Management of Metadata in Linguistic Fieldwork: Experience from the ACLA Project

Baden Hughes[1], **David Penton**[1], **Steven Bird**[1], **Catherine Bow**[1], **Gillian Wigglesworth**[2], **Patrick McConvell**[3] **and Jane Simpson**[4]

[1]Department of Computer Science and Software Engineering, University of Melbourne
{badenh, djpenton, sb, cbow}@cs.mu.oz.au

[2]Department of Linguistics and Applied Linguistics, University of Melbourne
gillianw@unimelb.edu.au

[3]Australian Institute of Aboriginal and Torres Strait Islander Studies
patrick.mcconvell@aiatsis.gov.au

[4]Department of Linguistics, University of Sydney
jhs@mail.usyd.edu.au

Abstract

Many linguistic research projects collect large amounts of multimodal data in digital formats. Despite the plethora of data collection applications available, it is often difficult for researchers to identify and integrate applications which enable the management of collections of multimodal data in addition to facilitating the actual collection process itself. In research projects that involve substantial data analysis, data management becomes a critical issue. Whilst best practice recommendations in regard to data formats themselves are propagated through projects such as EMELD, HRELP and DOBES, there is little corresponding information available regarding best practice for field metadata management beyond the provision of standards by entities such as OLAC and IMDI. These general problems are further exacerbated in the context of multiple researchers in geographically-disparate or connectivity-challenged locations. We describe the design of a solution for a group of researchers collecting data on child language acquisition in Australian indigenous communities. We describe the context, identify pertinent issues, outline the mechanics of a solution, and finally report the implementation. In doing so, we provide an alternative model and an open source software application suite which aims to be sufficiently general that other research groups may consider adopting some or all of the infrastructure.

1. Introduction

In this paper, we report the development of a model for an integrated offline and online application which allows multiple geographically dispersed field linguists working on collaborative research projects to centrally collate, manage and query metadata pertaining to digital language resources.

The context for the solution described here is a collaborative research project entitled "Australian Child Language Acqusition" (ACLA) which is investigating the impact of exposure to multilingual input on child language development, using case studies from indigenous Australian languages. Additional avenues of enquiry include examining issues of language shift, language maintenance and language change resulting from a multilingual environment. The nature of this project is multi-phase and longitudinal - fieldworkers will collect data in different indigenous communities over a period of three years; recording children's informal interactions with other children and adults, using digital audio and digital video. Transcription will occur in parallel with collection, whilst a separate analytical phase will be undertaken.

In such a project there are a number of widely-applicable requirements which are derived from data management, research methodology and technology domains which need to be adequately addressed. Here we report progress towards a solution which addresses these requirements, whilst seeking to remain consistent with the principles of best practice promoted by other efforts such as EMELD[1], HRELP[2], and DOBES[3].

The predominant goal of the implementation is to allow the generation and query of structured metadata pertaining to language resources in both offline and online modes using a single interface. Subsidiary goals include ensuring the integrity and consistency of metadata, the longer term benefit being flexible analysis through structured and semi-structured queries. The solution is implemented as a distributed client-server application, which complements local data entry and query functions with remote synchronisation of an individual fieldworker's data with a central repository. Through the adoption of cross-platform, open-source technologies, it is hoped that other projects facing similar problems may be able to leverage the existing application and customise it to suit their own requirements.

2. Requirements

In this section we describe the context for linguistic metadata management within the ACLA project, and out-

[1]Electronic Metastructures for Endangered Language Documentation, http://www.emeld.org

[2]Hans Rausing Endangered Language Documentation Project, http://www.hrelp.org

[3]Documentation of Endangered Languages, http://www.mpi.nl/DOBES

line a number of requirements which a viable solution must address. These requirements are variously derived from the domains of data management, research methodology, and technology.

2.1. Data Management Requirements

There are a number of requirements which are exhibited from the domain of data management. First, the application must facilitate handling complex multimodal data - the metadata collected encompasses not only multimodal sources, but multimodal sources in which there are multiple participants who engage in multiple different interaction activities within a single session. (For a discussion of the multimodal complexity relevant to this particular project, see McConvell (2003)). Second, the application must also enable the capture of complex relational information regarding the informants and their wider social identity in terms of both sanguineal and sociological affiliations. Third, the data structures must allow clear sociolinguistic delineation between the various classes of participants, including informants, researchers and intermediaries such as indigenous assistants who form the conduit between the indigenous communities and the fieldworkers themselves. Fourth, the application needs to provide a management interface to complex human subjects data including publication permissions and accessibility information, to monitor compliance with ethical guidelines and cultural sensitivity restrictions. Finally, the application needs not only to manage metadata, but also to generate outputs such as media cataloguing codes, media file names and transcript file names which assist with the electronic and physical management of media and transcripts in both on and offsite locations.

2.2. Research Methodology Requirements

The ACLA project has identified CLAN[4] as the tool of choice for analysis. The close alignment of a research methodology with a particular analytical tool has a number of impacts, not least of which is the need for integration of metadata held in external systems with data held in the chosen analysis tool, but also for complementary analytical approaches which leverage both components efficiently. The data captured should not exclude particular approaches to multi-dimensional and semi-structured enquiry at a later point in time in either distributed or centralised modes. In essence the query model requires two complementary parts: first metadata is queried using the ACLA-DB (to find sessions exhibiting a particular phenomema within the entire data set), and second, the results from this first query feed the second query in CLAN (to find the occurrences of the phenomena within media and transcripts). A third requirement is support for "extensible controlled vocabularies", a requirement which at first may seem paradoxical. Within the project there is a centrally maintained list of activities which can be supplemented with local variants or derivatives. An example of this requirement is for focus child activities, where a centrally maintained list of activities needs to be supplemented by fieldworkers to suit local conditions. Finally, the application must support a range of user-defined lists, predominantly for tracking temporal aspects such as educational progress.

2.3. Technology Requirements

Within the domain of technology there are a range of different requirements which need to be addressed. First, data entry and querying needs to be fully supported in both online and offline modes, ideally through an identical interface. In the ACLA project, the locale of data collection is in remote indigenous communities which often lack basic telecommunications infrastructure. In this environment, any system which requires online access to provide even basic functionality would be rendered useless. Second, while the metadata collected needs to be able to be expressed in a project specific manner initially, this should not preclude the alternative expression of this data in compliance with broad metadata standards for language resources such as OLAC[5] and IMDI[6]; and archive specific standards such as CHILDES[7]. Third, the application instance must be able to be easily installed and used on multiple combinations of operating systems and hardware. Finally, whilst all metadata is stored in a relational database which uses an XML based published schema, it should be possible to export the data set in a structured XML format, allowing maximum flexibility in terms of approaches to querying the data itself.

3. The Data Model

Many of the issues raised in the preceeding sections impact on the design and implementation of ACLA-DB. In this section we specifically discuss the data model which underlies the application - rather than provide a schematic of the entire data model, we instead focus on several aspects which illustrate various complexities. The full data model is available online. (Software developers may note that DBDesigner[8], an open source, XML based tool with a high level of integration with a range of open source database engines, including MySQL[9], was used in this project.)

First, whilst it is common to conceive of a data model to encode the relations between a recording session, its media and relevant transcripts, this model provdes inadequate in the context of ACLA-DB. Within this particular project, a session may have multiple media and transcripts, juxtaposed with the possibility that there may be a number of sessions on a single media. In order to encode these relations, in the ACLA-DB model, a two-node hierarchy is used whereby a session can consist of one or more segments; each segment consists of one or more media and one or more transcripts. This allows the construction of relations between sessions, media and transcripts where session boundaries do not coincide with media boundaries.

Second, in the ACLA-DB data model there are many participants, yet only a smaller number are actually in focus

[4]CLAN, http://childes.psy.cmu.edu/clan/

[5]Open Language Archives Community, http://www.language-archives.org

[6]ISLE Metadata Initiative, http://www.mpi.nl/IMDI

[7]Child Language Data Exchange System, http://childes.psy.cmu.edu/

[8]DBDesigner, http://www.fabforce.net/dbdesigner4

[9]MySQL Database Engine, http://www.mysql.com

in terms of analysis (being named "focus children" for this very reason). Whilst it could be argued from a scientific methodology perspective that it would be ideal to have a single focus child per session, it is common for number of focus children appear in a single session, with only a single individual actually being the locus of the particular session.

A third challenge in terms of data structure is that it is not uncommon for members of indigenous communities to have incomplete knowledge of their date of birth. This contrasts with the widely accepted belief in language acquisition research that there are distinct age brackets during which different influences come to bear on the development of language in children. In the ACLA project we need to support the entry of possibility inaccurate DOB entries eg where day, month or year of birth is approximate rather than canonical. This effectively renders inbuilt data types such as date inadequate, requiring a project-specific solution.

Fourth, one of the key challenges in the codification of educational standards in Australian indigenous communities is that there is widespread non-linear progression through an educational system. The data model therefore requires a flexible approach to the tracking of non-linear progressions with a temporal extent - this is required for other fields such as participant location as well.

Finally, a significant complexity which affects the development of the data model is the need to encode both sanginueal and sociological kin relations at number of levels of granularity. The database must store kin designators (a extensible, but controlled vocabulary), reciprocal relationships and a high-level description of the relationships between participants and focus children.

4. Implementation

The solution entails building network connectivity-independent interfaces for data entry, and integrating the data created using offline and online modes. This strategy facilitates synchronising multiple offline data sources in the field and allowing the fieldworkers to document the collection process without necessarily having network access. This solution also allows flexibility in the later stages of project where other researchers can consult the full database simultaneously from multiple locations. We discuss the architecture of the solution; the underlying technology used; and the structure of the application.

4.1. Architecture

The architecture of the ACLA-DB application is client-server, but with some notable variants from the standard approach. The client application instance consists of a database engine, a scripting language, and a web server. The server application instance is identical, although it is installed onto a central server rather than a workstation or a laptop. All data entry occurs on the client instance, and query functions are also supported. Data can be synchronised with the server once an internet connection is established, and by this method each fieldworker's data is propagated to the server, and thence to other fieldworkers.

4.2. Technology Platform

The current release of ACLA-DB is based on open source software; notably the Apache web server[10], the PHP scripting language[11], components from PHP Extension and Application Repository (PEAR)[12] and the MySQL database engine. The initial deployment context for ACLA-DB is on Mac OS X, although courtesy of cross-platform heritage of the technologies used, the same application will also run on Windows and Unix variants (including Linux and Solaris).

4.3. Application Functions

The application has a number of functions, in this section we will briefly discuss these based on the order in which they are presented through the GUI - data entry forms; reports, queries and searches; exports; synchronisation; and administration.

4.3.1. Data Entry Forms

There are two main forms in ACLA-DB, those for Participant, and those for Session, both having integrated view and edit functions. The participant forms allow entry of data pertaining to the individual participants (or informants) who appear within the session. The session forms allow entry of data pertaining to the fieldwork session which is recorded on digital media. While the actual fields are derived from the project itself, there is a high degree of affinity between the data collected through these interfaces and that required to follow best practice for metadata creation described in Bird and Simons (2003a, 2003b).

4.3.2. Reports, Queries and Searches

All search and query functions display results ranked by relevance, and allow direct linking to the relevant forms for either editing or viewing (dependent on privileges of the end user). There are three components of the Search and Query function, namely Simple Reports, Advanced Reports and Full Text Query. Simple Reports are in essence a series of pre-composed queries executed on demand with the results filtered automatically by user. Reports in this category are typically administrative eg "Display List of Participants by Fieldworker"; or "Display Fieldworker Activity During Timeframe". Advanced Reports address the challenge of how to enable very powerful queries to be composed on demand by the end user, whilst retaining simplicity of user interface. (For example, user defined queries are possible using raw SQL syntax and data structures, yet such environments are not accessible to end users with lower technical literacy.) The solution adopted in this context is to provide a simple query design interface which allows a user to define a query based on prepopulated lists of database table and column names, with support for sorting preferences, and then to save this query for later re-use. The Full Text Search is a fine-grained database search function executed against an index of the entire database. It supports standard

[10] Apache HTTP Server, http://http.apache.org

[11] PHP Hypertext Preprocessor, http://www.php.net

[12] PHP Extension and Application Repository, http://pear.php.net/

AND (+), NOT (-) and literals ("x"), features common with many web search engines.

4.3.3. Exports

There are three primary functions included in the Exports category - first to generate file headers for the CLAN analysis tool, second to generate physical media labels and third to generate transcript file names. Generating CLAN headers involves selecting the relevant session for transcription, selecting all of the participants in the particular session of interest and exporting the participant codes into a new transcript file or as window for cut-and-paste. This function effectively generates the '@participants' line in the CLAN transcript header. Second, generating media labels involves selecting the session in focus, and combining the relevant metadata pertinent to the media artifact into a label for physical media according to a project-defined schema. Third, generating transcript file names requires a selection of the session in focus, and combining the relevant metadata pertinent to the transcript artifact into a label for file names. The latter two functions are closely linked by virtue of the Session-Segment-Media-Transcript construct discussed earlier.

4.3.4. Synchronisation

The function of synchronisation between a local client instance of the database and the central server is conducted in two different operations, although the process is symmetrical. For client to server synchronisation, a local SQL query identifies all relevant changes which have been made since the last synchronisation request and the results serialised as XML, compressed, a checksum is generated, and the data transferred to the server over an authenticated HTTP session. Once the data arrives at the server, the checksum is verified, the data uncompressed, serialised from XML to SQL statements, and imported into the server database instance. Copies of the serialised XML transactions are kept on the client and the server as transaction logs. In the mode of server to client synchronisation, an SQL query is executed on the server, identifying all relevant changes which have been uploaded by other users since the last synchronisation request, from which point the process is identical to the client to server synchronisation method described above.

4.3.5. Administration

Administration functions include the ability of authorised users to edit system data, including the extensible controlled vocabularies for languages, geographical locations, focus child activities, and various access rights for other categories of user.

5. Related Work

The distributed client-server architecture utilised by ACLA-DB is not new, but rather the application instance joins a small number of other linguistic applications which take advantage of such technology. In particular, ACLA-DB has significant affinity with Kepler (Maly et al, 2001), an OAI[13]-oriented metadata creation and management tool which has been adapted for use within the language archives community, but adopts a pull, rather than push, model to metadata publishing. Other tools such as the IMDI tools, adopt similar architectures but vary in terms of their implementation approach, most notably requiring a manual synchronisation after which an individual's metadata is published centrally. ACLA-DB contrasts with tools produced by EMELD (eg FIELD[14] and LinguistList (eg ORE[15]) by virtue of its offline mode of operation.

6. Conclusion

In this paper we have reported the design and development of ACLA-DB, a management tool for field linguistic metadata, which supports the principled creation of metadata at the point of language resource creation. The ACLA-DB application provides full support for a distributed mode of operation including synchronisation of data between fieldworkers via a client-server architecture. The model proposed may provide other similar projects with a sufficiently general, and hence adaptable, methodology which would be of benefit, especially in light of the increasing requirements to collect metadata at the point of language resource creation in field research contexts. In reporting ACLA-DB, we hope to provide an alternative model based on open source software which is sufficiently general that other research groups may consider adopting some or all of the infrastructure.

(Additional resources, including the software described in this paper are available from http://www.cs.mu.oz.au/research/lt/projects/acla-db).

7. References

Steven Bird and Gary Simons, 2003a. Seven Dimensions of Portability for Language Documentation and Description. Language 79: pp 557-582

Steven Bird and Gary Simons, 2003b. Extending Dublin Core Metadata to support the description and discovery of language resources. Computing and the Humanities 37: pp 375-388.

Kurt Maly, Mohammad Zubair and Xiaoming Liu, 2001. Kepler: An OAI Data/Service Provider for the Individual. D-Lib Magazine 7(4), April 2001.

Patrick McConvell, 2003. Multilingual Multiperson Multimedia: Linking Audio, Video and Transcription for Analysis and Archives. In Proceedings of the PARADISEC Digital Audio Archiving Workshop, 2003. Sydney, Australia. 30 September - 1 October, 2003. [http://www.arts.usyd.edu.au/departs/rihss/daa.html]

8. Acknowledgements

The research described in this paper has been supported by the Australian Research Council Discovery Project Grant DP0343189.

The authors wish to acknowledge the contributions of Felicity Meakins and Samantha Disbray (U.Melbourne), and Karin Moses (Latrobe U.), who form the initial user group for this application.

[13] Open Archives Initiative, http://www.open-archives.org

[14] Field Input Environment for Linguistic Data, http://cf.linguistlist.org/cfdocs/emeld/tools/field/beta

[15] http://victoria.linguistlist.org/php4/ore-new/login.php4