## MCCRABB — READ

## <0 figures>

## READ—Entity Relationship Model

McCrabb, I.

This presentation will précis the underlying entity relationship model of the Research Environment for Ancient Documents (READ) and will outline some of the research methods enabled by the platform.

The READ project commenced in 2013 with development support from a consortium of universities (Ludwig Maximillian University, Munich, the University of Washington, Seattle, and the University of Lausanne), which, together with the University of Sydney, are involved in the study and publication of ancient Buddhist documents preserved in the Gāndhārī language that originate from Afghanistan and Pakistan. READ leverages 10 years of development on gandhari.org by Stefan Baums and Andrew Glass—the site that currently supports the Gāndhārī Dictionary, Bibliography and Catalog, as well as a comprehensive textual corpus. READ model and UI design are nearing completion, and development is on track for delivery in late 2015.

READ has been designed as a comprehensive multi‐user research workbench and publishing platform for ancient Sanskrit and Prakrit texts: manuscripts, inscriptions, coins, and other documents. READ is based on open-source software and built-to-open standards. It provides an extensible entity model, TEI support, and a published API for integration with related systems.

This presentation will focus on design aspects as encapsulated in the entity relationship model (ERM). An ERM is an abstract way of describing a relational database, most often represented as a flowchart accompanied by precise descriptions of each entity. This approach allows one to model the entities and their relationships and determine the most effective and flexible way of structuring the data to support authoring, storage, maintenance, analysis, reporting, and publishing.

The design approach has been to build a comprehensive set of entities, mapped to real-world objects, in order to seamlessly model both physical and textual domains. The underlying design principle is atomisation of data to its smallest indivisible components and the linking and sequencing of these entities.

As an illustration, in the physical realm, manuscripts or other inscribed Items have Parts, Fragments, and Surfaces. Images of these Surfaces are segmented to provide a fixed reference system, a BaseLine much like the grid laid out at an archaeological excavation. These Segments are then sequenced into Spans across a Surface. Surfaces can then be aligned and the Spans sequenced into complete Lines. The mapping between physical and textual is where Syllables are identified with *akṣara* (the graphical unit of the *kharoṣṭhī* script these texts are rendered in) Segments. Syllables (deconstructed into Graphemes) are sequenced into Tokens (words or compounds) that are ultimately sequenced up through textual divisions into full Editions.

The philological process model adopted is one of defining each entity by applying classifying metadata and progressively sequencing these entities from the smallest upwards. This approach allows for attribution and annotation at every entity to record scholarly contribution at the finest level of granularity. Different editors’ interpretations of *akṣara* Segments, Syllable assignation, and Token grammatical deconstruction are all attributed, and multiple versions of all entities exist in parallel to support the publishing of any number of alternative editions.

In summary the design approach rests on a small number of basic principles:

1. By designing entities and their relationships to reflect real-world objects in the domain, one can flexibly map system functions to actual philological processes.

2. By completely atomizing physical and textual objects, one can flexibly sequence these components into higher-level entities.

3. The application of metadata and progressive sequencing of entities from the smallest component upwards through scale supports flexibility and granularity.

The technology stack includes PostgreSQL with backend development in PHP and UI development using JQuery—all mainstream open-source development environments. A conventional software architecture has been adopted with

• A data storage system built on a relational database where entities are realized as tables with constraints and triggers to implement model logic and integrity.

• A storage abstraction layer implemented as server side libraries to expose entities and entity aggregates needed to support the web services.

• A variety of services including import, export, index, query, and data management. Services are being implemented as using Ajax for complex interactive UI functions.

• A JavaScript User interface using JQuery frameworks.

• A JavaScript API to provide access to libraries and core services to allow for system extensions and integration.

A great deal of work has been undertaken on the design and development of a comprehensive system ontology. The ontology provides standardized terms for both physical and textual domains—everything from object shapes, mediums, and types to grammatical categories such as declension and conjugation values. These standard terms manifest as constraints within the system and ensure data consistency and quality.

The system ontology has been designed to be extensible and configurable to allow READ to encompass alternative schema and new research objectives. Users can develop their own term sets within the ontology to address their own research questions. For example, my research encompasses the development of a metadata model for the analysis of formula structures in relic inscriptions, others wish to address syntactical structures, and still others questions of metre.

A practical example of the flexibility in the ontology is the support for languages other than Gāndhārī. READ has been designed from the ground up to support all languages that have Sanskrit as their primordial. Branching in the system ontology allowing for alternative grammatical deconstructions enables READ to be easily configured for Sanskrit or Pali corpora—indeed, any Prakrit language.

Fundamental to the design of READ is support for the import and export of TEI. The method adopted has been the development of services to export a complete XML rendition of stored data with XSLT transformation to EpiDoc TEI specifications. Import is the same process in reverse.

A highly sophisticated parser was developed to read the existing corpus of transcriptions from gandhari.org as text strings and shred these into READ database entities maintaining linkages. This parser can be finessed for other corpora that might currently only exist in unstructured formats like Word. Fundamental to READ’s design is support for import of existing datasets and export as XML and specifically TEI.

Indeed READ is positioned as a workbench for ancient documents, complementary to TEI-based textual repositories such as SARIT and integrated with existing Sanskrit dictionaries. Whatever format existing transcriptions were developed in, these can be consumed, elaborated upon, analyzed, and then published as research output in TEI or pdf. The data remains open-source and can be exported as a full XML archive.

In summary, READ has been designed to function as

• A linked repository of images, transcriptions, translations, metadata, commentary, and bibliographic records.

• A content management system encompassing multiuser editing, maintenance, and version control.

• A collaboration platform with comprehensive access and visibility control to support draft development, workgroup collaboration, and public presentation.

• A research workbench with access to a dictionary, corpus of texts, catalogs, glossaries, concordances, and bibliography.

• A publishing platform for individual transcription renditions or full scholarly editions, both print-ready and online.

• The kernel of an integrated research network interfacing with related dictionaries, repositories of parallel texts, GIS, data visualization, image rendering, and palaeographic analysis systems.

READ enables a very granular data structure, and whilst the level of elaboration is dependent upon the research requirement, deriving the most benefit entails an initial investment in the application of metadata at each level of sequenced entities. The dividend is that this enables a range of automated analysis outputs that support palaeographic, phonological, grammatical, orthographical, and morphological research in addition to the opportunities opened up for formulae and syntactical analysis.