FAIMS PROJECT

Creating eResearch tools for Archaeologists

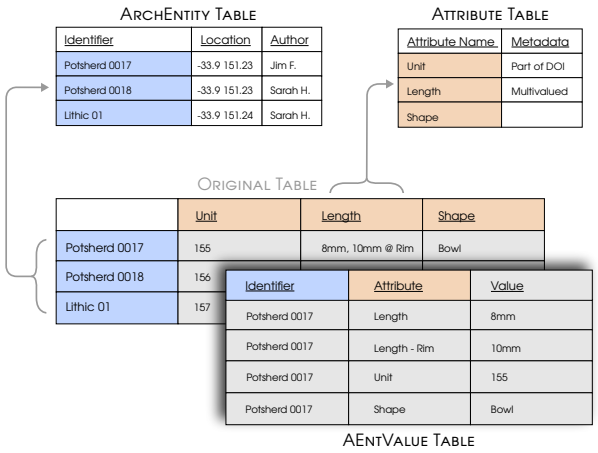
Relational and Nonrelational Program Independent Schemas:

Supporting arbitrary archaeological data models

1 Explored Approaches

We initially explored "noSQL" approaches to data storage (e.g., native XML, graph, protobuffers), but the need to support GIS functionality forced our hand with regard to choosing a database management system. Widely-used GIS software currently expects tabular or relational data. Considering the available options available for Android, we chose Spatialite (based on Sqlite) as our mobile device data store.

Spatialite is a relational database, but the arbitrary structure of the incoming data - with every user requiring a custom schema based on his or her own knowledge representation - presented a challenging problem in database design. We employ the SQL database as a key-value store, associating a series of "attributes" (as keys) with a specific ArchEnt (the atomic unit of archaeological practice discussed above). In order to implement a key-value store, we have designed our database to Domain-Key (Sixth) Normal form. In order to provide nuanced version control and multiple client updates of the same data without data loss, we have designed our database as an "append only store" - no record is ever deleted, it is merely superseded by newer entries that can be reviewed and "rolled back" if needed.



2 DKNF Implementation

The central data relationship in our database is comprised of four tables. However, it is best thought of as a single, virtual table. Imagine an Excel spreadsheet with the top row and the first column locked. The unique identifier for an Archaeological Entity resides In the first column, designating a row that will contain all of the information about that entity. The "ArchEntity" table acts as the first column, providing an identifier (along with identifier-specific metadata) to all information about that archaeological entity. The "AttributeKey" table provides a constrained set of attributes concerning the entity. It can be thought of as the frozen row at the top of the sheet with column names represing the things you want to record about an object. The "AEntValue" table represents the data held in the body of the table (each cell being an intersection of an "ArchEntity" row and the "AttributeKey" column). The relationship between these three tables provides a flexible data schema that does not need to be altered within the database to meet a wide range of use scenarios; only the values contained in the ArchEntity and AttributeKey tables need to be defined at project startup in order to define the types of archaeological entities being recorded along with all of their possible attributes. By loading columns into the attribute key table at project-start, no aspects of the database need to be altered and no fundamental database access logic needs to be altered, allowing significant customization in return for no investment of developer time.

The final supporting table, "Vocabulary" provides for the controlled vocabularies that will govern archaeological entities and their attributes. It would have been simpler to abstract this table into a separate Arch16n file, thereby tremendously simplifying the database. Unfortunately, the controlled vocabulary terms may change in the middle of a project. For example, in Bulgaria the

The University of New South Wales is proud to be in partnership with the National eResearch Collaboration Tools and Resources (NeCTAR) project to develop a comprehensive information system for archaeology, creating a central online platform for the capture, exchange, analysis and storage of archaeological data.

PARTNERS:

nectar UNSW AUSTRALIA

LA TROBE UNIVERSITY INTERSECT

THE UNIVERSITY OF SYDNEY VeRSI

CONTRIBUTORS:

Australian Heritage Management Service Pty Ltd nsw ⋆ Australian Cultural Heritage Management Pty Ltd vic ⋆ Archaeological Map of Bulgaria (bg) ⋆ University of California Berkeley (us) ⋆ University of Canberra ⋆ Flinders University ⋆ Godden Mackay Logan ⋆ Indiana University of South Bend (us) ⋆ James Cook University ⋆ University of Leicester (uk) ⋆ Macquarie University ⋆ Michigan State University (us) ⋆ University of New England ⋆ University of Queensland ⋆ Snappy Gum Heritage Services ⋆ Southern Cross University ⋆ University of Sydney ⋆ Vanderbilt University (us) ⋆ University of Western Australia

SUPPORTERS:

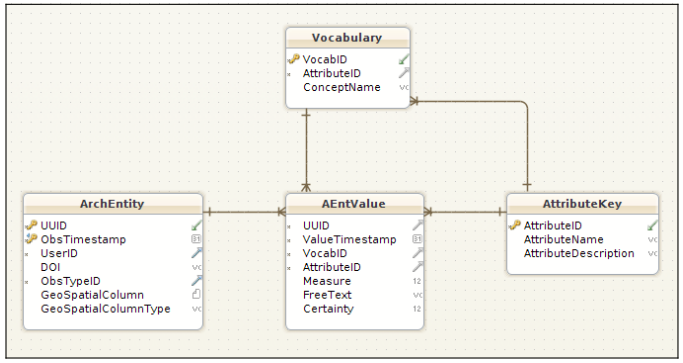
Archaeology Data Service ⋆ Digital Antiquity ⋆ University of Texas, Austin (us) ⋆ Australasian Institute for Maritime Archaeology ⋆ Australian Archaeological Association ⋆ Australasian Society for Historical Archaeology ⋆ ACT Heritage Project ⋆ ACT Heritage Unit ⋆ NSW Heritage Branch ⋆ NSW Office of Environment and Heritage ⋆ Monash University ⋆ University of Melbourne ⋆ University of Sydney Library ⋆ Earth Imprints Consulting

INFO@FEDARCH.ORG

WWW.FEDARCH.ORG

preferred transliteration from Cyrillic into Latin characters for sevaral towns and topynyms changed in the middle of our project, requiring spelling updates in our databases. Other examples might include the definition of new artefact types requiring updates of incorrect or temporary typological terms. Instead of filling the append only store with a record of such changes (as would be necessary if localisation is done programatically through a definitions file), a chance to the controlled vocabulary in a linked localisation table is a far more elegant solution to this problem.

The "AEntValue" table, where the data about archcaeological entities is stors, implements a trinary many to many relationship between the three other tables, is the primary key-less AEntValue. While the lack of a primary key is a violation of first normal form, tuples can be identified by the triple determinant of: UUID, Timestamp, and Attribute ID. The lack of a primary key allows us to provide for searchable "multi-valued" attributes: a set of the same measurements with the same UUID and Timestamp will all be displayed at the same time, naturally capturing multi-selects and other common multi-valued UI designs. Each value will also contain the ability to include a freetext annotation or hyperlink as well as a measure of certainty.



3 Arch16n

The Vocabualry table interfaces with the Arch16n definition document, which maps local vocabularies used by individual projects against common archaeological concepts. This localisation will use well established i18n techniques, looking for and replacing keyed strings, for example, finding "{StratigraphicUnit}" and replacing all instances found with "Context", in the user interface for that module. By replacing strings on display, we maintain a shared vocabulary independent of, but supporting, different terminologies. By moving concept mapping to the data creation phase of a project, we can produce compatible datasets without the problematic and time consuming ontology mapping at the time of ingest into a repository.

4 Append Only Datastore

As noted above, we have implemented an append only data store that incorporates full versioning. Updates and deletes in this database take a novel form. All updates are written as new records, with a timestamp. The "current" records are merely the set of all values with the latest timestamp for a given UUID and Attribute pair. Deletions work similarly. For example, to delete a record for an archaeological entity, a record is wrtten to the ArchEntity table with a new timestamp is and a"deleted" boolean set to true. The views (querries) we have designed for normal access to the database will simply disregard all deleted records - but the records will remain in the database. This approach makes viewing changes and recovering data trivial. The append-only datastore, a concept championed by Google with their protobuff-driven databases, makes unrecoverable errors very difficult to execute, as the complete transaction history of every significant change can be traced through every entity.