

UNITING SOFTWARE TOOLS FOR THE ARCHIVAL,
MANAGEMENT AND ANALYSIS OF LINGUISTIC DATA:
LESSONS FROM DEVELOPING THE LANGUAGE DATA
REPOSITORY

A Senior Honors Thesis

By

MICHAEL NEAL AUDENAERT

Submitted to the Office of Honors Programs
& Academic Scholarships
Texas A&M University
In partial fulfillment of the requirements of the

UNIVERSITY UNDERGRADUATE
RESEARCH FELLOWS

April 2001

Group: Computer Science

UNITING SOFTWARE TOOLS FOR THE ARCHIVAL,
MANAGEMENT AND ANALYSIS OF LINGUISTIC DATA:
LESSONS FROM DEVELOPING THE LANGUAGE DATA
REPOSITORY

A Senior Honors Thesis

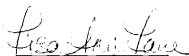
By

MICHAEL NEAL AUDENAERT

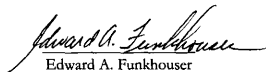
Submitted to the Office of Honors Programs
& Academic Scholarships
Texas A&M University
In partial fulfillment of the requirements
For the Designation of

UNIVERSITY UNDERGRADUATE
RESEARCH FELLOW

Approved as to style and content by:



Lisa Ann Lane
(Fellows Advisor)



Edward A. Funkhouser
(Executive Director)

April 2001

Group: Computer Science

1. ABSTRACT

Uniting Software Tools for the Archival,
Management and Analysis of Linguistic
Data: Lessons from Developing the
Language Data Repository. (April 2001)

Michael Neal Audenaert
Department of Computer Science
Texas A&M University

Fellows Advisor:
Dr. Lisa Ann Lane, Department of English

The complex tasks involved in both the production and use of linguistic data tend to be highly repetitive and tedious. These are the tasks at which computers excel, and at which humans perform very poorly. In this thesis I will begin by describing the current state of the art in computational support for the documentation and description of language. To do this I will outline the three major areas in which computers are currently being used to support linguistic research: data archiving, data analysis and data management, discuss some of the current projects and tools in each of these areas, and present a summary of work currently being conducted by the Open Language Archive Community to support open web based access to the resources available to the linguistic community. Following this I will describe some of the limitations presented by the current approach to this field and present a vision for system that will support analysis, management and archival of linguistic data in a single “universally” accessible system, providing specific examples from the system I have been working to develop, the Language Data Repository (LDR). I will conclude by laying out areas of future work, both for the development of the LDR and for the community in general.

2. TABLE OF CONTENTS

TITLE PAGE	i
APPROVAL PAGE	ii
1. Abstract	iii
2. Table of Contents	iv
3. Introduction	5
4. Overview of Requirements and Current Resources	8
4.1. DATA ARCHIVAL	8
4.1.1. <i>Large Archives vs. Personal Collections</i>	8
4.1.2. <i>Moving Away from the Publication Paradigm</i>	10
4.2. DATA MANAGEMENT	12
4.3. DATA ANALYSIS	15
5. Designing a Unified Approach: The LDR System	17
5.1. SEPARATING USER FUNCTIONALITY	18
5.2. SEPARATING DATA STORAGE	19
5.3. EXTENDING THE DATA MODEL	20
6. The System in Use: The Story of Bob	22
7. Current Progress and Future Work	24
7.1. FUTURE WORK	25
8. Conclusions	27
9. References	29
10. Vita	30

3. INTRODUCTION

The¹ current state of software tools that support linguistic research leaves much to be desired. The Language Data Repository is intended to provide a framework that more adequately meets the needs of the linguistic community by providing a general-purpose computational system for archiving and analyzing linguistic data. A short list of core requirements for such a system have served as the primary catalyst behind the development of the concept for the LDR.

- The system must provide for the reliable archiving of data collected from any spoken human language and facilitate access to data stored in multiple physical locations.
- The system must support management of data by multiple users, working together and independently, located both centrally and disparately while accurately attributing the work of the individual researchers who collected and catalogued the data.
- The system must allow for analysis from many different theoretical perspectives.

These requirements break the task of developing tools for linguistic data into three primary domains: data archiving, data management and data analysis. Current efforts in developing software for linguists focus primarily on a single domain and produce tools that most often have very little potential for integration with tools or data formats developed by other projects or of being extended to meet new needs of the community as they arise. It is the goal of the LDR project to provide a single system that integrates solutions to each of these domains and that can be easily extended to meet the wide variety of needs of the linguistic community. The approach that I have taken in

¹This thesis follows the style and format of *Language*

developing this system is significantly different than the current approaches being taken to develop tools to support linguistic research.

The LDR project is working to develop a system that will provide a unified solution to all three of these domains. It approaches this task by starting with a core architecture that supports a set of functionality that is common to all archival, management and analysis tasks regardless of the specific data involved. This architecture can be extended and specialized by independent developers to meet the specific needs of the community. The core architecture and independently developed extension combine to form the LDR system. This system will provide a set of reliable data management tools to help individual researchers manage personal collections of data. These data collections will be stored in Internet accessible repositories that the researcher can manage to grant or restrict access to the data that s/he has collected. These Internet accessible repositories will provide computational support to facilitate sharing information and collaborating on research in ways not currently possible. Tools to support the analysis of data stored in these repositories can be plugged-into the system to aid the researcher in the task of linguistic analysis. This unified approach to these three domains will result in a system that provides significantly more functionality than do existing systems and more adequately addresses the needs of the linguistic community. Moreover the ability to easily extend the LDR system will allow it to keep pace with changes in theoretical and analytic approaches to linguistic research and documentation.

This paper will present an overview of the general needs of the linguistic community, discuss the approaches currently being taken in developing tools to support that community and highlight some of the design features of the LDR system intended to improve on the shortcomings of the current work. The section "Designing a Unified Approach" describes the concept of the LDR system by examining how various elements of the system's functionality have been separated from the core architecture to allow for the extensibility of the system. "The Story of Bob" examines a potential

scenario in which the system might be used in the field. The paper concludes by presenting the current status of system development and some of the work that will be needed in the future.

UNITING SOFTWARE TOOLS FOR THE ARCHIVAL,
MANAGEMENT AND ANALYSIS OF LINGUISTIC DATA:
LESSONS FROM DEVELOPING THE LANGUAGE DATA
REPOSITORY

A Senior Honors Thesis

By

MICHAEL NEAL AUDENAERT

Submitted to the Office of Honors Programs
& Academic Scholarships
Texas A&M University
In partial fulfillment of the requirements of the

UNIVERSITY UNDERGRADUATE
RESEARCH FELLOWS

April 2001

Group: Computer Science

UNITING SOFTWARE TOOLS FOR THE ARCHIVAL,
MANAGEMENT AND ANALYSIS OF LINGUISTIC DATA:
LESSONS FROM DEVELOPING THE LANGUAGE DATA
REPOSITORY

A Senior Honors Thesis

By

MICHAEL NEAL AUDENAERT

Submitted to the Office of Honors Programs
& Academic Scholarships
Texas A&M University
In partial fulfillment of the requirements
For the Designation of

UNIVERSITY UNDERGRADUATE
RESEARCH FELLOW

Approved as to style and content by:

Lisa Ann Lane
(Fellows Advisor)

Edward A. Funkhouser
(Executive Director)

April 2001

Group: Computer Science

1. ABSTRACT

Uniting Software Tools for the Archival,
Management and Analysis of Linguistic
Data: Lessons from Developing the
Language Data Repository. (April 2001)

Michael Neal Audenaert
Department of Computer Science
Texas A&M University

Fellows Advisor:
Dr. Lisa Ann Lane, Department of English

The complex tasks involved in both the production and use of linguistic data tend to be highly repetitive and tedious. These are the tasks at which computers excel, and at which humans perform very poorly. In this thesis I will begin by describing the current state of the art in computational support for the documentation and description of language. To do this I will outline the three major areas in which computers are currently being used to support linguistic research: data archiving, data analysis and data management, discuss some of the current projects and tools in each of these areas, and present a summary of work currently being conducted by the Open Language Archive Community to support open web based access to the resources available to the linguistic community. Following this I will describe some of the limitations presented by the current approach to this field and present a vision for system that will support analysis, management and archival of linguistic data in a single “universally” accessible system, providing specific examples from the system I have been working to develop, the Language Data Repository (LDR). I will conclude by laying out areas of future work, both for the development of the LDR and for the community in general.

2. TABLE OF CONTENTS

TITLE PAGE	i
APPROVAL PAGE	ii
1. Abstract	iii
2. Table of Contents	iv
3. Introduction	5
4. Overview of Requirements and Current Resources	8
4.1. DATA ARCHIVAL	8
4.1.1. <i>Large Archives vs. Personal Collections</i>	8
4.1.2. <i>Moving Away from the Publication Paradigm</i>	10
4.2. DATA MANAGEMENT	12
4.3. DATA ANALYSIS	15
5. Designing a Unified Approach: The LDR System	17
5.1. SEPARATING USER FUNCTIONALITY	18
5.2. SEPARATING DATA STORAGE	19
5.3. EXTENDING THE DATA MODEL	20
6. The System in Use: The Story of Bob	22
7. Current Progress and Future Work	24
7.1. FUTURE WORK	25
8. Conclusions	27
9. References	29
10. Vita	30

3. INTRODUCTION

The¹ current state of software tools that support linguistic research leaves much to be desired. The Language Data Repository is intended to provide a framework that more adequately meets the needs of the linguistic community by providing a general-purpose computational system for archiving and analyzing linguistic data. A short list of core requirements for such a system have served as the primary catalyst behind the development of the concept for the LDR.

- The system must provide for the reliable archiving of data collected from any spoken human language and facilitate access to data stored in multiple physical locations.
- The system must support management of data by multiple users, working together and independently, located both centrally and disparately while accurately attributing the work of the individual researchers who collected and catalogued the data.
- The system must allow for analysis from many different theoretical perspectives.

These requirements break the task of developing tools for linguistic data into three primary domains: data archiving, data management and data analysis. Current efforts in developing software for linguists focus primarily on a single domain and produce tools that most often have very little potential for integration with tools or data formats developed by other projects or of being extended to meet new needs of the community as they arise. It is the goal of the LDR project to provide a single system that integrates solutions to each of these domains and that can be easily extended to meet the wide variety of needs of the linguistic community. The approach that I have taken in

¹ This thesis follows the style and format of *Language*

developing this system is significantly different than the current approaches being taken to develop tools to support linguistic research.

The LDR project is working to develop a system that will provide a unified solution to all three of these domains. It approaches this task by starting with a core architecture that supports a set of functionality that is common to all archival, management and analysis tasks regardless of the specific data involved. This architecture can be extended and specialized by independent developers to meet the specific needs of the community. The core architecture and independently developed extension combine to form the LDR system. This system will provide a set of reliable data management tools to help individual researchers manage personal collections of data. These data collections will be stored in Internet accessible repositories that the researcher can manage to grant or restrict access to the data that s/he has collected. These Internet accessible repositories will provide computational support to facilitate sharing information and collaborating on research in ways not currently possible. Tools to support the analysis of data stored in these repositories can be plugged-into the system to aid the researcher in the task of linguistic analysis. This unified approach to these three domains will result in a system that provides significantly more functionality than do existing systems and more adequately addresses the needs of the linguistic community. Moreover the ability to easily extend the LDR system will allow it to keep pace with changes in theoretical and analytic approaches to linguistic research and documentation.

This paper will present an overview of the general needs of the linguistic community, discuss the approaches currently being taken in developing tools to support that community and highlight some of the design features of the LDR system intended to improve on the shortcomings of the current work. The section "Designing a Unified Approach" describes the concept of the LDR system by examining how various elements of the system's functionality have been separated from the core architecture to allow for the extensibility of the system. "The Story of Bob" examines a potential

scenario in which the system might be used in the field. The paper concludes by presenting the current status of system development and some of the work that will be needed in the future.

4. OVERVIEW OF REQUIREMENTS AND CURRENT RESOURCES

This section presents an overview of the current state of the archival, management and analysis domains, discusses the shortcomings of current approaches, and outlines the approach taken by the LDR system to improve on the current approach.

4.1. Data Archival

To work with data, those data must first be collected and stored somewhere. Currently, there are many projects working to develop digital archives of data that range in size from a few megabytes to terabytes. There are many, many more archives that are not now and likely never will be digitized. These archives are used for everything from developing empirical methods for natural language processing to serving as sources of data for detailed theoretical analysis. The ability to interact with these archives (both digital and non-digital through the use of digital proxies) is one of the most critical issues in developing software to support the linguistic community. This section discusses some of the ways in which the general directions currently being taken by data archivists seems to be neglecting significant areas of data archival. Two problematic areas are: (1) emphasis on large archives as the primary repositories for linguistic data to the neglect of smaller personal archives and (2) the use of the publication paradigm as the primary model for thinking about archives.

4.1.1. Large Archives vs. Personal Collections

The archival projects enumerated by Simmons and Bird (2000) in their "Survey of the State of the Art in Digital Language Documentation and Description" are, without exception, organization-wide projects working to build large repositories, typically of all languages spoken on a continent or in a region. The movement toward building digital archives of data and making those archives web-accessible seems to be directed almost exclusively toward these large programs. Such projects are conducting useful work in the effort to provide documentation about the languages of the world, but the

nearly exclusive emphasis on organizational projects neglects the potential impact of that personal data collections can have. Vast collections of data sit unnoticed in the closets of thousands of linguists scattered around the world. Granted, these personal collection may vary in quality, but much of the data are likely useful, even with different methodological approaches having been adopted. By providing a system that supports the development of Internet accessible archives of personal data collections, the countless hours invested by many linguists may find far more wide-reaching applications than the original scope of research conducted by the linguist who collected the data.

Making these personal collections publicly available has at least two significant benefits for the linguistic community at large. First, it allows for documentation of focused, special interest issues. Such issues are common to linguistic research (e.g. Wennerstrom's (2001) analysis of the role of prosody in English discourse, Zantella's (1997) study on second language acquisition Puerto Rican immigrants in New York city or Mendoza-Denton's (1997) work in socio-cultural identification among teen-age Latino girls in northern California). The artifacts of such focused research may prove to be interesting in ways never imagined by the researcher and the potential variety of topics ensures that no single project will ever span the entire scope of such issues. Large archival projects, as useful as they are, cannot meet all of the linguistic community's needs for archived language data. Second, much work in documenting endangered and minority languages is done by individual linguists who are not affiliated with major documentation and description projects. The work these individuals have put into preserving our collective linguistic heritage is invaluable, yet much of that is inaccessible to the linguistic community. A system designed to promote archival of individual collections of data would do much to help preserve the work of these researchers.

The LDR system was originally intended to serve primarily as a tool to help individual linguists build and share repositories of personal data collections. It quickly

became apparent that major archival projects would need to be accounted for in the system design. While the LDR system can be easily extended to support large archives (see section 5.2), it remains very much committed to its original purpose of providing archival solutions for personal data collections.

4.1.2. Moving Away from the Publication Paradigm

Much of the work currently being conducted in language archival community comes from the publication paradigm of storing and sharing information. In developing and refining the concept for the LDR system, I have become convinced that this paradigm does not adequately address the needs of the linguistic community because it treats data in ways that are fundamentally different from the way in which those data were originally obtained and are likely to be used.

The development of the LDR system approaches the task of building linguistic archives from a paradigm that more closely resembles the way linguists interact with data on a day to day basis, and that can be extended to provide unified access to many independently developed repositories. I have defined the publication paradigm as:

An approach to disseminating information where that information is given to some individual (a publisher) who edits, formats, and records it onto some media (e.g. books, journal, publications, videos, CDs) that is reproduced and distributed to “clients” who are interested in this information.

This paradigm can be seen in both physical and digital realms. In the physical realm, the publication of linguistic data ranges from grammars and lexicons to journal articles to audio recordings of ethno-musical traditions and video of traditional ceremonies. These physical publications have parallels in the digital realm that use a variety of formats and distribution methods, for example web based publishing (e.g. Speech Accent Archives (Weinberger 2000), Linguistic Atlas Projects) and distribution of CD-ROMs (e.g. SILs Ethnologue, Linguistic Data Consortium). These digital publication often (but not always) offer a significantly higher level of user interaction

than their complementary physical publications, but they remain very much within the publication paradigm.

The need for a different paradigm for archiving arises from the fact that the publication paradigm does not accurately reflect the way linguists interact with data on a day to day basis. In the physical realm, linguists use index cards, loose leaf paper and audio and video recordings (typically scattered over a desk or shoved into a shoebox). These are not the finished results of hours upon hours of work that we see in the publication model, yet it is this information and the process of changes and revisions to this information, not just the end publication that need to be represented in the digital realm. And it is this data, to the extent that ethical and political constraints allow, that needs to be made available to the linguistic community. To adequately reflect the way in which data are used, a new paradigm is needed. I have called this the artifact based paradigm which is defined to be:

An approach to disseminating information where the artifacts of the research process (transcriptions, audio/video, field notes, sketches, etc) are made available in their "unedited" form to "clients" who are interested in this information.

This is the most common method for sharing information in the physical realm (at least among members of a research team) and exemplified by members rummaging through each other's file cabinets, perusing boxes of audio tapes and casually discussing various artifacts. Despite the commonality of this paradigm in the physical realm, there is no analog for it in the digital realm (at least for linguistic data).

There are a number of significant difficulties in translating this paradigm from the physical realm into the digital realm, especially a Internet accessible digital realm, and there are no tools currently available to the linguistic community to support this paradigm. There is the challenge of digitizing artifacts. Some of these artifacts can be relatively easily digitized (e.g. audio/video, transcripts, sketches). Others, such as arrows, plants, totem poles and casual conversations, are not so easy to digitize. There

are also ethical challenges. While some individuals may need (and have a right to) access to all information that is archived, only in the rarest of cases will it be possible to grant full access to the unedited artifacts. It must be possible then to restrict access to the archived data. The responsibility for this falls on the shoulders of both the linguist building the archive and the developers of the system being used for the archive. The linguist must recognize the ethical issues involved in storing his or her data in a semi-public archive and must take the appropriate steps to restrict access to sensitive information. S/he will need tools to make this process as easy as possible. The developers of the system are responsible for making it possible to restrict access to data and for ensuring that access restrictions are reliably enforced across the system. These ethical issues must be addressed both by the linguistic community developing ethical standards and by the software development community developing software that supports those standards. Due to these and other difficulties, tools that support an artifact based paradigm for archiving and disseminating linguistic data are not currently available.

The LDR system is being developed to support an artifact based paradigm. This represents a fundamentally different approach to archiving linguistic data than is taken by other systems and will provide significantly better support for future work in developing tools to facilitate collaborative work.

4.2. Data Management

The challenge of working with even small collections of data can be overwhelming. Organizing data collections and finding information stored in those collections is difficult under the best of circumstances. For collections for which digital representation of data (or at least digital cataloguing of data) is possible, software that supports data management has the potential of reducing the difficulties associated with data management. This software aids the linguist by providing two main services: storage and query.

The storage services free the user from managing the details of how data is stored in the file system. This differs from archiving in that the storage service of a data management system is the logical abstraction provided for the user that frees him or her from the direct manipulation of the archiving mechanisms. An example of this feature of data management services can be seen in Netscape's e-mail client. It allows users to manage large collections of email messages and to organize those message into various folders. The details of how e-mail messages are stored on the file system are completely hidden from the user. The e-mail client also supports a variety of services specific to e-mail such as replying to a message, forwarding a message, entering information about a sender of a message into an address book, etc. This provides the user with a much simpler, cleaner and more powerful solution than does saving each e-mail as a text file.

The query services provide the user with the ability to rapidly search or browse the data being managed by the system. An example of this feature of data management systems is demonstrated by SIL's Ethnologue. The Ethnologue provides a data management solution for descriptions of all of the world's living or recently dead languages (i.e. not ancient Greek, Latin etc.). This system manages a large amount of information and presents it to the user in an understandable fashion.

The tasks associated with storing and searching through data are exceptionally time consuming and tedious without the support of a computer and they are tasks that computers are particularly well suited to deal with. Despite the potential advantages of data management systems, software systems that attempt to provide the linguist with such tools are few and far between. Those that do exist are typically tightly coupled with the archival system they use to store data. It is difficult or impossible to use these systems for data that do not fit the data model developed by the creator of the system. For example, Netscape cannot be extended (at least not by someone outside of the Netscape development team) to support a system that manages chat client messages or a catalogue of physical mail in the same system that it manages e-mail messages. Chat

messages simply are not part of the data model that Netscape was designed to handle. This is fine (though perhaps annoying) for an e-mail client, but a system designed to support research that cannot handle the data needed for that research is more than inconvenient; it is unusable.

The challenge of developing a data model that is capable of representing all possible linguistic data is one that is impossible to solve. A single, static system simply cannot be developed that supports all data. As a result, systems are targeted to a specific user base in the hope that, by restricting the user community, the data model supported by a particular system will more closely meet the needs of the smaller user base. This approach creates other problems. Users with needs outside the targeted user base, are disenfranchised from that system and new systems must be built to meet their needs. For each of these new systems, much work is spent re-implementing functionality already provided by existing systems. In order for users of various targeted systems to use each other's data, those users must learn to use each other's software or must find a way to convert data between formats. The latter option is frequently unsupported and the former option is time consuming for the linguist who must learn how to use a new program. The development of multiple data management systems for multiple data models also fails to provide support for data management services that needs a hybrid solution (e.g. search for all words in a given language in two different systems).

As a result of these and other difficulties, in practice most linguists store their data using standard file systems or a desktop database systems such as Access. In this case the linguist who owns the collection must assume the burden of providing the functionality of a data management system by organizing data on the file system, maintaining indices of the data and searching through the collection when s/he needs to find data. This is difficult and time consuming for the owners of data collections and imposes severe limitations on their ability to share information between researchers. These limitations are somewhat less acute where researchers are in the same area and

have easy access to each other, but the vast majority of linguists are separated by either time or distance or both, making it extremely difficult for data being managed by one linguist to be made available to the linguistic community at large.

The LDR system addresses the problem of a static data model by providing an implementation for the management of an abstract data model. This abstract model can be extended to take into account new or focused needs of the linguistic community. Specific implementations for data storage and representation can be developed and implemented independently of the abstract system. Since the extensions to the data model are incorporated into the abstract data model, queries that request data from areas of the data model developed for different groups can be managed by the system. The user interacts with the data management services through plug-in tools that provide a graphical interface for working with the system. These tools can be developed to meet specific data management needs and to take advantage of advances in the field of human computer interaction. This approach allows the LDR system to provide a sophisticated data management system that allows for an extensible, yet unified data model. A more detailed explanation of how this is achieved is provide in section 5.

4.3. Data Analysis

Data analysis, at one level or another, is at the heart of linguistic computing. The objective of data management and data archival is to store data so that it can be analyzed. Accordingly, there are many analysis tools available. This is most evident when one simply considers the section titles on the linguistlist's software links page:

- Software Directories
- Text Analysis
- Phonetic Analysis
- Speech Analysis
- Lexical & Morphological Analysis
- Natural Language Processing
- Other Software

Two problems exist with the current state of affairs in data analysis tools. First, these tools must provide their own algorithms for reading and writing data. They may choose to read/write standardized formats (probably most do) or they may use proprietary data formats. Either way, the tool developer is responsible for developing the input/output algorithms of dealing with data and with managing that data while the system is using it. If data and data formats are common, and if every tool that is built must develop software to read and write that data, then much effort is being spent designing and developing software that has already been designed. The LDR system manages data as software objects. The system manages input and output for these objects so that if a tool needs to use a particular type of data, it simply creates an object of that type and lets the LDR system worry about how to save and restore that object. The tool developers then need only to write the software necessary for analyzing that data.

Second, these tools, being designed and developed independently, function independently. The LDR system will provide an architecture that will support inter-tool communication so that any tool running on the system can send a message to any other tool on the system. How to respond once a message is received is a decision left to the implementers of that tool.

5. DESIGNING A UNIFIED APPROACH: THE LDR SYSTEM

Traditional methods in the development of software involve an individual or organization who creates a single monolithic application to perform a specific task or a suite of applications to perform several tasks in a specific domain. Microsoft's Office Suite is an excellent example of a tightly integrated suite of applications that provides (or attempts to provide) a comprehensive solution to all tasks in a specific domain. This approach is of limited value in the field of linguistics. There are simply too many different types of data, too many theoretical approaches taken in the analysis of data, too many personal preferences with regard to display and organization of data for a single monolithic application, or even a closed suite of applications to adequately meet the needs of the community. On the other hand, the current approach of developing a new application every time a new type of data or a new analytic technique is needed is time consuming and results in a tremendous amount of duplicated labor. Both of these approaches to developing a system to support managing, archiving and analyzing data have serious drawbacks. An alternative solution that minimizes these drawbacks is needed. The LDR project is dedicated to developing such a system.

This section will present a conceptual overview of the LDR. It begins by considering the LDR system as a single, monolithic application and proceeds to describe a series of modifications made to the monolithic architecture model that have resulted in the present design of the system and how those changes support the flexibility and extensibility of the system. This program provides users with a variety of options and views to accomplish the linguistic tasks they need to complete. These tasks range from entering data, planning data elicitation for the next day's meeting with an informant, managing the long term objectives and schedule of a major field research project, conducting a phonological analysis of some language, etc. The system manages the data, saves it to and retrieves it from a data store, all out of sight of the user. Because this monolithic program is be used by many different users around the world simultaneously, it is divided into components that are distributed across network.

Since (as Sun Microsystems is fond of saying) the network is the computer, the aspects of the system associated with the network are irrelevant to the concept of the system. The remainder of this section will examine how the linguistic specific functionality is abstracted from this monolithic system resulting in a core architecture and independently developed software modules that plug into this architecture.

5.1. Separating User Functionality

As has already been mentioned, no matter how complex and all encompassing a program may be, it can never provide all the functionality that the linguistic community will need. The LDR system solves this problem by de-coupling user level functionality from the core data management system and providing that functionality through plug-in tools. The LDR architecture provides mechanisms needed to integrate these tools into the system. This includes methods for finding, retrieving, modifying and creating data, for communicating with other tools and organizing the display of the graphical elements of the tool interface. The core architecture is extended by a tool developer, which may be an individual or organization with no formal affiliation with the LDR development group, to provide software that meets a specific set of user requirements. The use of plug-in tools to provide specific functionality is roughly analogous to the use of plug-ins by modern web-browsers, except that in the LDR system, all user-level functionality, not just extensions to “standard” functionality, is provided by the plug-in tools. Separating the user-level functionality from the core architecture of the system solves a number of the problems mentioned above. Notably, the resources provided to the user by the system can be developed and extended independently of the core architecture, allowing the user to more completely leverage the novel contributions of many researchers. Using tools plugged into a core architecture also allows software developers to reuse the existing data representation and storage components of the system. As long as those components of the system meet their needs (see sections 5.2 and 5.3 for more details) they are free to focus their efforts on developing the specific functionality they are trying to provide.

The LDR architecture provides two major components for integrating tools into the system: the tool manager and the layout manager. The tool manager handles creation, registration and destruction of tools. It allows tools to be loaded into the system while it is running without interrupting its operation and facilitates access to tools by other components of the system. The layout manager will provide a number of different methods for displaying a tools user interface to meet the preferences of the user and help to ensure a common look and feel across the various tools being run on the system.

5.2. Separating Data Storage

The LDR system stores data in repository modules for long term storage. The data storage features of the system, encapsulated by the repository, are then de-coupled from the core system in much the same way as the user functionality (see section 5.1 above). This de-coupling allows the development of specialized repositories to meet specific needs (e.g. a slimmed down repository to be used on a personal digital assistant (PDA), or a repository implementation for an existing corpus). The system accesses these multiple repositories through a network manager so that this interaction seems to the user like interaction with a single repository. These repositories, like the plug-in tools, can be added or removed while the system is running without interrupting its operation.

By allowing existing data collections to be incorporated, the LDR system is able to capitalize on the tremendous amounts of effort that the linguistic community has put into building these corpora. Developing repository modules that are designed to interact with existing corpora is not a trivial task, but the alternative is developing a system that makes no provision for existing corpora or for data that, for one reason or another, were not originally intended for use in the system. By providing mechanisms by which data from different corpora can be accessed using a common² data model, the

² The repository is developed to export data to the rest of the system using a data format in the LDR data model. In this way repositories that implement independently developed data models can fit into the unified model of the LDR system. Where the LDR data model is insufficient, the data model can be extended as noted in section 5.3.

LDR system will promote cooperation between different research efforts, a collaboration not currently available. Ideally, though perhaps unrealistically, the LDR system hopes to provide access to all data stored in all archives in all the world. In practice, it is unlikely that this type of comprehensive archive will ever be fully achieved, but by using the approach described here it will be feasible to approximate this; developing a large network of interconnected repositories around the world that can all be accessed via the LDR system.

5.3. Extending the Data Model

Unfortunately, developing a model that fully represents the entire spectrum of linguistic data is no more possible than developing a finite set of tools that meets all the analytical needs of the linguistic community. Like tools, the LDR system allows for independent development of data elements in the data model. The LDR architecture supports the representation of linguistic data through its persistence framework. This framework, which provides an abstract representation of linguistic data, allows data to be passed between modules and saved to and restored from repositories. It provides two main “hooks” from which the actual implementations of linguistic data are attached to the system. The first of these “hooks” are the data classes. These classes are the primary unit of software that will be used to represent the linguistic data. Each class represents a specific type of linguistic data and provides a software interface so that any given realization of that type of data can be viewed and manipulated. The plug-in tools will then use the interface provided by these data classes to manipulate the data on behalf of the user. The second of these “hooks” are the peers. A peer is a software unit that is capable of handling save and restore requests for one or more class of data. Peers abstract the details of saving data to a particular repository implementation. Each repository must provide peers for the types of data that it supports. By using peers to interact with a repository, neither the system nor the data class needs to have any knowledge about how a particular type of data will be stored. All the system needs to do is send the data to the appropriate repository. The

repository then selects the appropriate peer and that peer formats the information to be saved. Both data classes and peers can be dynamically loaded by the system, allowing them to be constructed by third party developers independently of the development of the core architecture of the system.

One common problem in providing an open-ended data model is fragmentation of data formats, that is, many different data formats to represent the same data. Fragmentation is a problem that currently plagues efforts to unify both digital and physical archives of linguistic data. This was a significant concern in choosing to pursue an open data model for the LDR system. A major factor contributing to the current fragmentation of data formats is the lack of a standards body for linguistic data. While a standards body is not likely to be formed, the Open Language Archives Community³ is currently beginning to work on developing best practice recommendations that will help identify those data formats that most effectively meet the needs of the community. These best practice recommendations, coupled with the fact that developers will be more likely to use existing data formats than to develop their own if the existing formats meet their needs, will minimize the risk of fragmentation of the data model.

³ The Open Language Archiving Community was founded at the Workshop on Web-Based Language Documentation and Description, Philadelphia, December 2000. It is an international project to construct the infrastructure to support language archives linked by community-specific metadata and centralized union catalogs and builds on the Open Archives Initiative and the Dublin Core Metadata Initiative. For more information see <http://www.language-archives.org/>.

6. THE SYSTEM IN USE: THE STORY OF BOB

This section will attempt to restate my vision for the LDR system, not from the technical perspective employed so far, but rather as a (quite fictional) story: the story of Bob. Bob is a linguistic anthropologist in Papua New Guinea studying the life and language of the Bena Bena people. When we join Bob, he is out in a small village learning about pigs and the role of pigs in Bena Bena society. He is taking notes on his personal digital assistant using a version of the LDR system configured to run in a lightweight computing environment. Actually, the version he is using is the same as any other version except that he had to change a couple of configuration scripts as described in the user manual and he is using a lightweight repository model that does not handle large collections well, but is ideal for a couple of weeks of notes. He is not terribly worried about getting all the information down in precise detail, just his general impressions. His PDA is recording everything that is said, so he can go back over it when he gets "home." After an hour or so, he decides that he has more information than he will be able to work with for a while. Bob then gets on a public motor vehicle to head back to his apartment in Goroka. On his way and with road conditions permitting, he tries to do some initial organization of his thoughts and make some notes about the things he wants to explore later. Back at his apartment, he uses a plug-in tool on his PDA that uses the LDR architecture to connect to a new repository (the one on his laptop) via the infrared port to download all the notes he took today. Later, when he has a faster network connection, he will back everything up to his university's main repository in Singapore. For now, he begins his analysis process using the (relatively) slow internet connection he has available to search for information on status symbols. One of the things he learned, in the field today, is that to kill a man's pig (the primary symbol of wealth among the Bena Bena people), intentionally or not, is not just destruction of property, but is actually a threat to kill the owner of the pig. Bob uses the LDR system to search through archives around the world for similar observations in other cultures.

He also sends a message back to a computer scientist, Fred, at his university who assists the linguistics program. Thinking that being able to track the number of pigs each informant owns will be informative to the ethnolinguistic aspects of his study, he asks Fred to create an extension to the person data type currently existing to keep track of this type of information. Fred creates a new data type called pigsOwned that links a person to a number of pigs and the necessary classes to go along with it. He also makes a few modifications to some existing data entry / viewing tools to use this new data. All said, the changes are fairly simple and a few days later Bob gets a message that tells him the data and tools he requested are ready, and provides a short LDR script that automatically installs the needed files from the university's server in Singapore. Bob runs the script and starts storing information about the number of pigs each of his informants own.

7. CURRENT PROGRESS AND FUTURE WORK

So far this paper has laid the conceptual foundation for the development of the LDR system and has discussed how that concept differs from the approaches currently being taken by other projects. This section discusses the current status of system development and some areas of work that remain both for the project and the community.

The first system prototype is currently ending its system integration phase and should be released by early May. Like all early release software, this will be unstable and is intended only to provide the community with an early look at how the LDR system is being developed. This prototype attempts to demonstrate three main features of the system:

- Pluggable Components: Tools, Data, Repositories
- Distributed Architecture
- Three Layers of Access

As discussed in this paper, one of the major contributions of the LDR system is its ability to be extended through pluggable components. The prototype system will support each of the three pluggable components discussed in section 5 (tools, data and repositories) with simple implementations of each type of component. By making the application programming interfaces (API) for these components available it is hoped that community feedback will provide useful insights in speeding the process of standardizing these interfaces.

Section 5 mentions that the distribution of system components over a network is irrelevant. While, this is true for the system concept, the distributed aspects of the system are important implementation details. The prototype system will have three major distributable components, a client module, a data server module and a repository

module. The client serves as the user's point of entry to the system and hosts plug-in tools. The client module will connect to a data server to access the persistent framework and the system's support data management. The data server will support connections from many clients and will connect to many repository modules. This entire architecture is scalable. For example, if a user does not want to connect to system through a data server, the system can be configured so that a single computer serves as both the client and the data server and connects remotely to potentially many repositories. In the story of Bob, Bob's PDA served as a completely telescoped system, providing the functionality of the client, data server and repository all on a single machine. The amount of telescoping of the distributed components of the system is governed by a set of configuration scripts that are run when the system starts. The prototype system will provide basic support for the distributed aspects of the system and for telescoping of distributed components.

Every relevant feature of the LDR system will be accessible through three layers of access: a graphical interface, a scripting language and an API. The graphical interface provides user-friendly access to the system with a relatively shallow learning curve. The scripting language allows users with more technical experience to automate repetitive or common tasks. The JavaScript language is being used as the scripting language for the current LDR prototype. The API allows tools to be developed that can interact with all aspects of the system. These three layers help ensure that the system is both accessible to novice users, yet powerful enough for more experienced users.

7.1. Future Work

There are four major issues left for future development of the LDR system: (1) security - uniquely identifying users and insuring that access to data can be restricted as needed based on a user's identity, (2) query mechanisms - searching across multiple repositories for data that meets certain search criteria and ranking the data based on how closely it meets those criteria, (3) tool layout - organizing the graphical display of tools so that it is visually pleasing and user customizable and (4) network support -

allowing humans and tools to communicate with each other across the network. A number of other issues including (but not limited to) thread safety, optimization of algorithms and user identification will also need to be more thoroughly addressed in future releases of the system.

In addition to the work that is needed to develop the core architecture of the system, a considerable amount of effort will be need by the linguistic community to maximize the potential of the system. Core areas in which community support is need are: (1) developing a core data model that meets the majority of the community needs and standard extensions to that model to support more focused needs, (2) the development of tools to help present data in ways that are easily accessible to users and that support specific tasks in data analysis, (3) development best practice guidelines for data storage and analysis techniques, especially where multiple data formats and tools may claim to offer the same functionality and (4) ethical guidelines regarding online documentation and description of languages.

8. CONCLUSIONS

In this paper I have presented the current direction that linguistic community is taking to develop tools to support the documentation and description of languages and have discussed some of the shortfalls of the current approaches. In presenting those shortfalls, I have also described some possible improvements and explored the development of a system, the Language Data Repository, that is being built to take advantage of those improvements. The LDR system is composed of a core architecture and various pluggable components that provide the linguistic specific functionality of the system. These pluggable components can be developed independently of the core architecture. With respect to archiving data, the LDR system departs from the currently prevalent publication paradigm and the emphasis on large organization-wide archives. This shift allows the LDR system to represent data in ways that more closely reflect the ways in which linguists work with data on a day-to-day basis. These archives of data, encapsulated by a repository module, can be plugged into the system while it is running without interrupting its operation. The system utilizes an extensible data model that can be expanded as new data types are needed for specific research. These data types are accessed through the use of plug in tools, which are developed independently of the core architecture to provide the user with state of the art support for data analysis. Together, the core architecture and the independently developed components help individuals manage personal collections of data, ease the process of developing new tools to support linguistic research, facilitate access to data collected by others, and allow linguists to work with data in the digital realm in a way that more closely resembles the way they work with data in the physical realm. The power and flexibility provided by the LDR system significantly improves on other approaches to developing software to support linguistic research. The approach to developing the LDR system is approach represents a significant departure from traditional approaches to developing software to support language documentation and allows the system to be extended to keep pace with changes in theoretical and analytic approaches to linguistic

research and documentation. The core architecture, pluggable tools, repositories and data elements work together to provide a unified approach to the three major domains involved in working with linguistic data: data archiving, data management and data analysis.

9. REFERENCES

- Mendoza-Denton, Norma. 1997. *Chicana/Mexican Identity and Linguistic Variation: An Ethnographic and Sociolinguistic Study of Gang Affiliation in an Urban High School*. Ph.D. Dissertation, Stanford University.
- Weinberger, Steven H. 2000. *Speech Accent Archive: Issues and Methods*, in *Proceeding from the Workshop on Web-Based Language Documentation and Description*. Philadelphia, PA. Dec 2000.
- Wennerstrom, Ann. 2001. *The Music of Everyday Speech: Prosody and Discourse Analysis*. Oxford University Press, to appear.
- Zentella, Ana Celia. 1997. *Growing Up Bilingual*. Blackwell Publishers.

10. VITA

Michael Neal Audenaert

110 Cavendish Cir, Madison, AL 35758

EDUCATION

B.S. Computer Science, 2001

Texas A&M University, College Station, TX

Minor: Linguistics

Thesis: The Language Data Repository: Machine Readable Storage for Spoken Language Data

Thesis: Uniting Software Tools for Analysis, Management and Archival of Linguistic Data:

Lessons from Developing the Language Data Repository

GRANTS, HONORS AND AWARDS

2000-2001 Graduate Student Research and Presentation Grant

2000-2001 University Undergraduate Research Fellows

1999-2000 University Undergraduate Research Fellows

1997-2001 President's Endowed Scholarship

1997-2001 National Merit Scholarship

1997-2001 Engineering Scholars Program

2000 1st Place, Undergraduate Computer Science Poster Session, Student Research Week,
Project: Language Data Repository: Machine Readable Storage for Spoken Language
Data

1999 1st Place, Engineering Scholars Program Poster Competition, Project: Language Data
Repository: Machine Readable Storage for Spoken Language Data

1998 New Medic of the Year, Texas A&M Emergency Care Team

1997-1998 Thomas C. Lingeman Scholarship

PUBLICATIONS

Audenaert, Neal (2000). The Language Data Repository: Project Abstract. In *Proceedings of the Workshop on Web-Based Language Documentation and Description*, pages 36-42. Philadelphia, PA: University of Pennsylvania

PROFESSIONAL EXPERIENCE

1999, 2000 Summer Intern **Dynetics, Inc.** Huntsville, AL

Enhanced THAAD Project Office IDEF0 model of the THAAD Battery Battle Management and C3I functional requirements, and designed and implemented new table management system for the Air Force Joint Modeling and Simulation System.

ACTIVITIES

2000 Grace Bible Church: International Student Outreach, Web Development Team

1999 International Student Ministries Coordinator, Baptist Student Ministry

1997-2001 Texas A&M Emergency Care Team, Events Coordinator (Fall 1998-Spring 1999),
Operational Supervisor, Medic in Charge, CPR Instructor, First Aid Instructor,
Webmaster (Fall 1998-Spring 1999)

1990-2000 AWANA Children's Program, helped lead activities for children aged 3 through high
school, co-director middle school 1998-2000, director high school group 1999-2000

1998 Conducted field research on the language and culture of the Yagarian, BenaBena, and
Bahinemo people groups in Papua New Guinea.