMA therefore uses HTML-like tag names in the context of its XML representation (note the use of all-lowercase tag names here), albeit with some extra attributes, and avoids the direct incorporation of full dialects or namespaces associated with the above. The <Text> element will accept the following HTML-like mark-up in order to support structure and content type, including a primitive level of formatting and emphasis.

Element, and attributes	Description
** narrative-text **	Bold text
[sic='boolean']	As-it-was (see below)
** **	Line-break
<col/>	Start of next, or specific, column.
[align='L|R|C']	Text alignment. Default=L
[num='n']	Column index (1:n), must be in range of <page> setting.
[x='x%', y='y%']	Position of item relative to top-left of image
<colgroup> table-columns </colgroup>	Defines the columns of a table using <tcol>
[width='w%']	Table width as percentage of available width. Default 100%
** narrative-text **	Emphasis (usually italic in text)
** narrative-text **	Italic text
[sic='boolean']	As-it-was (see below)
<indent/>	Indent start of current line
** narrative-text **	List item. Ignored outside of or elements.
<line/>	Start of a line in a transcription

[num='n']	Line number
[posn='L|R']	Position of displayed line numbers
[x='x%', y='y%']	Position of item relative to top-left of image
<ms> narrative-text </ms>	Transcription or extract from manuscript text
[id='id']	Distinguishes different contributors
[scheme='tag']	Distinguishes different styles
list-items	Ordered (numbered) list
[start='n']	Starting number. Default=1
[type='chr']	Numbering type (1, A, a, I, i)
narrative-text	Paragraph
[align='L|R|C']	Text alignment. Default=L
[indent='n']	Indent left margin
[num='n']	Paragraph count (1:n) in column
[x='x%', y='y%']	Position of item relative to top-left of image
<page/>	Start of a page in a transcription
[cols='n']	Number of columns. Default=1
[id='id']	Page identification. May be non-numeric
[posn='T|B']	Position of any displayed page identification
<posn/>	Generic synchronisation point within image, rather than for a specific element
x='x%',y='y%'	Position of location relative to top-left of image
<s> narrative-text </s>	Strikethrough (deleted) text
[sic='boolean']	As-it-was (see below)
[style='value']	Number of strikes (1, 2, *=heavy)
narrative-text	Strong (usually bold in text)
_{narrative-text}	Subscript text
^{narrative-text}	Superscript text
<table> body </table>	Defines a table
<tcol/>	Defines a table column. Ignored outside of <colgroup>.
[align='L|R|C']	Text alignment. Default=L
[width='w%']	Column width as percentage of table width.
<td> data-cell </td>	Table data cell. Ignored outside of <table> element.
[colspan='n']	No. of columns for data cell
[rowspan='n']	No. of rows for data cell
<th> header-cell </th>	Table header cell. Ignored outside of <table> element.
[colspan='n']	No. of columns for header cell

[rowspan='n']	No. of rows for header cell
[scope='term']	Type of header cell ('row'/'col')
<time/>	Synchronisation point in transcript of a recording
stamp='[[hh:]mm:]ss'	Number of hours, minutes, and seconds since start of recording
<tr> row </tr>	Table row. Ignored outside of <table> element.
<ts> narrative-text </ts>	Transcription or extract from typescript text
[id='id']	Distinguishes different contributors
[scheme='tag']	Distinguishes different styles
<u> narrative-text </u>	Underlined text
[sic='boolean']	As-it-was (see below)
[style='value']	Number of underlines (1, 2, *=heavy)
list-items	Unordered (bullet) list
[type='type']	Marker ('disc', 'circle', 'square')
<voice> narrative-text </voice>	Transcription or extract from voice
[id='id']	Distinguishes different contributors
[scheme='tag']	Distinguishes different tones
[overlap='boolean']	Overlapping nested contributions, each transcribed. 'id' and 'scheme' not applicable with this attribute
[bg='boolean']	Untranscribed contribution (identified by 'id' attribute) is background to nested contributions. Exclusive of 'overlap'.

General Formatting

Explicit mark-up for visual rendering in authored narrative, such as colour and font, is not recommended since STEMMA's descriptive and semantic mark-up allows software products to select them in a consistent way using some sort of style gallery.

The <p> and
 elements behave as per their HTML equivalents in authored work, including the fact that
 is not recommended to simulate paragraphs. The <b>, <i>, <u>, and <s> elements provide some primitive visual attributes similar to older HTML definitions, but they all have more functional uses within transcripts and extracts (i.e. when sic='1'). For instance, <u> and <s> can distinguish single, double, or heavier use of lines; also, <s> indicates deleted text rather than specifically the use of strikethrough. Although these could have been done using the 'scheme' attributes (see below), they were allowed in transcription since their presence for other purposes would undoubtedly mean they would get applied in textual transcription too.

The sic attribute (short for Latin: *sic erat scriptum*, meaning "thus was it written") is used to distinguish between voluntary use of the respective elements (e.g. for citation formatting) and usage describing a transcribed prior

document. The default value is determined by an enclosing <ts> and <ms> (implying sic='1'), or otherwise (sic='0').

Because of their vague interpretation, the and elements are designed for formatting in authored work rather than for representing transcribed material.

Text Structure
For authored work, a narrative article basically uses section headings, and paragraphs within sections. In acknowledgement of HTML, <p> elements nested within the same stream (i.e. excluding out-of-line text, such as notes) are ignored.

For textual transcription (including transcribed extracts), the structure is: pages, columns, paragraphs, and lines. Each of these may be related to specific positions in an image of the original material using SVG-like x and y coordinates. These specify percentage displacements from the top-left corner of the last image identified through a ResourceRef element with Mode='SynchImage'. The fact that <page> also allows this means that a single image may show multiple pages.

Two main approaches to transcription are supported: line-based, which relies on
 and <line> elements (possibly within <p>), and paragraph-based, which relies on flowed text within <p>.

The <page> element marks the head of a new page. The id is the actual page identification, if any, e.g. id='12–2'. It is not incremented automatically as it may be non-numeric. The optional case is for unidentified pages. Line numbers are not reset on a new page.

<col> and <p> numbers run sequentially, starting at 1, unless explicit on these attributes.

The <line> element sets a new line count within a transcription. These usually count non-blank lines from the start of the outermost <Text> element but this can be changed, e.g. to record lines in a specific page, column, or paragraph. Line numbers are automatically incremented but this element may be used periodically to keep the count in step. If the document being transcribed already identifies line numbers, and the mark-up is mirroring them, then the posn attribute identifies whether they were displayed in the left or right margin. This setting carries though until the next posn attribute or the end of the <Text> element; note that alternate pages usually switch from left to right margins in practice.

The units for paragraph indentation (see <p>) and line indentation (see <indent>) are based on some externally-configurable unit, such as the width of four spaces.

Transcription source

The <ts>, <ms>, and <voice> elements distinguish between transcriptions of typescript, manuscript, and audio data. Their ‘id’ attribute distinguishes different contributors, and can therefore mark different (written) hands or different voices. Their ‘scheme’ attribute further distinguishes different styles, and can therefore mark different colours, different fonts, or different tones. The specific nature of these schemes is less important than their identification, and their interpretation must be part of the analytical process. STEMMA does not mandate how these two attributes are used, but it is recommended that they be described in commentary accompanying a transcription.

The use of the ‘id’ and ‘scheme’ attributes effectively separates structure and content from presentational or stylistic issues in a transcription, analogous to the similar goals for formatted text in HTML5.

NB: these three elements are automatically ‘off’ in any new <Text> element; this means that they’re initially ‘off’ within nested <Text> elements, but the previous status is un-stacked at the end of the <Text> element. As with <b>, <i>, etc., the settings for nested cases of each element are merged, and unmerged as each inner element is closed.

Audio Transcription
Support for audio recordings may be divided into the following broad areas:

- Specific audio contributions (such as the voice of an individual) – <voice>
- Anomalous contributions from an individual that cannot be represented as text, including noises, pauses, and gestures – <Anom>
- Alternative word meanings, clarifications, or other notes – <Alt> and <NoteRef>, exactly as with textual transcription
- Time synchronisation – <time>. This is analogous to <posn>, and other x/y coordinates, used in textual transcription.

The ‘scheme’ attribute allows intonation or emotional changes to be marked in the transcript, e.g. fast/slow, loud/soft/whisper, laughing, singing, false accent, imitation.

The ‘overlap’ and ‘bg’ attributes provide two different ways of representing overlapping audio contributions. The ‘overlap’ attribute’ defines a container <voice> element for multiple transcribed contributions. The ‘bg’ attribute identifies an untranscribed background contribution to its nested transcribed contributions.

See the example at Dialogue Transcription for a more detailed illustration.

Tables
The <table>, <colgroup>, <tcol>, <tr>, <th>, and <td> elements are very similar to their HTML equivalents (<tcol> analogous to HTML <col>). They attempt to focus on the structure and content rather than the presentation — something that’s not relevant until the STEMMA representation has been

transformed into a particular visual representation, such as HTML+CSS, Word, or PDF.

Enclosed <Text> segments with Class='Caption' are used as the table caption. Ones with Class='Tablenote' cause tablenotes to be deposited at the foot of the table.

Lists
The <ul>, <ol>, and <li> are very similar to their HTML equivalents.

6.3.4 Probabilities

Accepted genealogy certainly accommodates qualitative assessments such as primary/secondary information, original/derivative source, impartial/subjective viewpoint, etc. STEMMA supports each of these assessments in the Source entity. They can assist when assessing evidence in order to associate a level of confidence with an item of data (e.g. a date or relationship). That level of confidence will obviously affect conclusions and inferences derived from it. STEMMA goes further, though, by allowing qualitative assessments to be expressed quantitatively using probabilities (written as percentages).

This is viewed as a controversial feature by some people since the selection of base probabilities is somewhat subjective, although the mathematics of combining them to investigate scenarios is well-defined. We accept the use of probabilities for gambling, even on subjective issues like the "form of a horse", but we may feel it implies inappropriate accuracy in our genealogical research. Our brains mostly handle probabilities in an analogue fashion, and we have no issue with ordered, non-numeric scales such as "very likely, likely, probably, maybe, unlikely, improbable". The step of associating numeric probabilities is a relatively small one from that perspective.

The subject of "Structured Indications of Uncertainty" is discussed in the context of TEI here: Structured Uncertainty in section 17.1.2. A further discussion directly related to genealogy may be found at: You're Probably Right.

A STEMMA rationale for using percentages in the Surety attribute rather than simple integers was partly so that it allows some basic arithmetic to assess derived data. For instance, if A => B, and B => C, then the absolute surety of C is surety(A) * surety(B).[3] Another potential advantage, though, is that of

[3] The probability of 'A or B' being true is expressed as:
$p(A \cup B) = p(A) + p(B) - p(A \cap B)$
where $p(A \cap B)$ is the probability of 'A and B' being true. If A and B are independent of each other then:
$p(A \cap B) = p(A) * p(B)$
If A and B are mutually exclusive then:
$p(A \cap B) = 0$
and
$p(A \cup B) = p(A) + p(B)$

'collective assessment'. Given three alternatives, X, Y, & Z, simple integers might allow an assessment of X against Y, or X against Z, but not X against all the remaining alternatives (i.e. Y+Z).

6.4 Person

The Person entity represents a specific unique person. Most Properties will be dependent on the date of some activity or registration, and these should be included through a <SourceLnk> element of an Event (or Eventlet) entity. The only direct data are the core ones such as their name, birth-sex, birth, death, and biological parentage. These are primarily conclusions drawn from information in the cited sources. SourceLnk can also be used to associate static Properties directly with the Person, and an example might be their blood group or other congenital detail.

PERSON=

```
<Person Key='key'>
        [ <Title> person-title </Title> ]
        [ <Sex [DATA_ATTRIBUTE] ... > boolean </Sex> ]
        [ NAME_VARIANTS | { <PersonalName [DATA_ATTRIBUTE] ... >
        personal-name  </PersonalName> } ]
        [ <FatherPersonLnk Key='key' [DATA_ATTRIBUTE] ... >
                [ TEXT_SEG ] ...
        </FatherPersonLink> ]
        [ <MotherPersonLnk Key='key' [DATA_ATTRIBUTE] ... >
                [ TEXT_SEG ] ...
        </MotherPersonLink> ]

        [ <Birth>
                EVENT_LNK  |  EVENTLET
                [ TEXT_SEG ] ...
        </Birth> ]
        [ <Death>
                EVENT_LNK  |  EVENTLET
                [ TEXT_SEG ] ...
        </Death> ]
        [ <MemberOf  Key='key' [RANGE_FROM] [RANGE_TO]>
                [ TEXT_SEG ] ...
        </MemberOf> ] ...

        [ EVENTLET ] ...
        [ SOURCE_LNK ] ...
        [ CONTACT_DETAILS ]
        [ EXTERNAL_ID ] ...
        [ TEXT_SEG ] ...
</Person>
```

Bayesian probability takes this to a much deeper level where a simple true/false is inadequate.

It should be noted that STEMMA Persons may be disjoint from a lineage point of view. That means that the Persons in a given Dataset do not necessarily constitute a single tree or connected set. This is designed to accommodate both Persons of uncertain relationship and Persons who are not family members at all but who may still have contributed to their history.

The <Title> element provides a unique descriptive title for the Person. This controls how you want to see the Person identified in any genealogical reports or charts. It is not the same as their full formal name or their name variations — both of which may be date-dependent (see Personal Names). It could default to their full name at birth but this might also be annotated, say, with their date-of-birth if there were multiple people with the same title. For instance: “John Smith (1892)”. It may also be used to distinguish people in different generations, or branches, with the same name, or different siblings with the same name (e.g. when one has died), or unnamed children who died at birth. See also One Name to Rule Them All.

NB: The Person’s name may not be known, and similarly with their sex and biological parentage.

The birth and death Events are specifically identified since there cannot be more than one of each (in contrast to other Event-types) and as an aid to creating traditional family trees and pedigree charts.

The <Sex> element represents the Person’s birth sex. Although Sex is not really a Boolean, STEMMA avoids assigning English Male/Female initials and uses a 1/0 replacement instead. Sex is actually tri-state since there is no default when it is not specified. The assumption made here is that the birth sex — however indeterminate — is a single value. If that person undergoes some later gender reassignment then that would be recorded through ‘Medical’ Events. If the person adopts some lifestyle outside of traditional bi-gender roles (e.g. LGBT) then that can be recorded through biographical narrative in conjunction with relevant Events. See also No Sex Please, We're Genealogists!

The association with one-or-more Groups may be constrained to a range of dates by specifying their entry and exit within <MemberOf> elements. The dates default to their birth and death of the Person if a range has been left open.

6.4.1 Properties

Properties are items of data extracted and summarised from a given data source, such as name, age, or occupation. As extracted forms of information they therefore share the same requirements as marked-up references in terms of supporting transcription issues (e.g. uncertain characters, strikethrough) and separating the interpretation from the original value.

The value of virtually all Properties, whether they’re applicable to a Person, Animal, Place, or Group, will be relevant to a given date. When these subject

entities are associated with an Event then it will provide that date context, in addition to other context such as a Place and supporting sources. See Time-dependent Attributes for further discussion.

Static Property values may be associated directly with a subject entity, but these are rarer. An example for a Person entity might be their blood group.

The full syntax for defining Properties, and for providing values, may be found at: Extended Properties. This scheme is fully open to custom properties. The ones presented below are the predefined ones associated with the http://stemma.parallaxview.co/person-property namespace.

PERSON_PROPERTY=

<PropertyDef Name='Name' Type='PersonRef'/>
The personal name of the person, as recorded in the supporting source. On a birth certificate, this would also be used for the mother's maiden name(s).

<PropertyDef Name='Age' Units='y,m,w,d' Type='Measure'/>
The age of the person, usually at the start of the current event. Default unit is 'y' (years) but also valid are 'm' (months), 'w' (weeks), and 'd' (days). The value may be fractional, e.g. 6.5 years, in which case the decimal point character is strictly prescribed under Locale-independence. Currently, no multi-unit ages are allowed, e.g. 3y 2m.

Note that the recorded age may follow different conventions in different sources and some narrative comment would be well placed. For instance, in the 1841 census of England and Wales, the age was rounded down to the nearest multiple of 5 for over fifteens. In the Canadian census, the requested age was at "next birthday" before 1881 and at "last birthday" from 1881,

<PropertyDef Name='Occupation' Type='Text'/>
The occupation, profession, or rank of the person.

<PropertyDef Name='Employer' Type='Text'/>
The name of the person, organisation, or institution employing the person.

<PropertyDef Name='WorkPlace' Type='PlaceRef'/>
Place of work or employment.

<PropertyDef Name='Role' ItemList='1' Type='EnumList'/>
The role(s) of a person in a multi-subject Event. Role values depend on the context but include Head, Visitor, Inmate, Boarder, Lodger, Servant, Child, Bride, Groom, Witness, and Informant — from the http://stemma.parallaxview.co/person-role/ namespace. See Event Types and Roles for more details, and see Extended Vocabularies for custom roles.

<PropertyDef Name='Status' ItemList='1' Type='Enum'/>
Status values depend on the context but includes the predefined values: Married, Unmarried, Divorced, Widow(er), Stillborn, Deceased, Absent

(crossed-out), and Implied (i.e. mentioned but not present) — from the http://stemma.parallaxview.co/person-status/ namespace. The default is blank, i.e. none. See Extended Vocabularies for custom status values.

<PropertyDef Name='Relationship' ItemList='1' Type='PersonEL'/>
The relationship of a person to another subject in a multi-subject Event. Relationships are always relative to a specific person (e.g. Wife, Brother, Son-in-law) and contrast with Roles which are relevant to the Event itself. See Relationships for details of the http://stemma.parallaxview.co/person-relationship/ namespace, and see Extended Vocabularies for custom relationships.

<PropertyDef Name='ResidencePlace' Type='PlaceRef'/>
Residential address. This is not the same as the Place that the Event occurred at.

<PropertyDef Name='BirthPlace' Type='PlaceRef'/>
Place of birth.

<PropertyDef Name='CauseOfDeath' Type='Text'/>
A description of the cause of death.

<PropertyDef Name='Disability' Type='Text'/>
A description of some debility, infirmity, or disability.

<PropertyDef Name='GroupEnter' Type='GroupRef'/>
<PropertyDef Name='GroupLeave' Type='GroupRef'/>
Indicates that the person was becoming a member of the indicated group, or leaving it. Although this information will eventually contribute to the Person's <MemberOf> element, note that these Property values are evidence rather than conclusion, and so may differ in different sources.

<PropertyDef Name='DateOfReg' Type='Date'/>
Date of registration or recording of the information source. This is not the same as the Event date derived from that source. For instance, date of birth registration as opposed to date of birth itself. NB: This generic Property should be used sparingly as using an equivalent Event provides more options, e.g. using Constraints to sequence, say, a birth event before the civil registration of that birth.

<PropertyDef Name='RegPlace' Type='PlaceRef'/>
Place of registration or recording of the information source. This is not the same as the Event location derived from that source. For instance, place of marriage registration as opposed to place of the wedding itself.

PROPERTY_VALUE
Any custom Person properties can also be used in the context of PERSON_PROPERTY. See Extended Properties.

Note that property names such as ‘FathersName’, ‘MothersAge’, etc., are strongly discouraged. Properties should be associated directly with a relevant Person entity, not indirectly.

Q: How should the living/non-living status of a Person best be represented? It's obviously problematic since the absence of a death date may just mean you don't know rather than the person is living. Adding a simple "is-living" flag would not work because it could never be kept up-to-date, especially if copies of the data are persisted (in STEMMA) or exchanged with other products.

6.4.2 Lineage

Formats like XML provide an automatic way of depicting a top-down hierarchical relationship. Unfortunately, genealogical lineage is more of a 'network' than a pure 'hierarchy', as many researchers will appreciate, so a simple nesting of "offspring" under their associated "parents" is insufficient. In fact, to be strictly accurate, genealogical lineage is really a “directed acyclic graph” (DAG) since there cannot be any loops, e.g. a child being their own grandparent etc. In contrast to lineage-based genealogy, micro-history associations between Persons, Places, Animals, Groups, and Events are a lot more general and do form an arbitrary network.

There's also a problem with any top-down approach (where biological parents point to children) unless a specific union between two people has a single representation in the data, but that then causes further problems with the nature and the lifetime of that union. In fact, the only universal events for any person are their birth and death. All the others, such as baptism, christening, marriage (civil & religious), divorce, burial, cremation, etc., are culturally dependent and so cannot be expected to be present. Multiple concurrent marriages may even be legal in some cultures. The alternative bottom-up approach (where children point to their respective parents) is the only practical representation.

Each person has just one progenitive father and one progenitive mother[4] and so the associated Person entity can have upward links to its appropriate two parents (where known) using <FatherPersonLnk> and <MotherPersonLnk> elements. These links are permanently valid (i.e. have no date-limited applicability) and form the basis of simple genealogical lineage. They do not define any sort of ‘family unit’ and this is where representations that quietly infer the concept of a family unit from the genealogical data would fall down. See Happy Families.

STEMMA can represent additional types of parentage such as guardians, foster parents, adopted parents, etc., that are designed to support the identification of family units. These are valid over specific periods of time

[4] Actually, technology is capable of engineering children with DNA from three or more “parents” (see uk-government-ivf-dna-three-people). When this becomes a strong requirement for genealogy, STEMMA will address it by having new up-link types other than FatherPersonLnk and MotherPersonLnk. The same approach can be used to handle the case of a surrogate mother, including the gestational type.

since family circumstances can change and so those time periods are determined by Event entities. Furthermore, because micro-history can sometimes be even more complicated than that, the narrative feature associated with all top-level entities will allow arbitrary person-to-person associations.

A typically difficult situation to handle would be a death notice that mentions a mourning grandson by name but gives no indication of which side of the family he is associated with. Such a Person could still be represented in STEMMA and associated with the relevant grandparent through the narrative support.

6.4.3 Personal Names

STEMMA has a single representation of subject names, whether for Persons, Animals, Places, or Groups. Although this section is primarily about the personal names for Person entities, it is using a mechanism that was designed to address cultural differences in naming, as well as being subject-neutral, and it will be used later for those other subject types.

Personal names around the world are not used in the same way as each other, and some things we take for granted in the West have no correspondence elsewhere. As well as variations due to married names, alternative spellings, nicknames, spellings in alternative languages, optional name parts, and stage names, the very structure of a name may be variable leaving it with little uniqueness and no obvious interpretation for our Western given-name/middlename/surname concepts. An in-depth discussion of the issues may be found under Worldwide Family History Data and at The Game of the Name.

The handling of personal names separates the acceptance and matching of the name variants from the generation of the canonical names (i.e. the preferred identifications) during output. Both of these also support the temporal dependencies of those names (e.g. changes during marriage, adoption, etc) and potential overlaps of those time periods.

As a generic approach that applies to all subject entities, STEMMA provides a prioritised set of patterns to match. A 'full name' is defined by a list of possible ‘token sequences'. These are in priority order and imply which should be tested first. Each ‘token sequence' is an ordered set from the following token types:

name	- simple name token, e.g. Tony
{name, ...}	- mandatory selection from alternative tokens
[name, ...]	- optional selection from alternative tokens

The following example might belong to someone called Grace Ann Murphy who doesn't always use her middle name and sometimes goes as Gracie. However, she's Irish and also has an Irish version of her name. This would require the following two 'token sequences':

{Grace,Gracie} [Ann] Murphy
Gráinne [Ann] Ní Murchú

Tokens in each sequence are matched against those in the name from head-to-tail. I emphasise this because some cultures do not write left-to-right.

An interesting issue here concerns the variations of individual name parts. In this example, Grace accepts "Gracie" as an informal version of her forename. However, the difference between Ann and Anne is more of a spelling error, during either recording, transcription or a subsequent lookup. This should be handled by the software unit, just as a *soundex* match might be.

Such patterns are stored in STEMMA using the following elements:

NAME_VARIANTS=

<Names>
<Sequences [RANGE_FROM] [RANGE_TO] [Type='name-type']
[Culture='cultural-style'] [DATA_ATTRIBUTE] ... >
<Canonical [Mode='name-mode'] [SortAs='sort-as'] >
canonical-name </Canonical> ...
<Sequence [NAME_ATTRIBUTE] ... >
<Tokens [Optional='boolean'] [Initial='boolean']>
{ <Token> name-token-ucf-text </Token> } ...
</Tokens> ...
[TEXT_SEG] ...
</Sequence> ...
</Sequences> ...
</Names>

RANGE_FROM=

AfterEvent='key' | FromEvent='key' | After='std-date' | From='std-date'

RANGE_TO=

BeforeEvent='key' | UntilEvent='key' | Before='std-date' | Until='std-date'

NAME_ATTRIBUTE=

Language='code' | Phonetic='boolean' | Romanised='boolean'

As with Event constraints, After is >, From is >=, Before is <, and Until is <=.

When software loads a <Names> element then it should tokenise the canonical names in addition to the explicit token sequences. This enables a certain level of simplification for the cases where there are no accepted token sequences beyond those implied by the canonical names.

The SortAs attribute allows the sort-order to be overridden when it is not determined solely by the available characters (e.g. in Japanese). The string consists of a token-by-token specification with '-' indicating 'no change from the equivalent canonical token'. For instance: "- - Souza".

The Culture attributes is yet to be defined. It is designed to indicate the general style of name and its handling. It therefore implies a prevailing Language code for cases where it has not been overridden. Q: Do we need a default Culture in the Dataset header?

The name-mode may be one of Formal, SemiFormal (default), Informal, and Listing, where 'Listing' is for sorting and collation purposes (e.g. Proctor, Anthony Charles). See http://stemma.parallaxview.co/name-mode namespace.

The name-type may be one of the following. See Extended Vocabularies for defining custom name-types.

- Alias — General pseudonym, including also-known-as, *nom de plume*, pen name, and *nom de guerre*. Some cases may have a specific *type* available for them.
- Married — Name adopted after a marriage ceremony, or other type of union.
- Nickname — Informal alias.
- Personal (default) — Normal personal name, as assigned at birth.
- Petname — Hypocorism. A term of endearment used in more intimate circumstances.
- Private — For cases where a personal name is only used within certain circles, as with some Native American tribes.
- Professional — Includes stage name.
- Public — Some Native American tribes distinguish a private name, used within their own tribe, from a public name used outside of it.

The Initial attribute controls whether individual tokens may be recognised by their initials. When canonical names are being tokenised, this is implied by the Culture setting. Note that initialisms are not applicable in all languages, and even when a foreign name has been Romanised. It is not even the case that subsequent given names following the first may all be placed with initials; a familiar example in genealogical circles being D. Joshua Taylor

The default setting for the Optional attribute is '0' (i.e. False). The optional Event range attributes allow the applicability of a set of sequences to be constrained by relevant Events. The default attributes imply those sequences are always valid. A typical use of these is to differentiate maiden names from married names but they would be applicable for any type of name change. During name matching, it is recommended that the Event range attributes are ignored in order to provide a more relaxed operation. However, in order to derive a Person's full formal name then they should be honoured and in the order they are written, just in case there's any overlap due to fuzzy Event dates.

The name-attributes identifying the language, or whether the representation is phonetic, etc., probably need some clarification. There are a number of terms that often aren't distinguished as well as they should be:

- Transcription - Systematic representation of language (either spoken or prior textual form) in written form. May be phonetic transcription (mapping sounds) or orthographic transcription (mapping spoken words).

- Translation - Conversion of a source language to a target language. Deals with the meaning expressed by the language.

- Transliteration - Conversion of text from one script to another.

- Romanisation (or Latinisation) - Representation of language (either written or spoken) using the Latin script. May use transliteration for written text, or transcription for spoken words.

The <PersonalName> element is provided as a much simplified alternative to the <Names> element for the case where there are no variations and the matched name is identical to just one canonical name. A personal-name specified by a <PersonalName> element is wholly equivalent to a 'SemiFormal' Canonical name provided by the <Names> element.

Do we need to identify a subset of the tokens in a canonical name for highlighting as a surname, or family name, in software products? Note that a blind approach to marking tokens for highlighting avoids all the pitfalls associated with the rigorous categorisation of all name tokens.

In our example, Grace Murphy might be stored as follows, although the first <Sequences> element could be inferred at load-time from the canonical name:

```
<Names>
    <Sequences>
        <Canonical>Grace Ann Murphy</Canonical>
        <Sequence>
            <Tokens Initial='1'>
                <Token>Grace</Token>
                <Token>Gracie</Token>
            </Tokens>
            <Tokens Optional='1' Initial='1'>
                <Token>Ann</Token>
            </Tokens>
            <Tokens>
                <Token>Murphy</Token>
            </Tokens>
        </Sequence>
```

```
            <Sequence Language='gle'>
                <Tokens>
                    <Token>Gráinne</Token>
                </Tokens>
                <Tokens Optional='1'>
                    <Token>Ann</Token>
                </Tokens>
                <Tokens>
                    <Token>Ní</Token>
                </Tokens>
                <Tokens>
                    <Token>Murchú</Token>
                </Tokens>
            </Sequence>
        </Sequences>
</Names>
```

This approach would be familiar to anyone with some knowledge of computer-language parsers. The interpretation of the tokens as given names, etc., might be done by a genealogical product but it is not inherent in the stored data.

Character matching should be relaxed here, as for Place and Group names. The most obvious case of this to people speaking in a Latin-based language is a case-blind match. However, when looking at other Western locales, the next most common instance of a relaxed match is an accent-blind one. This basically means treating, say, A-acute the same as A, etc. This is common in some locales where the accents are routinely dropped for uppercase. There are also characters that have very different representations in upper and lower case. For instance, the German lowercase sharp s in "straße" (known as eszett) usually (there are exceptions) uppercases to "SS", i.e. "STRASSE". After that, there are symbols with both "composed" forms (i.e. one Unicode character) and "decomposed" forms (i.e. 2 or more Unicode characters). For instance, the following should all be treated the same:

212B (Å) ANGSTROM SIGN
00C5 (Å) LATIN CAPITAL LETTER A WITH RING ABOVE
0041 (A) LATIN CAPITAL LETTER A + 030A (°) COMBINING RING ABOVE

Unicode makes specific recommendations about which composed and decomposed forms should be equivalent:
http://www.unicode.org/reports/tr15/.

In summary, any pair of tokens being compared must both be normalised to a "flattened" form that treats each of these categories as equivalent. Only the normalised forms should then be directly compared.

A final note on tokenisation of a name prior to applying the name-matching algorithm: Certain punctuation characters should be used to separate the tokens but should not be present during the matching, e.g. spaces, apostrophe, hyphen, and non-breaking space. Hence, *Henri Cartier-Besson*

should be tokenised as the set [Henri, Cartier, Besson]. An exception to this might be the period which would have to be retained. Hence, *James O. O'Seven* would be tokenised as the set [James, O**.**, O, Seven] to ensure the initial is distinct from a single-character token. See Worldwide Family History Data for further discussion.

6.4.4 Contacts

A special type of person is represented by the Contact entity. These are typically other researchers or contributors who are not part of your family history data. If someone falls into both categories then the CONTACT_DETAILS can be added to their respective Person entity.

A Contact entity can be referenced by PersonRef mark-up, and by Properties or Params of type PersonRef, but not via FatherPersonLnk or MotherPersonLnk connections from a Person entity.

The only data associated with a Contact is their personal name and a set of ContactDetails. If a researcher or contributor also happens to be represented by a Person entity then a set of ContactDetails can be provided in there instead of via a separate Contact entity.

```
CONTACT=

<Contact Key='key'>
        <Names>
                { <Name [Mode='name-mode']> personal-name </Name> } ...
        </Names>
        [ CONTACT_DETAILS ]
        [ TEXT_SEG ] ...
</Contact>

CONTACT_DETAILS=

<ContactDetails>
        [ <Address>
                [ <AddressLine1> line1 </AddressLine1> ]
                [ <AddressLine2> line2 </AddressLine2> ]
                [ <AddressLine3> line3 </AddressLine3> ]
                [ <TownOrCity> town-or-city </TownOrCity> ]
                [ <StateOrProvince> region </StateOrProvince> ]
                [ <PostalCode> postal-code </PostalCode> ]
                [ <CountryCode> iso-3166-1 </CountryCode> ]
        </Address> ]
        [ <Phone>
                { <Number Tag='tag'> E-123-phone-number </Number> } ...
        </Phone> ]
        [ <Emails>
                { <Email Tag='tag'> email-address </Email> } ...
        </Emails> ]
```

```
        [ <Web>
                { <URL Tag='tag'> url </URL> } ...
        </Web> ]
        [ <Messaging>
                { <Message Tag='tag'> account </Message> } ...
        </Messaging> ]
        [ TEXT_SEG ] ...
</ContactDetails>
```

The StateOrProvince address term should be interpreted as the top-level administrative division used in an address for the respective country, e.g. a State or a County. The name-mode is the same as for the <Canonical> element at NAME_VARIANTS.

The phone numbers are E.123 format. The Messaging systems are either instant-text or audio-visual systems. The tag is a user-defined or third-party identifier.

6.4.5 Person Groups

See Group.

6.5 Event

An Event entity represents a date, or range of dates, for which source information exists. This is slightly different to the everyday usage as something that happened at a particular time and place but the definition is deliberate in order to construct substantiated histories, and effectively provides the where-and-when context for subject-entity references within those sources.

A given source may support a number of Events to different levels:

- Directly - as in a census supporting a census event, or a birth certificate supporting a birth event.
- Indirectly - as in a letter that records a previous residence, or a military record that records a date and location of a marriage.
- Implied - as in an age from a census implying a birth event.

Events are not specific to a single subject entity, such as a Person, and as many subject entities as required can be associated with the same Event. Effectively, those subject entities are the thing of interest mentioned in the supporting sources. Information derived from a source is nearly always going to be associated with a given time and place — even if implicit — and this is why STEMMA associates sources with its Event entities (see Evidence and Where to Stick It for more details).

There may even be zero subject entities associated with an Event, as in the case of a country-wide census. This allows a high-level definition of the census Event that fixes the date, place, event-type, etc. Other, more-specific Events can then be defined in terms of the higher-level one in order to refer to

a particular household, and hence individual Persons. Although the associated Place is syntactically optional, for a non-abstract Event one must at least be inherited from a base Event, or be determinate from the largest common Place of any child Events, or be provided by the context (e.g. the use of an Eventlet within a Place entity). In other words, all Events must have an effective Place. When loading the definition of a specific Event, though, any explicit start/end dates or place specified by a hierarchical parent must be correctly bounded any explicit equivalents in the loaded Event; for dates, this might involve the use of implicit constraints between the parent and child Events.

The narrative sections of a Person, Place, Animal, or Group could describe an event in an informal long-hand style, and could even embed computer-readable dates. However, if the event is deemed to be of sufficient importance (e.g. affecting the lives of more than one Person) then the description should be formalised as a top-level Event entity. That Event entity can then be connected to the relevant subject entities, and this is recommended since that common reference point then links the subject entities to give a better account of their history.

Although less common, an Event can be associated with a Place that is a subject entity rather than the Event's primary Place. An example would be the commencement of a voyage that records the intended final destination.

If an event is only relevant to a single subject entity then it can still be formalised without having to create a top-level Entity for it. The Eventlet is a slightly cut-down version of an Event that can be embedded within the relevant subject entity, and which does not require a unique Key associated with it. The other main differences from an Event entity are that an Eventlet does not have any hierarchical relationship, or a <BaseEventLnk> element, or any external IDs. They are primarily a syntactic convenience and offer no functional advantage over top-level Events. Every Eventlet could be converted to a valid Event if necessary. NB: A <SourceLnk> element within an Eventlet must refer to an enclosing subject entity (e.g. by a <PersonLnk> element), without an explicit Key value, and no others of the same entity type; other types being unrestricted. This prevents an Eventlet trying to represent a shared event.

Simple Events represent just one date but *protracted Events* represent both a start and an end date. This means protracted Events have a duration associated with them.

EVENT=

<Event Key='key' [Abstract='boolean']>
[<Title> event-title </Title>]
[<Type> event-type </Type>]
[<SubType> event-subtype </SubType>]
[PLACE_LNK]
<When [Value='std-date'] [DATA_ATTRIBUTE] ... >

```
            [ DATE_ENTITY ] [ EVENT_CONSTRAINTS ]
            [ TEXT_SEG ] ...
        </When>
        [ <Until [Value='std-date'] [DATA_ATTRIBUTE] ... >
            [ DATE_ENTITY ] [ EVENT_CONSTRAINTS ]
            [ TEXT_SEG ] ...
        </Until> ]
        [ <ParentEventLnk Key='key'>
            [ TEXT_SEG ] ...
        </ParentEventLnk> ] ...
        [ <BaseEventLnk Key='key'>
            [ TEXT_SEG ] ...
        </BaseEventLnk> ]
        [ EV_SOURCE_LNK ] ...
        [ EXTERNAL_ID ] ...
        [ TEXT_SEG ] ...
</Event>

EVENTLET=

<Eventlet>
        [ <Title> eventlet-title </Title> ]
        [ <Type> event-type </Type> ]
        [ <SubType> event-subtype </SubType> ]
        [ PLACE_LNK ]
        <When [Value='std-date'] [DATA_ATTRIBUTE] ... >
            [ DATE_ENTITY ] [ EVENT_CONSTRAINTS ]
            [ TEXT_SEG ] ...
        </When>
        [ <Until [Value='std-date'] [DATA_ATTRIBUTE] ... >
            [ DATE_ENTITY ] [ EVENT_CONSTRAINTS ]
            [ TEXT_SEG ] ...
        </Until> ]
        [ EV_SOURCE_LNK ] ...
        [ TEXT_SEG ] ...
</Eventlet>
```

The Properties associated with each subject entity mentioned in the supporting sources are attached to the corresponding Event-to-subject link (see Timelines, Multi-Source Events, and Personal Events). If there is no corresponding subject entity (e.g. a reference to an incidental person) then an unkeyed <PersonLnk>, <AnimalLnk>, <PlaceLnk>, or <GroupLnk> element may be used.

One of the design goals of STEMMA was to not simply date individual items, such as photographs and Property values, but to associate the aggregate with a single Event (or Eventlet). If, for instance, a source described someone changing their occupation, residence, and personal name — all at the same time — and there were several photograph taken at that same time, then the

aggregate knowledge could be associated with the corresponding change Event; this being more informative than simply duplicating the same date and source details on each item.

Each Event can have one-or-more <SourceLnk> elements to represent summarised information from specific supporting sources. References to people, animals, places, or groups, can be assembled into these elements as sets of associated Property values (e.g. name, occupation). These summary forms can be connected to specific subject entities if a correspondence has been deduced. A Property that references another named subject entity, such as a place or relative, can be done using a PlaceRef-type Property with a Key attribute. NB: Properties for the Event itself (e.g. the dates and the place) would be provided separately in the <SourceLnk> element. When SourceLnk is employed within an <EventLet> element then only a single un-keyed <PersonLnk> element (or <PlaceLnk>, etc., depending on the enclosing entity type) can be specified of the same type as the enclosing entity.

EV_SOURCE_LNK=

<SourceLnk Key='key'>
[PERSON_LNK] ...
[ANIMAL_LNK] ...
[PLACE_LNK] ...
[GROUP_LNK] ...

[EVENT_PROPERTY] ...

[TEXT_SEG] ...
</SourceLnk>

The <BaseEventLnk> element may nominate an Abstract Event from which data may be inherited by the current Event, in much the same vein as base classes and derived classes in software programming. An Abstract Event must define no embedded Keys and can only reference other abstract entities. It is also allowed to omit the <PlaceLnk> element.

The event-type provides a coarse grouping of similar event-subtypes such as Union, Birth, Death, Religious, Legal, Travel, and Military. See Event Types and Roles for a list of these and their relationship to roles, and see Extended Vocabularies for custom event types.

The start of the Event can be specified either through a combination of a date and Event constraints, or by reference to a different Event. A date may have error margins associated with it (see Dates), and Event constraints (see Constraints) allow it to be limited by the dates of one-of-more other Events. These may be used separately or together to express something similar to 'Event1 happened in c1810 but it was after Event2 and before than Event3'. When the start of an Event is simply defined in terms of a specific other Event then it inherits the start date definition of that other Event.

Notice that the Event has an optional definition of an end date. This effectively defines a *protracted Event* that spans a period of time. Examples of this might include a long sea voyage which has both a date of emigration and a date of immigration, or WWI which has a date that war broke out and a separate date for Armistice Day. Each of the start and end dates can be defined in terms of discrete Events. This approach allows the two end-points to have independent Place references, which then has an obvious application for a journey. It further allows the individual end-points to be referenced separately. For instance, if something in a person's family history was relevant specifically to the outbreak of WWI or to the ceasing of hostilities then a direct reference to the sub-Event can be made.

For a protracted Event, if the <Until> element is defined in terms of a specific other Event then it inherits the end date definition of that other Event. If the specified Event is a *simple Event* then its only date is always applicable.

Dates may be specified either as a compact date-value (using the Value= attribute) or a more powerful date-entity (using the DATE_ENTITY structure). These should be considered mutually exclusive.

EVENT_PROPERTY=

<PropertyDef Name='When' Type='Date'/>
The date at which the event commenced.

<PropertyDef Name='Until' Type='Date'/>
The date at which the event finished. Defaults to the commencement date.

<PropertyDef Name='Where' Type='PlaceRef'/>
The place at which the event took place.

PROPERTY_VALUE
Any custom Event properties can also be used in the context of EVENT_PROPERTY. See Extended Properties.

6.5.1 Dates

There are two distinct date representations in STEMMA: the date-value and the date-entity. These are equivalent for many purposes but the date-entity affords greater flexibility and scope. A date-value is encoded in a single text string, while a date-entity is a combination of elements and attributes.

The general representation of date-values is mentioned under Locale-independence. STEMMA must accommodate multiple calendars but, at the time of writing, no international standard yet exists. It therefore introduces a practical date-value string representation for world calendars and differentiates the ISO and STEMMA forms accepted in the STEMMA syntax as follows.

- **iso-date** — date subset of ISO 8601 Gregorian standard, i.e. yyyy[-mm[-dd]].

- **iso-datetime** — full ISO 8601 Gregorian date and time, i.e. yyyy-mm-ddThh:mm:ssZ.
- **std-date** — STEMMA date referenced in specific calendar, analogous to **iso-date**. Calendar can be specified by optional prefix, as described under Dates and Calendars research notes, or implied by the syntactic context. The reserved value "?" indicates *not known*.
- **std-fulldate** — full STEMMA date reference in specific calendar. That is, with a granularity of only one day. The reserved value "?" indicates *not known*.

When a date is referenced in an element, it may have both a *granularity* and an *imprecision* (i.e. a margin of error) associated with it. The granularity is implicit in the date-value string (see under Dates and Calendars). The imprecision can be represented by a +/- offset or more explicit min/max limits in a date-entity.

DATE_ENTITY=

<Date>
{ <Value [Calendar='name'] [Calc='boolean'] [Margin='err']> std-date </Value>
|
{ <Range [Calendar='name'] [Calc='boolean']> <Min> std-fulldate </Min> <Max> std-fulldate </Max> </Range> } } ...
[TEXT_SEG] ...
</Date>

The default calendar name is "Gregorian". The calendar may be specified explicitly in the Calendar attribute or in the date-value (as described above), but they must not conflict.

This date-entity effectively allows a specific date to be represented using a value, or a range of values, from one-or-more calendars, and this is used in modelling *synchronised dates* (aka: dual dates). The Calc attribute indicates that the value for that calendar was calculated as opposed to being recorded as part of the original information. A discussion of this, with examples, may be found at: Synchronised Dates.

A date-value may imply a granularity other than one day using truncated forms. For Gregorian dates, this includes the normal yyyy-mm and yyyy, as in the ISO standard, but also yyyy-mm:xx and yyyy:xx. For comparative purposes (e.g. sorting and collation) the truncated variants are equivalent to a corresponding pair of <Min> and <Max> elements. The default error margin is ± 0. The margin units depend on the granularity of the date-value. Hence, a full yyyy-mm-dd specification would expect a margin in days. If the date-value is truncated to yyyy-mm: then any margin would be in months. If the date value is truncated to yyyy: then any margin would be in years. The <Min> and <Max> must always be full-length dates (e.g. yyyy-mm-dd in the Gregorian case).

A representation of yearly quarters (e.g. Q1 = January to March) is noticeably absent from the ISO 8601 standard. Given the way that it represents week numbers, it should have made provision for the format yyyy-Qq, e.g. 1956-Q2. STEMMA acknowledges the importance of this granularity for certain records and so accommodates it in its own world-calendar syntax. The units of any margin would then be quarters of course.

The following table indicates how a Margin specification is interpreted in the context of the date units to yield equivalent Min/Max values.

Date form	Margin units	Equivalent Min	Equivalent Max
yyyy-mm-dd	Days	The day - margin	The day + margin
yyyy-mm	Months	First day of (month - margin)	Last day of (month + margin)
yyyy	Years	First day of first month of (year - margin)	Last day of last month of (year + margin)
yyyy-mm:03	Quarters	First day of first month of (quarter - margin)	Last day of last month of (quarter + margin)

When deterministic dates, such as our normal Gregorian ones, are loaded into some type of indexing system, like a database, it is expected that they will all be stored as pairs of internal 'timestamp' values, i.e. one each for the upper and lower limits. Timestamps represent points-in-time along an absolute timeline, starting at some arbitrary base date (aka: epoch). Since these are usually represented as binary long-integers then it means issues such as the external date representation, imprecision, TZ, etc., all become irrelevant and the values can all be handled efficiently in the same manner.

The following table indicates how comparisons should be implemented between dates when either one of them may be a simple discrete date (e.g. A) or an inclusive date range with an upper and lower limit (e.g. [A1,A2]). In the context of a date range, equality is roughly translated as “some degree of overlap”.

A op B	[A1,A2] op [B1,B2]	A op [B1,B2]	[A1,A2] op B
A > B	A1 > B2	A > B2	A1 > B
A < B	A2 < B1	A < B1	A2 < B
A = B	A2 >= B1 & A1 <= B2	A >= B1 & A <= B2	A2 >= B & A1 <= B
A >= B	A1 >= B1	A >= B1	A1 >= B
A <= B	A2 <= B2	A <= B2	A2 <= B
A <> B	A2 < B1 or A1 > B2	A < B1 or A > B2	A2 < B or A1 > B

Q: Should the Dataset header specify a default Calendar or simply assume Gregorian as the default? Most Datasets will involve dates from one predominant Calendar and so it would be more convenient to specify a default for cases when no explicit one has been provided. See Locale-independence for potential Calendar names.

6.5.2 Constraints

Being able to place limits on Events that relate to other Events is a powerful feature. The <Constraints> element allows an Event to be sequenced relative to one-or-more other Events.

```
EVENT_CONSTRAINTS=

<Constraints>
        <Constraint { AfterEvent='key'  |  BeforeEvent='key'
                    | FromEvent='key'   |  UntilEvent='key'
                    | AtEvent='key' } >
                   [ TEXT_SEG ] ...
        </Constraint> ...
</Constraints>
```

The interpretations of the elements are as follows:

- **AfterEvent**. Current event > specified event
- **BeforeEvent**. Current event < specified event
- **FromEvent**. Current event >= specified event
- **UntilEvent**. Current event <= specified event
- **AtEvent**. Current event = specified event

If there are causal relationships between Events then the <Text> elements can be used to describe it.

The context of the Constraint, i.e. whether it appears in a <When> or <Until> element, determines whether it relates to the start or end date. If the specified Event is a *simple Event* then its only date is always applicable. Only one constraint for a given context may be established between two particular Events (e.g. no duplication for a start date), and they must not be self-referential.

Q: Should the relationship between events involve an optional duration? For instance: event-B is after event-A + 3 months. GenTech V1.1 (appending C3) suggested something similar: “Date data can be relative. Dates can be relative to an event, such as “four months after marriage”.
Q: Do we need multi-unit durations, such as 3y 120d? This would affect the ‘Age’ Property of both Person and Animal.

6.6 Place

Place entities can describe a household address, or a street, a village, a town, a city, a whole country, etc. They have a strong hierarchical structure which effectively factors out common data. This reduced duplication helps to ensure consistency of the data and better supports geographical analysis. For instance, simply being able to see who else in a Dataset lived near the same address. It also allows relevant material (links, documents, images, maps, etc.) to be associated with the correct place, and not simply some related place.

PLACE=

```
<Place Key='key'>
        [ <Title> place-title </Title> ]
        [ <Type> place-type </Type> ]
        [ <Category> place-category </Category> ]
        [ NAME_VARIANTS  |  { <PlaceName [DATA_ATTRIBUTE] ... > place-
        name </PlaceName> } ]
        [ <Location [Open='boolean'] [RANGE_FROM] [RANGE_TO]>
                coordinates
                [ TEXT_SEG ] ...
        </Location> ] ...
        [ <ParentPlaceLnk [RANGE_FROM] [RANGE_TO] Key='key'>
                [ TEXT_SEG ] ...
        </ParentPlaceLnk> ] ...
        [ <Creation>
                EVENT_LNK  |  EVENTLET
                [ <JoinFrom Key='key'/> ...  ]
                [ TEXT_SEG ] ...
        </Creation> ]
        [ <Demise>
                EVENT_LNK  |  EVENTLET
                [ <SplitTo Key='key'/> ...  ]
                [ TEXT_SEG ] ...
        </Demise> ]

        [ EVENTLET ] ...
        [ SOURCE_LNK ] ...
        [ <RelatedTo>
                <PlaceLnk Key='key'/> ...
                [ TEXT_SEG ] ...
        </RelatedTo> ]
        [ EXTERNAL_ID ] ...
        [ TEXT_SEG ] ...
</Place>
```

The location of the Place may be specified using optional <Location> elements — more than one of a location has changed over time. This normally contains an ISO 6709 longitude/latitude pair (for a point) or an

ordered closed list of such values describing a polygon (for an irregular area — requires minimum of 3 points). An Open='1' attribute may be added to declare an ordered open list for representing roads and streets (Open='0' is the default for 3 or more points). For instance:

```
<Place Key='wGladeHill'>
        <PlaceName> Glade Hill </PlaceName>
        <ParentPlaceLnk Key='cNottm'/>

        <Location> +53.0020-001.1517/+53.0018-001.1510/
        +53.0012-001.1510/+53.0012-001.1522/+53.0017-001.1527/
        </Location>

        <Text>
        A differentiation between this place and nearby
        <PlaceRef>Bendigo's Ring</PlaceRef> can be found at <Link
        URL='http://parallax-viewpoint.blogspot.com/2013/11/where-is-
        bendigos-ring.html'>Where is Bendio's Ring?</Link>
        </Text>
</Place>
```

The ISO standard mandates a trailing '/' at the end of a pair and so an ordered list is simply a concatenated series of them. If a different coordinate system is required, such as the Ordnance Survey National Grid of Great Britain, then it must use an explicit namespace prefix. Support for alternative systems can be important if interfacing to local map resources; especially through a software interface. Alternatively, an authoritative set of coordinates can be imported from some external Place Authority using one of the EXTERNAL_ID values.

Items of information about the place that have been extracted from relevant sources may be specified using the following Properties. These are defined here using the syntax used for Extended Properties.

PLACE_PROPERTY=

<PropertyDef Name='Name' Type='PlaceRef'/>
The name of the place, as recorded in the supporting source.

<PropertyDef Name='Parish' Type='Text'/>
The name of the ecclesiastical parish associated with the Place.

<PropertyDef Name='Role' ItemList='1' Type='EnumList'/>
The role(s) of a place mentioned in Event sources. Role values currently include Destination, from the http://stemma.parallaxview.co/place-role/ namespace. See Extended Vocabularies for custom roles.

PROPERTY_VALUE
Any custom Place properties can also be used in the context of PLACE_PROPERTY. See Extended Properties.

Places may be part of several flavours of hierarchy such as registration/statistical (registration district, in registration county), or religious (ecclesiastical parish, in diocese, in archdiocese), electoral wards, etc., but the Place entity primarily focuses on the geographical/administrative hierarchy. The underlying premise is that every place has a unique bounding parent place at any given time. The other jurisdictional zones could be represented as simple properties, although alternative hierarchies can also be constructed as long as that premise is not violated. The Place entity also support non-hierarchical relationships (see Related Entities) that may be used to tie alternative hierarchies together, if necessary. A typical example that is used to justify an entity having multiple concurrent parents — contrary to what was said above — is that of a street being in both an administrative area and a religious parish. However, it's not uncommon to have a single street split between two such zones, especially where electoral divisions have been defined. This is actually a good case for linking hierarchies of different types using the <RelatedTo> element.

The <Title> element provides a unique descriptive title that will be used to identify the Place in genealogical reports. This is not the same as the Place-hierarchy-path as reconstructed from the Place-hierarchy (see Place Names) which may be date-dependent.

A Place entity does not demand a completely determinate address but where the address is known then it is connected to a parent Place, and so on. The termination point of this hierarchy is a matter of choice. For example, in the following Nottingham address, we could have continued the parentage all the way up to Country (England). This would have to be a consideration if a Dataset spanned several countries since no Place-hierarchy-path should be ambiguous:

> [Nottinghamshire, Nottingham, Manning Grove, 15]

The following place-types are predefined:

- Small-scale geographical
  - Apartment — Apartment name or number.
  - Building — A named building. This may be a house, church, institution, company building, or a set of apartments or flats.
  - Number — A numbered building on a street.
  - Street — Street or road name.
  - Hamlet — Small group of households.
  - Village — Village name.
  - Area — Neighbourhood or local area within town or city.
  - Town — Town or city name. Classification may change over time.
- Administrative divisions
  - Authority, Borough, Canton, Colony, County, Department, Dependency, District, Island, Municipality, Parish (Civil),

Province, Region, Republic, Shire, States, Territory, Townland, Township.

- Large-scale geographical
  - Country — Country name. May not be in ISO 3166 list.
  - Vessel — A ship or other named vehicle travelling between locations (bounding location not applicable).
  - Unknown.

For instance: “Flat 18 (apartment), Da Vinci House (building), 44 (number) Saffron Hill (street), London (town)”. For programmers in particular, a very useful list of presumptions that they often make about places and addresses may be found at: Falsehoods Programmers Believe About Addresses. See Extended Vocabularies for defining custom place-types.

Note that the hierarchical link to the parent Place can be made dependent upon the date. For instance, a Place may have moved from one county to another. An example of this is presented for the Place Key=’wStapenhill’ in the STEMMA Example section. The <Creation> and <Demise> elements can detail when a place was first defined, or ceased to exist. An example of this is presented below for Key=’wJessamineCottages’.

Q: Should place-type be time-dependent? we avoid the change of status of many towns to cities by having a single term (Town) for both.

Complex issues such as places being split up, or joined together, or having some other type of connection to a different place, are handled using the <SplitTo> elements (in Demise), the <JoinFrom> elements (in Creation), and <RelatedTo> elements. More details can be found at: Related Entities.

By contrast with the place-type, the place-category is looser. It does not constitute a hierarchy, and the values are not necessarily subordinate to place-type. The following values are currently defined: Cemetery, Church, Hospital, School, Ship, Prison, and Workhouse. Again, see Extended Vocabularies for defining custom place-categories.

Custom properties can be used as in the following illustration (see Extended Properties):

```
<Dataset Name=’Example’ xmlns:x=’http://superproduct.com/properties’>

<ExtendedProperties>
        <PlaceProperties>
                <PropertyDef Name=’x:Ward’ Type=’Text’/>
        </PlaceProperties>
</ExtendedProperties>

<Place Key=’wJessamineCottages’>
        <Title>Jessamine Cottages</Title>
        <Type>Street</Type>
        <PlaceName> Jessamine Cottages </PlaceName>
```

```
        <ParentPlaceLnk Key='wNottm'/>
        <Demise>
            <Eventlet><When Value='1956'/></Eventlet>
        </Demise>
</Place>

<Event Key='eCensus1901'>
        <SourceLnk Key='sCensus1901'>
            <PlaceLnk Key='wJessamineCottages'>
                <Property Name='Parish'> Nottingham St Nicholas
                </Property>
                <Property Name='x:Ward'> Market </Property>
            </PlaceLnk>
        </SourceLnk>
</Event>
```

Some notes on various other properties that are often associated with a place or an address (see also the contact details support under Contacts):

Postal code
Postal codes or zip codes differ widely between countries and so can only be handled as plain strings. Some are based on geographical coordinates while others are based on postal sorting offices zones. At the time of writing, some countries did not have such a system, and Ireland only launched one on 13 July 2015, so it should never be mandated.

Telephone numbers
All stored telephone numbers should be fully international. The E.164 standard specifies how to represent an international telephone number, e.g. +15551234567. It does not specifically separate the ISD country dialling code, trunk code, or subscriber number. Also, it does not represent any trunk prefix required within that country. Formats such as +44 (0)1728 123456 should be avoided as the parenthesised trunk prefix is a UK/Ireland-centric way of representing numbers. E.123 is similar to E.164 but allows for some restricted punctuation for readability. Presentation differs greatly between countries but formatting them for a given locale has never been provided as part of software locale systems; it has always been added as an extra layer on top.

Geographic Coordinates
ISO 6709:2008 supports point location representation through the use of XML but, recognizing the need for compatibility with the previous version of the standard, ISO 6709:1983, it also allows for the use of a single alphanumeric string. The ISO string representations can be concatenated to generate an ordered set of coordinates for describing a polygon, and it is therefore used in the Place entity's <Location> element. In general, the coordinates for well-established places should be left to a Place Authority rather than duplicated in all instances of personal data.

6.6.1 Place Names

Place names are handled almost identically to Person names (see Personal Names) and so automatically support name variants and temporal dependencies. This sharing therefore includes the syntax of the <Names> element, Event/date range attributes, and the relaxed name-matching semantics.

The <PlaceName> element is provided as a much simplified alternative to the <Names> element for the case where there are no variations and the matched name is identical to just one canonical name. This is very useful, for instance, in the case of a simple house number or house name. A place-name specified by a <PlaceName> element is the equivalent of a 'SemiFormal' Canonical name provided by a <Names> element. When constructing a *Place-hierarchy-path* then it is this name-type (subject to From/To ranges) that is used for each of the terms. If not provided then the corresponding fall-back sequence is documented under PLACE_REF.

For instance:

```
<Place Key='wNotts'>
        <Title>Nottingham County</Title>
        <Type>County</Type>
        <Names>
                <Sequences>
                        <Canonical>Nottinghamshire</Canonical>
                        <Sequence>
                                <Tokens>
                                        <Token>Nottinghamshire</Token>
                                        <Token>Notts</Token>
                                </Tokens>
                        </Sequence>
                </Sequences>
        </Names>
</Place>

<Place Key='wNottm'>
        <Title>Nottingham City</Title>
        <Type>Town</Type>
        <Names>
                <Sequences>
                        <Canonical>Nottingham</Canonical>
                        <Sequence>
                                <Tokens>
                                        <Token>Nottingham</Token>
                                        <Token>Nottm</Token>
                                </Tokens>
                        </Sequence>
                </Sequences>
        </Names>
```

```
        <ParentPlaceLnk Key='wNotts'/>
</Place>

<Place Key='wManningGrove'>
        <Title>Manning Grove</Title>
        <Type>Street</Type>
        <Names>
                <Sequences>
                        <Canonical>Manning Grove</Canonical>
                        <Sequence>
                                <Tokens>
                                        <Token>Manning</Token>
                                </Tokens>
                                <Tokens>
                                        <Token>Grove</Token>
                                        <Token>Gr</Token>
                                </Tokens>
                        </Sequence>
                </Sequences>
        </Names>
        <ParentPlaceLnk Key='wNottm'/>
</Place>

<Place Key='wManningGrove15'>
        <Title> Family Home </Title>
        <Type> Number </Type>
        <PlaceName> 15 </PlaceName>
        <ParentPlaceLnk Key='wManningGrove'/>
</Place>
```

In this case, the Place-hierarchy-path reconstructed from the Place-hierarchy would be:

15, Manning Grove, Nottingham, Nottinghamshire

Note that the comma separator is culturally dependent and so is not prescribed by STEMMA. Nor is the ordering (big-to-small or small-to-big). It's worth emphasising that a Place-hierarchy is not an address, and so issues such as the comma being dropped after the number, or the number appearing after the street in Netherlands, are not prescribed here.

Q: Do we need an optional no-show attribute to omit certain Places when generating a Place-hierarchy-path?

6.7 Group

STEMMA implements a generic Group concept that can model an organised real-world concept such as a regiment. It qualifies as a *subject entity* because

historical sources may contain references to that group, irrespective of whether people, or any other subject entity, are mentioned within it.

A Group shares many of the same features as a Place, such as time-dependent hierarchical structure, non-hierarchical relationships, and alternative names.

A Set of subject entities (currently only Persons and Animals — see Extended Schema) can be associated with a Group over specific periods of time, and manipulated using Set operators to create derived Sets. One of the uses employing Sets is to model a Family concept. As the subject matter under Worldwide Family History Data explains, the concept of a *family unit* is very subjective, and has more in common with general membership of groups rather than lineage or marriage. See also Family Units, Happy Families, and Revisiting the Family Group.

Individual Persons and Animals can be linked to one-or-more Groups using the <MemberOf> element within their respective entities These specify a *begin* and *end* by either date or Event, and these default to the birth and death of that subject if the range is left open-ended. Set operators may be used to adjust the content based on other Groups. This happens after any explicit associations have been recorded and is primarily designed to support the creation of derived Groups. For instance, Persons in both Group A *and* B, or Persons in Group A *but not* B, etc.

GROUP=

```
<Group Key='key'>
        [ <Title> group-title </Title> ]
        [ <Type> group-type </Type> ]
        [ <SubType> group-subtype </SubType> ]
        [ NAME_VARIANTS  |  { <GroupName [DATA_ATTRIBUTE] ... > group-
        name </GroupName> } ]
        [ <ParentGroupLnk [RANGE_FROM] [RANGE_TO] Key='key'>
                [ TEXT_SEG ] ...
        </ParentGroupLnk> ] ...
        [ <Creation>
                EVENT_LNK  |  EVENTLET
                [ <JoinFrom Key='key'/> ...  ]
                [ TEXT_SEG ] ...
        </Creation> ]
        [ <Demise>
                EVENT_LNK  |  EVENTLET
                [ <SplitTo Key='key'/> ...  ]
                [ TEXT_SEG ] ...
        </Demise> ]

        [ EVENTLET ] ...
        [ SOURCE_LNK ] ...
        [ <RelatedTo>
```

```
            <GroupLnk Key='key'/> ...
            [ TEXT_SEG ] ...
        </RelatedTo> ]
        [ <Operation Name='group-op' Key='key'/> ] ...
        [ EXTERNAL_ID ] ...
        [ TEXT_SEG ] ...
</Group>
```

The <SplitTo>, <JoinFrom>, and <RelatedTo> elements are described further under Place.

Named items of information about the group as a whole that may be extracted from an associated source can be specified using the following Properties. These are defined here using the syntax used for Extended Properties.

GROUP_PROPERTY=

<PropertyDef Name='Name' Type='GroupRef'/>
The name of the group, as recorded in the supporting source.

PROPERTY_VALUE
Any custom Group properties can also be used in the context of GROUP_PROPERTY. See Extended Properties.

The <GroupName> element is provided as a much simplified alternative to the <Names> element for the case where there are no variations and the matched name is identical to just one canonical name. A group-name specified by a <GroupName> element is the equivalent of a 'SemiFormal' Canonical name provided by a <Names> element.

There is a broad set of group-types including Family, Education, and Military. There is also a set of predefined group-subtypes that includes those family-related ones discussed in the aforementioned section:

- Matrilocal. A mother and her children.
- Nuclear (or Conjugal family). A husband, his wife, and children.
- Extended (or Consanguineal family). In which parents and children co-reside with other members of a parent's family.
- Blended (or Step-family). Families with mixed parents. For instance, where one or both parents remarried, bringing children of the former family into the new family.

There are also predefined group-subtypes for Regiment, and Household (e.g. inclusive of boarders, lodgers, and staff). See Extended Vocabularies for creating custom group-types and group-subtypes.

The Person and Animal associations are time-dependent so each Group is not a static SET. When interrogated for membership, a date must therefore be provided.

Each group-op must be one of:

- Union — (A U B)
- Intersect — (A ∩ B)
- Differ — (A U B) - (A ∩ B)
- Exclude — (A - B)

.
These are not time-dependent and so can be applied directly after all the Group associations have been loaded, or after a subsequent change in a Group association.

6.8 Animal

This entity is strongly modelled on the Person entity: each have an associated hierarchy that depicts their biological lineage, each have alternative names, each have birth and death Events, and each can be members of a Group. The main differences, other than some slightly renamed elements, are that the Animal entity does not have any CONTACT_DETAILS, but it does have <Type> and <SubType> elements.

ANIMAL=

<Animal Key='key'>
[<Title> animal-title </Title>]
[<Type> animal-type </Type>]
[<SubType> animal-subtype </SubType>]
[<Sex [DATA_ATTRIBUTE] ... > boolean </Sex>]
[NAME_VARIANTS | { <AnimalName [DATA_ATTRIBUTE] ... > animal-name </AnimalName> }]
[<FatherAnimalLnk Key='key' [DATA_ATTRIBUTE] ... >
[TEXT_SEG] ...
</FatherAnimalLink>]
[<MotherAnimalLnk Key='key' [DATA_ATTRIBUTE] ... >
[TEXT_SEG] ...
</MotherAnimalLink>]

[<Birth>
EVENT_LNK | EVENTLET
[TEXT_SEG] ...
</Birth>]
[<Death>
EVENT_LNK | EVENTLET
[TEXT_SEG] ...
</Death>]
[<MemberOf Key='key' [RANGE_FROM] [RANGE_TO]>
[TEXT_SEG] ...
</MemberOf>] ...

[EVENTLET] ...
[SOURCE_LNK] ...

[EXTERNAL_ID] ...
[TEXT_SEG] ...
</Animal>

The animal-type is the species or breed (e.g. Dog, Cat) and is defined by the partially-controlled vocabulary http://stemma.parallaxview.co/animal-type. The animal-subtype represents the importance of the animal to the associated Persons (e.g. Pet, Mascot) and is represented by the partially-controlled vocabulary http://stemma.parallaxview.co/animal-subtype. See Extended Vocabularies for creating custom animal-types and animal-subtypes.

The <AnimalName> element is provided as a much simplified alternative to the <Names> element for the case where there are no variations and the matched name is identical to just one canonical name. An animal-name specified by an <AnimalName> element is the equivalent of a 'SemiFormal' Canonical name provided by a <Names> element.

The full syntax for defining Properties, and for providing values, may be found at: Extended Properties. This scheme is fully open to custom properties. The ones presented below are the predefined ones associated with the http://stemma.parallaxview.co/animal-property namespace.

ANIMAL_PROPERTY=

<PropertyDef Name='Name' Type='AnimalRef'/>
The personal name of the animal, as recorded in the supporting source.

<PropertyDef Name='Age' Units='y,m,w,d' Type='Measure'/>
The age of the animal, usually at the start of the current event. Default unit is 'y' (years) but also valid are 'm' (months), 'w' (weeks), and 'd' (days). The value may be fractional, e.g. 6.5 years, in which case the decimal point character is strictly prescribed under Locale-independence. Currently, no multi-unit ages are allowed, e.g. 3y 2m.

<PropertyDef Name='Role' ItemList='1' Type='EnumList'/>
The role(s) of an animal in a multi-subject Event. Role values depend on the context but include: Parent, Offspring, and Deceased — from the http://stemma.parallaxview.co/animal-role/ namespace — or any of the sub-types from the http://stemma.parallaxview.co/animal-subtype/ namespace mentioned earlier. See Extended Vocabularies for custom roles.

<PropertyDef Name='Status' ItemList='1' Type='Enum'/>
Status values depend on the context but includes the predefined values: Deceased, and Implied (i.e. mentioned but not present) — from the http://stemma.parallaxview.co/animal-status/ namespace. The default is blank, i.e. none. See Extended Vocabularies for custom status values.

<PropertyDef Name='Relationship' ItemList='1' Type='AnimalEL'/>
The relationship of an animal to another subject in a multi-subject Event. Animal Relationships are always relative to a specific person (e.g. Pet) or

animal (e.g. Offspring, Parent, Sibling) — using the http://stemma.parallaxview.co/animal-relationship/ namespace — and contrast with Roles which are relevant to the Event itself. See Extended Vocabularies for custom relationships.

<PropertyDef Name='ResidencePlace' Type='PlaceRef'/>
Residential address. This is not the same as the Place that the Event occurred at.

<PropertyDef Name='BirthPlace' Type='PlaceRef'/>
Place of birth.

<PropertyDef Name='CauseOfDeath' Type='Text'/>
A description of the cause of death.

<PropertyDef Name='GroupEnter' Type='GroupRef'/>
<PropertyDef Name='GroupLeave' Type='GroupRef'/>
Indicates that the animal was becoming a member of the indicated group, or leaving it. Although this information will eventually contribute to the Animal's <MemberOf> element, note that these Property values are evidence rather than conclusion, and so may differ in different sources.

<PropertyDef Name='DateOfReg' Type='Date'/>
Date of registration or recording of the information source. This is not the same as the Event date derived from that source. For instance, date of birth registration as opposed to date of birth itself. NB: This generic Property should be used sparingly as using an equivalent Event provides more options, e.g. using Constraints to sequence, say, a birth event before the registration of that birth.

<PropertyDef Name='RegPlace' Type='PlaceRef'/>
Place of registration or recording of the information source. This is not the same as the Event location derived from that source. For instance, place of marriage registration as opposed to place of the wedding itself.

PROPERTY_VALUE
Any custom Animal properties can also be used in the context of ANIMAL_PROPERTY. See Extended Properties.

6.9 Source

The Source entity is a core component of STEMMA's informational sub-model — that is, the support that relates source descriptions, assimilation of information therein, analysis, transcriptions, and citations. More than that, though, it also provides a link from conclusions back through the reasoning and evidence, to the information involved, and to the underlying sources.

The Source entity is primarily built from a number of linked items, called *profiles*, which relate source fragments to concepts and relationships in a humanly-readable manner. Such information may be used to implement a

Graphic Organiser that allows those relationships to be analysed visually in a process known as Link Analysis.

SOURCE=

```
<Source Key='key'>
        [ <Title> source-title </Title> ]
        SRC_FRAME
        [ SRC_PROTO_SUBJ | SRC_COMMENTARY | SRC_DATE ] ...
        [ SOURCELET ] ...
        [ TEXT_SEG ] ...
</Source>
```

SRC_FRAME=

```
<Frame>
        [ <Where DetLnk='key'/> ]
        [ <When DetLnk='key' | Value='std-date'/> ]
        { CITATION_LNK | RESOURCE_LNK } ...
        [ <Credibility> information-credibility </Credibility> ]
        [ <Reliability> information-reliability </Reliability> ]
        [ <Quality> source-quality </Quality> ]
        [ TEXT_SEG ] ...
</Frame>
```

SOURCELET=

```
<SourceLet [Key='key']>
        [ <Title> sourcelet-title </Title> ]
        SRC_FRAME
        [ SRC_PROTO_SUBJ | SRC_COMMENTARY | SRC_DATE ] ...
        [ TEXT_SEG ] ...
</SourceLet>
```

SRC_PROTO_SUBJ=

```
<ProtoPerson { DetKey='key' || Key='key' }>
        [ <Title> proto-title </Title> ]
        SRC_LINK ...
        [ TEXT_SEG ] ...
</ProtoPerson>
<!-- or ProtoAnimal, ProtoPlace, ProtoGroup, ProtoEvent -->
```

SRC_COMMENTARY=

```
<Commentary DetKey='key'>
        [ <Title> commentary-title </Title> ]
        SRC_LINK ...
        [ TEXT_SEG ] ...
```

</Commentary>

SRC_DATE=

<ProtoDate DetKey='key'>
[<Title> date-title </Title>]
SRC_LINK ...
[TEXT_SEG] ...
</ProtoDate>

SRC_LINK=

<Link [Type='link-type'] { DetLnk='key' || Value='value' }>
[TEXT_SEG] ...
</Link>

A ProtoPerson profile basically corresponds to what is described as a "persona" elsewhere, but there are also equivalent profiles for references to animals, places, and groups.

The various profiles — prototype subjects, prototype events, prototype dates, and commentary — have by one or more links; links to source fragments or to other profiles. Each link has a link-phrase that is used to describe the semantics of its data. Any outputs from a profile (i.e. Value and/or DetLnk) are associated with this link-phrase, and by default all links from an input profile are carried through to higher-level profiles, creating *threads of information*. For instance, a prototype person may have links to several source fragments that use the same link-phrase of "name". By default, they would be merged, in their declared order, to yield a single Value and/or DetLnk, but an output link may make an inference to change the exported value for that thread. This is basically what happens in any profile: the links are acted up in-order to control the output threads, but the profile as a whole may be thought of as a "black box" that just relates inputs to outputs. There are very few rules and the connectivity is subjective, according to the assimilation-analysis process of the user.

The distinction between "links", which are mechanically created using the <Link> element, and "threads", which are a notional concept, may appear confusing at first. The easiest way to visualise it is that the threads carry named items of information, and the profiles are snapshots of that information and knowledge at a particular point in the assimilation and analysis. The links indicate a dependency of one profile on another.

In software terms, the threads constitute notional named tuples of the form link-phrase={Value,DetLnk} that are carried by the links. The links must constitute a Directed Acyclic Graph (DAG), meaning no circularity.

The following link-types are currently predefined in the partially-controlled vocabulary http://stemma.parallaxview.co/source-link-type.

- Source — Provides a link to a source fragment identified by mark-up elements such as: <Mark>, <PersonRef>, <DateRef>, etc. The Value attribute may be used to provide a normalised interpretation of the fragment, if necessary.
- Input (default) —Link to another profile used as input. For instance, to which something is being added or changed, or from which some inference is derived. All the associated threads have continuance, and could be used to build multi-tier “Personae”. The Value may be used to add an additional thread, or modify a prior one, if a link-phrase is provided.
- Reference — Identical to Input but with no continuance of threads from the input profile. An output DetLnk is effectively a ‘Reference’ to the respective profile.
- Reading — A reading of information from the previous Source-type link in the profile. Any Value and/or DetLnk is effectively an output from the profile, if a link-phase is provided. Together, for instance, they may describe a relationship.
- Inference — Comment, observation, conclusion, etc., on the inputs to the current profile. Any Value and/or DetLnk is effectively an output from the profile, if a link-phase is provided.

One of the functions of the link-type is to support applications that want to display different types of connector in a graphic organiser, or to allow a user to filter them based on the contents of the link-phrase. Others could be defined to support the evidence categories in Evidence Analysis Process Map, or distinguish link according to whether they relate to dates/events, persons, places, etc. See Extended Vocabularies for creating custom link-types.

The <Frame> element enumerates a list of materials relevant to the current source (such as citations, images, and transcriptions), and specifies the *where* and *when* relevant to the general source. The <Where> and <When> elements will usually refer to some corresponding profile in the body of the corresponding Source or SourceLet, but <When> may also take an explicit date value if it is known for the source but not visible in the transcription (e.g. for a census page).

The <Quality>, <Credibility>, and <Reliability> elements characterise the confidence in a source, and of information derived from it Note that these do not relate to a specific datum from the source. The Surety data-attribute is provided for that case. See Extended Vocabularies for defining custom values.

- Quality — source quality:
  - Original — Material in its original recorded form.
  - Copy — Facsimile of original, e.g. image copy, certified copy.
  - Derivative — Manipulated version of original, e.g. translation, abstract, extract.
  - Authored — Narrative work using other sources but providing independent conclusions.

        - Unknown — Unknown or unspecified assessment.
    - Reliability — information reliability:
        - Primary — Details provided by someone with first-hand knowledge.
        - Secondary — Details provided by someone with second-hand or more-distant knowledge.
        - Unknown — Unknown or unspecified assessment.
    - Credibility — credibility of information author, compiler, or reporter:
        - Expert — Information from someone with relevant expertise.
        - Questionable — Questionable credibility of information, as in interviews and oral genealogies, or with potential for bias as in an autobiography.
        - Trusted — Information from a trusted source.
        - Unsubstantiated — Claims or opinions.
        - Unknown — Unknown or unspecified assessment.

When a source has disjointed parts — such as a multi-page census household, or a book's pages — or it contains anterior (*from a previous time*) references — such as a diary, chronological narrative, or recollections during a story — then smaller sets of linked profiles can be specified in corresponding <SourceLet> elements. These will have their own <Frame> elements that are more precise in relation to the associated material, and which may specify a different *where* and *when*. For instance, their citation may specify an actual page or entry, and any Resource entity may include a specific transcription of it. In order to facilitate their use for more-localised citations, any Parameters used in associated <CitationLnk> or <ResourceLnk> elements are deemed to be inclusive of those specified in the main <Frame> for the same entity reference. In other words, it may only be necessary to include a page Parameter and leave the remainder implicit. Also, the <Where> and <When> elements inherit from the main <Frame> if unspecified.

At the lowest level, a ProtoPerson profile, for instance, may relate to a specific person reference in a transcription, and allows the details of that reference to be collected together, including the subject's relationships. For example, a simple statement of "John was the neighbour of Samuel" might be represented as follows:

```
<ProtoPerson DetKey='dpJohn'>
        <Link DetLnk='dsJohn' Value='John' Type='Source'>
                <Text>name</Text>
        </Link>
        <Link DetLnk='dsJohnRel'  Value='neighbour' Type='Source'>
                <Text>relation 1</Text>
        </Link>
        <Link DetLnk='dpSamuel' Type='Inference'>
                <Text>relation 1</Text>
        </Link>
</ProtoPerson>

<ProtoPerson DetKey='dpSamuel'>
```

```
        <Link DetLnk='dsSamuel'  Value='Samuel' Type='Source'>
            <Text>name</Text>
        </Link>
</ProtoPerson>
```

What this represents are two prototype persons: dpJohn and dpSamuel. The first has a <Link> element identifying the name of “”John” in the transcribed source, and another pair of links identifying the relationship to dpSamuel, again beginning with a relevant source fragment as their input.

Note that the link-phrase may be chosen freely by the researcher, and may be expressed in multiple languages if required. Also, the link-phrase is not constrained by any taxonomy or controlled vocabulary.

The associated transcription would have labels to which these links would be connected. For instance:

```
<PersonRef DetKey='dsJohn'>John</PersonRef> was the <Mark
DetKey='dsJohnRel'>neighbour of</Mark> <PersonRef
DetKey='dsSamual'>Samuel</PersonRef>
```

A link-type of Input may be used to build up a multi-tier persona-like profile. For instance:

```
<ProtoPerson DetKey='dpJohnSmith'>
        <Link DetLnk='dpJohn' Type='Input'/>
        <Link DetLnk='dpMrSmith' Type='Input'/>
        <Link Value='John Smith' Type='Inference'>
            <Text>name</Text>
        </Link>
</ProtoPerson>
```

NB: This merged prototype effectively embraces all the threads from the lower-level prototypes (merged in order), and an inferred name was established to override the merged threads that have link-phrase “name”.

The same principles apply identically to ProtoAnimal, ProtoPlace, and ProtoGroup profiles. If they are defined with a DetKey of their own then they can be referenced in higher profiles in the detail-link network (see below). At the very top, a Key attribute may be specified that links to a conclusion entity of a corresponding type, and either Key or DetKey, or both, must be specified. For instance:

```
<ProtoPerson Key='pJohnSmith'>
```

That would indicate an end to that prototype as a direct connection to a conclusion entity had been established.

The profiles ProtoEvent and/or ProtoDate might be connected using link values (again, free-form) to indicate their relative ordering. Elements of logic

and inference can be specified on their own via the Commentary profile if necessary.

A more complex example may be found at: Census Roles.

Detail Linkage

Support for the Source and Matrix entities uses an independent set of keys to chain conclusions to reasoning, to evidence, and to the original information. The entity keys used elsewhere are strongly typed; when something expects a Person key then only a Person key will do. The detail-links, on the other hand, are un-typed, but they are scoped — meaning they have referential containment. For clarity, attributes that define these keys are called DetKey, and the ones that reference them are called DetLnk.

The DATA_ATTRIBUTE syntax includes a DetLnk attribute and this means that it can be added to many conclusion items in a subject entity, including Property values. Because Property values are defined within a SourceLnk, their DetLnk instances can only refer to DetKey keys defined in the respective Source entities, or in an associated SourceLet. Other DetLnk instances can refer to DetKey keys in any Matrix or Source entity.

DetLnk instances in a Matrix entity can only refer to DetKey keys defined in the Source entities specified in its <Frame> element. DetLnk instances in a Source or SourceLet can only refer to DetKey keys in the same entity (or a lower SourceLet) or in a transcription associated with a Resource specified in the respective <Frame> element.

6.10 Matrix

The Matrix[5] entity supports cross-source correlation and analysis. Its <Frame> element enumerates a set of Source entities that are to be analysed together.

```
MATRIX=

<Matrix Key='key'>
        [ <Title> matrix-title </Title> ]
        MAT_FRAME
        [ MAT_PROTO_SUBJ | MAT_COMMENTARY | MAT_DATE ] ...
        [ TEXT_SEG ] ...
</Matrix>

MAT_FRAME=

<Frame>
        SOURCE_LNK ...
```

[5] "An environment or material in which something develops; a surrounding medium or structure", from *Oxford Dictionaries Online* (http://www.oxforddictionaries.com/definition/american_english/matrix : accessed 28 Oct 2015), s.v. "matrix", alternative 1.

```
        [ TEXT_SEG ] ...
</Frame>

MAT_PROTO_SUBJ=

<ProtoPerson { DetKey='key' || Key='key' }>
        [ <Title> proto-title </Title> ]
        MAT_LINK ...
        [ TEXT_SEG ] ...
</ProtoPerson>
<!-- or ProtoAnimal, ProtoPlace, ProtoGroup, ProtoEvent -->

MAT_COMMENTARY=

<Commentary DetKey='key'>
        [ <Title> commentary-title </Title> ]
        MAT_LINK ...
        [ TEXT_SEG ] ...
</Commentary>

MAT_DATE=

<ProtoDate DetKey='key'>
        [ <Title> date-title </Title> ]
        MAT_LINK ...
        [ TEXT_SEG ] ...
</ProtoDate>

MAT_LINK=

<Link [Type='link-type'] { DetLnk='key' || Value='value' }>
        [ TEXT_SEG ] ...
</Link>
```

The Matrix entity is typically used to compare/contrast the profiled sources from a number of Source entities. This would usually be to solve a specific problem, or to study a given person or family, and contrasts with the Source entities themselves which are designed to assimilate one source each for any future study.

6.11 Resource

A STEMMA Resource represents any digital or physical artefact in your collection. The term is more often used in the context of electronic documents, pictures, scans, and recordings. However, in STEMMA it may also be a physical object, or artefact (i.e. an object of human creation), such as actual letters, original documents, medals, portraits, original photographs, personal possessions, and family heirlooms.

RESOURCE=

```
<Resource Key='key' [Abstract='boolean']>
        [ <Title> resource-title </Title> ]
        [ <URL [ContentType='content-type']> url </URL> ]
        [ <Type [Artefact='boolean']> resource-type </Type> ]
        [ <DataControl>
                [ <Sensitivity> level </Sensitivity> ]
                [ <Copyright> copyright-notice </Copyright> ]
                [ <Permission> permission-notice </Permission> ] ...
                [ <Prohibition> prohibition-notice </Prohibition> ] ...
                [ <Attribution> prohibition-notice </Attribution> ] ...

                [ TEXT_SEG ] ...
        </DataControl> ]
        [ <Params>
                { PARAM_DEF ... } | { PARAM_VALUE ... }
        </Params> ]
        [ <BaseResourceLnk Key='key'>
                [ TEXT_SEG ] ...
        </BaseResourceLnk> ]
        [ TEXT_SEG ] ...
</Resource>
```

Digital resources are located by a URL (Uniform Resource Locator). The 'scheme' prefix of the URL allows it to be applied to different file stores, e.g. file:// for local files[6] and http:// for the Internet/Intranet ones accessed via the HTTP protocol. If the digital data-type cannot be determined by the file type (e.g. by its file extension) then the ContentType attribute must be employed to specify a corresponding Internet Media Type, e.g. 'text/plain', 'image/jpeg'. A URL is not required for wholly physical artefacts, and is optional for Abstract Resource entities.

Irrespective of whether the Resource is digital, non-digital, or both, the nature of the resource is described by the <Type> element. This includes: Award, Clothing, Document, Furniture, Jewellery, Letter, Map, Music, Painting, Photograph, Recording, and Video. See Extended Vocabularies for an example of defining custom resource-types.

Non-digital resources are indicated by the presence of the Artefact attribute. A value of '1' (true) indicates you have a non-digital resource, such as an original letter, portrait, or item of clothing. The default is '0' (false). If a URL is also specified then it indicates that a digitised copy is also held. This will usually be a digital image but it could also be a digital sound recording or

[6] RFC 1738 requires an explicit host name in a URI employing the file scheme, e.g. file:///host/g:/folder/file.jpg, even if the value is left blank. RFC 3986 allows this to be optional, e.g. file:/g:/folder/file.jpg. This means a relative file specification does not require a leading '/' separator at all, e.g. file:folder/file.jpg.

video. Original non-digital recordings and video should be considered artefacts. When copying a collection for transmission to someone else, or for long-term storage, consideration should be given as to whether the Artefact attribute needs to be dropped if the physical items are not included.

This simple example provides a definition of a photographic resource and the URL through which it may be accessed, plus a sample reference to it:

```
<Resource Key='MyPhoto'>
        <Title>Photograph of myself</Title>
        <URL> file:mydir/MyPhoto.jpg </URL>
</Resource>

<ResourceRef Key='MyPhoto'/>
```

The <DataControl> element provides any notices that must be displayed to the end-user before the associated resource is copied or transmitted to another user. Software components are not expected to act on those notices themselves. They should merely be displayed in order to prevent an accidental breach of trust or copyright. Permission and Prohibition are designed for informal control, such as when a family member asks that their photographs not be passed outside of the immediate family. Copyright is a formalised type of prohibition, usually applied to works of artistic, academic, or commercial value. These concepts are discussed under Worldwide Family History Data. These settings should be honoured when bundling a collection, or part of, for transmission to another researcher. The possible Sensitivity levels are exactly as defined for the Sensitivity data attribute at DATA_ATTRIBUTE.

A Resource definition may be parameterised so that multiple resource references can reuse common information. This scheme allows the URL, resource-title, narrative elements, and Parameter values to utilise ${param-name} markers. For instance, the following definition and reference selects a specific family photograph from a common collection:

```
<Resource Key='rPhotos' Abstract='1'>
        <Title>Family photograph: ${PhotoName}</Title>
        <URL> file:myphotos/family/${PhotoName}.jpg </URL>
        <Params>
                <Param Name='PhotoName'/>
        </Params>
</Resource>

<ResourceRef Key='rPhotos'>
        <Param Name='PhotoName'> Me </Param>
</ResourceRef>
```

This effectively creates an unnamed transient Resource with the modified parameter context. It would be functionally identical to the creation of a named permanent Resource.

The next example goes to a hypothetical web site to retrieve census images for England and Wales.

```
<Resource Key='rCensusImage' Abstract='1'>
        <Title>1851-1901 Census Images of England and Wales</Title>
        <URL>http://www.census.com/image?series=${Series}&piece=${Pie
        ce}&folio=${Folio}&page=${Page}</URL>
        <Params>
                <Param Name='Series'/>
                <Param Name='Piece' Type='Integer'/>
                <Param Name='Folio' Type='Integer'/>
                <Param Name='Page' Type='Integer'/>
        </Params>
</Resource>
```

The valid Parameter data-types are documented at: Data Types. The same ItemList approach to lists is taken as for Property values. The semantic type is indicated by the SemType attribute which may use the Dublin Core vocabulary, e.g. SemType='DC:title' or SemType='DC:publisher'.

Any <Param> element may specify a default value if necessary. The default value for the Optional attribute is '0' (i.e. False) which means a non-blank value must be provided. When an ItemList Parameter is substituted then the result is a comma-separated list of the component Items.

The <BaseResourceLnk> element may nominate an Abstract Resource from which data may be inherited by the current Resource, in much the same vein as base classes and derived classes in software programming. An Abstract Resource must define no embedded Keys, can only reference other abstract entities, and must contain Parameter definitions rather than Parameter settings. The previous example using a <ResourceLnk> element with Parameters could have been replaced with two distinct Resource entities: one being the generic representation of a photograph from a particular folder, and the other being a specific photograph from that folder. For instance:

```
<Resource Key='rMyPhoto'>
        <BaseResourceLnk Key='rPhotos'/>
        <Params>
                <Param Name='PhotoName'> Me </Param>
        </Params>
</Resource>
```

Application of any Parameter substitution must therefore occur after the inheritance process has completed. If an implementation of this mechanism creates a temporary conglomerate entity in memory by doing a physical

merge then it must not be persisted back to the data file, otherwise it constitutes a data corruption.

Electronic resources attached to a data collection present a specific issue during data exchange, i.e. import/export. Subject to privacy controls, the relevant resources should be bundled with an exported STEMMA Document and transmitted along with it to whoever the recipients are. They cannot be included in the body of the Document in any practical way. Although this area still needs work, there are several existing document container file mechanisms available. For instance:

- Open Office XML. This is a zipped, XML-based file format developed by Microsoft. Initially standardised as ECMA-376 and later as ISO/IEC 29500.

- MHTML, or MIME HTML. This is a Web page archive format used to combine resources that are typically represented by external links (e.g. images) together with HTML code into a single file. It is used extensively for rich-text email messages. MHTML is a proposed standard, circulated in a revised edition in 1999 as RFC 2557.

- Java archives, i.e. jar files.

- ISO/IEC NP 21320-1. This standard was still under development at the time of writing. It is expected to be a standardised version of the *zip* compressed file encoding.

Some basic functional requirements include: compression, optional encryption, preservation of relative directory structure, custom name-value properties per item, ability to keep STEMMA <Resource> references valid after unpacking, and ability to address each item separately from outside the container. Some interesting discussion on this topic may be found on the BetterGEDCOM wiki at: BG Container Formats and Packaging Data.

6.12 Citation

A Citation entity identifies the source of some information mentioned in the Dataset. Examples might include books, newspaper articles, BMD certificates, census data, tax records, court records, film, tombstones, military service records, journey manifests, cemetery records, oral history, church records, pension records, land or property transfers, etc.

These are common historical sources, and there are accepted printed citation formats applicable to each of them, but this Citation entity goes further; it can also identify a collection of works, a repository or institution, or even represent attribution to an individual.

In the citations of normal written or printed works, there are two main citation modes that may be employed within text: document labels and source labels, being applicable to documents and images respectively. Citations may involve

reference notes linked to inline superscript indicators in the main text. Alternatively, they may involve a source list or bibliography at the end of the work. Parenthetical in-text citations — such as “Smith (2004, p. 39) claims that...”, or “...(Smith 2004, p.39)...” if all details are parenthesised — are commonly associated with published sources in academic work and are less appropriate for genealogical or historical citations. This is because they do not accommodate the source provenance or analytical notes that are frequently required.

There are citation conventions that apply to different source types and scenarios in order to present some consistency, and these have specifications for their layout, quotation marks, punctuation, and use of italics. Several citation styles are in common use. For instance, in the humanities there are: Modern Language Association (MLA), Harvard referencing, Modern Humanities Research Association (MHRA), and the Chicago Manual of Style (CMOS). There are other styles commonly used in law or the sciences too.

The Board for Certification of Genealogists (BCG) recommends the use of both CMOS and EE for family history. EE is a style devised by Elizabeth Shown Mills in *Evidence Explained: Citing History Sources from Artifacts to Cyberspace* (Baltimore: Genealogical Publishing Co., 2009) to cover the wider range of historical and unpublished sources used in family history.

It should be understood, though, that all these citation styles and modes relate to the final-form written or printed citations. Their application is therefore relevant to a specific end-user rather than to computer storage. Since those final-form citations are designed to be humanly-readable, they also embody elements of a specific locale, culture, and preferred style. This is a problem for electronic documents as they are not computer-readable, and so cannot be adjusted to suit the locale or preferences of an arbitrary end-user. It is therefore necessary to go back to the essence of a citation rather than consider specific physical implementations — i.e. to provide sufficient information through a digested citation to uniquely identify a source, its characteristics, and any analytical assessment. These *citation-elements* — implemented through STEMMA’s Parameter mechanism — should be sufficient to support the formatting appropriate for any given end-user.

Although it is possible to generate citations of different style, and for different locales, from the discrete citation-element values, there are many complications in the real world. A citation sentence may contain different layers describing the provenance of the source and its information, or it may contain analytical notes. A reference note may contain multiple citation sentences — a tour of these scenarios was covered in Cite Seeing. Subsequent references to the same source would typically use a shortened form of the associated reference note (see ‘WhereIn’ attribute), or the author may have employed an explicit *hereinafter-cited-as* term, or the Latin abbreviation *Ibid*. A footnote may have woven two source references into the same piece of text. Certain parts of a citation may not have been available (e.g. an undated document), or were erroneous, and so the citation would need to override any simple template-like formatting. In effect, authors of

narrative work are loath to delegate generation of their citations to a piece of software working blindly from a set of data values. It is therefore necessary to support hand-crafted forms, and change the focus of citation-elements to that of correlation and interrogation rather than formatting.

The scheme presented here is a generalised computer-readable one that would cope with all possible source types and scenarios. It does not strive to enumerate all possible source types, or specify what elements they require, or mandate a particular presentation style; the main goals of this scheme are to keep it open-ended so that source types can be defined freely, to parameterise the scheme so that it can interface to external citation-templates, and to give it a hierarchical structure for representing different layers of a citation (e.g. for provenance or location).

CITATION=

```
<Citation Key='key' [Abstract='boolean']>
        [ <Title> citation-title </Title> ]
        [ <URI> source-type-uri </URI> ]
        [ <Params>
                { PARAM_DEF... } | { PARAM_VALUE ... }
        </Params> ]
        [ <DisplayFormat [Mode='citation-format-mode']>
                TEXT_SEG ...
        </DisplayFormat> ] ...
        [ PARENT_CITATION_LNK ]
        [ <BaseCitationLnk Key='key'>
                [ TEXT_SEG ] ...
        </BaseCitationLnk> ]
        [ TEXT_SEG ] ...
</Citation>
```

PARENT_CITATION_LNK=

```
<ParentCitationLnk Key='key' [Type='layer-type']>
        [ PARAM_VALUE ] ...
        [ PARENT_CITATION_LNK ]
        [ TEXT_SEG ] ...
</ParentCitationLnk>
```

PARAM_DEF=

```
<Param Name='name' [Type='type']  [SemType='sem-type']
        [ItemList='boolean'] [Optional='boolean'] [WhereIn='boolean']>
        default-value
</Param>
```

PARAM_VALUE=

{ <Param Name='name' [Key='key'] [Subst='substitution']>
value
</Param> }
|
{ <Param Name='name' [Subst='substitution']>
{ <Item [Key='key']> value </Item> } ...
</Param> }

The parameterisation is available in the citation-title, the format-strings, narrative elements, and the values of Parameters themselves (i.e. within a Params element).

Note that Parameter names are local to the corresponding source-type. There is no sharing of Parameter names between different source-types, and no implied semantics in any of their names. If two source-types each have a Parameter called 'Publisher' then they are each interpreted in the context of their respective source-types. In effect, no semantics are conveyed directly by the Parameter name — that is the purpose of the optional SemType attribute.

The valid Parameter data-types are documented at: Data Types. The same ItemList approach to lists is taken as for Property values. The semantic type is indicated by the SemType attribute which may use the Dublin Core vocabulary, e.g. SemType='DC:title' or SemType='DC:publisher'. The default value for the Optional attribute is 0 (i.e. false) which means that a non-blank value must be provided. The 'WhereIn' attribute flags parameters that identify a location or entry within a source, as opposed to the source itself, its provenance, or its location. The 'Subst' attribute allows the formatting of a value to be overridden, and is especially useful for unknown values. For instance, an undated document might be represented with a date Parameter having a value of "?" but a substitution of "n.d." or "[1832]".

The <BaseCitationLnk> element may nominate an Abstract Citation from which data may be inherited by the current Citation, in much the same vein as base classes and derived classes in software programming. An Abstract Citation must define no embedded Keys, can only reference other abstract entities, and must contain Parameter definitions rather than Parameter settings. Any application of Parameter substitution must therefore occur after the inheritance process has completed. If an implementation creates a temporary conglomerate entity in memory by doing a physical merge then it must not be persisted back to the data file, otherwise it constitutes a data corruption. See Inheritance and Parameters for more information.

It is important retain a clear view of the distinction between a Citation and a Resource. As an example, consider UK BMD references. These might be linked to the defining body, say with something like http://www.gro.gov.uk/, in order to create a unique source citation. However, if you wanted to be able to

pull up the appropriate census page from some Web site then that would be done via a corresponding Resource entity.

Some related articles may be found at: Cite Seeing and Citations for Online Trees.

Semantic Typing

The simple Dublin Core (see Dublin Core Metadata Initiative) terms cannot clearly distinguish, say, the title of an article from the title of a journal containing that article, or provide a clear indication of other data related to the containing journal such as publication date as distinct from the article submission date, or the volume and issue numbers. That same page recommends the use of the OpenURL (ANSI/NISO standard, Z39.88-2004) ContextObject for representing the context of a bibliographic citation, although it does not take this to the level of a hierarchical chain. The OpenURL concept is designed to provide the context of a citation in a machine-readable form that can be resolved by an unspecified library or archive. In other words, the Dublin core recommendation doesn't cite a source directly but as a library-independent hyperlink to content. At best, it constitutes a reference to an *indefinite source*.

The SemType attribute associates such semantic information with the individual citation-elements (i.e. Parameters) but leaves the Parameter names to be chosen independently to suit the source-type. Other semantic types could be applied using the same attribute, but with a different namespace.

The STEMMA scheme described here is fully in keeping with those Dublin Core recommendations but is not specifically tied to it. It allows each type of source to be represented by a source-type-uri. Parameters can be applied to build up a citation description for a specific instance of that source-type. The source-type-uri also acts as a global key for retrieving localised text for soliciting Parameter values, data-types for validating the Parameter values, and for interfacing to a citation-template system in order to generate a formatted string for the user. If omitted then an effective one must be available through inheritance.

Citation Chain

Citations may be linked to describe the provenance of a source, the provenance of the information itself, where the originals are held, and any analytical comments. These are known as citation layers and the associated chain forms part of a hierarchy created through the use of the <ParentCitationLnk> element.

Note that STEMMA Citation chains do not differentiate between citing a specific source of information, citing a collection or work that the information was contained within, or citing a repository or institution hosting that work or collection — they are all citing something in the more literal sense. They do

not mandate the juxtaposition of *definite* and *indefinite sources*,[7] or the ordering of original and derivative references (see below). Supporting citation layers avoids duplication and provides a stronger representation overall.

The Dublin Core Metadata Initiative has encountered the issue of a chain but has tried to solve it by adding additional terms and namespaces (see dc-citation-guidelines/).

The links between the layers may be characterised using the Type='layer-type' as follows. Note that this doesn't describe the layers themselves — which should be obvious from their content — but rather the relationship between the layers.

Layer-type	Comments
AbstractOf	A brief summary or a précis of --
Citing	Information cited by the source. Source-of-the-Source.
Comment	Analytical comments.
ConsultedAs	Consulted through derivative, usually online or in database
ExtractOf	Extracted portion from --
ImagedAt	Consulted through general image copy
MediaCopy	Media conversion from --
Provenance	Other provenance information, differing from 'Citing'.
Repository (default)	Location of original source.
ReworkOf	Revised, abridged, or otherwise modified from --
TranscriptionOf	Transcribed details from --
TranslationOf	Translated details from --

These should cope with instances of image derivatives where the emphasis is placed on the image or the original document. This choice is covered in detail by Elizabeth Shown Mills at "QuickLesson 19: Layered Citations Work Like Layered Clothing", *Evidence Explained: Historical Analysis, Citation & Source Usage* (https://www.evidenceexplained.com/content/quicklesson-19-layered-citations-work-layered-clothing : posted 4 Sep 2014, updated 5 Mar 2016, accessed 4 Apr 2017), under "Online Records at State-Agency Sites".

Display Format

STEMMA allowed preferred hand-crafted citations to be specified in the <CitationRef> element (see CITATION_REF). This is particularly useful when there are shortened forms (e.g. employing 'hereinafter cited as ...') which cannot be generated directly from the Parameters representing the citation-elements. Individual Parameters may also override their default formatting, say for substituted text in the absence of a value, or for abbreviated list formatting. Taking the onus off formatting allows the Parameter settings to be used moer for correlation and interrogation.

[7] Note that academic citations, such as those in journals, often refer to an indefinite source. This allows them to be much briefer but it only works because such sources are published and easily accessible; it makes no difference where the article or paper was obtained from.

Citation entities will require formatting to a given style and locale before they can be displayed. A later version may allow styles to be automatically selected from Citation Style Language (CSL) templates — CSL is an open XML-based language for defining the parameters and formatting for different citation types. Such styles can be browsed and searched via the Zotero Style Repository, although it currently has no concept of a URI string which is unfortunate because it would be a convenient handle to distinguish the templates and applicable source-types in the repository. A problem with such citation-template schemes is that they try to format plain textual elements into a simple template, whereas STEMMA assumes that *objects* (in the OOP sense) representing, say, a Person, Place, or Contact can be provided. The advantage of this scheme is that the template system can call-back on well-defined methods to obtain a particular style of name, or specific contact details; otherwise the genealogical software product is assumed to have intimate knowledge of the specific template.

In the absence of any external formatting support for citations, or any explicit hand-crafted citations, the <DisplayFormat> element can also be used as a simple STEMMA-defined citation-template. It allows a number of language-specific text strings to be defined for different formatting modes (e.g. full reference note — the default), and these can make use of mark-up and parameterisation to employ them in multiple scenarios. Although some brief examples are presented below, a fuller example may be found at: Citation Template. NB: this template feature is purely declarative and currently contains no decisional control over the generation of the citation text.

Examples

Here's a simple example of a traditional book citation:

```
<Citation Key='cOldNottm'>
        <Title>Old Nottingham Notes</Title>
        <URI> http://stemma .parallaxview.co/source-type/book/ </URI>
        <Params>
                <Param Name='Author'>James Granger</Param>
                <Param Name='Title'>OLD NOTTINGHAM : Its Streets, People,
                etc</Param>
                <Param Name='Publisher'>Nottingham Daily Express
                Office</Param>
                <Param Name='Date'>1904</Param>
                <Param Name='Pages'/>
        </Params>
        <Text>
                Reprinted from the Nottingham Daily Express, October 3rd,
                1903 – July 9th, 1904.
        </Text>
</Citation>
```

A corresponding citation invocation, for a specific page, might appear as:

```
<CitationRef Key='cOldNottm'>
        <Param Name='Pages'>46-48</Param>
</CitationRef>
```

Whether this generates a long or short reference note depends on whether the same source is referenced earlier in the current <Narrative> element.

Citations can become very complex since the author will not only want to cite the source, and the information obtained from that source, but the context of how it substantiates or contradicts their assertions and conclusions. This often involves some type of analytical commentary in the citation. For instance:

> Death notices, *Ulster Gazette* and *Daily National Intelligencer*, both dated 24 January 1815. Corra Bacon-Foster, "The Story of Kalorama," *Records of the Columbia Historical Society* (1910), 108, states Louisa left four children; three have been identified. In 1810, Charles "Cating" and a female, both over 44, were enumerated with one male and female aged 26-44; one male and female aged 16-25; and one male under 10 - suggesting that George, Louisa, and their first son may have been living in the Catton household. See 1810 U.S. census, Ulster County, New York, New Paltz, p. 116, line 6; NA micropublication M252, roll 37.

Each reference note may contain multiple “citation sentences” (separated by periods), and each of these may contain multiple layers (separated by semicolons). See Cite Seeing for a deeper discussion.

6.12.1 Inheritance and Parameters

Inheritance is a mechanism that allows an entity to share the definition of an abstract base entity. An abstract entity must define no embedded Keys (e.g. in embedded <Text> elements) and can only reference other abstract entities. There may be other restrictions imposed depending on the entity type. In STEMMA, inheritance occurs for Events, Citations and Resources. A lighter introduction may be found at: Genealogical Inheritance.

Inheritance should not be confused with *parent entities*. Some entities have parents that help construct bottom-up trees (e.g. Person/Animal lineage, Place/Group hierarchies, Event hierarchies, and Citation chains).

When an entity initially inherits the definition of a base entity, its explicit parts override any corresponding inherited parts. NB: Nothing is actually copied from base entity to the derived entity, meaning the result of the inheritance must not be persisted in any subsequent export operation; this overriding occurs in memory only.

Parameterisation is a mechanism where the Parameter values are applied to an entity in order to modify its context. This applies to both Citations and Resources. The Parameters may be inherited from a base entity, declared explicitly in the body of an entity, or applied to a link from one entity instance to another (see below). All of these schemes can be used together.

The set of valid Parameter names and types must be defined in an Abstract Citation or Resource, as appropriate. In the case of a Citation entity, the URI implies one higher level of inheritance from an external repository. If the URI is defined in such a repository then the initial Parameters names and types are retrieved from there. If both definitions are available then they must be identical. One Abstract entity inheriting from another can add default values to existing Parameters, or define new Parameters. For non-Abstract, or concrete, entities, only Parameter settings, rather than definitions, are allowed, and the names and types are validated against their original definitions. The Parameter order is unimportant.

During an entity link, the associated <ResourceLnk>/<ResourceRef> or <CitationLnk>/<CitationRef> elements may provide a partial set of explicit Parameter values. These merely override any equivalent Parameters in the target entity in order to create an unnamed transient entity. NB: it is an error if the Parameters in those links do not correspond to the declared Parameters in the target entity. In the Citation case, the transient entity may be given an explicit chain of parent entities, and this will override any specified in the Citation entities themselves.

An important point regarding the application of parameter substitution is that it always occurs after the inheritance process has completed. Hence, the following distinct stages may occur:

1. Inheritance of fields from the base entity.
2. Overriding (in memory) with explicit fields from the derived entity.
3. Creation of a transient unnamed entity from the parameter settings in a *Lnk/*Ref element.
4. Substitution of current parameter values, in the source-order of their substitution markers.

Consider the case of a short-hand source citation for a general census page of England & Wales for a particular year, e.g. [*RG 13/3178/51/12*]. While this particular form is not recommended, a purely catalogue-reference example makes an illustration easier to read.

```
<Citation Key='cCensus1901' Abstract='1'>
        <Title> 1901 Census of England and Wales </Title>
        <DisplayFormat Mode='RefNote'>
                <Text Language='eng'>
                [<Subs><i>${Series}/${Piece}/${Folio}/${Page}</i></Subs>]
                </Text>
        </DisplayFormat>
        <URI> http://www.nationalarchives.gov.uk/census </URI>
        <Params>
                <Param Name='Series'>RG 13</Param>
                <Param Name='Piece' Type='Integer'/>
                <Param Name='Folio' Type='Integer'/>
                <Param Name='Page' Type='Integer'/>
```

```
        </Params>
</Citation>
```

This abstract citation provides a default citation-title, a simple citation-template for reference-note citations, the applicable source-type URI (contrived for this example), and a list of required Parameters — one of which has a default value.

We could then create a derived Citation for a specific household as follows:

```
<Citation Key='cCensus1901ManningGrove'>
        <Title> 1901 Census for Manning Grove </Title>
        <BaseCitationLnk Key='cCensus1901'/>
        <Params>
                <Param Name='Piece'>3178</Param>
                <Param Name='Folio'>51</Param>
                <Param Name='Page'>12</Param>
        </Params>
</Citation>
```

The BaseCitationLnk causes all the data from cCensus1901 to be inherited. The title is then overridden with a more specific one. The values of the last three Parameters are also given explicit values. The first Parameter retains the default value declared in the base Citation.

Now consider an associated Source entity for the 1901 census of this particular English household. Its <Frame> element could refer directly to the derived Citation, created above, or create an unnamed transient equivalent directly from the base Citation.

```
<Source Key='eCensus1901ManningGrove'>
        <Frame>
                <!-- Alternative 1 -->
                <CitationLnk Key='cCensus1901'>
                <Param Name='Piece'>3178</Param>
                <Param Name='Folio'>51</Param>
                <Param Name='Page'>12</Param>
                </CitationLnk>

                <!-- Alternative 2 -->
                <CitationLnk Key='cCensus1901ManningGrove'/>
        </Frame>
</Source>
```

In alternative-1, we use the generic Citation for the 1901 census and simply pass explicit Parameters (ignoring the first which has a default value). In alternative-2, we reference a specific Citation for the corresponding household.

There are some subtle differences here: Alternative-1 only allows the creation of a modified transient entity through parameterisation, either from an Abstract or “concrete” (non-Abstract) entity. You cannot add an explicit title, or add narrative. Alternative-2 is the recommended approach in this situation because it provides a named Citation to which other data can be added. A typical example where alternative-1 might be used is for referencing discrete pages of a single book.

Note that during parameter substitution the search for a matching name will use the parent hierarchy if necessary. In the case of a Citation, this means the search will follow the Citation chain in order to locate an appropriate Parameter name.

6.12.2 Source-Type URI

The URI string is a global identifier for the source-type — one that can be used as a key to retrieve its meta-data or the formatting template associated with the source-type.

When the URI string is used as a key, or handle, it may be employed to access an external definition of the source-type, or a citation-template to format the parameters of the citation. For instance, the following identify possible libraries or interfaces that might be interrogated using that key, and the information that might be contained there:

- **Definition (URI):** [possibly a local symbolic name for the source-type, element ids, element data-types]. This contains the meta-data associated with the source-type.

- **Data-entry (URI, locale):** [source-type description, element descriptions, element hints]. This contains locale-specific data to support selection of a source-type from a list (e.g. using the source-type descriptions) or data-entry for a specific citation.

- **Presentation (URI, element-data, locale, citation-style, citation-mode):** [readable reference note]. This would return some formatted version of a citation for a given locale and citation style. That style could conceivably be a private one, as opposed to say CMOS, if someone wanted guaranteed perfect transportability of the final form whilst retaining the ability to analyse and correlate the machine-readable citations. The citation-mode determines whether a first/subsequent reference note or a bibliography entry is required.

Some URIs could be the responsibility of the institution or organisation that catalogued the corresponding records or allocated the associated reference codes. However, that would require global adoption of this scheme which is more than a little naïve. STEMMA will therefore define its own URIs based on the http://stemma.parallaxview.co domain until such a time as alternatives are published.

6.12.3 Source Citation Categories

A number of STEMMA source-types are used in the online examples. Although not proposed as standards, I document them here for completeness:

http://stemma.parallaxview.co/source-type/blog
An online blog. Parameters: [Author], Title, [Host], Blog, Date, URL, Accessed.

http://stemma.parallaxview.co/source-type/book
A printed book. Parameters: Author, Title, Publisher, PublisherAddr, Date, Page.

http://stemma.parallaxview.co/source-type/correspondence
Written correspondence. Parameters: From, To, Date.

http://stemma.parallaxview.co/source-type/folklore
Folklore or legend. Original source unknown. Parameters: Source, Date.

http://stemma.parallaxview.co/source-type/newspaper
Newspaper article. Parameters: [Author], Title, Paper, Date, Page, [Column].

http://stemma.parallaxview.co/source-type/testimony-1
First-hand testimony. Parameters: Source, Date.

http://stemma.parallaxview.co/source-type/testimony-2
Second-hand testimony. Parameters: Source-1, Date-1, Source-2, Date-2 (i.e. recollection as given to Source-1 by Source-2 on Date-2).

http://stemma.parallaxview.co/source-type/web-media
Online news service. Parameters: Author, Title, Publisher, PublisherAddr, Date, URL, Accessed.

6.12.4 Transcriptions

If you want to include transcriptions, including transcribed extracts, in your data then ideally they should appear in the Resource entity that describes the associated digital or physical artefact; they should not appear in any Citation entity that references the data source. Although the transcription could appear in a top-level Narrative entity, this is not recommended since it disassociates it from any digital image of the text and other descriptive information about it. When the Source entity is used for source analysis, it designed to work with transcriptions identified by <ResourceLnk> elements. In principle, the Source entity could work directly from an image — especially if it contained a transcription within its meta-data, possibly linked to the image using SVG — but there are currently no examples for this. The x/y attributes available on selected elements at Descriptive Mark-up may be used to keep transcriptions and document scans in step when displaying them to the end-user.

For a multi-page source then there may be separate Resource entities but these can be grouped using STEMMA's inheritance mechanism. See

Hierarchical Sources. The Source entity's <SourceLet> elements also accept an abbreviated Parameter list in their own <ResourceLnk> elements (sharing the remainder with the main <Frame> element) in order to make it convenient to cite individual parts of a source.

In summary, the transcription would appear in a <Text> element within the Resource entity, and it would use the mark-up described under Narrative Structure to capture original presentational attributes, anomalies such as marginalia and corrections, and the semantics of subject references, dates, etc. Some of the semantic mark-up takes a DetLnk='key' attribute that allows that field to be labelled for analysis and correlation within the Source and Matrix entities. The Resource entity can even indicate that you have the physical original as well as an image of it. See Handling Transcriptions, and also Structured Narrative for a worked example.

7 Validation

There are various types of validation that can be performed on one of these STEMMA source files.

Syntax

Whenever a software unit loads the data into an XML DOM (Document Object Model) then the syntax of the XML is automatically validated to be "well formed". It should not be possible for a source to be other than well-formed unless it has been edited by hand or corrupted.

Schema

The structure of the XML content can be validated against an XSD Schema defining the XML representation of the STEMMA data model. This is a manual (non-automatic) software operation but not a difficult one. It should be made clear that this differs from the automatic validation of the XML being well-formed since it is validating the particular XML dialect as defined here.

If there are formal extensions to the XML schema (see Extended Schema) then they must have associated XSD definitions that can additionally be used to validate them.

Semantic

This is the validation of the data content itself and the implications of the data stored in the source file. Here are some aspects for the data that could be validated:

- Symbolic names (e.g. Keys) have valid format
- All referenced Key names actually defined in a given Dataset.
- No duplicate Key names defined in a given Dataset.
- No imported Key names are already defined in a Dataset.
- Key type matches reference type.
- No circularity in biological lineage. Links should constitute a DAG.
- No circularity in hierarchies for Places, Groups, Events, or Citations.
- No circularity in derived Groups.

- No circularity in inheritance relationships.
- No circularity in DetLnk/DetKey references (see Source/Matrix entities).
- Event constraints are valid (see below).
- All STEMMA tag values (e.g. types, Property names, modes) are valid.
- All Property values match the defined data-type.
- Parameter names and types match their definitions.
- Links between top-level entities are unique (see Dataset Structure).

Validating the constraints in the Events to ensure that there are no contradictions or impossibilities is an exercise in graph theory. It has been solved before for ‘scheduling constraints’ in project management software.

8 Extensibility

Any source format for micro-history data must be extensible in order to accommodate unforeseen circumstances or cultural variations.

There are three main types of extension considered here: Extensions to the schema itself to define new element types, extended properties for Person, Animal, Place, Group, or Event entities, and extended vocabularies for types, etc. The latter two are deliberately made easy since they will be more common. Extending the core schema is controversial, and not really recommended, but the possibility is included here for completeness.

Note that there is no real need to introduce other types of Person-to-Person linkage to supplement the existing ones that represent biological lineage and association via Role and Relationship through shared Events. The fact that narrative text can reference any Person, Animal, Place, Group, or Event means that the data model already has a fundamental capability to link elements for arbitrary reasons. This, in turn, is necessary for it to be applicable to micro-history in general. Although the conclusional sub-model, in the form of Properties, describes relationships in terms of a normalised vocabulary, the informational sub-model, in the form of the Source entity, can describe any subject-to-subject relationship using the informal vocabulary of the user or the source.

8.1 Extended Vocabularies

STEMMA uses a number of partially controlled vocabularies for its types, sub-types, and other taxonomies — collectively referred to here as “tag values” as they are specified within element content or attribute values. With the exception of source-type, these all belong to default implicit namespaces. Those namespaces are implicitly rooted on the versioned default namespace specified in the standard xmlns attribute, e.g. http://stemma.parallaxview.co/2017-04. The following namespaces are currently defined by STEMMA. The ones highlighted in blue constitute fully controlled vocabularies that cannot be extended.

Namespace	Examples
http://stemma.parallaxview.co/animal-name-mode	Title
http://stemma.parallaxview.co/animal-property	Name, Age, Role

http://stemma.parallaxview.co/animal-relationship	Sibling, Parent
http://stemma.parallaxview.co/animal-role	Deceased, Offspring
http://stemma.parallaxview.co/animal-status	Deceased, Implied
http://stemma.parallaxview.co/animal-subtype	Pet, Mascot
http://stemma.parallaxview.co/animal-type	Cat, Dog
http://stemma.parallaxview.co/anomaly-mode	Footnote, Marginalia
http://stemma.parallaxview.co/citation-format-mode	RefNote, ShortRefNote
http://stemma.parallaxview.co/citation-layer-type/	ImageCopy, Repository
http://stemma.parallaxview.co/citation-mode	Footnote, Inline
http://stemma.parallaxview.co/data-type	Text, Date, Integer
http://stemma.parallaxview.co/date-mode	Short, Long
http://stemma.parallaxview.co/event-mode	Title
http://stemma.parallaxview.co/event-property	When, Where
http://stemma.parallaxview.co/event-subtype	Marriage
http://stemma.parallaxview.co/event-type	Union, Travel, Birth
http://stemma.parallaxview.co/group-name-mode	Title
http://stemma.parallaxview.co/group-subtype	Nuclear, Blended
http://stemma.parallaxview.co/group-type	Military, Family
http://stemma.parallaxview.co/group-op	Union, Exclude
http://stemma.parallaxview.co/info-credibility	Trusted, Questionable
http://stemma.parallaxview.co/info-reliability	Primary, Secondary
http://stemma.parallaxview.co/name-mode	Formal, SemiFormal
http://stemma.parallaxview.co/note-mode	Link, Inline, Footnote
http://stemma.parallaxview.co/person-name-mode	Title
http://stemma.parallaxview.co/person-name-type	Alias, Nickname
http://stemma.parallaxview.co/person-property	Role, Age, Occupation
http://stemma.parallaxview.co/person-relationship	Wife, Brother, Mother
http://stemma.parallaxview.co/person-role	Head, Bride
http://stemma.parallaxview.co/person-status	Married, Widow
http://stemma.parallaxview.co/place-category	School, Cemetery, Ship
http://stemma.parallaxview.co/place-name-mode	Title, Hierarchy
http://stemma.parallaxview.co/place-property	Parish
http://stemma.parallaxview.co/place-role	Destination
http://stemma.parallaxview.co/place-type	Street, District, Country
http://stemma.parallaxview.co/resource-mode	Title, Large
http://stemma.parallaxview.co/resource-type	Artefact, Letter
http://stemma.parallaxview.co/sensitivity	Public, Family
http://stemma.parallaxview.co/source-link-type	Source, Inference
http://stemma.parallaxview.co/source-quality	Original, Derivative
http://stemma.parallaxview.co/source-type	Testimony-1
http://stemma.parallaxview.co/text-class	Caption, H1, Tablenote

Custom tag values may be defined by declaring a new namespace using an xmlns attribute in the <Dataset> header element. The prefix associated with that namespace can then be used to introduce custom tag values without any fear of clashing. For instance:

```
<Dataset Name='Example'
```

```
        xmlns:MyEvents='http://mydomain.com/myevents'>
...etc...
<Event>
        <Type> MyEvents:xyz </Type>
```

This mechanism uses the same XML namespace feature that prevents clashes between element names and attribute names from different origins. XML tag names (elements and attributes) are deemed to belong to a given namespace and must be qualified using a namespace prefix if this is not the default one, e.g. <xs:annotation>. The qualified form is referred to as a QName. When the mechanism is employed within attribute values or element content then it is more correctly called a CURIE ("Compact URI"), and the qualified value may not even be a valid identifier — in the STEMMA case then this occurs with Place coordinates.

Although the namespace prefix is bound to a namespace URI, the XML standard does not define how to map a QName to an equivalent URI specification. The XML Schema language (XSD) concatenates the local tag name and the namespace URI using a '#' separator to create a Fragment identifier (e.g. http://stemma.parallaxview.co#Dataset) but it is not clear what happens if the namespace URI already ends with a '#'. The RDF model, on the other hand, simply concatenates the namespace URI and the local tag name with no separator (e.g. http://stemma.parallaxview.coDataset). Most RDF namespace URIs already end with a '#' (or even a '/') but not always. This is a well-known problem and a possible solution has been proposed at QNameQuagmire.

This does not directly impact the STEMMA use of namespaces though. The above custom Event-type is defined by the pair (http://mydomain.com/myevents, xyz) and the predefined Event-type 'Union' is defined by the pair (http://stemma.parallaxview.co/event-type, Union). The main differences here are that these namespaces apply to tag values, and the non-default namespaces are local to the associated Dataset.

An XML parser normally discards any such prefixes once the XML has been loaded since they usually just connect names to their respective namespace declarations. The exception to this is when they have been employed in attribute values or element content, as is the case in STEMMA and SOAP. The prefix-to-namespace mapping then has to be retained and made available to the program loading the XML. This is why the tag-value namespaces are expected to be associated with the enveloping Dataset element rather than any of elements below it.

Note that tag values should not be displayed directly in the UI of a genealogical product. See Locale-independence.

The namespace prefix DC='http://purl.org/dc/terms/' is required for the Dublin Core semantic types mentioned in this documentation. The use of a namespace prefix, rather than simply "DC.", accommodates other semantic-type systems if necessary.

See Digital Freedom for a related discussion.

On a technical note, this usage of a URI is sometimes referred to as a URN although, strictly speaking, a URN is a particular form of URI that uses a "urn:" scheme prefix and is designed to support hierarchical naming of objects. A much-quoted example of it is ISBN book references. The syntax is therefore more rigid (e.g. urn:xx:yy:zz) and the allowable characters more restricted. The associated namespace also has to be officially registered and that administrative burden tends to lessen its usage.

What we've used here is still a URI but employed for naming purposes, and is hence not the same as a URL. W3C don't really have a separate category for this although the historical use of the term URN in this context is accepted. Some material describes it as a "namespace name" but that's not universal. The most familiar example is the "namespace URI" in XML bodies.

Essentially, this form of URI names an object, or a type, and is guaranteed to be unique by virtue of the ownership of the domain used. For instance, a URI of http://stemma.parallaxview.co is unique to STEMMA because the author owns the parallaxview.co domain. It is also possible to derive URIs from a private email address, e.g. mailto:name@emaildomain?subject=types. Such namespace URIs are not designed to be dereferencable and so the scheme prefix isn't implying any access protocol.

In summary, this style of URI can be created in a decentralised manner (unlike real URNs), it is extensible (supporting derivative names and types), and it can be versioned. This contrasts with the use of raw UUIDs, or even ones wrapped as URNs (e.g. urn:uuid:d8e6a531-5dee-47a1-a0e2-ca5dbffd87c0), since they are amorphous and isolated See URNs, Namespaces, and Registries for more details, and Uniform Resource identifier for a mention of URC.

See Uniform Resource Identifier for a good summary.

8.2 Extended Properties

The <ExtendedProperties> element defines additional property names and types for usage within the current Dataset. These are effectively implemented as name-value pairs that can be used in subsequent Person, Animal, Place, Group, or Event entities.

EXTENDED_PROPERTIES=

<ExtendedProperties>
[<PersonProperties> PROPERTY_DEF … </PersonProperties>]
[<PlaceProperties> PROPERTY_DEF … </PlaceProperties>]
[<GroupProperties> PROPERTY_DEF … </GroupProperties>]
[<AnimalProperties> PROPERTY_DEF … </AnimalProperties>]
[<EventProperties> PROPERTY_DEF … </EventProperties>]
</ExtendedProperties>

PROPERTY_DEF=

<PropertyDef Name='prefix:name' [Type='type'] [Units='units']
[SemType='sem-type'] [ItemList='boolean'/>

PROPERTY_VALUE=

{ <Property Name='name' [Key='key'] [Value='value']
[Units='units'] [DATA_ATTRIBUTE] ... >
orig-text
</Property> }
|
{ <Property Name='name'>
orig-text
{ <Item [Key='key'] [Value='value'] [Units='units']
[DATA_ATTRIBUTE] ... >
orig-text
</Item> } ...
</Property> }

For Properties that accept the optional Value attribute, the value defaults to the original written value if the attribute is absent. This presumes that the original written value is syntactically valid for the given data-type. For multi-value Properties, the orig-text may be a single datum if an individual item cannot be separated entirely. Any individual non-blank text is taken to be more specific than an amalgamation.

For custom Properties, as opposed to the predefined STEMMA ones, a namespace prefix should be present. See Extended Vocabularies. In the case of Properties of type EnumList (including its derivations) then each term may have such a prefix, e.g.

Value="pre1:Term1.pre2:Term2"

The list of valid data-types may be found at: Data Types. See Extended Vocabularies for defining custom data-types. The semantic type is indicated by the SemType attribute which may use the Dublin Core vocabulary, e.g. SemType='DC:title' or SemType='DC:publisher'.

Q: Should there be a variant of the 'Text' data-type that corresponds to text in particular language. For instance, in order to distinguish the name of an article — which may be in English — from a textual reference code (e.g. RG13) which would be language-independent?

When defining a Property, the Units attribute can provide a list of valid unit names, separated by commas, with the default being the first one. An example of this is used to define the STEMMA 'Age' property at Properties.

If ItemList is true (i.e. ‘1’) then the value may be a list of items, each of which is represented using an <Item> element. If the value has associated Units then the respective attribute will be on the <Item> element. In the case of a list containing a single value, the non-ItemList syntax is acceptable even though ItemList was true.

For example:

```
<Dataset Name='Example' xmlns:t='http://example.com/types'>

<ExtendedProperties>
        <PersonProperties>
                <PropertyDef Name='t:PlaceOfEducation' Type='PlaceRef'/>
        </PersonProperties>
</ExtendedProperties>
```

A sample reference might be:

```
<Event Key='eGraduation'>
        <Type> Education </Type>
        <SubType> Enrolment </SubType>
        <When> 1974-06 </When>
        <SourceLnk Key='sGraduation'>
                <PersonLnk Key='pMe'>
                        <Property Name='Name'> A. C. Proctor </Property>
                        <Property Name='t:PlaceOfEducation'
                        Key='wCollege'/>
                </PersonLnk>
                <PlaceLnk  Key='wPeoplesCollege'>
                        <Property Name='Name'> People's College </Property>
                </PlaceLnk>
        </SourceLnk>
</Event>
```

Notice that as well associating a reference to a named person with an equivalent Person entity, it also associates the reference to a named place with a corresponding Place entity. This has significant advantages if you want to make use of organised Place entities rather than simple place names. The lesser alternative would have been to declare the PlaceOfEducation Property as having a mere Text value. See also the Source entity which provides a way of assembling profiles from the subject references in a source — thus allowing analysis to be performed before making those associations to actual conclusion entities.

Let’s look at an example for Height and Weight Properties which would require the optional Units attribute.

```
<ExtendedProperties>
        <PersonProperties>
```

```
            <PropertyDef Name='t:Height' Type='Measure'
            Units='cm,inch'/>
            <PropertyDef Name='t:Weight' Type='Measure'
            Units='kg,lb'/>
        </PersonProperties>
</ExtendedProperties>
```

A sample reference might be:

```
<Event Key='eEnlistment'>
        <Type> Military </Type>
        <SubType> Enlistment </SubType>
        <When> 1914-06 </When>
        <SourceLnk Key='sEnlistment'>
            <PersonLnk>
                <Property Name='Name'> W. H. Jones </Property>
                <Property Name='t:Weight' Units='kg' Value='72.3'>
                11st 5lb </Property>
                <Property Name='t:Height' Units='inch' Value='71'> 5ft
                11in </Property>
            </PersonLnk>
        </SourceLnk>
</Event>
```

No list of accepted units is prescribed here, although the Property definition can declare the accepted units using a comma-separated list in a Units attribute. Either the full name or the standard abbreviation may be used, together with the correct prefix if relevant (e.g. cm). Care should be taken not only to differentiate metric from imperial units, but also the UK/US differences for gallons, pints, tons, ounces, etc., wet and dry differences such as ounces versus 'fluid ounces', and nautical miles.

8.3 Extended Schema

This mechanism would be used where new elements need to be added to the standard schema. An example might be a requirement to represent vehicles through a new <Vehicle> element. Under these circumstances, the author designing the extension needs to provide an associated XSD schema file defining the syntax of those new elements. The new XML elements and attributes are then referenced with the prefix of the associated namespace. For instance:

```
<?xml version="1.0"?>
<Datasets xmlns='http://stemma.parallaxview.co/2017-04'
        xmlns:vh='http://www.vehicle-history/2012'>

        <Dataset Name='Example'>
            <vh:MotorCycle Key='Harley1'>
                ...data parts...
            </vh:MotorCycle>
```

```
            <Person>
                <Text>
                    Text with embedded Vehicle reference
                    <vh:VehRef Key='Harley1'/>
                </Text>
            </Person>
        </Dataset>
</Datasets>
```

A schema validation operation on such XML would use both the standard schema associated with the default unnamed namespace and the extended schema associated with the additional namespace.

STEMMA implements a number of *subject entities* — Person, Place, Animal, and Group — and it is this area where schema extensions are most likely to be proposed. The existing entities are treated uniformly using OOP techniques such that they all inherit from a base subject-entity class, with intermediate base classes to differentiate inanimate and animate subjects.

9 Data Types

The following data-types are defined for Property values (see PROPERTY_VALUE) and Parameter values (see PARAM_VALUE). Although quite similar, the table indicates where there are necessary differences.

Properties embrace 'orig-text' and annotate it with a combination of 'Key' and/or 'Value' attributes. The 'Value' is considered optional if the orig-text contains the same textual representation, and a 'Key' is optional in those circumstances where an entity association is not available.

Parameters embrace an explicit 'value' and can annotate some cases with a 'Key' attribute. The 'Key' references are mandatory when interfacing to citation-templates if an *object* is to be passed. Parameters can also override the 'value' formatting using their 'Subst' attribute. During local Parameter substitution, the precedence is: 'Subst' attribute, any explicit 'value' (formatted according to data-type), any entity key name.

Data-type	Description	Property	Parameter
Text	Simple text item (default data-type).	[Value]	value
Integer	Non-negative whole number.	[Value]	value
Number	Decimal number.	[Value]	value
Measure	Decimal number with units.	Value, Units	--
Boolean	1=true, 0=false.	[Value]	value
Date	Requires an std-date date-value, or a <Date> date-entity (see Dates).	[Value]	value
Enum	Enumerated set of possible values.	[Value]	--

EnumList	As per Enum but multiple entries can be concatenated with a period to create a dependent list, i.e. where each term is interpreted in the context of the previous one.	Value	--
PersonRef	Person (or Contact) reference.	[Key]	Key || value (one or both)
PlaceRef	Place reference.		
AnimalRef	Animal reference.		
GroupRef	Group reference.		
EventRef	Event reference.		
PersonEL	Subject entity together with a relative EnumList value.	Key, [Value]	--
PlaceEL			
AnimalEL			
GroupEL			

Prior to V4.1, a Date Parameter was restricted to ISO Gregorian dates; however, the full range of calendars is required for older sources, and a common example is citing newspapers that predate the Julian-to-Gregorian changeover.

10 Event Types and Roles

The table below identifies some of the main event-types and event-subtypes of interest. The default in both instances is ‘Unknown’. The table also indicates typical associations of person-roles with event-type.

The predefined event-types are ones for which direct evidence is likely to be available. STEMMA deliberately makes it easy to define new event-types and subtypes, partly so that custom entities can be defined for describing detailed hierarchical events. The Persons associated with an Event are ones mentioned in a supporting source and that usually means they were present but not always. The Status Property can be used for deceased/implied/absent/ persons (e.g. a deceased parent on a marriage certificate, a person mentioned — *implied* — but not present, or a crossed-out — *absent* — person in a census household).

The values for event-subtype are subordinate to an event-type. In software terms, the event-subtype may be represented as a child of the event-type, e.g. Union.Marriage. The reason this was not carried through to the person-role (e.g. Union.Marriage.Bride) was due to the number of combinations and the undue rigidity.

Event Type	Event sub-type	Event Role	Notes
Birth	Birth	Child, Mother, Informant	
	StillBirth	Mother	
	Miscarriage		
Death	Burial	Deceased	

	Cremation		
	Death	Deceased, Informant	
Dissolution	Annulment	Husband, Wife, Partner	
	Divorce		
	Separation		Also applies to unmarried couples
Education	Enrolment	Subject	
	Graduation		
Group	Begin		
	Change		
	End		
Legal	Court	Subject	
	Imprisonment		
	Probate		
	Release		
	Will		
Medical		Subject	
Military	Campaign	Subject	
	Discharge		
	Disciplinary		e.g. Court martial
	Engagement		Not same as Union.Engagement
	Enlistment		
	Rank		Promotion, Demotion
	Recognition		Medals, awards, citations
	Travel		
Place	Begin		
	Change		
	End		
Religious	Baptism	Child	
	Bar-mitzvah		
	Bas-mitzvah		Also Bat-mitzvah
	Blessing		
	Christening		
	Communion		
	Confirmation		
Responsibility	Adoption	Child, AdoptedParent, FosterParent	
	Fostering		
	Guardianship	Dependant, Guardian	
Social	Correspondence		
	Meeting		
Survey	Census	Head, Apprentice, Assistant,	Normally just one Head per household

		Boarder, Governess, HouseBoy, HouseKeeper, Inmate, Keeper, Lodger, LodgerHead, Member, Nurse, NurseChild, Officer, Orphan, Porter, Probationer, Pupil, Servant, Visitor	
	Directory		
Travel	AddressChange		
	Emigration		
	Immigration		
	Naturalisation		
Unknown	Unknown		
Union	Affair	Partner	
	Banns	Fiancee, Fiance	
	Cohabitation	Partner	
	Engagement	Fiancee, Fiance	NB: Fiancée is a woman, fiancé is a man
	Marriage	Bride, Groom, Witness, Guest, BestMan	Where civil/religious distinction not relevant
	MarriageCivil		
	MarriageContract		
	MarriageReligious		
	PrenuptialAgreement	Fiancee, Fiance	
	SameSex	Partner	
Work	Commencement	Subject	
	Retirement		
	Status		Promotion, Demotion
	Termination		

A description of how custom event-types and roles are accommodated may be found at Extended Vocabularies. Note that Unions involving more than two people, or people of the same sex, can be determined by looking at the Roles involved. They are not given a specific event-type.

Roles and Relationships are both normalised items of information about a subject reference, but a Role is something relative to the Event itself, whereas a Relationship is something relevant between two subject references, such as

between two persons. A more readable work on Roles and Relationships can be found at Role of the Role.

Q: Do we need a recognised event-type for a gender reassignment procedure? The <Sex> element describes the “birth sex” (however indeterminate) but we still need to be able to record such a procedure. See No Sex Please, We’re Genealogists!.

Q: Should we handle change of slave ownership under the event-type ‘Responsibility’?

11 Relationships

STEMMA has the capability to represent relationships between entities of different types. This is done using Properties in a <SourceLnk> element of an Event, where the extracted and summarised forms are assembled. However, person-to-person relationships are far more common than the others, and attempting to decide whether something is a group-relative *relationship* or an event-relative *role* may be a subjective decision.

The Person Property called Relationship has predefined values for biological relationships, and a number of other predefined ones for common non-biological cases. The biological ones may be recognised and acted upon by software, but all others are interpreted blindly as 'relationships'. If a researcher determined that some of those relationships were more binding, or more significant, then a Group entity could be used to collect them together and document why. A typical example might be a “family unit” since no program can infer it all by itself.

The following person-to-person relationships must be relative to a selected person reference and so the Relationship Property has a data-type of PersonEL. That means it must have a Key attribute identifying the selected person reference, and a list of one-of-more relationship terms.

Husband,Wife
Brother,Daughter,Father,Mother,Sister,Son
Grand{Daughter,Father,Mother,Son}
Half{Brother,Sister}
Step{Brother,Daughter,Father,Mother,Sister,Son}
{Brother,Daughter,Father,Mother,Sister,Son}InLaw
Cousin,Uncle,Aunt,Nephew,Niece
{Foster,Adopted}{Parent,Sibling,Child}
Guardian,Dependant
Friend, Neighbour

For instance, the following would be read as (John).Wife.Friend:

```
<Property Name=’Relationship’ Key=’pJohn’ Value=’Wife.Friend’>
        Friend of John’s wife
</Property>
```

Note that the “Relation to Head of Family” column in the census of England and Wales is overloaded by representing both event roles and personal relationships. A more detailed discussion can be found at Role of the Role.

12 Glossary

This section defines the terms used in this STEMMA data-model description. Some of these, such as attribute, element, and document, also have XML semantics. The usage here has tried to be compatible with those semantics even though they are technically independent. Also see the supplemental glossary for Dates and Calendars.

Abstract Entity
An *entity* (q.v.) with an Abstract=’1’ attribute. Such entities are designed to be used as base entities for *inheritance* (q.v.) and must not contain any Key references.

Address (or Postal Address)
A sequence of terms that direct traditional mail (e.g. letters, packages, etc) to a particular recipient. Contrast with *place* and *location (qq.v.)*.

Animal
The representation of a unique physical non-human animal, including their properties, parentage, event history, and biographical narrative.

Annotation
Text annotation is the addition of meta-data (related textual or other information) to a body of text. See also *Mark-up* (q.v.)

Attribute
An item of data or meta-data associated with an *element* (q.v.). Effectively a name-value pair. This definition is in keeping with the XML interpretation, although technically distinct.

Bundle
A STEMMA *document* (q.v.) and any local attachments (e.g. images) that it references, bundled together for transmission as a whole.

Calendar
A mechanism by which dates are reckoned in a given culture. For instance, Gregorian, Julian, etc.

Citation
A written or printed identification of a source of information. See *citation mode*, *citation style, and layered citation* (qq.v.). In STEMMA, a Citation *entity* (q.v.) is a generalised description of a *source* (q.v.), the location of a source, or the location of *information* (q.v.) within a source.

Citation Element

A specific datum, or iterated list of data (as with page numbers), that are required to generate a *citation* (q.v.). Also see *citation template*.

Citation Mode
The written or printed mechanism employed to achieve a *citation* (q.v.). For instance, a source label, a reference note, or a bibliographic entry.

Citation Reference
In STEMMA, a reference to a Citation *entity* (q.v.) with parameters that identify the associated source, source location, or item of information in that source.

Citation Style
The style of a written or printed citation, including the order of the parameters, punctuation, and use of italics etc. Examples in the humanities include CMOS, Harvard referencing, MLA. *Evidence Explained* (EE) is the most common for genealogy.

Citation Template
A specification for how *citation elements* (q.v.) should be processed in order to generate a formatted *citation* (q.v.).

Conclusion
See *inference* (q.v.).

Constraint
A relational connection between two Events that orders them and places limits on their dates, even when those dates may not be fully determined.

Controlled Vocabulary
A fixed set of predefined termed used for the description or classification of data. See Controlled vocabulary. Contrast with *partially controlled vocabulary* (q.v.).

Credibility
The confidence in the information from a source as being unbiased and unembellished. Contrast with *Quality*, *Reliability*, and *Surety* (qq.v.).

Dataset
A named, self-contained sub-section of a STEMMA *document* (q.v.).

Date Entity, Date Value
A computer-readable date as reckoned according to a given *calendar* (q.v.). A date value is a specific date encoded in a text string, whereas the date entity also represents *granularity* (q.v.), *imprecision* (q.v.), and *synchronised dates.*

Date String
A date as originally written, including transcribed version thereof. A *date value* (q.v.) or *date entity* (q.v.) formatted for readability by a user.

Deep Semantics
The nature of a datum (e.g. a date reference, or a person reference) including a conclusion identifying the target (e.g. the actual date, or the actual person). Contrast with *shallow semantics* (q.v.).

Definite Source
Identification of a specific *source* (q.v.) that was accessed or consulted during research. Contrast with *indefinite source* (q.v.). Analogous to the *definite article* in grammar.

Discursive Notes
Commentary or notes that digress from the main subject. Usually presented using footnotes or endnotes.

Document
A complete STEMMA file, or its representation in memory or in a communications network. This definition is in keeping with the XML interpretation, although technically distinct.

Drill Down
Originally a BI process where selecting a summarised datum or a hierarchical field — usually with a click in a GUI tool — revealed the underlying data from which it was derived. The term was used routinely during the early 1990s when a new breed of data-driven OLAP product began to emerge. Used here to describe the process of "opening" a conclusion to see the supporting evidence (including any proof argument), the raw information that was used as evidence, and eventually to the underlying source.

Element
A logical part of a STEMMA *document* (q.v.). A Document is effectively a tree of nested elements. This definition is in keeping with the XML interpretation, although technically distinct.

Endnote
See *footnote* (q.v.).

Entity
A data-model representation of a *micro-history* (q.v.) item. These are top-level elements in a *dataset* (q.v.) and usually have a Key by which they can be referenced. See *Person*, *Animal*, *Place*, *Group*, *Event*, *Citation*, *Resource, Source*, and *Matrix* (qq.v.). See also *abstract entity* (q.v.).

Event
In everyday life, an event is something that happened at a particular place and time, or over a span of time. The Event *entity* (q.v.), though, represents a date, or range of dates, for which source information exists. It effectively provides a where-and-when context for the referenced subjects. *See simple Event*, *protracted Event*, and *hierarchical Event* (qq.v.).

Evidence

Information (q.v.) that either supports or contradicts a statement or claim.

Family History
A type of *micro-history* (q.v.) concerned with people's lives within a family context. See *genealogy* (q.v.).

Footnote
Paragraph of text appearing at the end of a page (footnote) or chapter, volume, or whole text (endnote). General mechanism for adding commentary, notes, or source references linked to a location in the main text.

Genealogy
In its literal sense, the study of biological lineage. In its more generic usage, any type of research involving families. See f*amily history* (q.v.).

Granularity
The date unit or concept that a date-value is representing, e.g. a day, a month, a decade, or a century. This is similar to a 'period' in the GEDCOM model. Contrast with *imprecision* (q.v.).

Group
A STEMMA *entity* (q.v.) representing an organised real-world entity, such as a regiment, organisation, or school/class. Individual *subject entities* (q.v.) — currently only Persons and Animals — may be associated with a Group over given spans of time. It may, therefore, be used to model different interpretations of a family unit.

Hierarchical Event
A p*rotracted Event* (q.v.) which is the parent of one or more other Events. This mechanism allows the finer structure of a p*rotracted Event* to be described.

Imprecision
The uncertainty or range of possibilities for a date specification. This is a different concept to the date unit being referenced. This is similar to a 'range' in the GEDCOM model. Contrast with *granularity* (q.v.).

Indefinite Source
A generic identification of a *source* (q.v.) that may be available from many different places, and for which provenance and analytical notes are irrelevant. Common in scientific journals. Contrast with *definite source* (q.v.). Analogous to the *indefinite article* in grammar.

Inference
A reasoned judgment or decision based on *evidence* (q.v.) derived from *information* (q.v.) obtained from *sources* (q.v.). The judgement includes both the reasoning and any associated conclusion.

Information
Semantic data obtained from a *source* (q.v.).

Inheritance
A mechanism that allows an *entity* (q.v.) to share parts of the definition of a parent entity. In STEMMA, this currently occurs for *Events*, *Citations*, and *Resources* (qq.v.).

Layered Citation
A form of citation used to describe the different derivative states of a consulted source through to its original form; usually separated by semicolons. May also include analytical notes.

Location
A fixed geographical point or area, usually referenced by its coordinates. Contrast with *place* and *address* (qq.v.).

Mark-up
The scheme by which text *annotation* (q.v.) is represented or encoded.

Matrix Entity
Entity representing the *profiled* (q.v.) information from multiple *sources* (q.v.).

Micro-history
History on a smaller scale than world history. Often researching smaller units such as local places, families, ordinary people, surnames, and the fine-grained events that interrelate them.

Namespace
A container for a set of names or other identifiers. Each namespace typically has an identifier of its own unless it is the default one.

Parameter
The name-value pairs employed by a Citation or Resource reference to identify specific information or data.

Parameterisation
A mechanism where parameter values are applied to an entity in order to modify its context. This is supported for *Citations* and *Resources* (qq.v.). Parameters may be inherited from a base entity, specified in the body of an entity, or specified in a reference to that entity. All of these schemes work together.

Partially Controlled Vocabulary
A set of core predefined terms for the description or categorisation of data that allows for extensions. Extensions are usually defined within alternative namespaces. Contrast with *controlled vocabulary* (q.v.).

Person
The representation of a unique physical person, including their properties, parentage, event history, and biographical narrative.

Persona (pl. Personae)

A description of some person from one specific information source, and with no interpretation. See Genealogical Persona Non Grata.

Place
A named point or area deemed to have significance to humans. Contrast with *location* and *address* (qq.v.).

Place Hierarchy
A description of a *place* (q.v.) in which each component of the reference is linked to a parent *place* with a broader context, e.g. house to street, to town, to county, to state, to country. See A Place for Everything.

Place Hierarchy Path
An ordered list of *place* (q.v.) names from a *Place hierarchy* (q.v.) that uniquely identifies the *place*. For instance, “15, Manning Grove, Nottingham, Nottinghamshire, England”. NB: the order and separator characters are both culturally dependent. See A Place for Everything.

Profile
(noun) The representation of a subject, object, statement, or concept from a *source* (q.v.) — such as a *prototype subject* (q.v.), event, date, or general words/phrase — and its relationship to other profiles. (verb) To profile a source is to create these profile items for it.

Property
A named datum representing extracted and summarised *information* from a *source* and associated with a *Person*, *Animal*, *Place*, *Group*, or *Event* (qq.v.). They are slightly more than simple name-value pairs since selected data-types may include a unit of measurement, an entity reference, or a list of associated values.

Protracted Event
An *Event* (q.v.) that has both a start and an end date.

Prototype Subject
Correlated references to some subject (person, animal, place, group) derived from one or more sources. Contrast with *Subject Entity* and *Subject Reference* (qq.v).

Quality
The confidence in a source as an accurate version of an original. Contrast with *Credibility*, *Reliability*, and *Surety* (qq.v.).

Reference Note
Source reference note. Source citation or commentary provided via a *footnote* or *endnote* (qq.v.).

Reliability
The confidence in the information from a source as being first-hand or less direct. Contrast with *Credibility*, *Quality*, and *Surety* (qq.v.).

Resource
Entity (q.v.) describing digital and/or physical artefacts, including images and transcriptions.

Shallow Semantics
The nature of a datum (e.g. a date reference, or a person reference) without any conclusion being offered that identifies the target (e.g. the actual date, or the actual person). Contrast with *deep semantics* (q.v.).

Simple Event
An *Event* (q.v.) that has a single date.

Source
Origin of *information*, such as an artefact, book, newspaper, person, census, photograph, Web site, etc. See *definite* and *indefinite source* (qq.v.).

Source Entity
Entity representing the *profiled* (q.v.) information from a given *source* (q.v.).

Source Format
A definitive plain-text machine-readable version of data that can be used for multiple purposes (e.g. backups, exchange between different products) or from which derivatives can be generated (e.g. loading into an indexed database product, or conversion to alternative formats). The term is analogous to *source code* in a programming context, which is a definitive representation that can be compiled for different machines, and in any locale.

Structured Narrative
Rich-text that uses semantic mark-up to link references to persons, animals, places, groups, and events, to their respective STEMMA *entity* (q.v.) representations, or to simply mark them for analysis purposes.

Subject Entity
Description of a subject mentioned in historical sources, e.g. *Person*, *Animal*, *Place*, *Group* (qq.v.), built up from aggregated evidence from multiple sources. Contrast with *Subject References* and *Prototype Subjects* (qq.v).

Subject Reference
Reference to some subject (person, animal, place, group) from a given *source* (q.v.). Contrast with *Subject Entity* and *Prototype Subject* (qq.v).

Surety
A numeric estimation of the confidence in a piece of *evidence* or a *conclusion*. For instance, a source may have contained original errors, and conclusions may include conjecture and speculation. Contrast with *Credibility*, *Quality*, and *Reliability* (qq.v.).

Vocabulary

The allowable values for a given genealogical datum, such as event-type, person-property, place-type. See *Controlled-* and *Partially Controlled Vocabulary* (qq.v.).

13 STEMMA Example

This section presents a simplified example for a William Elliott, born in Uttoxeter c1841. The example is abbreviated for compactness and so does not include other people present in each census, or provide for alternative Place name spellings, or make full use of the inheritance mechanism. A fuller version of the same example may be taken from Downloads.

Note that the birth event for William is a conclusion only since it is not directly backed any source such as a birth certificate. However, it includes a small amount of narrative in the associated Event entity to explain how the date relates to his age in various other sources. A more involved version of this example may be found at Evidence, Reasoning, and Conclusion where the Source and Matrix entities are used to perform correlation of those supporting sources, and leave an explanatory trail that can be followed via a drill-down operation.

```
<?xml version="1.0"?>
<Datasets xmlns='http://stemma.parallaxview.co/2017-04'>
    <Header>
        <Created> 2015-11-01T17:00:00Z </Created>
        <Product>
            <Id></Id>
            <Name> Hand Crafted</Name>
            <Version></Version>
        </Product>
        <Text>
            Example STEMMA Document.
        </Text>
    </Header>

    <Dataset Name='Elliott_Example'>
        <Content>
            <Created> 2015-11-01T17:00:00Z </Created>
            <Author> Tony Proctor </Author>
            <Version> 2.0 </Version>
            <Locale> en_GB </Locale>
            <Text>
                Example STEMMA Dataset.
            </Text>
        </Content>

        <Person Key='pSarahWoods'>
            <Title> Sarah Woods </Title>
            <Sex> 0 </Sex>
            ... etc ...
```

```
</Person>

<Person Key='pSarahWildgoose'>
        <Title> Sarah E. Wildgoose </Title>
        <Sex> 0 </Sex>
        ... etc ...
</Person>

<Person Key='pWilliamElliott'>
        <Title> William Elliott </Title>
        <Sex> 1 </Sex>
        <Names>
                <Sequences>
                        <Canonical>William Elliott</Canonical>
                        <Sequence>
                                <Tokens><Token>William</Token
                                ></Tokens>
                                <Tokens><Token>Elliott</Token>
                                </Tokens>
                        </Sequence>
                </Sequences>
        </Names>

        <Birth>
                <EventLnk Key='eBirthWilliamElliott'/>
        </Birth>
</Person>

<Citation Key='cCensusEngWales' Abstract='1'>
        <Title>Census of England and Wales, 1841-
        1901</Title>
        <URI> http://www.nationalarchives.gov.uk/census
        </URI>
        <Params>
                <Param Name='Series'/>
                <Param Name='Piece' Type='Integer'/>
                <Param Name='Book' Type='Integer'
                Optional='1'>0</Param>
                <Param Name='Folio' Type='Integer'/>
                <Param Name='Page' Type='Integer'/>
        </Params>
</Citation>

<Citation Key='cCivilMarriage' Abstract='1'>
        <Title> Civil Marriage Registration </Title>
        <URI> http://www.gro.gov.uk/marriage </URI>
        <Params>
                <Param Name='Year' Type='Integer'/>
                <Param Name='Quarter'/>
```

```
                <Param Name='Volume'/>
                <Param Name='Page' Type='Integer'/>
        </Params>
</Citation>

<Event Key='eBirthWilliamElliott'>
        <Title>Birth of William Elliott </Title>
        <Type> Birth </Type>
        <SubType> Birth </SubType>
        <When><Date>
                <Value> 1841 </Value>
                <Text Inference='1'>
                William's date of birth is derived from his age in
                1851 (10), 1861 (20), 1862 (21), 1871 (31), and
                1881 (35). The latter is an outlier due to being
                estimated.
                </Text>
        </Date></When>
        <PlaceLnk Key='wUttoxeter'/>
</Event>

<Event Key='eCensusElliott1851'>
        <When Value='1851-03-30'/>
        <Title>1851 census for William Elliott</Title>
        <Type> Survey </Type>
        <SubType> Census </SubType>
        <PlaceLnk Key='wTinkersLane'/>
        <SourceLnk Key='sCensusEngWales'>
                <PersonLnk Key='pWilliamElliott'>
                        <Property Name='Age'> 10 </Property>
                        <Property Name='Occupation'> Scholar
                        </Property>
                        <Property Name='BirthPlace'
                        Key='wUttoxeter'/>
                        <Property Name='Relationship'
                        Key='pTimothyElliott'> Son </Property>
                        <Property Name='Status'/>
                </PersonLnk>
        </SourceLnk>
</Event>

<Source Key='sCensusEngWales'>
        <Frame>
                <CitationLnk Key='cCensusEngWales'>
                        <Param Name='Series'> HO107 </Param>
                        <Param Name='Piece'> 2010 </Param>
                        <Param Name='Folio'> 113 </Param>
                        <Param Name='Page'> 8 </Param>
                </CitationLnk>
```

```
        </Frame>
</Source>

<Event Key=’eCensusElliott1861’>
        <When Value=’1861-04-07’/>
        <Title>1861 census for William Elliott</Title>
        <Type> Survey </Type>
        <SubType> Census </SubType>
        <PlaceLnk Key=’wRussellStreet’/>
        <SourceLnk Key=’sCensusEngWales1861’>
                <PersonLnk Key='pWilliamElliott'>
                        <Property Name=’Age’> 20 </Property>
                        <Property Name=’Occupation’> Labourer
                        </Property>
                        <Property Name=’BirthPlace‘
                        Key='wUttoxeter'/>
                        <Property Name=’Relationship’
                        Key='pTimothyElliott'> Son </Property>
                        <Property Name=’Status’> Unmarried
                        </Property>
                </PersonLnk>
        </SourceLnk>
</Event>

<Source Key='sCensusEngWales1861'>
        <Frame>
                <CitationLnk Key='cCensusEngWales'>
                        <Param Name='Series'> RG09 </Param>
                        <Param Name='Piece'> 2505 </Param>
                        <Param Name='Folio'> 129 </Param>
                        <Param Name='Page'> 27 </Param>
                </CitationLnk>
        </Frame>
</Source>

<Event Key=’eCensusElliott1871’>
        <When Value=’1871-04-02’/>
        <Title>1871 census for William Elliott</Title>
        <Type> Survey </Type>
        <SubType> Census </SubType>
        <PlaceLnk Key=’wSiddalsLane62’/>
        <SourceLnk Key=’sCensusEngWales1871'>
                <PersonLnk Key='pWilliamElliott'>
                        <Property Name=’Age’> 31 </Property>
                        <Property Name=’Occupation’> Labourer
                        in Iron works </Property>
                        <Property Name=’BirthPlace’
                        Key='wUttoxeter'/>
```

```
                    <Property Name='Role'> Boarder
                    </Property>
                    <Property Name='Status'> Married
                    </Property>
                    <Text Inference='1'>
                    Head of household is Elizabeth Wildgoose
                    (b. c1802) and is almost certainly a
                    relative of Sarah Elliott, nee Wildgoose.
                    </Text>
                </PersonLnk>
        </SourceLnk>
</Event>

<Source Key='sCensusEngWales1871'>
        <Frame>
                <CitationLnk Key='cCensusEngWales'>
                    <Param Name='Series'> RG10 </Param>
                    <Param Name='Piece'> 3565 </Param>
                    <Param Name='Folio'> 82 </Param>
                    <Param Name='Page'> 35 </Param>
                </CitationLnk>
        </Frame>
</Source>

<Event Key='eCensusElliott1881'>
        <When Value='1881-04-03'/>
        <Title>1881 census for William Elliott</Title>
        <Type> Survey </Type>
        <SubType> Census </SubType>
        <PlaceLnk Key='wCarringtonSq14'/>
        <SourceLnk Key='sCensusEngWales1881'>
                <PersonLnk Key='pWilliamElliott'>
                    <Property Name='Age' Surety='20%'> 35
                    </Property>
                    <Property Name='Occupation'> Striker
                    Iron Foundry </Property>
                    <Property Name='BirthPlace'
                    Key='wUttoxeter'/>
                    <Property Name='Role'> Lodger
                    </Property>
                    <Property Name='Status'> Married
                    </Property>
                    <Text Inference='1'>
                    William's age probably estimated to be 35
                    </Text><Text Inference='1'>
                    William and Sarah Wood didn't marry
                    until October but they were recorded as
                    man and wife here. This was because their
                    son, William, was already born by January.
```

```
                    </Text>
                </PersonLnk>
        </SourceLnk>
</Event>

<Source Key='sCensusEngWales1881'>
        <Frame>
                <CitationLnk Key='cCensusEngWales'>
                        <Param Name='Series'> RG11 </Param>
                        <Param Name='Piece'> 3404 </Param>
                        <Param Name='Folio'> 18 </Param>
                        <Param Name='Page'> 30 </Param>
                </CitationLnk>
        </Frame>
</Source>

<Event Key='eMarriageElliott1862'>
        <When Value='1862-03-12'/>
        <Title>Marriage of William Elliott and Sarah
        Wildgoose</Title>
        <Type> Union </Type>
        <SubType> Marriage </SubType>
        <PlaceLnk Key='wDerbyRegOffice'/>
        <SourceLnk Key='sMarriage1862'>
                <PersonLnk Key='pWilliamElliott'>
                        <Property Name='Age'> 21 </Property>
                        <Property Name='Occupation'>
                        Hammersman </Property>
                        <Property Name='ResidencePlace'
                        Key='wVictoriaStreet'/>
                        <Property Name='Role'> Groom
                        </Property>
                        <Property Name='Status'> Unmarried
                        </Property>
                </PersonLnk>
        </SourceLnk>
</Event>

<Source Key='sMarriage1862'>
        <Frame>
                <CitationLnk Key='cCivilMarriage'>
                        <Param Name='Year'> 1862 </Param>
                        <Param Name='Quarter'> Q1 </Param>
                        <Param Name='Volume'> 7b </Param>
                        <Param Name='Page'> 444 </Param>
                </CitationLnk>
        </Frame>
</Source>
```

```
<Resource Key='rPhotoTinkersLane'>
        <Title>Photograph of Tinker's Lane (c1860)</Title>
        <Type> Photograph </Type>
        <URL> file:mydir/TinkersLane.jpg </URL>
</Resource>

<Source Key='sPhotoTinkersLane'>
        <Frame>
                <ResourceLnk Key='rPhotoTinkersLane'/>
        </Frame>
</Source>

<Place Key='wTinkersLane'>
        <Title>Tinkers Lane</Title>
        <Type> Street </Type>
        <PlaceName> Tinkers Lane </PlaceName>
        <ParentPlaceLnk Key='wUttoxeter'/>
        <SourceLnk Key='sPhotoTinkersLane'/>
</Place>

<Place Key='wUttoxeter'>
        <Title>Uttoxeter</Title>
        <Type> Town </Type>
        <PlaceName> Uttoxeter </PlaceName>
        <ParentPlaceLnk Key='wStaffs'/>
</Place>

<Place Key='wBurtonParishChurch'>
        <Title>Parish Church of Burton-on-Trent</Title>
        <Type> Building </Type>
        <PlaceName> Parish Church </PlaceName>
        <ParentPlaceLnk Key='wBurton'/>
</Place>

<Place Key='wBurton'>
        <Title>Burton-on-Trent</Title>
        <Type> Town </Type>
        <PlaceName> Burton-on-Trent </PlaceName>
        <ParentPlaceLnk Key='wStaffs'/>
</Place>

<Place Key='wStaffs'>
        <Title>Staffordshire</Title>
        <Type> County </Type>
        <PlaceName> Staffordshire </PlaceName>
</Place>

<Place Key='wRussellStreet'>
        <Title>Russell Street</Title>
```

```
        <Type> Street </Type>
        <PlaceName> Russell Street </PlaceName>
        <ParentPlaceLnk Key='wLitchurch'/>
</Place>

<Place Key='wCarringtonSq14'>
        <Title>14 Carrington Square</Title>
        <Type> Number </Type>
        <PlaceName> 14 </PlaceName>
        <ParentPlaceLnk Key='wCarringtonSq'/>
</Place>

<Place Key='wCarringtonSq'>
        <Title>Carrington Square</Title>
        <Type> Street </Type>
        <PlaceName> Carrington Square </PlaceName>
        <ParentPlaceLnk Key='wLitchurch'/>
</Place>

<Place Key='wLitchurch'>
        <Title> Litchurch </Title>
        <Type> Town </Type>
        <PlaceName> Litchurch </PlaceName>
        <ParentPlaceLnk Key='wDerbys'/>
</Place>

<Place Key='wSiddalsLane62'>
        <Title>62 Siddals Lane</Title>
        <Type> Number </Type>
        <PlaceName> 62 </PlaceName>
        <ParentPlaceLnk Key='wSiddalsLane'/>
</Place>

<Place Key='wSiddalsLane'>
        <Title>Siddals Lane</Title>
        <Type> Street </Type>
        <PlaceName> Siddals Lane </PlaceName>
        <ParentPlaceLnk Key='wDerby'/>
</Place>

<Place Key='wVictoriaStreet'>
        <Title>Victoria Street</Title>
        <Type> Street </Type>
        <PlaceName> Victoria Street </PlaceName>
        <ParentPlaceLnk Key='wDerby'/>
</Place>

<Place Key='wDerbyRegOffice'>
        <Title>Derby Register Office</Title>
```

```
            <Type> Building </Type>
            <PlaceName> Register Office </PlaceName>
            <ParentPlaceLnk Key='wDerby'/>
        </Place>

        <Place Key='wDerby'>
            <Title> Derby </Title>
            <Type>Town</Type>
            <PlaceName> Derby </PlaceName>
            <ParentPlaceLnk Key='wDerbys'/>
        </Place>

        <Place Key='wStapenhill'>
            <Title> Stapenhill </Title>
            <Type> Town </Type>
            <PlaceName> Stapenhill </PlaceName>
            <ParentPlaceLnk Key='wDerbys' Before='1889'/>
            <ParentPlaceLnk Key='wStaffs' From='1889'/>
            <Text>
                In 1889 the part of the parish of Stapenhill in the
                borough of Burton upon Trent became part of
                Staffordshire, and in 1894 the remaining
                Derbyshire parts of the parish became part of the
                parishes of Bretby and Drakelow, so that
                thereafter Stapenhill was wholly in Staffordshire.
            </Text>
        </Place>

        <Place Key='wDerbys'>
            <Title> Derbyshire </Title>
            <Type> County </Type>
            <PlaceName> Derbyshire </PlaceName>
        </Place>
    </Dataset>
</Datasets>
```