# EATS
# an entity authority tool set

Jamie Norrish

December 7, 2007

Authority control has long been used in library catalogue systems to handle the problem of ambiguous identifiers for entities, especially authors. With electronic and networked collections the need for authority control is only increased, and made more important by the possibility of extending defined entities to cover the content of full-text collections.

The Entity Authority Tool Set (EATS) is a web application for recording, editing, and using authority information about entities. EATS was developed at the New Zealand Electronic Text Centre (NZETC) and is now used to manage the 30 000+ entities represented in the NZETC online collection of significant New Zealand and Pacific Island texts. This paper sets out the requirements for an entity authority system and makes the case that no existing system suits all of those needs. It presents the EATS model for representing entity authority information, and describes the NZETC implementation of the model along with associated tools.

## Developments in authority control

Authority control is the practice of grouping multiple terms for the same entity into a single record for the purposes of disambiguation and collocation. In a library context, the entities are authors and books, and the terms are their different names and titles. In card catalogues, the primary requirement is to have a single authorised form of the name or title, under which the appropriate cards are filed. The other, non-authorised forms are included in the catalogue only as pointers to the authorised form, directing the user to the right place to look for the actual bibliographic records — this is the collocation function. So, for example, searching for the author "Iris Wilkinson" should result in a card pointing to "Robin Hyde", under which will be listed all of her books. Likewise, looking for "Voina i mir" (or indeed "Война и миръ") should result in a card pointing to "War and peace". The disambiguation function is achieved through the provision of extra information about an entity that can serve to distinguish between two entities which have the same name. So in the case of two authors, both named "Adam Smith", dates might be added (1723–1790 for one, 1930– for the other) to disambiguate them.

With an electronic catalogue, it is possible to make several improvements to a card catalogue system. The two most important of these are the aggregation of all information about an entity into a single record, and exact, direct linking

between records for different entities. The first means that a search for "Iris Wilkinson" would bring up a link to the record for "Robin Hyde" — there is no need to have the intermediate stage of going to a record for "Iris Wilkinson" only to be pointed at the record for "Robin Hyde". With all of her names on the single record, a search for any name will return the single record. The second means that on a record for "Inquiry into the nature and causes of the wealth of nations" will link directly to the record for the Adam Smith born in 1723 as the author — there is no need for the user to choose between the records for the different Adam Smiths.

The consequence of this is that the authorised form of an entity's name becomes much less important, since the same record will be found directly from a search on any of the names, authorised or not. Taking its place is the identifier for the entity's record — this must be a unique, persistent identifier for each entity that can be dereferenced in order to get the full details of that entity. This identifier *must not* be the same as or derived from any information in the record (such as the authorised form of a name); in such a case when the information changes (as it always does) either the identifier must change (thus breaking persistence) or it will be misleading. Identifiers must therefore not carry any information.

A good example of a catalogue that implements suitable identifiers is the Internet Movie Database (`http://www.imdb.com/`). So the identifier for the 1956 film "War and Peace" is tt0049934, with the record at `http://www.imdb.com/title/tt0049934/`, while the 1991 TV series of the same name is tt0371388, with the record at `http://www.imdb.com/-title/tt0371388/`. By exposing these identifiers (in their URL form), the entities are usable not only within the IMDb system, but also by others in their work (whether published on the web or not). That is, the work in disambiguating entities with the same name that IMDb have done is available to anyone — without regard to what name with which that outside user refers to the entity.

In both a card catalogue and a conventional electronic catalogue, the authority data is much the same, and there is little requirement that it be processed by a machine (beyond the level of being able to extract information of a particular type, such as date, name or note). Records are created manually and the links that should be made to other records (such as to the author of a work in that work's record) is determined by the person doing the cataloguing. In a full-text electronic collection, such as the NZETC's, a manual approach to linking the many instances of a name to an authority record is simply not feasible. Therefore the authority data must be amenable to automatic processing. Further, it is important to be able to link not only within one's own collection but also to resources which are held and catalogued in other systems.

## Requirements for an authority control system

From the preceding discussion it follows that an authority control system (that is, software which allows for the creation and use of authority control records) has two major requirements: it must expose (or be able to expose) its records publicly using a persistent identifier, and it must support full handling of the actual data that makes up an authority record.

There are, of course, a number of systems and schemas for handling authority

data. Unfortunately these all disappoint in various ways. Given that the Entity Authority Tool Set was developed only because there was no existing system that fulfilled the needs of the NZETC, it is worth going over these issues; their solution are the truest requirements of EATS.

The Metadata Authority Description Schema (MADS, `http://www.loc.gov/mads/`) was used at the NZETC prior to the development of EATS, and prompted the search for a better data model. In its earlier form it suffered from the idea, inherited from MARC, that the authorised form of a name should be the identifier for a record — a fundamental mistake that they have thankfully corrected since, though other problems remain. Non-authorised names may only be variants of the authorised name, and not variants of each other (the transliteration of a translation of the authorised form, for example). Further, while names from multiple authorities may be included, there is no way of linking a particular variant name to a particular authorised form, nor indeed of providing an identifier for the authority record the name belongs to (again a consequence of the MARC view that the authorised form *is* the identifier).

Encoded Archival Context (EAC, `http://www.iath.virginia.edu/eac/`) fares rather better than MADS in terms of coping with multiple authorities and relationships between names. Unfortunately it is biased towards content creators, rather than being generic enough to apply to any entity type. The NZETC currently tracks people (including animals), organisations, places, works, and ships; to this events and ideas will later be added. Even if it might be made to fit these entities, it is not a natural fit. It also has not moved out of draft form, though work is now underway to revamp the schema.

The Library of Congress Authorities database (`http://authorities.loc.gov/`) suffers from a couple of problems. While it provides a wealth of information in its area, it confines itself to the Latin script with diacritics, meaning that much of the world's literature and many of its people are represented using a transliteration of their names. Support and use of Unicode is a requirement for any modern system. Secondly, it fails to have persistent, non-meaningful identifiers. This makes linking to particular records a largely futile exercise. The OCLC Linked Authority File project (`http://alcme.oclc.org/eprintsUK/`) seems a step in the right direction for accessing the records in a structured way, but the service still has stability and display problems.

The Virtual International Authority File (`http://www.oclc.org/research/projects/viaf/default.htm`) research project uses various mechanisms to match records from multiple systems to each other, thus offering a linked federated search, but does not provide a referenceable point of access to the linked records. That is, there is no single record which encompasses the information of all of the relevant records, and so nothing that can be referenced (though each record links to the others, so navigation is not thwarted).

Beyond the data schema, there must be a system to actually create, edit, and make use of the stored information. None of the aforementioned projects have a (publicly available) interface except for searching. EATS is both a schema and a system; both parts deserve elucidation.

# The EATS Data Model

EATS models entities and authority information in a robust and generic fashion. There is no restriction on what types of entity may be recorded, and the core properties are not specific to any given type. All information about an entity is in the form of an authorised property assertion. An authorised property assertion links an authority record with a property (such as a name) and an entity. An authority record is a pointer to a record in an authority's system (such as a Dictionary of New Zealand Biography entry). It may optionally also include references to the sources used to justify the assertion.

So, for example, the NZETC asserts, in a record with the ID "name-208662", that a particular entity has the name "Katherine Mansfield". The DNZB asserts, in a record with the ID "M169" and the URL `http://www.dnzb.govt.nz/dnzb/default.asp?Find_Quick.asp?PersonEssay=3M42`, that this same entity lived from 1888 till 1923. Each assertion is of a single property, and multiple assertions about an entity may be made by the same authority (so that the DNZB also asserts that the entity's name is Katherine Mansfield).

There are a number of uses that this approach allows:

1. Linking collections together without requiring any change to the identifiers used in each collection. This is similar to the aims of the VIAF project mentioned earlier, with the difference that in EATS the equivalence of two records in different systems is not the product of an automated guess, but a definite statement within the EATS record itself.

2. Greater scope for disambiguation: not only can plenty of information be recorded within an EATS record, but the authority records and URL references provide pointers to further sources of information.

3. Access control based on association with a particular authority, so that members of one organisation might only have permission to modify the property assertions associated with one of their authority records.

For the most part the EATS model meshes with the Functional Requirements for Authority Records conceptual model (`http://www.ifla.org/VII/d4/wg-franar.htm`). The main differences are that EATS does not cover the rules an authority ("Agency" in FRAR) uses to create its names and identifiers, and that an access point is constrained to a URI.

The properties that EATS models are: existence, type, name, relationship between two names, relationship between two entities, note information, and URL reference (simply a pointer to a resource that pertains to the entity). There is also scope for user-defined properties. Each property may have one or more dates associated with it. The more important properties are described below.

## Existence

It might seem that the existence of an entity is implicit in the there being an entity to apply this property to. However, it makes sense in a multiple authority system, when an entity may be part of one organisation's collection, and not part of another's. Further, it provides a handle for dates, which may also vary by authority.

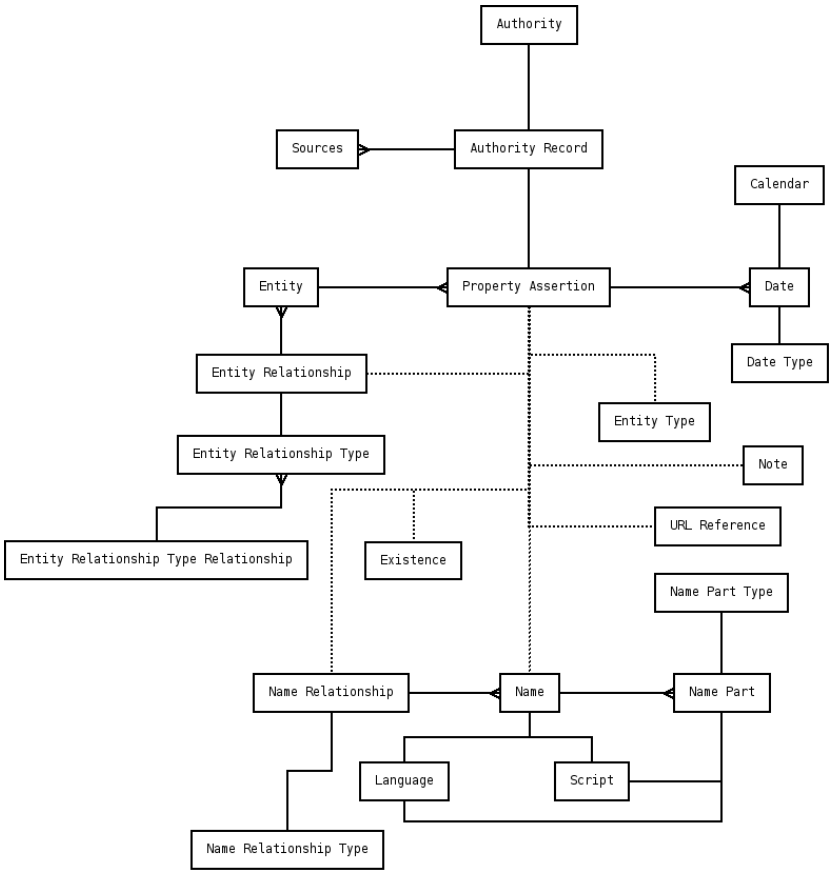


Figure 1: The EATS objects and basic relationships

## Type

While it may at first seem that the type of an entity is not a matter of debate (it is improbable that one authority will consider Katherine Mansfield a person, while another states that she is an organisation), it is entirely possible that different organisations will have different terms for the same entity type, or make finer-grained distinctions than others. In the case where a single EATS installation is used by multiple organisations, it is important to allow each to work using their own preferred terms.

## Name

As mentioned above, when dealing with a full-text collection in which names within the text are marked up as pointing to an authority record, some sort of automatic processing is required. In the case of personal names, this means being able to cope with the different forms a name may take. So, for example, "Katherine Wilson Sheppard", "K. Sheppard", and "Sheppard, Katherine W." must be recognised as being the same personal name. EATS therefore stores such names not as a whole, but as broken down into component parts, with rules for their correct reassembly. To complicate matters, these rules are language and script dependent, so that information is recorded also. For those cases where a personal name is too complicated to be automatically reassembled into a full form, a display form can be supplied.

## Relationship between two entities or names

Relationships between entities or names are important pieces of information. While EATS is not intended to be a store of biographical information, such information can be very useful for the purposes of identification and disambiguation.

For example, in the NZETC collection there are texts referencing six people called Te Heuheu, all from the same family. Some of these references are oblique, and the importance of even a sketchy whakapapa[1] cannot be overstated when determining which family member is meant.

## Note information

While every effort has been made to codify information within EATS in a structured, machine-processable way, there is always information which cannot be captured this way. This material goes into notes, whether as its own property associated with an entity, or as an addition to a date or name.

## User defined properties

Where structured information needs to be recorded that is not covered by an existing property, EATS offers a generic mechanism for user-defined properties. Naturally these do not have any associated special handling (as there is for the assembly of names from parts, for example), but it does provide a basic level of extensibility. It would be trivial, for example, to add a "sex" property to

[1] Whakapapa is a Māori word meaning, roughly, "family tree".

record a person's sex (and allow, as EAC does, for this property to have a date or dates associated with it, to accomodate any changes that might occur during an entity's existence).

### Dates

Dates in EATS are designed to cope with the common cases, allowing for date ranges, single dates, different calendars, circa and floruit, and confidence. There is support for terminus post and ante quem dates, but only as applying to start and end dates respectively. It is not, therefore, possible to state that the start date of a range occurred before some date, or that the end date of a range occurred after some date. A date may have a note associated with it to give more detail in the complex cases. Confidence is expressed as a binary property.

## Implementation

EATS is implemented as a web application, written in Python using the Django framework, with the data stored in a PostgreSQL relational database. The only access to the data is via the web application, which provides both a web API for getting and setting data and an admin interface for modifying records.

On the client side there are a number of tools for interacting with the EATS system. There is a search which operates over names, and a search for those entities which have a property asserted by a particular authority record. The most frequently used application is a Java program for looking up entities based on a name and creating TEI XML markup from the result — this is at the heart of the NZETC's markup of its texts to allow for the interlinking of those texts and the entity topic pages on the website. Another web tool supports the discovery and markup of multiple names within a text.

It must be said at this point that the screenshots given here are from a release of EATS which has an older form of the data model than that described above. In this model various pieces of data are not handled as property assertions, but as data intrinsically associated with an entity.

EATS does not currently have a complete serialisation format. There is the facility to export to and import from MADS, but given the limitations of that schema, it is not possible to export full record details from EATS into MADS.

There are a number of parts of the schema which are not yet exposed in the user interface: entity and name relationships, user-defined properties, and associating source material/evidence with assertions. Fortunately the addition of these elements requires only writing some bits of code — it does not involve touching the database schema or the data at all, and indeed the NZETC installation has name and entity relationship information in it, even though it is not currently editable through the standard interface.

EATS has been used in production at the NZETC since July 2007. It enables the Centre to manage entity identification information and link instances of names in the tens of thousands of pages of text that are added to the collection each year to rich authority records. Combined with the NZETC Topic Map online

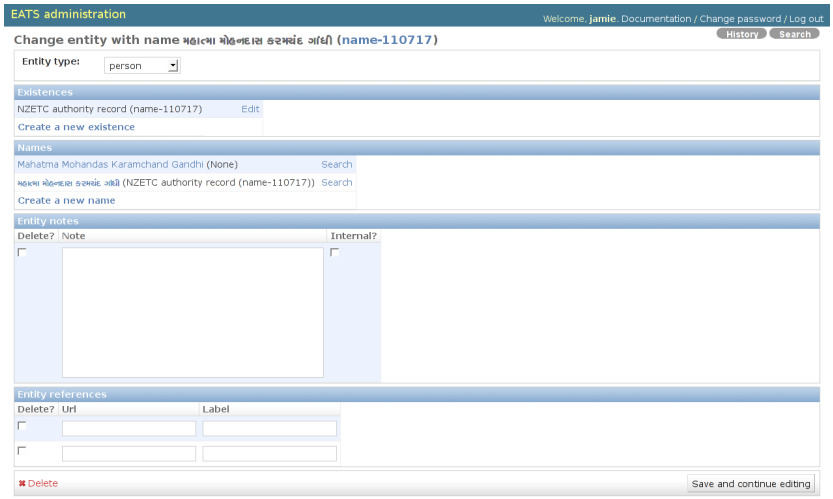


Figure 2: The main admin screen for editing an entity

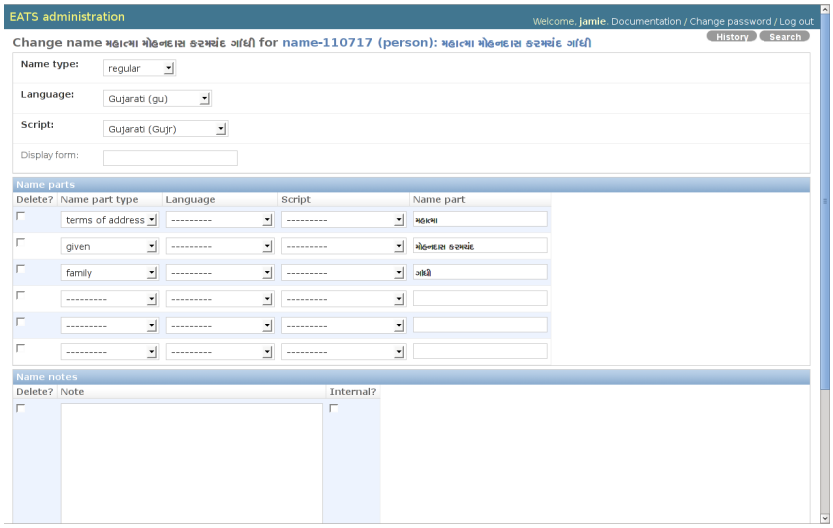


Figure 3: Screen for editing an entity's name. Name parts are used on personal names wherever possible, to aid in automatic matching. Unicode is used throughout.

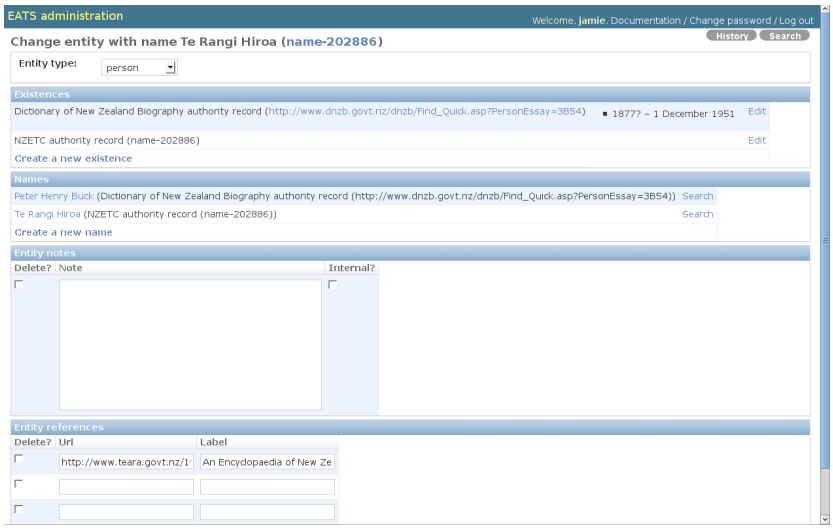


Figure 4: An entity may have multiple names authorised by different authorities.

delivery system, EATS allows the Centre to use identified entities as the core of a navigational framework providing access to resources in multiple, distributed collections. It is hoped that as more organisations recognise the value in using authority control systems suitable for online digital resources, more persistent, unique identifiers will be created and shared. This would greatly increase the opportunities for cross-collection searching and the linking of related resources producing a rich space for research and discovery.


Figure 5: An NZETC topic page creates links to resources from EATS data as well as texts.