MacVisSTA: A System for Multimodal Analysis of Human Communication and Interaction

Richard Travis Rose

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science

In

Computer Science

Francis Quek, Chair
Christopher L. North
Edward A. Fox

July 27, 2007

Blacksburg, Virginia

Keywords: Imagery and Video Analysis, Information Access, Visualization, Temporal Analysis, Discourse Production, Mental Imagery, Embodied Imagery, Communication, Interaction

MacVisSTA: A System for Multimodal Analysis of Human Communication and Interaction

Richard Travis Rose

Abstract

The study of embodied communication requires access to multiple data sources such as multistream video and audio, various derived and meta-data such as gesture, head, posture, facial expression and gaze information. This thesis presents the data collection, annotation, and analysis for multiple participants engaged in planning meetings. In support of the analysis tasks, this thesis presents the multimedia Visualization for Situated Temporal Analysis for Macintosh (MacVisSTA) system. It supports the analysis of multimodal human communication through the use of video, audio, speech transcriptions, and gesture and head orientation data. The system uses a multiple linked representation strategy in which different representations are linked by the current time focus. MacVisSTA supports analysis of the synchronized data at varying timescales for coarse-to-fine observational studies. The hybrid architecture may be extended through plugins. Finally, this effort has resulted in encoding of behavioral and language data, enabling collaborative research and embodying it with the aid of, and interface to, a database management system.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation Research Experience for Undergraduates program, and the DTO VACE-II grant 665661: From Video to Information: Cross-Modal Analysis of Planning Meetings.[1] Significant portions of this thesis were written after the author began duty at NIST.[2] In this document, the following terms are trademarks: Java, Python, QuickTime, Theme, SignStream, Vicon, Viper; the trademark symbol (^{TM}) is omitted. The author expresses his gratitude to the following individuals for their steadfast support and encouragement of this work: Bennett Bertenthal, Rachel Bowers, Susan Duncan, Edward Fox, Jon Fiscus, John Garofolo, Yonca Haciahmetoglu, Mary Harper, Tom Huang, Sari Karjalainen, Dan Loehr, David McNeill, Chris North, Francis Quek, Greg Sanders, Deborah Tatar, Ron Tuttle, and Laurian Vega, as well as the multitude of friends and colleagues I've met along the way.

Special thanks to my advisor, Dr. Francis Quek, and committee members, Dr. Edward Fox and Dr. Chris North, for their time, patience, and helpful feedback during the preparation of this thesis.

[1]DTO: Disruptive Technology Office, VACE: Video Analysis and Content Extraction
[2]NIST: National Institute of Standards and Technology

Disclaimer

Any opinions, findings, or conclusions expressed in this thesis are those of the author and do not necessarily reflect those of Virginia Tech or the U.S. Government.

Table of Contents

Page

Disclaimer iv

List of Figures viii

List of Tables ix

Chapter

1 Introduction 1

1.1 Data Sources 2

1.2 Requirements for Multimodal Analysis 4

1.3 Motivation for Creating a New Tool 5

1.4 Organization 6

2 Literature Review 7

2.1 Multimodal Annotation Tools 7

2.2 Transcription tools 21

2.3 Psycholinguistic Theory 24

2.4 Speech and Gesture 26

3 Methodology 28

3.1 Psycholinguistic Annotation 29

4 Design and Implementation 34
4.1 Data Types 36
4.2 Project-based Representation 37
4.3 Display Components 38
4.4 Plug-in architecture 38
4.5 Interaction design 39
4.6 Music-Score Annotation 41
4.7 Notes and Notebooks 42
4.8 Open Source Components 42
5 Corpus Building and Database Management 45
5.1 Embedded SQL 50
5.2 Database Table Structure 51
5.3 Significance of Database Management Systems . . . 52
5.4 Relation to other Information Technology Frameworks 52
5.5 Coupling of Multimodal Corpora to Information Visualization and Visual Schema 53
6 Annotation and Metadata Standards 55
6.1 AGTK 55
6.2 ATLAS 56
6.3 MATE 57
6.4 NITE and NXT 57
6.5 MPEG-7 58

6.6 VERL and VEML 59
6.7 W3C's Annotea 59
6.8 Summary 60

7 Round-trip Conversion of MacVisSTA Annotations to and from Annotation Graphs 62
7.1 Towards Greater Interoperability 62
7.2 Annotation Schema 62
7.3 Implementation 63
7.4 Loss-less conversion 65
7.5 Observations 66

8 Temporal Analysis 68
8.1 Temporal Queries 69

9 Conclusion 72
9.1 Impact 73
9.2 Future Work 74

Bibliography 75

List of Figures

2.1 A timeline of selected multimodal annotation tools 11

4.1 VisSTA with all components 35

4.2 MacVisSTA interface 36

4.3 Sample configuration with AFIT meeting data 39

4.4 Sample configuration with NIST meeting data 40

4.5 Annotation editor 41

4.6 Key modules in MacVisSTA's design 44

5.1 Workflow diagram of for management of source and derived data . . . 47

5.2 MacVisSTA preferences 49

5.3 Visualization, coding, and analysis with database support 51

8.1 Sample Theme output 71

List of Tables

1.1 Source and derived data 4

1.2 Requirements of a multimodal analysis system 5

2.1 Desired features for annotation and analysis tools 10

2.2 A comparison matrix for annotation tools 23

3.1 Alternative hypotheses for intervals annotated for gesture 30

3.2 Alternative hypotheses for intervals annotated for gaze 31

3.3 Alternative hypotheses for intervals annotated for coreference 31

3.4 Multi-pass, hypothesis-driven approach to multimodal analysis 33

4.1 Open source components used in development of MacVisSTA 43

5.1 Annotation schema 50

5.2 Application of 5S framework to multimodal corpora and analysis . . . 54

8.1 Explanation of specialized temporal queries 70

9.1 Desired components for psycholinguistics research 73

Chapter 1

Introduction

As embodied beings, humans communicate through a variety of cues: speech, gesture, eye gaze, and body posture. The activity is inherently multimodal and behaviors that include hand, head, and eye movement are intricately related to each other, and to speech, by their time synchrony [20, 26]. Salient analysis of these aspects of communication requires investigation of multiple modes (or channels) of behavior, which necessitates a multi-layered and multi-pass approach. This thesis presents a new system for analysis that is general-purpose yet motivated by modern psycholinguistic theory, combining elements of information visualization, observation (as in the scientific method), database management, and multimedia, resulting in a comprehensive approach to understanding communication and interaction.

Because existing datasets of meetings were too few and variable for systematic study, the Cross-Modal Analysis For Planning Meetings project collected new data to support this research. These were recorded over several sesstions at the Air Force Institute of Technology (AFIT) during 2004-2006. Key features of this data include: scenario-based, problem-solving topics[1], known mission and doctrine, known participant roles and hierarchy. The range of interaction in these meetings varies from

[1] An approach employed at National Institute of Standards and Technology (NIST)

dyadic to multiple participants. As a result, discourse is constructed through social interaction, during which several factors can come into play, including social dominance (for example, competing for the floor), coalition formation and dissolution, and topical shifts that occur according to the flow of meeting dynamics. The variety of events that can arise in this setting requires a holistic approach to analysis and an overarching theory, which will be developed further in subsequent chapters. The following sections will describe the types of data we employ and our integration strategy for addressing the analysis needs in this research.

1.1 Data Sources

In order to better understand human communication and the factors that contribute to interaction, we require high-quality source data, as well as accumulating many fine-grained observations. Source data minimally consist of synchronized video and audio, with optional sensor data depending on configuration and available equipment. Recordings are made using one or more cameras for video in National Television Standards Committee format (NTSC 720 by 480 resolution, 29.97 frames/second), tabletop and/or "boom" microphones, and optionally head-worn microphones to collect audio (24 bit, 44.1 or 48 kHz). Our collection protocol varies depending on the setting. In some cases, monocular video may be all that is available (similar to home video). In other settings, two cameras may be used for stereo calibration and triangulation. We prefer to use at least two cameras for recording dyadic interactions and to support stereo computer vision experiments. This approach generalizes to n-

cameras with overlapping fields of view, allowing us to capture larger spaces with more people, such as multi-participant meetings. Xiong and Quek have developed a calibration method both for a single pair of cameras (for stereo vision) or n-pairs of cameras for a fully calibrated meeting room [44]. An advantage of using multiple overlapping fields is that it avoids the problem of selective capture, which can arise with one camera (e.g., shoulder-mounted camera in a classroom setting).

After recording, we process the data to extract low-level features. Examples include speech transcription and output of recognition algorithms, hand-tracking, head-tracking, etc. This enriches the source data with additional derived data layers. As a result of this process, we have source and derived data. In addition, data may be further characterized as continuous or discrete. In our work, we treat data that has a high sampling rate (such as audio/video, motion tracks) as continuous. Derived data may have the same or lower sampling rates, and may represent discrete events (i.e., segments). Observational data can be either points (i.e., instants in time) or intervals; these are also discrete data and are made with the limit of precision allowed by digital media.

Our meeting room corpus contains time synchronized audio and video recordings, features derived by computer vision algorithms and Vicon motion capture, audio features such as pitch tracking and duration of words, and coding markups. As described in [6], details on source and derived data appear in Table 1.1.

Table 1.1: Source and derived data for meeting room corpus

Source	**Derived**
Video	Video (raw and compressed) from 10 cameras
Audio	Audio from all microphones
Vicon	3D positions of head, shoulders, torsos, and hands
Computer vision	Head pose, torso configuration, and hand position
Audio	Processing speech segments, transcripts, alignments
Prosody	Pitch, word and phone duration, energy, etc.
Gaze	Gaze target estimation
Gesture	Gesture phase/phrase, semiotic gesture coding
Metadata	Language metadata, e.g., sentence boundaries, speech repairs, floor control

1.2 Requirements for Multimodal Analysis

The above datatypes call for an integration strategy for making sense of the multiple types of information – source as well as derived data. In previous work, this diversity of datatypes had already begun to be addressed in a system called VisSTA (Visualization for Situated Temporal Analysis), developed for a Unix platform running X-Windows.[2] Experience with the design and implementation of this system established an initial set of requirements for analysis [36, 38, 40], summarized in Table 1.2.

While VisSTA served as a useful system that supported previous research, we chose to re-implement the system for Apple Macintosh on OS X. Reasons for this

[2]Silicon Graphics, Incorporated (SGI)

Table 1.2: Requirements of a multimodal analysis system

Requirement	**Brief justification**
Construction of arbitrary views of data	Flexible visualization of heterogeneous datatypes
Creation of free-form observations	Visualization and segmentation of episodes
Creation of time-tagged labels	Visualization of observations, event data
Database management system (DBMS)	Support querying of observations
Import of speech transcripts	Visualization/analysis of speech data
Import of continuous data	Visualization/analysis of continuous plots
Individual and shared data spaces	Separation of project vs. reference/shared data
Multiple-linked representation	Support for data navigation and sense-making
Playback of multi-channel audio	Permit use of audio from multiple microphones
Project-based system	Organize analysis according to user goals/criteria
Simultaneous, multi-camera video	Playback from multiple vantage points
Support for collaborative work	Pooling observations and analyses across teams
Zoomable interface	Coarse-to-fine observation and analysis

choice of platform included built-in support for modern codecs through QuickTime; built-in graphics rendering with OpenGL; operating system based on Unix (Darwin kernel); availability of high-level frameworks and application programming interface (API) via Cocoa/Objective C; developer tools; and ability to combine modern languages (C++, Objective C, Java, etc.) in the same application.

1.3 Motivation for Creating a New Tool

Since this research focused on multimodal analysis of meetings with a team of experts in multiple disciplines, we desired a tool that would allow us to integrate several

sources of information. Specifically, we sought to create a hybrid system that would give analysts simultaneous access to source data (audio/video), low-level features such as 2D/3D tracking (obtained from automated and semi-automated algorithms), speech metadata, and analysis (through manual annotation). We developed new software that would support these needs, resulting in a new way to pool, visualize, and create multipl e streams of data. Thus, MacVisSTA embodies a new interface that supports *visualization, coding,* and *analysis.*

1.4 Organization

The remainder of this thesis is organized as follows: in order to situate this new design effort in the context of related work, a review of multimodal annotation tools and psycholinguistic theory appears in the literature review (Chapter 2), followed by methodology (Chapter 3). The system architecture is presented in Chapter 4, and corpus building/database management in Chapter 5. A review of metadata/annotation standards is given in Chapter 6. A method for converting MacVisSTA annotations to a general format is discussed in Chapter 7. A discussion of temporal analysis appears in Chapter 8, and conclusions appear in Chapter 9.

Chapter 2

Literature Review

2.1 Multimodal Annotation Tools

Over the years, there has been a steady evolution in the development of new tools for video annotation, visualization, and analysis. The following sections provide an overview of several of these (in alphabetical order); while not an exhaustive list, this review shows that there are several overlapping features shared by many of the tools. In addition, some tools used for speech transcription are included as examples. These play an important role in support of multimodal analysis. In particular, tools for video and audio analysis must often be used in complementary fashion because they have emerged from different communities having different research trajectories.

This survey is included to illustrate the multitude of approaches to computer-aided video analysis. Importantly, we did not adopt any of these tools for our multimodal annotation for the following reasons:

- Existing tools did not provide complete coverage of desired features
- Lack of support for compressed video (e.g., MPEG-4)
- Lack of support for long videos (e.g., more than 10 minutes)

- Lack of multi-camera video
- Lack of precise synchronization between multiple videos/dependent views
- Lack of flexibility resulting from closed architecture
- No appropriate database management system
- Not enough options for heterogeneous datatypes
- Not enough options for visualization (e.g., multiple views)
- Reliance on Java Media Framework (JMF) would have become rate-limiting

Further, in some cases software has reached the end of the development cycle and has been transitioned to other packages. Since many of the newest systems are still evolving, many of the capabilities of these multimodal tools are being refined in parallel (shown in Figure 2.1). This illustrates that multimodal analysis tool development is an active area of research, which has roots in on-going efforts spanning decades. In essence, the overarching goal of these kinds of software is to provide holistic systems for analysis, which has also served as a guiding principle for creating MacVisSTA (discussed in Chapter 4).

The diversity of tools for audio and video annotation has led to a situation where there are several tools with overlapping functions and independent file formats. Most of the transcription and annotation tools do not read the file formats of other programs. This would be a desirable feature because it would allow more analyses to be performed across data sets, and would also permit many different analyses to be performed and used together (pooled) more readily. As a result, several efforts

towards standardizing the annotations/metadata have emerged and are discussed in Chapter 6. In addition, a discussion of improving the compatibility of tools appears in Chapter 7, which explores an approach to annotation interchange.

Some comparisons across tools such as these have been made by Bigbee, Loehr, and Harper [2], and others [39]. Desired features of annotation and analysis tools are summarized in (Table 2.1), for example, use of XML (Extensible Markup Language) for tagging. Importantly, these features are consistent with our concept of a system for multimodal analysis; we established a baseline system (VisSTA) to meet an initial set of requirements as described in Chapter 1 (Table 1.2). This chapter reviews several related tools and their salient features, as well as Psycholinguistic theory, and implications for our system's design.

Table 2.1: Desired features for annotation and analysis tools (adapted from [2])

Videos time-aligned with annotation
Direct support of XML tags
Time-aligned audio waveform display
Acoustic analysis (e.g. pitch tracking) tools included
Direct annotation of video
Collapsable views
Annotation of different levels via programming interface
Modular, open architecture
Music-score display
Automatic tagging functions
Easy to navigate and mark start and stop frame of any video or audio segment
Segment start and stop points include absolute time values (not just frames)
User can make explicit relationships or links across levels
Can specify levels and elements (attribute / values)
Inclusion of graphics as an annotation level (i.e., ink, diagrams)
Support for overlapping and hierarchical structures in annotation
Easy to annotate metadata (annotator, date, time, etc.) at any given level or segment
Some levels time-aligned, others are independent but ordered in time
Support for working with multiple synchronized video, audio, and vector ink media sources
Import/export of all annotations
Multiple platform execution
Query and search of annotations

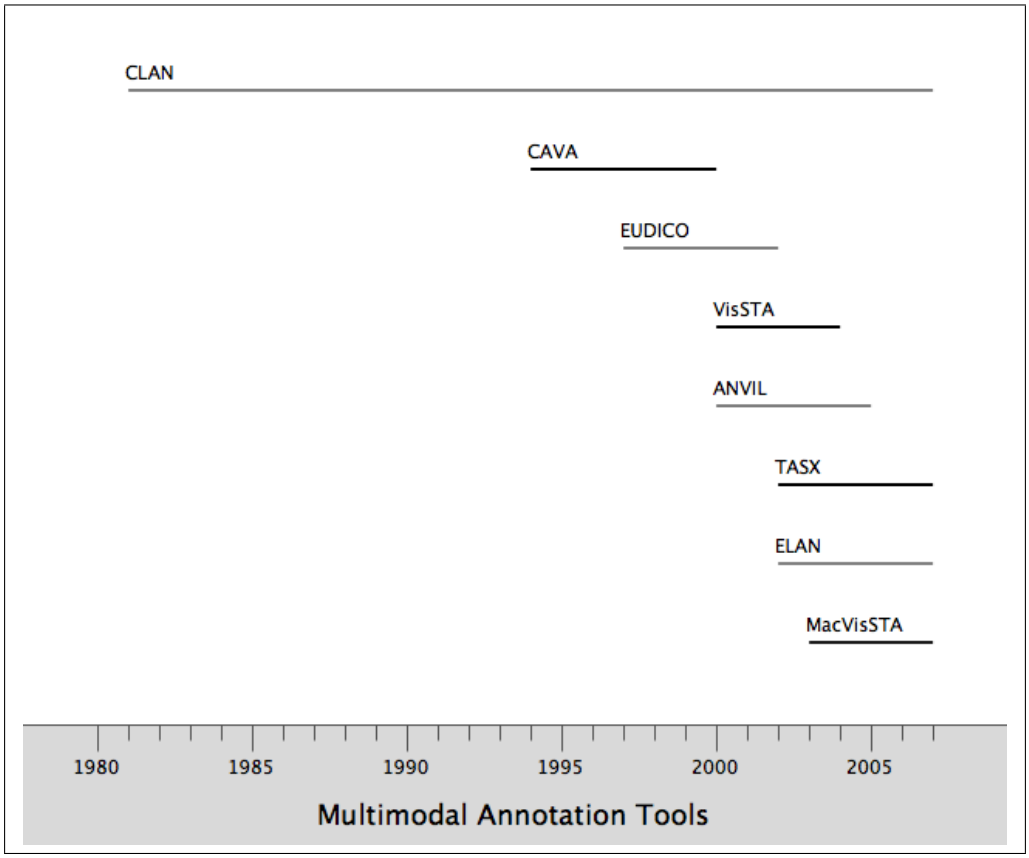

Figure 2.1: A timeline of selected multimodal annotation tools

2.1.1 Anvil

`http://www.dfki.de/~kipp/anvil/`

Anvil (Video Annotation Research Tool) is an annotation tool written in Java available since 2000 and designed to work with audio and video and provide visualization of supporting meta-data; it features frame-accurate annotation and is hierarchical with multiple user-defined layers. It uses color-coding on multiple tiers to represent

events, and can annotate links between tracks if desired. ANVIL was created for gesture research and has been applied in other domains (human computer interaction, linguistics, computer animation, etc.) It imports time-aligned speech markup from Praat (described below) and XWaves. The major supported video formats are Audio Video Interleave (AVI) and QuickTime. The software is downloadable as a Java executable from DFKI (German Research Center for Artificial Intelligence). Anvil was originally written and is currently maintained by Michael Kipp.

2.1.2 CAVA

`http://www.mpi.nl/world/tg/CAVA/CAVA.html`

Begun in 1994, CAVA (Computer Assisted Video Analysis) was created for use on both PC and Macintosh: in particular, there were two transcription tools, the Transcription Editor (TED) for use with the PC for transcribing analog video tape, and Media Tagger for working with digital video on the Macintosh. CAVA is a multiplatform system that can access data stored in an Oracle database on a Unix server. In addition, the CAVA tools are platform-dependent, use a proprietary data storage format, and are designed for single-site use (i.e., site-specific). The tools that comprise CAVA appear to be the necessary precursors to the European Distributed Corpora Project (EUDICO), described below. Both CAVA and EUDICO were created at the Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands.

2.1.3 CHILDES/CLAN

`http://childes.psy.cmu.edu/`
`http://childes.psy.cmu.edu/clan/`

The Child Language Data Exchange System (CHILDES) and Child Language Analysis (CLAN) are a suite for studying conversation and interaction. CHILDES is a large database of language acquisition data in more than 30 languages in a common format. It was established by Brian MacWhinney at Carnegie Mellon University. CLAN supports searching and frequency counts, and other functions (such as transcript editing). These functions are performed using a command-line interface. Designed for use with CHILDES, it uses a format for transcription called Codes for Human Analysis of Transcripts (CHAT). It may also be used with files in conversation analysis (CA) format. CLAN is available for both Windows and Mac OS X.

2.1.4 EUDICO

`http://www.mpi.nl/world/tg/lapp/eudico/eudico.html`

The Max Planck website provides a demonstration version of this tool. Begun in 1997, the EUDICO toolbox is still in development and source is not yet publicly available. EUDICO is intended to be platform-independent, employ multiple video formats, and support distributed, multiple-site operation via the internet. Concepts in EUDICO anticipated several distributed services with respect to analysis software, annotation, and digital media. Thus, EUDICO was oriented towards remote

collaboration. Written in Java, it uses the Java Development Kit and Java Media Framework. The engineering in EUDICO was leveraged in the design of EUDICO Linguistic Annotator (ELAN).

2.1.5 ELAN

`http://www.mpi.nl/tools/elan.html`

ELAN (EUDICO Linguistic Annotator) is an open source, cross-platform Java tool that was created for psycholinguistics research. It allows one to create, edit, visualize, and search annotations for video and audio data. It includes features such as display of audio and video *in situ* with annotations, time linking of annotations to media streams, linking of annotations to others, and an unlimited number of annotation tiers as defined by the users, as well as import, export, and search options. ELAN is under active development and is functional, was made to be extendible and support collaborative annotation/analysis. Many of ELAN's features make it a good example of what can be done using Java on modern computer platforms. ELAN is developed and maintained at the Max Planck Institute for Psycholinguistics.

2.1.6 EXMARaLDA

`http://www1.uni-hamburg.de/exmaralda/index-en.html`

EXMARaLDA (Extensible Markup Language for Discourse Annotation) is a package combining concepts, data formats, and tools for computer-aided transcription and

annotation of human language. It is being developed at the University of Hamburg as the core component of a database for a project at the Collaborative Research Center, and is freely available to the public. Key features include XML-based data formats, Java tools, and interoperability. In particular, EXMARaLDA interoperates with Praat (a speech transcription tool, section 2.2.1), ELAN, and TASX Annotator (sections 2.1.5 and 2.1.13; ELAN and TASX are both tools for video annotation).

2.1.7 FORM: A Kinematic Gesture-Annotation Scheme

`http://projects.ldc.upenn.edu/FORM/research.html`

FORM is a gesture annotation scheme designed by Craig Martell to capture the kinematic information in gesture from videos of speakers. The FORM project is currently building a detailed database of gesture-annotated videos stored in Annotation Graph format. This allows the gestural information to be augmented with other linguistic information, such as parse-trees of the sentences accompanying the gestures, discourse structure, intonation information, etc. FORM encodes the "phonetics" of gesture by giving geometric descriptions of location and movement of the right and left arms and hands, the torso and the head. Other kinematic information like effort and shape are also recorded. FORM uses Anvil as its engine (i.e., it has been created as a plugin). In the future a stand-alone FORM tool is feasible and would also be an open-sourced gesture annotation tool.

2.1.8 IBM MPEG-7 Annotation Tool

`http://www.alphaworks.ibm.com/tech/videoannex`

The Moving Picture Experts Group has developed a standard, MPEG-7, for capturing/describing features of multimedia content (described in Chapter 6). The IBM MPEG-7 Annotation Tool is for annotating vide o sequences with MPEG-7 metadata. It is based on segments of the video referred to as *shots*; a shot is a continuous stretch of video, and multiple shots combine to make a longer sequence.

In the IBM tool, shots in the video sequence can be annotated with:

- static scene descriptions
- key object descriptions
- event descriptions
- metadata from user-defined schema (which IBM refers to as *lexicons*)

All annotations associated with each video shot are stored in an XML file. The tool allows users to create, edit, download, or save customized lexicons. It requires an input video and a shot segmentation file, in which the video shots have been determined by detecting scene "cuts", dissolutions, fades, etc.

2.1.9 MacSHAPA

`http://www.aviation.uiuc.edu/institute/acadprog/epjp/macshapa.html`

MacSHAPA was developed at University of Illinois by Penelope Sanderson[1] This tool was created as general-purpose and can either be used with a VCR or QuickTime files, allowing creation of observational data, namely coding/coded events that describes human and other activities (such as system events). MacSHAPA represents ground-breaking effort for Exploratory Sequential Data Analysis applied to video. The software was actively developed through 1994 for the Macintosh running OS 9 [42, 41].

2.1.10 Multitool

`http://www.ling.gu.se/projekt/tal/multitool/`

Multitool is an open-source, cross-platform multimodal transcription and analysis tool written in Java. It can be used to create time syncronized transcriptions using audio and video. It is designed to allow import and export of transcriptions. The software supports amplitude analysis of audio waveforms, playback of Quicktime video, and multiple coding views. In particular, it provides flexible coding schema and colored coding in an animated window, which is synchronized with the video and/or audio waveform visualization.

2.1.11 Observer XT

`http://www.noldus.com/site/doc200401012`

[1]with Jay Scott, Tom Johnston, John Mainzer, Larry Watanabe, Jeff James, Vance Morrison, and Jeff Holden.

Observer is a software package licensed by Noldus Information Technology (Noldus) for the collection, analysis and presentation of observational data. Observer can be applied to study observable events, such as gesture, speech, gaze, expressions, movement, and social or human-computer interactions. Observations are entered into a project's database as part of a flexible, user-defined coding scheme. A set of observations can be exported to Statistical Package for the Social Sciences (SPSS), used for built-in report generation, or used with pattern analysis software such as MATLAB or Theme (section 2.1.14).

2.1.12 SignStream

`http://www.bu.edu/asllrp/SignStream/`

SignStream is a database tool for analysis of linguistic data captured on video. Although SignStream has been established specifically for working with data for American Sign Language, the tool may be applied to language data captured on video. SignStream should be suitable for the study of other signed languages as well as studies that include analysis of gesture. SignStream is supposed to simplify transcription of sign language video data and increases accuracy since the interface obtains timing information from the video. The software runs on Mac OS (9.0), and may potentially be ported to Mac OS X. (An older tool for sign language annotation/analysis, syncWRITER, was created for a similar purpose.) The SignStream project is led by Carol Neidle at Boston University.

2.1.13 TASX

`http://medien.informatik.fh-fulda.de/tasxforce/TASX-annotator`

Time Aligned Signal data eXchange (TASX) is an open source, cross-platform (Java) tool that provides a general framework for creating and managing corpora, XML-based annotation of multimodal data, transformation of non XML-annotations, and web-based analysis and dissemination of the data. TASX-annotator is intended to be a user-friendly program for multilevel annotation and transcription of (multi-channel) video and audio data. TASX development is managed by Alexandra Thies and Jan-Torsten Milde at the University of Bielefeld.

2.1.14 Theme

`http://www.noldus.com/site/doc200403003`

Theme differs from other tools because it is an analysis tool that can be used with a variety of data (including video annotations); it is licensed by Noldus and can be used with Observer. Theme was developed to detect and analyze patterns in time-series data. It can discover relationships in observational data that humans typically overlook and commonly accepted statistical methods (e.g., modeling distributions) cannot find. Theme uses an algorithm created especially for behavioral research, which is discussed in further detail below. Theme was originally written by Magnus Magnusson, now director of the Human Behavior Laboratory, University of Iceland.

2.1.15 Transana

`http://www.transana.org/`

Transana is a free, open source tool implemented in Python and downloadable from University of Wisconsin-Madison Center for Education Research; it can be used with audio and/or video. It has the ability to mark intervals of video called "clips" and labeling them. It allows the user to label clips and arrange them in collections. It can display audio as a waveform; as a result, Transana can be used for general purpose transcription of speech and other events. It can be used as a single or multiple-user program having a MySQL database. The Transana software was originally created by Chris Fassnacht and is now developed and maintained by David K. Woods.

2.1.16 Viper

`http://viper-toolkit.sourceforge.net/products/`

The Video Performance Evaluation Resource tool (ViPER), is developed at the Language and Media Processing Lab (LAMP) at the University of Maryland. ViPER is a toolkit that contains scripts and Java programs for creating spatial annotations, specifically "ground truth" for video, and systems to evaluate the performance of computer vision algorithms. Thus, ViPER offers an annotator with an interface (ViPER-GT, ground truth), and ViPER-PE (for performance evaluation) is a command-line tool. The tools are open source and are being used to support multi-site video evaluations (e.g., text recognition in video for Rich Transcription 2006).

Since ViPER is mostly oriented towards spatial rather than temporal annotation, it fills a special niche for video analysis and evaluation.

2.2 Transcription tools

Besides video annotation tools, study of human language phenomena also requires experience with speech transcription and a different toolset. Speech transcription and video annotation tools are currently packaged separately because they have evolved along different research trajectories, and typically in support of different communities and feature sets. Several of these may be used together with video analysis tools. Some example tools that are specialized for speech transcription are described next.

2.2.1 Praat

`http://www.fon.hum.uva.nl/praat/`

Praat is an open source, cross-platform tool for doing phonetic analysis of speech on the computer. It originates from Paul Boersma and David Weenink at the Institute of Phonetic Sciences, University of Amsterdam. Praat has a variety of built-in functions and machine learning algorithms for working with speech (audio), such as spectral analysis (spectrograms), pitch analysis, formant analysis, intensity analysis, etc. Praat provides its own scripting language, permitting additional functions to be added, and can be used to create speech transcriptions. It is implemented in C++ and uses X-windows/Motif, Carbon (for Mac OS), and QuickTime.

2.2.2 Transcriber

`http://trans.sourceforge.net/en/install.php`

Transcriber is open source, cross-platform software for manual annotation of speech. It was developed using Tcl/Tk scripting language (and C). It provides an interface for segmenting speech, transcribing, as well as labeling turns, topics, etc. It uses a toolkit called "Snack". Although it was created for broadcast news transcription, it can also be used as a general-purpose audio transcription tool.

2.2.3 WaveSurfer

`http://www.speech.kth.se/wavesurfer/`

WaveSurfer is an open source tool for sound visualization and analysis. It can be used in its default configuration as a stand-alone tool for transcription, or it can be extended through plugins. WaveSurfer can also be embedded in other applications. It uses a toolkit called "Snack Sound Toolkit". Both WaveSurfer and Snack are from the Department of Speech, Music and Hearing at the School of Computer Science and Communication, Royal Institute of Technology (Kungliga Tekniska Högskolan, KTH), in Sweden.

2.2.4 Comparison Of Selected Multimodal Annotation Tools

Several of the multimodal annotation tools reviewed in this chapter are compared/contrasted with MacVisSTA in Table 2.2.

	ANVIL	**ELAN**	**EXMARaLDA**	**MacVisSTA**	**TASX**	**Transana**
Annotation type	Structured	Simple	Simple	Simple	Simple	Simple
Built-in database	no	no	no	yes	no	yes
Continuous plots	waveform, pitch	waveform	waveform	yes	no	waveform
Multiple camera video	no	yes	no	yes	yes	no
Multiple channel audio	yes	yes	yes	yes	yes	no
Overlapping events possible	no	no	no	yes (as notes)	yes	no
Separate schema definition	yes	partly	no	no	no	no
Timeline	implicit	explicit	explicit	implicit	implicit	implicit
XML format	yes	yes	yes	yes	yes	no

Table 2.2: A comparison matrix for annotation tools

These tools were and compared with respect to several key features. The similarities and differences of these selected tools may be summarized as follows: annotation in Anvil is "structured" because it uses a schema definition, whereas the other tools use "simple" annotation, where the user's labeling of intervals is not restricted (the labeling is free-form). ELAN uses a controlled vocabulary. Built-in database support (using relational tables) is provided by MacVisSTA and Transana. Several tools support audio waveform display, but general support for continuous plots is found in MacVisSTA. Some of the tools support multiple-camera video or multiple-channel audio, including MacVisSTA. Finally, except for Transana, the tools in Table 2.2 store their data in XML.

2.2.5 Relation of Tools to Psychology and Language Research

Tools give researchers access to their data such that they can make new discoveries. Thus, development of new tools should be grounded in the theory, yet ideally be kept flexible. The following section reviews elements of Psycholinguistic theory that form the foundation for Chapter 3 (Methodology).

2.3 Psycholinguistic Theory

Psycholinguistics can be defined as the study of those mental processes that underlie the acquisition and use of language [26]. Research in this field relies on observational studies and hypothesis-driven coding, which is related to *grounded theory* [15]. As described in [37], the field is multi-disciplinary in nature and can be approached

from several different perspectives, including: phonetics/phonology, syntax, semantics, and the patterning of verbal and non-verbal behavior. An important issue at the interface of the verbal and nonverbal domains is that of defining an appropriate unit of analysis and ultimately explanation. On the verbal side there is the system of linguistic categorical description; while on the nonverbal side there exists a host of alternative descriptive frameworks developed for specific analytic goals that may or may not be translatable to one another [8]. To achieve understanding of the verbal-nonverbal interface we employ a unit we call the "hyperphrase," a nexus of converging, interweaving processes that cannot be totally untangled. The hyperphrase is thus a higher-level unit for understanding nonverbal and verbal behavior.

The hyperphrase unit is based on the growth point, GP, concept. The GP is a minimal theoretical unit of cognition during speech production and comprehension [28]. GPs are inferred from speech-gesture synchrony and co-expressivity. The GP is meant to be the initial form of a thinking-for-speaking unit out of which a process of speech organization emerges pulse-by-pulse. Further elaborating on a communication pulse, it can be defined as follows:

> A "pulse" is a unit of speaker effort, encompassing prosodic highlighting, discourse highlighting, a gesture phrase; also, gaze, posture, and other dynamic factors clearly, then, a judgment reflecting the analysts final hypothesis concerning the organization of the example under analysis. [12]

A catchment is a term for discourse units inferred from gesture information. Catchments are recognized when gesture features recur in at least two (not necessarily consecutive) gestures. We hypothesize that mental imagery in a GP generates the gesture features; recurrent imagery suggests a common discourse theme. A catchment is a kind of thread of consistent visuospatial imagery running through a discourse segment. By discovering a given speaker's catchments, we can see for this speaker what can be placed together into larger discourse units, what meanings are isolated and thus seen by the speaker as having *either* distinct or less related meanings.

For example, a given catchment could be defined by the recurrent use of the same trajectory and space with variations of hand shapes. This would suggest a larger discourse unit within which meanings are contrasted. Essentially, this is a heuristic model that explains how recurrent idea units (growth points) arise during discourse production. The hyperphrase may be summarized as a comprehensive view of verbal and non-verbal packaging in communication.

2.4 Speech and Gesture

The interplay of speech and gesture in communication can be seen as a dialectic (the two modalities exist in opposition to each other); while speech expresses the immediate and sequential, gesture expresses the global and synthetic [27]. In other words, this dialectic (and Growth Point theory in general) give rise to a heuristic model that gives criteria by which we can understand communication and interaction as a series of unfolding episodes, that mainly are constructed from cohesive units. Exceptions

to this occur when speech falters or shuts down: these occasions mark points where language functions as "stop order". The features of gesture that contribute to GP are recurring and, in concert with speech, serve to structure discourse [27, 28].

Chapter 3

Methodology

Because of the intricacies of human language production and embodied behavior, from the perspective of psycholinguistics, discourse analysis requires researchers to have a grasp of their source data (audio/video) and features of communication/interaction as evidenced by multiple annotation layers. It requires access to multiple types of information simultaneously and forms the basis for multimodal analysis. This makes the multiple-linked representation [23] a powerful way to make sense of multimodal data. In line with this, in order to support psycholinguistics research using computer-aided analysis, the following elements must therefore be assembled in a coherent system:

- information visualization (e.g., data plots, symbolic entities)
- media players
- interfaces for creating new annotations
- a database management system

We supported these features by employing a strategy of rapid deployment with iterative refinement during the evolution of the subsequent system (MacVisSTA). This

is an example of *user-centered* design (as in [33]), in which the author conducted several on-site interviews with expert annotators. One of the outcomes of this approach was to incorporate elements that would help such users in the design of the system. In addition, the author sought to understand the nature of the annotation tasks that occur in Psycholinguistics research. In the remainder of this chapter, we explore the nature of Psycholinguistic annotation, which directly influenced the evolution of MacVisSTA.

3.1 Psycholinguistic Annotation

Investigating several behavioral cues (gesture, speech, gaze) involves combining multiple layers of manual annotation and speech transcripts; these are all metadata that may be visualized on a common timeline, and importantly, are utilized in Psycholinguistics. The following section (3.1.1) explains Psycholinguistic coding as it relates to metadata visualization and explains the role of *hypothesis-driven* coding in Psycholinguistic research.

3.1.1 Coding and Metadata Visualization

After initial tests with annotating an early dataset (specifically, a multi-camera, multi-channel meeting recorded at AFIT on June 1, 2004) a request by the annotators for adding color to the time-tagged intervals resulted in a new capability for visualizing patterns in the event data. In particular, it became possible for a specific label to be associated with a color (e.g., gaze to C, gaze to D, etc.). However, an

alternative to this 1:1 mapping occurs when the color of the tag is used to carry information that is independent of the label. For example, when annotating gesture type, different colors can be chosen according to whether the gesture is a *beat*, *deictic*, *iconic*, or *metaphoric* – while the labels could be LH (left hand), RH (right hand), or BH (both hands). In other words, when gestures are annotated using color to indicate their type, the labels (LH, RH, BH) only have bearing on the gesture handedness. These aspects of annotation illustrate that the labeled intervals are important objects in their own right, but equally important are the secondary properties that may be embedded along with them – which are essentially *attribute-value* pairs – and act as supporting *metadata*.

The use of color also relates directly to the method of inquiry. Psycholinguistic annotation is *hypothesis-driven*, with observations assembled in order to test the multimodal language theory that is the basis of this research. This key feature gives rise to multiple, flexible coding schemes that can be illustrated in the following examples (Tables 3.1, 3.2, and 3.3).

Table 3.1: Alternative hypotheses for intervals annotated for gesture

H_0 The hand movement is not related to an idea unit
H_1 Gesture stroke phase is *deictic* (i.e., referring to some idea unit, usually concrete)
H_2 Gesture stroke phase represents something abstract as evidenced by the speech (i.e., the gesture is *metaphoric*)
H_3 Gesture stroke phase is meant to be pictorial and bears a close formal relationship to the semantic content (i.e., *iconic*)

All annotation is subject to revision. This is especially true when deciding how best to represent the meaning intended by a speaker as evidenced by their gestures,

Table 3.2: Alternative hypotheses for intervals annotated for gaze

H_0 Gaze at neutral space
H_1 Gaze at C
H_2 Gaze at C/D (i.e., the gaze fixation is indeterminate)
H_3 Gaze at D
H_4 Gaze at E
H_5 Gaze at F
H_6 Gaze at papers
etc.

Table 3.3: Alternative hypotheses for intervals annotated for coreference

H_0 no coreference (i.e., the interval is not marked)
H_1 *object*: coreference is a task-related object
H_2 *para*: coreference includes something about the discourse (e.g., "that")
H_3 *meta*: includes something about a speaker's viewpoint

gaze, and speech (syllable by syllable). The decision-making process is something that an analyst must learn as they gain experience observing *that* individual. As described in [12] the process necessitates a multi-pass approach to analysis (Table 3.4) and is "backward-adjusting" as the analyst proceeds through the video.

A key to carrying out psycholinguistic analysis is perhaps best explained as follows [12]:

> The exercise of gesture analysis and annotation is necessarily backward-adjusting. As the analyst moves forward through the narration from segment

> to segment, insights accumulate about how the particular speaker typically executes certain types of gestures, the speaker's handshapes, what is typical of the speakers gestures during intervals of dysfluency (for instance, holding versus repeating gestures across such intervals); on and on. Multitudes of tiny insights accumulate. An interval of gesturing at discourse segment no.47 may require annotation that calls into question how an interval at segment no.33 was annotated (at any level: gesture 'type', gesture meaning, any aspect). The analyst is obliged to return to segment no.33 and re-do the annotations or add a note of some kind.

The multi-pass approach gives rise to two important properties of this research: a requirement for information access across multiple tiers (as many as needed to capture interesting phenomena), and provision for multiple hypotheses that may need to be explored, accepted, or rejected as alternatives, possibly in parallel. These aspects give rise to the requirements for flexible visualization and annotation, as well as access to a database management system, topics which are explored further in Chapters 4 (Design and Implementation) and 5 (Corpus Building and Database Management).

Table 3.4: Multi-pass, hypothesis-driven approach to multimodal analysis (adapted from [12])

Watch the complete product of the elicitation
Transcribe the speech (including partials and unintelligibles)
Organize the speech into short utterances
Annotate points of primary peak prosodic emphasis
Bracket the gesture phrases
Annotate gesture strokes (i.e., preparation-stroke-retraction, holds, classify them, etc.)
Re-organize short utterances into speech/gesture "production pulses"
Revise earlier observations as needed to advance the analytic goals
(i.e., "backward-adjusting")

Chapter 4

Design and Implementation

This chapter presents an architectural overview of a system that has been implemented for use in multimodal visualization, annotation, and analysis. This was first developed for the Silicon Graphics, Inc. UNIX platform, and included several visualization and multimedia components (Figure 4.1).

The new system MacVisSTA (Macintosh Visualization for Situated Temporal Analysis) supports both visualization and annotation of time-based data in conjunction with audio and one or more videos [40]. This system (Figure 4.2) uses a multiple-linked representation where all components are time-synchronized and can act both as controllers and displayers [23]. The unifying factor in MacVisSTA for time-synchronous analysis is the current time focus. Each data object (video, audio, motion trace) may be represented within MacVisSTA in a display component. A data object can be multiply represented in more than one component simultaneously. The current time focus is represented in each display component in a manner that is compatible with that component. For a video display, the current time focus is displayed as a frame/time counter and by the current frame displayed. Employing multiple linked representation, each display component responds to changes in the current time focus and is able to manipulate the system-wide current time focus.

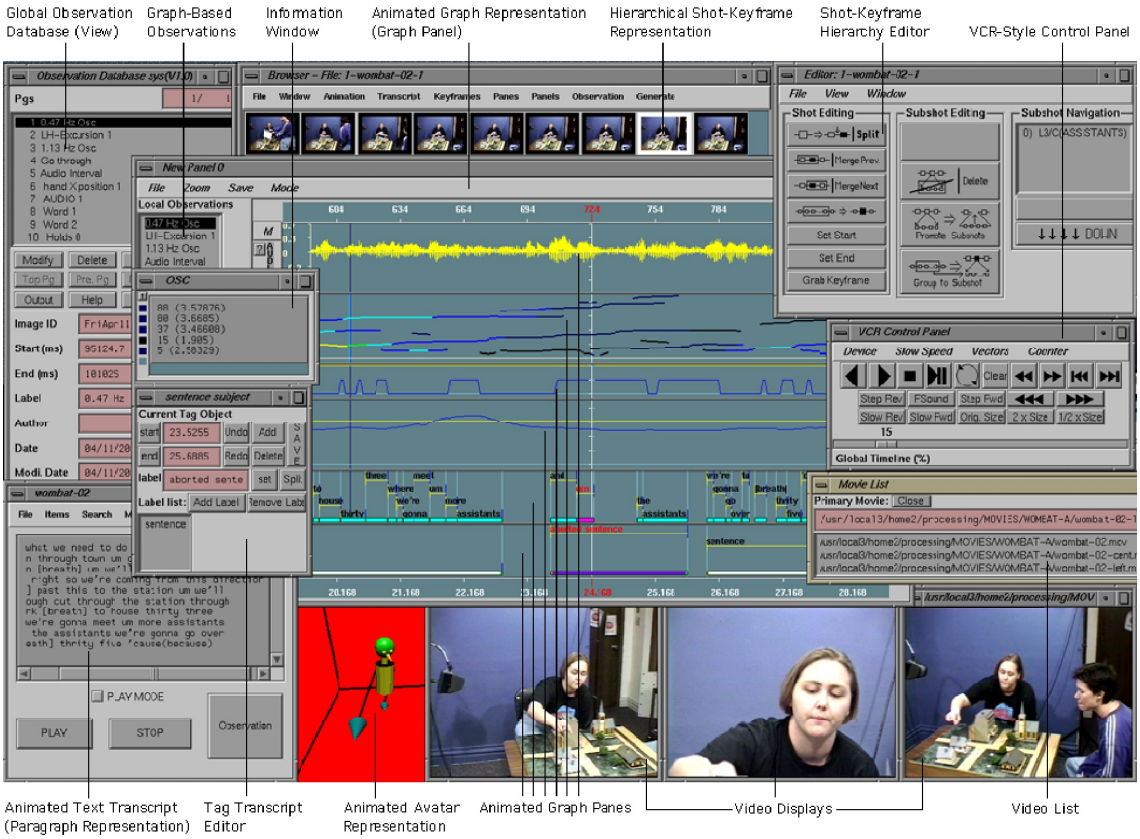

Figure 4.1: VisSTA with all components (photographs obtained from a research survey and are used with permission)

For example, the user can change the current time focus by clicking anywhere in a time-series plot, and all the system components will update to reflect the current time focus. This permits brushing and linking for multiple data types.

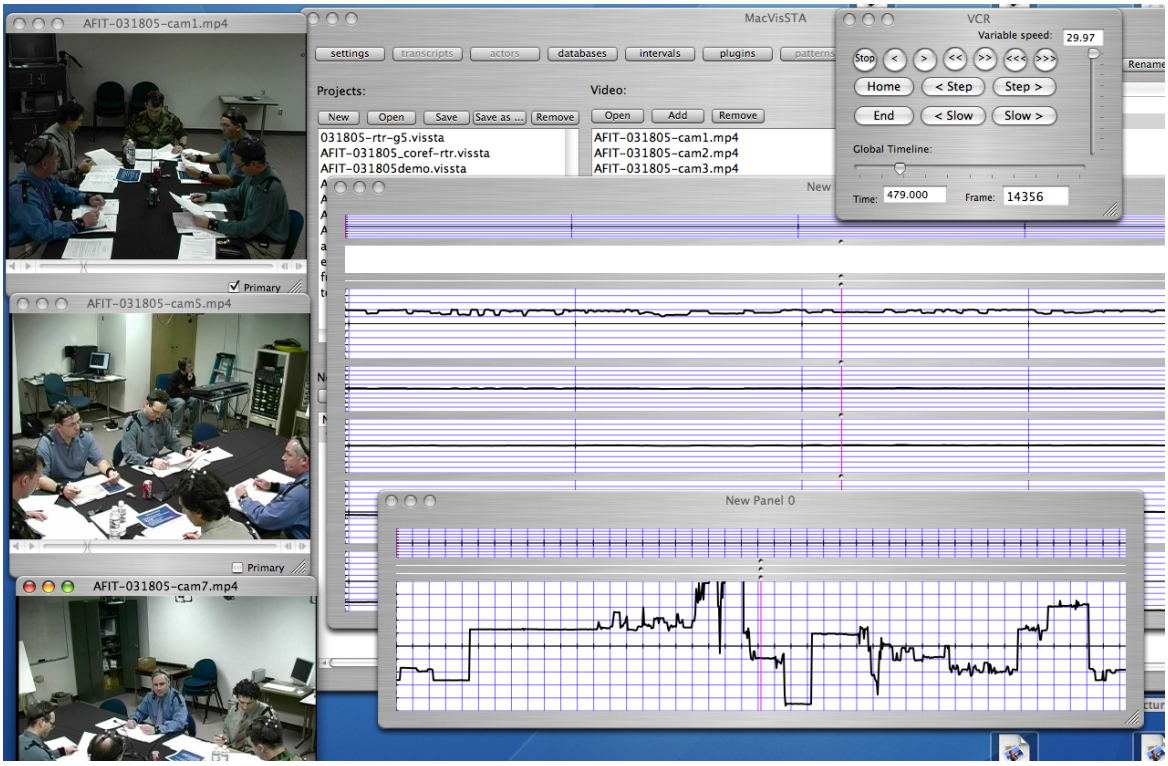

Figure 4.2: MacVisSTA interface (photographs obtained from a research survey and are used with permission)

4.1 Data Types

MacVisSTA handles four different time synchronous datatypes: multiple videos, audio data (embedded in the media players and as audio signal plots), time series data for continuous signals (e.g., x, y, z plots of hand position as a function of time), music-score (i.e., time-occupying) objects. In MacVisSTA, multimodal annotations

are represented as music-score objects in standoff XML. Each set of non-overlapping music-score objects is organized as a music-score tier.

In addition, MacVisSTA maintains two abstract meta-data objects: notes and notebooks. Notes are arbitrary time-occupying entities that may overlap one another in time. Each note contains a label, a free-text description, and a note source. Notes are conceived as units of user observation on the data. A note source is the identity of the data entity on which the note or observation was made. A notebook is a collection of notes. This permits the user to organize observations for analysis and for presentation. All of these notes and notebooks are stored as metadata in a project file.

4.2 Project-based Representation

MacVisSTA is a project-based system. A project file consists of a list of preferences, a list of table names for searching, a pane and panel library, observation/notebook data, and a list of associated movies. *Panes* are tiled surfaces for visualization; they are named as such because their borders appear similar to window panes. *Panels* are assemblies composed of individual *panes* (drawing surfaces).

The system separates user data from source data; this separation is necessary so that different users may have access to shared data yet prevent their annotations and project files from overwriting each other. Each project uses a dataset name as the prefix for the names of data files; since each data type has a unique extension, the system can build the appropriate filename to access the data as necessary.

4.3 Display Components

MacVisSTA exploits the multimedia capabilities of QuickTime to handle its video and audio media and to maintain system-wide temporal synchrony. MacVisSTA's video display component is essentially a QuickTime player window. Hence the system is able to handle any media type that QuickTime can play (e.g. MPEG-1, MPEG-2, MPEG-4, MP3, AAC, AIFF, WAV, MJPEG-A, MJPEG-B).

In addition to video playback, we use graphical plots to represent data such as audio waveforms and motion traces. We employ an animated "strip chart" metaphor to visualize the set of data types that may be represented in graphical form in which the graph horizontal axis represents time. These types include audio signals, continuous time-series signals, and annotation data (as music score objects). The display component that contains each strip-chart representation in MacVisSTA is a pane object. Panes are assembled into panels. This permits flexible visualization of only the desired data that an analyst may wish to explore by providing multiple alternative assemblies within and across datasets (for example, Figure 4.3 and 4.4).

4.4 Plug-in architecture

MacVisSTA uses an open architecture that supports "plug-ins", either as general-purpose media controllers that remain time-synchronized with the rest of the system, or at the level of panes. Using pane "plug-ins", new visualization components can be

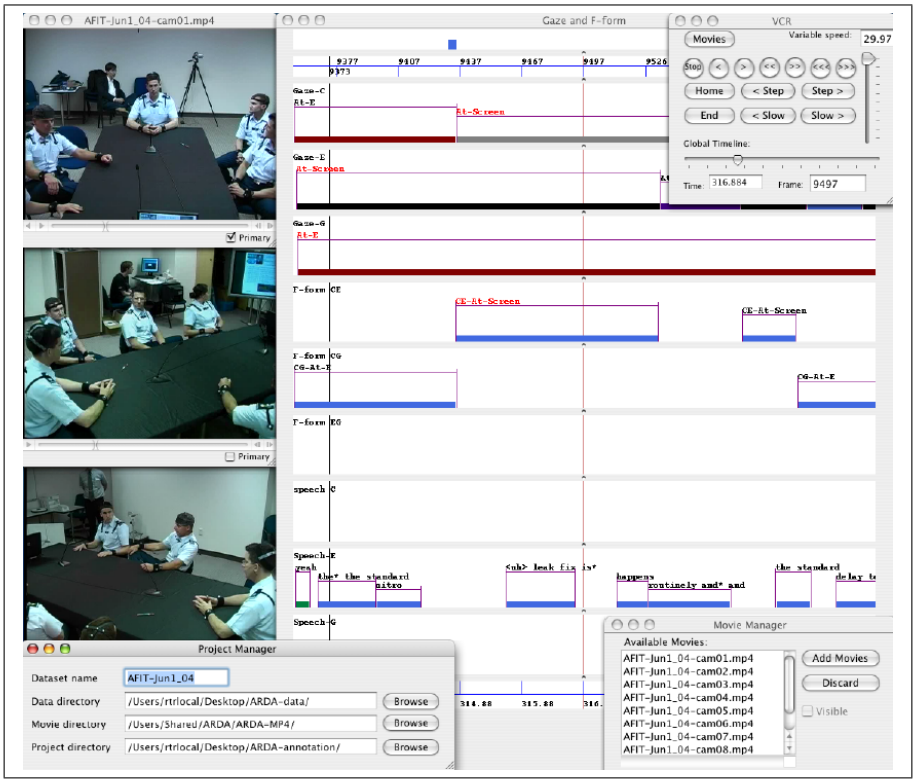

Figure 4.3: Sample configuration with AFIT meeting data (photographs obtained from a research survey and are used with permission)

created to extend MacVisSTA to handle multiple data types. Several consequences of using a plugin approach are explored further in section 5.5.

4.5 Interaction design

MacVisSTA integrates multiple videos that are time-synchronized with animated strip charts. Since multimodal analysis relies on video, we designed the system to

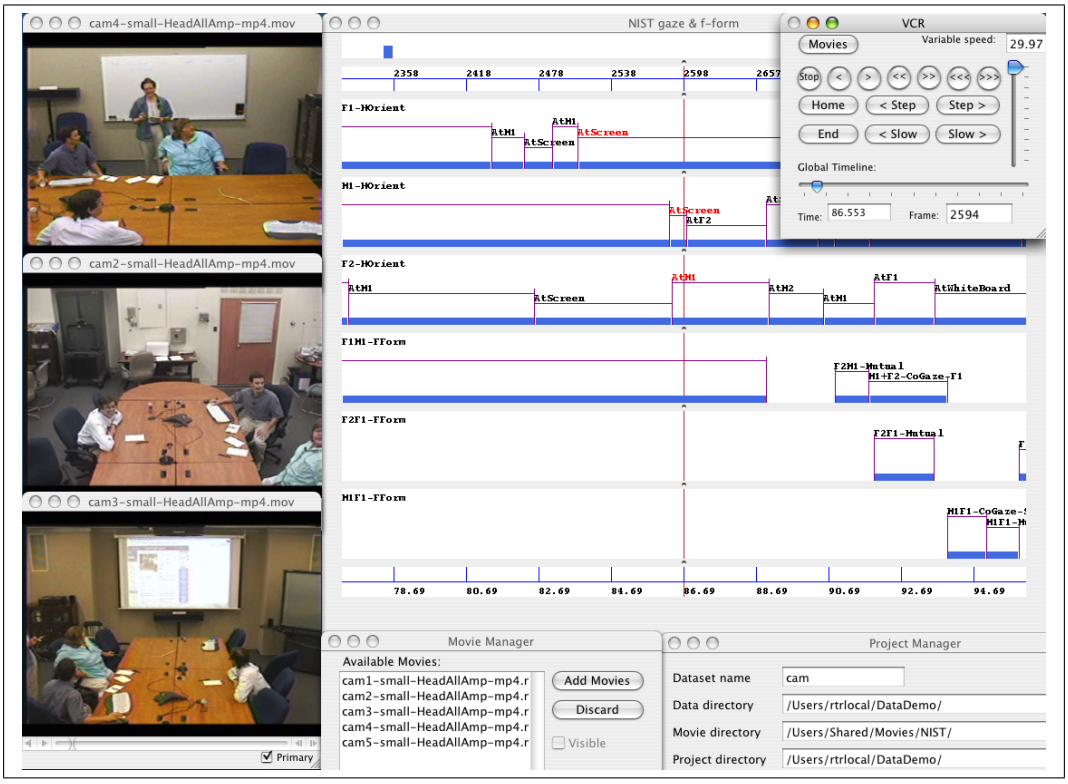

Figure 4.4: Sample configuration with NIST meeting data (photographs obtained from a research survey and are used with permission)

handle simultaneous display of multiple video streams that are time-synchronized. MacVisSTA is designed to handle analysis at varying timescales. This is relevant to human discourse production, which may be examined at the level of single utterances to longer topical segments. The key interface of the system is a VCR-style control panel through which the user can advance to any point in time, change the frame rate (i.e. frames per second) of video playback, or single-step forward or backward through

each video frame. MacVisSTA uses Apple's QuickTime and can play both forwards and backwards and at variable speeds. Finally, the graphical plots are rendered in a zoomable interface for navigation at multiple scales.

4.6 Music-Score Annotation

In annotation panes, the user can create and edit time-occupying objects. An example interface for editing an annotation appears in Figure 4.5.

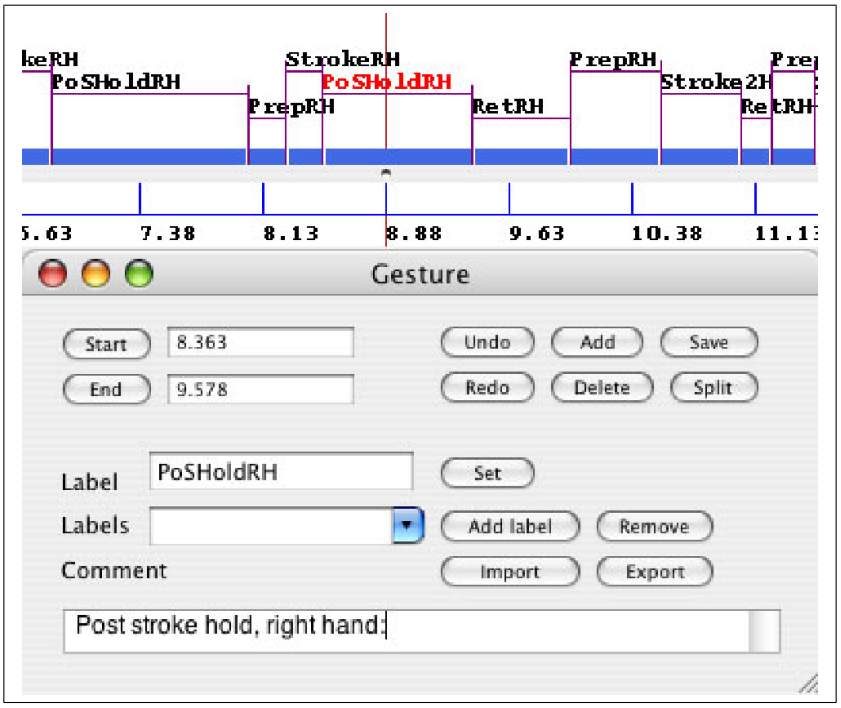

Figure 4.5: Annotation editor

Users can edit annotations by adding, deleting, or splitting objects. Each object has a begin time, an end time, a label, and comments field. The user can set the

extent of each time-occupying entity in three ways. For convenience, a label list can be created to store frequently-used labels that the user can choose from. Since annotation layers are managed as specialized panes, they are easily displayed in the same panel as graphical plots. As a result, the system provides a convenient way to combine annotation and visualization. In addition, annotations are stored in "standoff" XML format (i.e., the tags are stored separately and refer to intervals in digital media).

4.7 Notes and Notebooks

The notebook system was designed for marking time intervals that may span minutes, and is useful for quick, informal analysis. A user may highlight and store any number of time intervals in multiple panes, which correspond to phenomena in the video that the user is interested in. Highlighting is done interactively using click-and-drag (or keyboard shortcuts) to mark an interval in an animated pane. The user can enter a comment and give a label to a notebook entry. These notes are kept in a library that is part of the user's project.

4.8 Open Source Components

Besides several custom modules, MacVisSTA leverages several open source components that have been integrated into a comprehensive system, summarized in Table 4.1. An overview of the MacVisSTA architecture appears in Figure 4.6.

Table 4.1: Open source components used in development of MacVisSTA

Component	**Purpose**
Apple property list	Datatype for project settings, annotation, etc.
Drag and drop outline view	Drag and drop view for panes, panels, etc.
Drag and drop view	Drag and drop for importing files, etc.
Embedded Python interpreter	Provide Python bindings for future scripting additions
GraphX Framework	Plotting data in Objective C (scatter plots, charts, etc.)
Java Database Connectivity	Driver for uploading to MySQL database (JDBC)
MySQL C interface	Interface to MySQL database using C/C++
OpenGL bitmap fonts	Rendering text in OpenGL views
PAPluginProtocol	Protocol to register new plugins
RBSplitView	Provide flexible arrangement and re-sizing of tiled views
SQLite	Embedded SQL for querying annotation data
SQLite Objective C wrapper	A wrapper for working with SQLite tables etc.
VisSTA	C++ architecture, UML design, specialized routines etc.

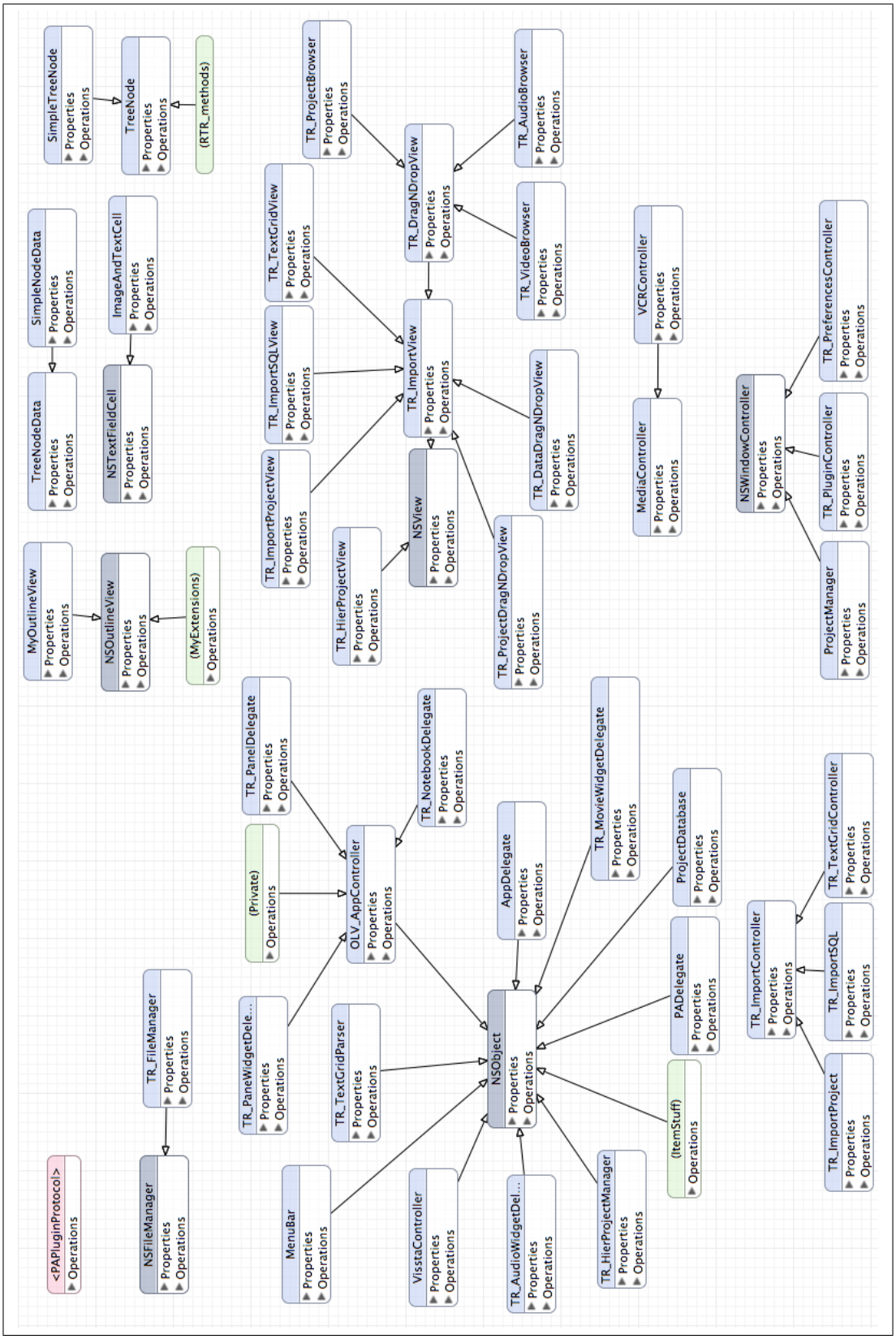

Figure 4.6: Key modules in MacVisSTA's design

Chapter 5

Corpus Building and Database Management

As described in Chapter 4, we created an interface providing access to the following data:

- Multimedia
- Continuous data (e.g., motion tracking)
- Supporting metadata (e.g., transcripts, annotations)
- Project details (e.g., the details of analysis: segmentation, visualization, etc.)

This requires a corpus-building process, in which we anticipate the need to separate source data from interesting metadata, such as a user's exploratory, evolving analysis/analyses in different stages of completion. We adopt a concept of data *spaces* that keeps user data apart from shared data.[1] As a result, datasets can be shared across users (and sites) in their canonical form, by keeping such data in privileged, read-only locations. The minimum requirement during this assembly process is to use the dataset name as the first part of all data files. This allows us to keep source data and metadata organized as the additional metadata layers are completed over time

[1]The separation of shared data from the user space also prevents collaborators from overwriting or contaminating each other's work.

(i.e., during corpus assembly). We keep working copies of annotation data in a user's local directory, until they are ready to be shared with the group. During assembly, we take the latest version of the outputs of different group members and enrich the dataset with the different data layers (e.g., speech transcriptions and forced alignments, sentence unit annotation, other multimodal behavioral and psycholinguistic coding).

A summary of the workflow necessary for a comprehensive treatment of the data appears below (Figure 5.1), which features all arms of the collaboration in this research: at AFIT[2], elicitation of the meeting and out-of-vocabulary resolution; Purdue/Maryland, rich metadata such as sentence unit (SU) markup, forced alignments of speech, analysis of floor control, etc.; University of Chicago, rich transcription, psycholinguistic analysis, etc.; VT[3] and UIUC[4], application of computer vision techniques to automate event detection.

A practical result of our collaboration across several institutions was that we needed to distribute source data first (i.e., by shipping MPEG-4 compressed videos and down-sampled audio), and then to make supporting metadata and processing results available to the entire group incrementally. We achieved this by using a collaborative, web-based tool developed at Virginia Tech, TeacherBridge [5, 21], which resulted in rapid delivery and immediate visibility of new metadata. In cases where files exceeded 10-12 megabytes (MB), we uploaded these to a different server and provided hyperlinks; further, we used individual web pages to organize information

[2]Air Force Institute of Technology
[3]Virginia Tech
[4]University of Illinois at Urbana-Champaigne

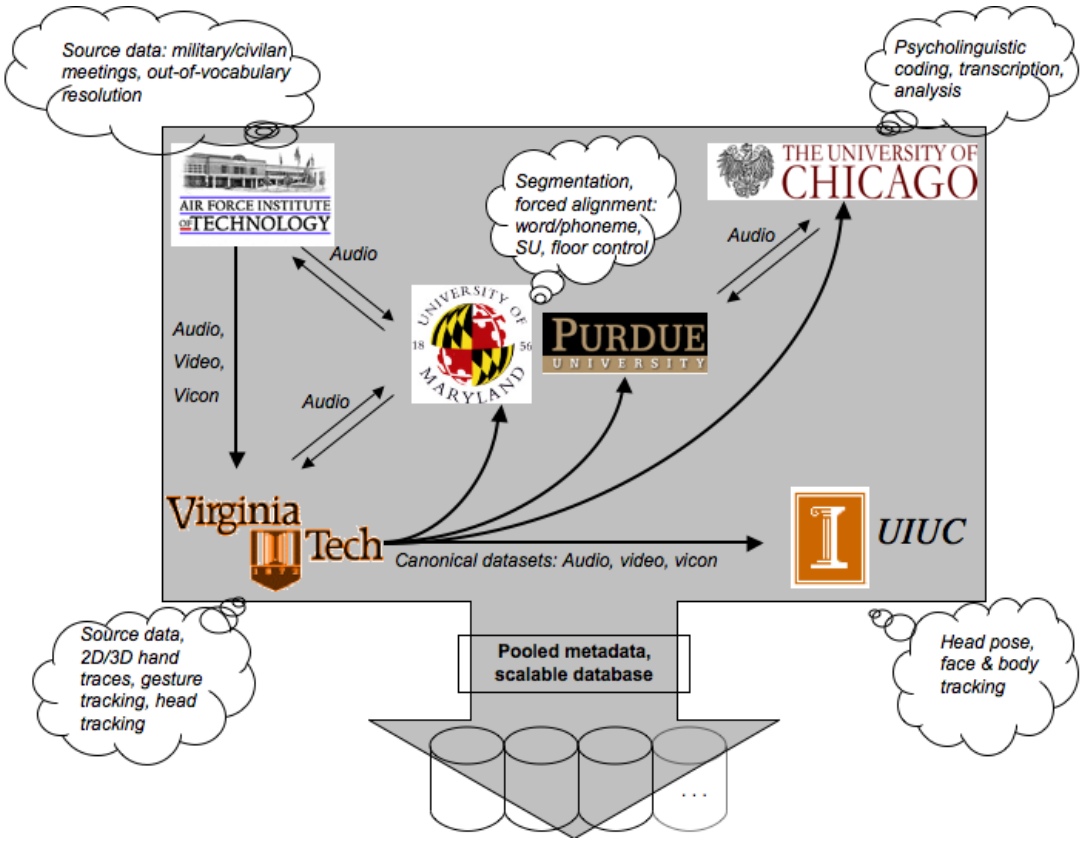

Figure 5.1: Workflow diagram of for management of source and derived data

about each dataset. TeacherBridge provided an effective means of sharing and documenting our data, and it was an important intermediate in building our multimodal corpora and, ultimately, our shared database.

With MacVisSTA we provide a lightweight interface for making observations about the data streams using a note/notebook metaphor. We also designed the interface for fine-grained analysis using non-overlapping, symbolic entities, thus permit-

ting accumulation of event data spanning several types of language and interaction phenomena (such as gesture, speech, gaze, co-reference, etc.). In order to aggregate the results of the many annotation layers and analyses for all of our datasets, and to facilitate further analyses of the metadata, we interfaced to a database management system (DBMS).

We decided to use databases that required minimal configuration from the user. Thus, we established a MySQL database[5] hosted on a server (vislab.cs.vt.edu). This allowed the author to manage the database (at Virginia Tech), and simply provide a username/password to collaborators for remote access, which could be specified from within MacVisSTA (Figure 5.2). We also decided to keep notes and notebooks apart from the shared database; instead, the shared database was strictly used for all of the fine-grained annotation layers (non-overlapping symbolic entities). We used MacVisSTA to assign the different behavioral streams to meeting participants. Annotations were mapped to a defined schema (Table 5.1) and committed to the database (Figure 5.3). This approach supported annotation and analysis among several researchers in the collaboration, permitting speech and psycholinguistic metadata to be pooled in a shared database.

[5]MySQL is an open source relational database management system that uses Structured Query Language (SQL)

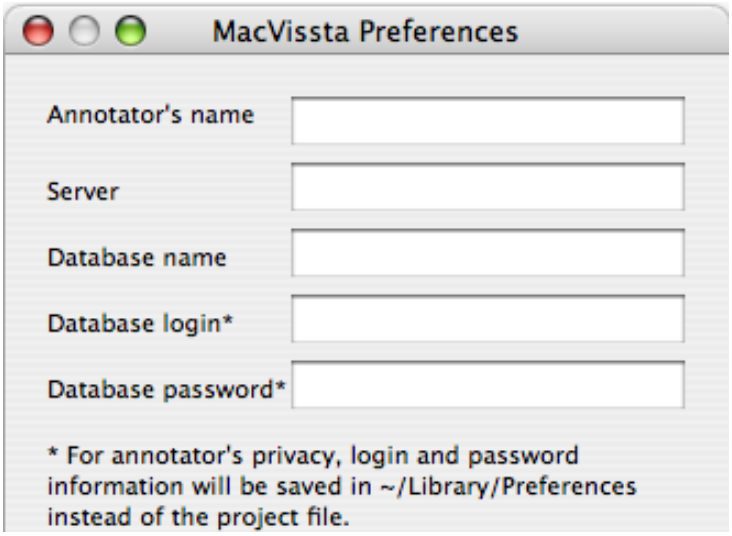

Figure 5.2: MacVisSTA preferences

Using a DBMS, we were able to write queries to obtain statistics such as the following:

- Numbers of each type of gesture annotated
- Number of gestures annotated in a meeting
- Number of gaze shifts observed for each participant in a meeting
- Number of certain vs. uncertain gaze fixations (i.e., estimating observer confidence)
- Frequency and rank order of references (elements in the discourse)
- A matrix of gaze sources vs. gaze targets in a meeting

Table 5.1: Annotation schema

Field	**Description**
dataset	Name of the dataset
actor	Name/alias of person in video
stream	Behavioral feature (e.g., gesture)
begin	Start time of the event
end	End time of the event
label	Event label
annotator	Name/initals of person who made the annotation
comment	Comment on the event (optional)
reserved	Reserved for future use (e.g., as a foreign key)
date	Date of upload/submission to the database
id	Primary key (unique, system-generated)

5.1 Embedded SQL

We added an embedded SQL query engine (using "SQLite", a lightweight SQL) and interface in order to support querying of the annotations *offline* (i.e., while working locally). Importantly, the embedded SQL capability required no configuration by the user. Besides gaining insight into several of the aggregate behavioral features by using standard queries, we sought to understand the patterning of behaviors in more detail, described in Chapter 8.

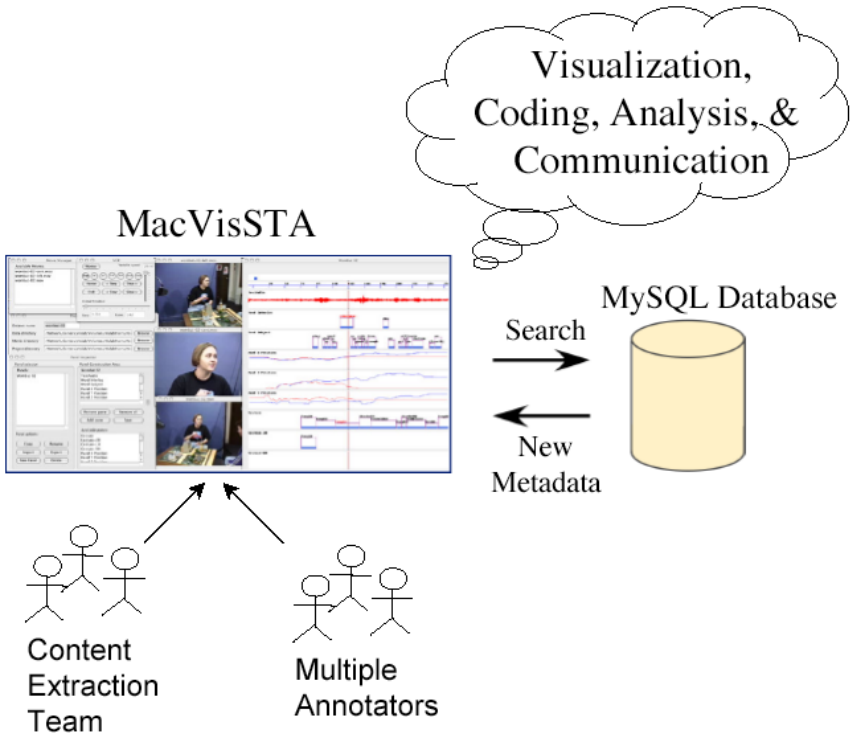

Figure 5.3: Visualization, coding, and analysis with database support

5.2 Database Table Structure

In both the SQLite and MySQL databases, the same table structure is used for all annotation data. By setting up the table this way, observational data can be pooled across datasets (when using MySQL). The embedded SQLite database is currently set up for use on a per project basis, but can easily change to be a persistent, local database. The table is created as follows:

```
CREATE TABLE 'mmdata' IF NOT EXISTS (
  'dataset' varchar(30) default NULL,
  'actor' varchar(20) default NULL,
  'stream' varchar(30) default NULL,
  'begin' float default NULL,
  'end' float default NULL,
  'label' varchar(50) default NULL,
  'annotator' varchar(20) default NULL,
  'comment' varchar(250) default NULL,
  'linktable' varchar(20) default NULL,
  'date' date default NULL,
  'id' int(11) NOT NULL auto_increment,
  PRIMARY KEY  ('id')
) TYPE=MyISAM
```

5.3 Significance of Database Management Systems

This chapter illustrates how we assemble metadata for analysis, which in turn can be realized collaboratively and, with sufficient engineering, transparently. MacVisSTA is a hybrid system that facilitates visualization, coding, and analysis, and interfaces with a DBMS to support many kinds of analytic goals.

5.4 Relation to other Information Technology Frameworks

Given the scope of the datatypes and the broader collaboration described above, management of source data, derived data, and analysis/analyses quickly becomes a complex information management problem. The process of *corpus assembly* described above may alternatively be viewed as a specific instance of creating and maintaining a Digital Library – in this case, for the purpose of enabling research in computer vision,

speech recognition, and multimodal behavior and psycholinguistics. The heart of the matter for this work, then, is both accumulating enough data, and mustering sufficient technical "assists" (e.g., through innovation in software – integrative interfaces, recognition algorithms, automation) to carry out cogent analyses of datasets. We are as much concerned with corpus assembly as we are with preserving the details of our analyses, which we expect will contribute directly to new scientific discoveries.

As a result, access to heterogeneous datasets can be thought of as miniature Digital Libraries. Treating multimodal corpora as Digital Libraries allows us to situate our research, and abstract from the focus of the research. Because of the strategic importance that Digital Libraries research has in multiple diverse fields, we can understand the multi-disciplinary research described in this work by viewing it in terms of a comprehensive framework known as 5S: Streams, Structures, Spaces, Scenarios, and Societies [16]. From this perspective, we can summarize the analysis problems, design issues, and interactions in Table 5.2.

5.5 Coupling of Multimodal Corpora to Information Visualization and Visual Schema

Because all of the data for each meeting recording are synchronized to a common timeline, multiple comparisons can be made across the different information streams. In MacVisSTA, visualizations are first-class citizens (objects), and are either instantiated as needed by the application, or can be inspected as *visual schema* [34]. As a result, this permits introspection of schema that are available ("live", at runtime)

Table 5.2: Application of 5S framework to multimodal corpora and analysis

Stream Model
Source data (i.e., multimedia); time series; continuous data; multimodal behavior (gesture, speech, gaze, etc.) and associated queries; transcripts; annotation and event data; continuous vs. discrete data
Structural Model
Annotation and database schema; structure of communication, interaction (e.g., episodes, discourse segmentation); role hierarchy
Spatial Model
2D, 3D reconstruction and person tracking; user vs. shared data space; multimodal cue density; probabilistic space; visualizations
Scenarios Model
Meeting scenario; analysis scenario; use case
Societies Model
Annotators; computer scientists; archivists; social scientists; interaction design; interface design; etc.

by plugins. Extension through plugins will allow future analysts to interact with the data sources in ways that have not been initially implemented. Leverage points for plugins exist at the level of media controllers, visualizations, and annotations, all of which are points of entry in MacVisSTA. In the next chapter, a review of annotation standards is presented in order to motivate an initial effort aimed at greater tool interoperability, described in Chapter 7.

Chapter 6

Annotation and Metadata Standards

The diversity of tools and approaches to digital media has resulted in several efforts to standardize annotation and metadata, which has broad relevance to multimedia bitstreams in general [19], such as how to process, encode, and access any multimedia content anywhere and anytime. This chapter presents standards that have made an impact on linguistic, multimodal, or other kinds of annotation, such as multimedia and hypertext.

6.1 AGTK

`http://agtk.sourceforge.net/`

The Annotation Graph Toolkit (AGTK) was created to facilitate development of new annotation tools based on Annotation Graphs (AG), which are a formal framework for representing linguistic annotations of time series data. AG provide an abstraction from file formats, coding schemes and user interfaces, resulting in a logical layer for annotation. Formally, AG consist of nodes and arcs, where the nodes are time references and arcs contain a record's information (such as a label). Annotation Graphs were created as a general-purpose data structure for linguistic annotation.

AGTK is implemented as a C++ library and has interfaces to Toolkit language (Tcl) and Python. Sample applications include MultiTrans for transcribing multi-party conversation, TableTrans for observational coding of audio, TreeTrans for syntactic annotation, and InterTrans for interlinear text transcription (this uses a subset of AG formalism; interlinear text is a standard form for displaying a source text aligned with other linguistic annotations such as phonological, morphological and syntactic analyses, glosses, and translations) [4].

6.2 ATLAS

`http://www.nist.gov/speech/atlas/index.html`

ATLAS (Architecture and Tools for Linguistic Analysis Systems) was an initiative involving NIST, Linguistic Data Consortium (LDC), and MITRE. A Java implementation of the data model is available as open source; jATLAS is on SourceForge: `http://sourceforge.net/projects/jatlas/`. ATLAS addresses an array of applications' needs spanning corpus construction, evaluation infrastructure, and multimodal visualization [3]. The ATLAS framework abstracts across diverse linguistic annotations and is also based on Annotation Graphs [4]. ATLAS consists of four parts: an annotation ontology, an application programming interface (API), an interchange format, and a Meta-Annotation Infrastructure for ATLAS (MAIA). Thus, an overarching goal of the project is to facilitate annotation interchange and reuse. The framework provides an API consisting of three layers: application, logical, and phys-

ical. The interchange may either use ATLAS Interchange Format (AIF) level 0, which is equivalent to AG, or AIF level 1, which is hierarchical.

6.3 MATE

`http://mate.nis.sdu.dk/`

MATE (Multilevel Annotation, Tools Engineering) was motivated by the need to standardize the annotation and analysis of dialogue corpora. It was created as a preliminary standard and workbench for language annotation. The MATE workbench (a Java tool) was based on an "interface engine" that would allow the behavior of its interface to be specified using stylesheets. MATE was essentially the precursor to the Natural Interactivity Tools Engineering (NITE) XML Toolkit (described below). Similar to ATLAS, it abstracts from a variety of linguistic annotations in an effort to permit interchange.

6.4 NITE AND NXT

`http://nite.nis.sdu.dk/aboutNite/`

NITE can be viewed as evolving from MATE (including use of a similar logo), and takes a comprehensive approach to human language. While MATE focused on annotation of spoken dialogue, NITE is oriented toward multimodal phenomena. In particular, the NITE tools are designed for multi-level and cross-modal annotation, information retrieval, and exploitation of multi-party interactive phenomena, including

human-human and human-machine dialogue. The NITE tools are the NITE workbench (for Microsoft Windows), the NITE XML Toolkit, and Noldus Observer.

The NITE XML Toolkit (NXT) was created for work with multimodal, spoken, or text language corpora, and includes a set of libraries and an end user tool written in Java. It evolved from a collaboration between the University of Edinburgh's Language Technology Group (LTG), the University of Stuttgart's Institut für Maschinelle Sprachverarbeitung (IMS), and the Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI). NXT is being used in support of the Augmented Multiparty Interaction (AMI) and International Computer Science Institute (ICSI) meeting corpora.

6.5 MPEG-7

`http://www.chiariglione.org/mpeg`

MPEG-7 is an International Standards Organization (ISO) standard developed by Moving Picture Experts Group (MPEG) [25]. Formally called the "Multimedia Content Description Interface", MPEG-7 is a standard for describing multimedia content data that supports interpretation of the meaning of information, and can be attached to timecode in order to mark events, which can be relayed to or parsed by a device or other software. Thus, it is not a standard for encoding moving pictures and audio, such as MPEG-1, MPEG-2 and MPEG-4. Rather, MPEG-7 is intended for use by a broad range of applications. In particular, MPEG-7 is designed to standardize:

- description schemes and their descriptors
- the language used to define the schema (i.e., Description Definition Language)
- a means of encoding the description(s)

The syntax for the Description Definition Language is XML.

6.6 VERL and VEML

The Video Event Representation Language (VERL) is a way of expressing video events in language that resembles first-order predicate logic [29]. For example, VERL allows one to define a taxonomy of entities and their properties, as well as rules that describe how entities interact. Events can be defined individually or as parts of *composite* events, and in general an event will be recognized when a set of conditions are met by the required entities and in the proper order. (VERL can express *single-threaded* or *multi-threaded* processes, the latter having utility for multiple actors.) The entity taxonomy and rules describing events are encoded in Video Event Markup Language (VEML), an XML standard.

6.7 W3C's Annotea

`http://www.w3.org/2002/12/AnnoteaProtocol-20021219`

The Worldwide Web Consortium (W3C) has developed Annotea as a system for creating and sharing annotations of web documents. Annotea uses hypertext, resource

description framework (RDF) and XML, working in concert with web browsers to allow users to create and publish their annotations without modifying the source document. The annotations may be visualized when the user returns to view the same web page, or they can be displayed or processed by an external tool/application. This protocol has been established as a way to supply annotations that would promote development of semantic web technology.

6.8 Summary

The metadata standards presented above have some overlapping goals and features. First, they each aim to provide a common framework to support video annotation/analysis research. Use of a common standard in different settings will permit researchers to more readily share data and observations. Second, each standard deals with representing video events as well as temporal relationships. They all have an XML format. Presently, there is no direct mapping from one standard to another, perhaps because of the difficulty in creating a standard that is generally applicable, yet flexible and easy to use.

Annotation Graphs (AG) provide a general data structure for linguistic annotation; ATLAS extended AG to provide a hierarchical representation (if needed) and API. The more recent NITE and NXT appear to have features in common with AG/ATLAS, with some differences in implementation. With respect to VERL/VEML and MPEG-7, these also have a rich capability for representing

multiple entities and attributes in audio/video, and may potentially be unified since these have been expressed in Web Ontology Language (OWL) [14, 29].

The next chapter presents inter-conversion of MacVisSTA annotations to and from AG format as a first step towards increasing interoperability.

Chapter 7

Round-trip Conversion of MacVisSTA Annotations to and from Annotation Graphs

7.1 Towards Greater Interoperability

Because of the variety of tools and formats now in existence, a desirable goal is to achieve greater interoperability using a common exchange format. This has been explored in related work specifically addressing linguistic annotation [3]. The Macintosh Visualization for Situated Temporal Analysis (MacVisSTA) software was created as a multimodal analysis and annotation tool, incorporating time-synchronized multimedia and visualization components [40]. This chapter describes an approach to converting MacVisSTA annotations to and from Annotation Graphs, in an effort to facilitate annotation interchange[1].

7.2 Annotation Schema

MacVisSTA and Annotation Graphs (AG) both use XML to represent their data. The Document Type Definitions (DTDs) for each are outlined below.

[1]This exercise was organized for the Third International International Society for Gesture Studies (ISGS) Conference, Multimodal Annotation Tools Workshop, 2007

7.2.1 MacVisSTA Annotations

MacVisSTA annotation files use Apple's DTD for property lists. This permits each annotation to be instantiated as a native Cocoa/Objective C data structure. Allowable elements for property lists include *arrays, dictionaries, reals, strings,* and *data*. In particular, MacVisSTA annotations are stored as an array of dictionaries. Each dictionary contains at least a begin time, end time, and label, plus *optionally* a comment or color information.

7.2.2 Annotation Graphs

The Annotation Graph (AG) format is an XML representation of the annotation graph data model, and is used as the official format for annotation graphs. It consists of at least one *AGSet* containing a list of *Anchors* (times) that refer to some continuous signal; in addition, the *AGSet* contains a series of *Annotations* that encode *Features* of interest. In general the AG format may also encode supporting *Metadata*, a designated *Timeline*, and reference to a *Signal*.

7.3 Implementation

The annotation schema noted above permit mapping of the annotation formats to facilitate round-tripping. The round-trip conversion is implemented using Python 2.3.5 running on Macintosh OS X (10.4.9). The conversion uses two independent modules that can be invoked as batch processes. Each operates on one or more input

files and uses the Document Object Model (DOM) to parse XML; in particular, the modules use the Python *xml.dom.minidom* library, which is a lightweight DOM implementation. The conversion modules are described in greater detail below.

7.3.1 Converting from MacVisSTA to Annotation Graph

Given a list of MacVisSTA annotation files as input, this module outputs a file in Annotation Graph (AG) format that contains multiple tiers. Each time the program (macvissta2ag.py) is invoked with a file list, the resulting AG file includes all input tiers. The outline of the algorithm is as follows:

- open the MacVisSTA source file S
- parse the source file S as a tree structure using DOM
- traverse the tree to obtain all annotations elements
- create the AG tree T
- for each MacVisSTA annotation, obtain the annotation begin and end times ($b,\ e$)
- insert the begin and end times $b,\ e$ into an 'anchor list'
- for each MacVisSTA annotation, create a corresponding AG annotation A
- obtain child elements and add them as 'features' to A
- add 'anchor list' to T
- add each annotation A to the AG tree T

7.3.2 Converting from Annotation Graph to MacVisSTA

Given a list of AG files as input, this module outputs the corresponding MacVisSTA annotations, where each tier originating in the AG results in a separate MacVisSTA file. The outline of the algorithm is as follows:

- open the AG source file
- parse the source file S as a tree structure using DOM
- for each tier in S, create a MacVisSTA tree T
- obtain the 'anchor list'
- for each AG annotation, create a corresponding MacVisSTA annotation A
- obtain the annotation begin and end times (b, e) from the 'anchor list' and add to A
- for each feature, add as a child element to A

7.4 Loss-less conversion

These procedures were tested on sample MacVisSTA annotations containing 1000 or more coded events per file. This resulted in loss-less conversion of the data when converting to Annotation Graph and back (round-tripping). An excerpt of a *round-trip converted* file appears below, dispayed in a format that MacVisSTA can read:

```
<dict>
<key>beginTime</key>
<real>31.164497375488281</real>
<key>color</key>
<data>
BAt0eXBlZHN0cmVhbYED6IQBQISEhAdOU0NvbG9yAISECE5TT2JqZWN0AIWE
AWMBhARmZmZmgz7MzM2DP0zMzQEBhg==
</data>
<key>created</key>
<string>2005-07-13 11:27:05 -0500</string>
<key>endTime</key>
<real>31.464797973632812</real>
<key>text</key>
<string>at other's paper</string>
<key>tier</key>
<string>0</string>
</dict>
```

7.5 Observations

MacVisSTA can be classified as a system that uses a single timeline and multiple tiers, similar to Praat, Anvil, ELAN, EXMARaLDA, and others. The AG format permits specification of arbitrary units such as "seconds" or "milliseconds". MacVisSTA expresses times in seconds, which also appears in the resulting AG files. However, when converting from AG, either seconds or milliseconds may be used. Finally, because each annotation in MacVisSTA contains its own begin/end time, this results in duplication of values (originating from adjacent tags) in the 'anchor list'. The converted files use a file naming scheme that combines the date expressed as *YYYY:MM:DD*, a random number ($<10^9$), a label indicating type of file (either

to AG or back to MacVisSTA format), and the time as *hh:mm:ss* and *.xml* extension.

In the present conversion modules, there is no provision for handling the *Metadata* or *Signal* in AG files; this is because this type of information is stored in MacVisSTA project files. (A project file also uses Apple's DTD for property lists.) A complicating factor involves the relation of one or more tiers to graphical views, which is many-to-many. Nevertheless, in future the *Metadata* and *Signal* elements may be used to generate supporting project files. This difficulty actually points to a larger "corpus management" problem that arises when working with multiple streams of data (e.g., audio, video, extracted features, etc.), as well as the semantic or other relations between streams and tiers. Approaches to working with multimodal and multimedia corpora (and their underlying database structures and relations) in different tools will ultimately need to be addressed as well.

This chapter describes an initial approach to converting MacVisSTA annotations to and from Annotation Graph format using Python and *xml.dom.minidom* (lightweight DOM). This work represents a first step towards providing MacVisSTA with greater interoperability with other annotation and analysis tools. Conversions to other metadata standard formats may be possible as well.

Chapter 8

Temporal Analysis

The common, unifying factor in our datasets is that each recorded meeting has a synchronized timeline. As a result, these datasets are amenable to *time-synchronous, micro-analysis*, or in general, temporal analysis, in which all of the information streams are situated with respect to time, and which may be examined at multiple resolutions. Temporal analysis may be approached in several ways. In this research, we employ temporal analysis techniques as an aid in understanding communication and interaction. Our goal is to discover meaningful patterns of interaction that are revealed by sequences of observational data. A classic example for detecting repeated patterns is to create subsets based on counts of their frequencies [1], recursively. However, this technique is difficult to apply to multimodal data because we generally have four kinds of event streams:

- Short duration, short gaps (continuous speech, gaze)
- Long duration, short gaps (meeting phases)
- Long duration, long gaps (speaker turns)
- Short duration, short gaps (gesture phrase/phase)

Thus, we see a hierarchy of event types that will impact analysis, namely because it affects which patterns we may perceive, as well as which kinds of algorithms that may be used.

One possibility is to use temporal logic (a form of modal logic) as a basis for reasoning about relations between events, with several computer science researchers offering solutions [7, 9, 22, 31, 30, 32, 35]. This would be compatible with the Video Event Representation Language [29], for example. The problem with this approach, however, is that it would only translate the events without offering much more insight into their structure. A body of work has evolved to deal with issues of pattern analysis for understanding discourse, communication, multimodal meetings, and sequences of events in general. In this research, we are interested in understanding sequences and clustering of events such as gesture, speech, and gaze (multimodal cues) and characterizing their relationships as in [10, 17, 18, 11]. The importance of temporal relationships for this purpose has been described [36, 38] and continues to be actively explored in this work and elsewhere [6, 36, 37, 39, 40, 43].

8.1 Temporal Queries

A problem with annotation tools is that subsequent analyses become constrained either by the lack of database support, or by the level of resolution. A tool that has a rich database structure, Transana, permits multiple users to work on the same dataset, but it constrains analysis to the level of sentences/phrases. Thus, existing tools are mostly designed for macro-level analysis, and are mostly limited in terms

of searches they could support, such as simple keyword searches. MacVisSTA was created to support *micro*-analysis at varying timescales, and extended to support specialized querying of the annotation data. We sought the following capabilities, described in [13]: (a) vertical net, (b) sequential probe. Definitions of these queries appear in Table 8.1.

Table 8.1: Explanation of specialized temporal queries

The *vertical net* query is best understood the following: During an interval (or instant in time), which behavioral streams are active? During an interval, what is the extent of overlap in the following behavioral streams?
The *sequential probe* asks: For a specific time interval, are there any *similar* episodes anywhere else in the data? For a specific time interval, what is the *next* predicted segment, based on properties of the events?

A vertical net query searches recursively for successive overlaps, and keeps the part of the interval that is contained by all of the streams. This is especially useful when searching for certain aggregate behaviors, such as instances of shared gaze. We implemented the vertical net query by allowing the user to specify streams of interest, and the rule for building successive overlaps using a graphical user interface. Thus, we have developed a method for querying overlaps in event data, without the user needing any knowledge of query languages.

Implementing a sequential probe is more difficult, however, for the following reasons: we do not try to model semantics or the discourse structure using classifiers. This stems from the sparse nature of the dataset, which does not give enough examples to train Hidden Markov Models (HMMs), for example. Many patterns exist

in gaze data, however. Currently Theme [24] is the only system that systematically explores all statistically significant patterns (built up from smaller "T-patterns" in which sequences are hierarchically organized using a genetic algorithm). Thus, Theme may potentially yield all possible sequences that are patterns of interest, organizing them into hierarchies. We are exploring whether this has any benefit to discourse segmentation, beginning with instances of shared gaze. An example of Theme's output appears in Figure 8.1.

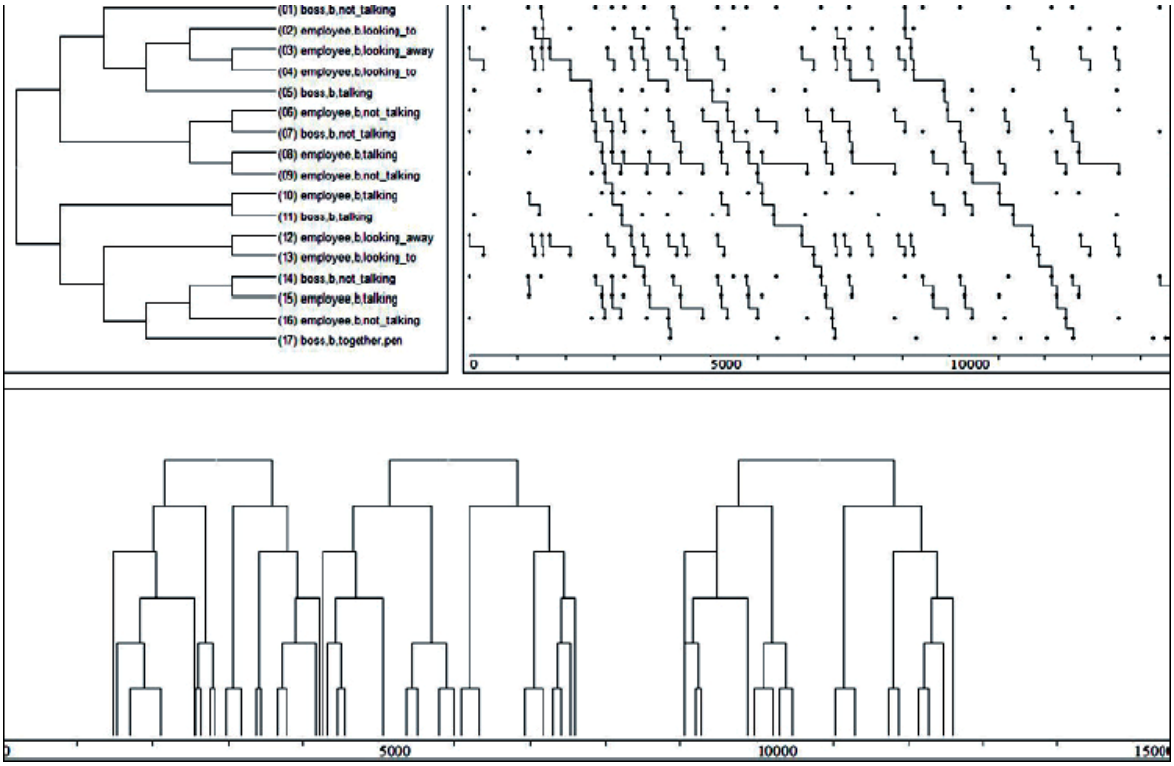

Figure 8.1: Sample Theme output

Chapter 9

Conclusion

The focus of this research involves several facets: multimodal corpus building, visualization, annotation and analysis. In support of these, the author created new software that employs a hybrid architecture, enabling researchers to have access to multiple information streams, as well as an integrated database management system. This thesis has reviewed existing tools and standards of annotation, as well as demonstrating the use of Annotation Graphs as a means of exchange. MacVisSTA was created based on multiple-linked representations [23] as a principled interaction model, and supports definition of a range of visual schema [34], as well as combinations of these in user-defined assemblies.

These facets exist because of our investigation of psycholinguistic theory and practice, both through collaborative research and interviewing expert annotators, which has led to an understanding of the requirements for time-situated, fine-grained, multimodal analysis. These are summarized in Table 9.1.

Table 9.1: Desired components for psycholinguistics research

database management system
flexible annotation system
free-form, overlapping observations for exploratory analysis (notes/notebooks)
minimal configuration by the user
music score interface
multiple-linked representation
non-overlapping, symbolic entities for fine-grained analysis of events
printing
time-accurate and time-synchronous playback
visualization of continuous data
visualization of observations on a timeline co-temporally with audio/video
visualization of time-aligned speech/language data

9.1 Impact

MacVisSTA was successfully used for annotation and analysis of multimodal meeting data. It supports working at multiple time-scales for coarse-to-fine annotation. The software can be extended through a plugin-loading mechanism. It is also one of the few tools that provides a built-in database infrastructure. The software was prototyped and, after moving to Mac OS X, has gone through iterative refinement with input from psycholinguist and speech researchers. This effort has involved the use of several API's (Application Programming Interfaces) for development, including Cocoa/Objective C, QuickTime, OpenGL, MySQL, etc. This software embodies the requirements for flexible, multimedia annotation and analysis in an integrated framework, and should be a lasting resource to the community.

9.2 Future Work

This thesis has presented MacVisSTA in the context of Psycholinguistic annotation; MacVisSTA can potentially be employed to investigate the utility of several annotation schemes. Given MacVisSTA's flexibility and extensibility, several other opportunities exist for future work. These include:

- exploration of configurable information visualization with multimedia
- interfacing to multiple databases
- collaboration in Digital Libraries and Multimodal Corpora research, as well as hierarchical pattern analysis information visualization techniques
- iterative algorithm development
- development of new metrics/measurement studies
- using observational and real-time (continuous) data for Behavioral or Human-Computer Interaction studies
- interfacing to other multimodal systems

We will continue with further elaboration of MacVisSTA, including creation of new and improved interfaces for "scrubbing" audio, further graphics optimization (using OpenGL techniques), and evaluation of MacVisSTA's usability for a range of benchmark tasks. Continued work in time-synchronous, multimodal analysis should make this system increasingly accessible and easy for researchers to use, which we anticipate will have utility in several domains.

Bibliography

[1] Rakesh Agrawal and Ramakrishnon Srikant. Fast algorithms for mining association rule. In *20th Very Large Databases Conference*, 1994.

[2] Tony Bigbee, Dan Loehr, and Lisa Harper. Emerging requirements for multi-modal annotation and analysis tools, eurospeech special event: Existing and future corpora acoustic, linguistic and multi-modal requirements, 2001.

[3] Stephen Bird, David Day, John Garofolo, John Henderson, Christophe Laprun, and Mark Liberman. ATLAS: A flexible and extensible architecture for linguistic annotation. In *LREC '00: Proceedings of the Second International Language Resources and Evaluation Conference*, 2000.

[4] Steven Bird, Kazuaki Maeda, Miaoyi Ma, and Haejoong Lee. Building annotation tools with the annotation graph toolkit, available on internet, *citeseer.ist.psu.edu/460030.html*.

[5] John M. Carroll, Chun W.Choo, Daniel Dunlap, Phillip Isenhour, Stephen T. Kerr, Allan MacLean, and Mary Beth Rosson. Knowledge management support for teachers. In *Educational Technology Research and Development (Education Module)*, number 51 in 4, pages 42–64, 2003.

[6] Lei Chen, R. Travis Rose, Fey Parrill, Xu Han, Jilin Tu, Zhongqiang Huang, Mary Harper, Francis Quek, and David McNeill. VACE multimodal meeting corpus. In *2nd Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, 2005.

[7] Aron Culotta and Andrew McCallum. Practical markov logic containing first-order quantifiers with application to identity uncertainty. Technical Report IR-430, University of Massachusetts, June 2005.

[8] Pavel Curtis and David A. Nichols. MUDs grow up: Social virtual reality in the real world. In *IEEE Computer Society International Conference*, pages 193–200, 1994.

[9] Artur S. d'Avila Garcez, Luis C. Lamb, and Dov M. Gabbay. A connectionist inductive learning system for modal logic programming. In *Proceedings of the 9th International Conference on Neural Information Processing*, 2002.

[10] Mehmet E. Donderler, R. Ulusoy, and Ugur Gkbay. A rule-based approach to represent spatio-temporal relations in video data. In *Advances in Information Systems, First International Conference, ADVIS*, pages 409–418, 2000.

[11] Christine du Toit and Andries van der Walt. Temporal grammars. In *SAICSIT '02: Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 205–211, Republic of South Africa, 2002. South African Institute for Computer Scientists and Information Technologists.

[12] Susan Duncan. Coding "Manual". Appendix C in David McNeill, *Gesture and Thought*, 2005.

[13] Susan Duncan, David McNeill, and K.E. McCullough. How to transcribe the invisible – and what we see. In *KODICAS/CODE special issue*, pages 75–94, 1995.

[14] Roberto Garcia and Oscar Celma. Semantic integration and retrieval of multimedia metadata. In *5th Knowledge Markup and Semantic Annotation Workshop, SemAnnot'05*, volume 185, pages 69–80. Central Europe CEUR Workshop Proceedings, 2006.

[15] Barney G. Glaser and Anselm L. Strauss. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, Aldine Publishing Company, 1967.

[16] M. A. Goncalves, E. A. Fox, L. T. Watson, and N. Kipp. Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. In *ACM Transactions on Information Systems*, number 22 in 2, pages 270–312, 2004.

[17] Marios Hadjieleftheriou, George Kollios, Petko Bakalov, and Vassilis J. Tsotras. Complex spatio-temporal pattern queries. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 877–888. VLDB Endowment, 2005.

[18] Janet Hitzeman, Mare Moens, and Claire Grover. Algorithms for analysing the temporal structure of discourse. In *Proceedings of the 7th European Meeting of the Association for Computational Linguistics*, pages 253–260, Dublin, Ireland,

1995.

[19] A. I. Joseph, I. Thomas-Kerr, S. Burnett, C. H. Ritz, S. Devillers, D. De Schrijver, and R. V. Walle. Is that a fish in your ear? A universal metalanguage for multimedia. *IEEE Multimedia*, 14(2):72–77, 2007.

[20] Adam Kendon. *The Relationship of Verbal and Nonverbal Communication*, chapter Gesticulation and speech: Two aspects of the process of utterance, pages 207–227. The Hague: Mouton and Co., 1980.

[21] Kibum Kim, Phillip Isenhour, John M. Carroll, Mary Beth Rosson, and Daniel Dunlap. Teacherbridge: Knowledge management in communities of practice. In *Proceedings of the IFIP TC9 WG9.3 International Conference on Home Oriented Informatics and Telematics (HOIT 2003)*, 2003.

[22] Robert Kowalski and Marek Sergot. *A logic-based calculus of events*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[23] R.B. Kozma, J. Russell, T. Jones, N. Marz, and J. Davis. The use of multiple, linked representations to facilitate science understanding. In *Fifth Conference of the European Association for Research in Learning and Instruction*, 1993.

[24] Magnus S. Magnusson. Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments, and Computers*, 32(I):93–110, 2000.

[25] Jose M. Martinez. MPEG-7 Overview, ISO/IEC report no. JTC1/SC29/WG11N5525, Int'l Organization for Standardization. (web

http://www.chiariglione.org/mpeg/standards/ mpeg-7mpeg-7.htm), March 2003.

[26] David McNeill. *Hand and Mind: What Gestures Reveal About Thought*. Univ of Chicago Press, 1992.

[27] David McNeill. Gesture and language dialectic. In *Acta Linguistica Hafniensia green*, 2002.

[28] David McNeill and Susan Duncan. *Growth points in thinking-for-speaking*, pages 141–161. Cambridge University Press: Cambridge, MA, 2000.

[29] Ram Nevatia, Jerry Hobbs, and Bob Bolles. An ontology for video event representation. In *Conference on Computer Vision and Pattern Recognition Workshop*, volume 27, pages 119–129, 2004.

[30] Linh Anh Nguyen. Multimodal logic programming and its applications to modal deductive databases, manuscript, available on internet at *http://www.mimuw.edu.pl/ nguyen/papers.html*.

[31] Linh Anh Nguyen. Constructing the least models for positive modal logic programs. *Fundamenta Informaticae*, 42(1):29–60, 2000.

[32] Linh Anh Nguyen. The modal logic programming system MProlog: Theory, design, and implementation, manuscript (submitted), available on internet at *http://www.mimuw.edu.pl/ nguyen/mprolog/jmpl-long.pdf*, 2006.

[33] Donald A. Norman. *The Design of Everyday Things*. Doubleday, 1986.

[34] Chris North, Nathan Conklin, and Varun Saini. Visualization schemas for flexible information visualization. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 15, Washington, DC, USA, 2002. IEEE Computer Society.

[35] Mehmet A. Orgun and Weichang Du. Multi-dimensional logic programming: Theoretical foundations. *Theoretical Computer Science*, 185(2):319–345, 1997.

[36] Francis Quek, David McNeill, Travis Rose, and Yang Shi. A coding tool for multimodal analysis of meeting video. In *4th International Conference on Language Resources and Evaluation*, 2004.

[37] Francis Quek, R. Travis Rose, and David McNeill. Multimodal meeting analysis. In *2005 International Conference on Intelligence Analysis*, 2005.

[38] Francis K. H. Quek, Robert K. Bryll, Cemil Kirbas, Hasan Arslan, and David McNeill. A multimedia system for temporally situated perceptual psycholinguistic analysis. *Multimedia Tools and Applications*, 18(2):91–114, 2002.

[39] K. J. Rohlfing, D. Loehr, S. Duncan, A. Brown, A. Franklin, I. Kimbara, J.-T. Milde, F. Parrill, T. Rose, T. Schmidt, H. Sloetjes, A. Thies, and S. Wellinghoff. Comparison of multimodal annotation tools: Workshop report. *Gesprachsforschung*, 7, 2006.

[40] R. Travis Rose, Francis Quek, and Yang Shi. MacVisSTA: a system for multimodal analysis. In *ICMI '04: Proceedings of the 6th International Conference on Multimodal Interfaces*, pages 259–264, New York, NY, USA, 2004. ACM Press.

[41] Penelope M. Sanderson and C. Fisher. Exploratory sequential data analysis: Foundations. *Human-Computer Interaction*, 9:251–317, 1994.

[42] P.M. Sanderson, M. McNeese, and B. Zaff. Knowledge elicitation and observation in engineering psychology: MacSHAPA and COGENT. *Behavior Research Methods, Instruments, and Computers*, 26:117–124, 1994.

[43] Yang Shi, R. Travis Rose, and Francis Quek. A system for situated temporal analysis of multimodal communication. In *4th International Conference on Language Resources and Evaluation*, 2004.

[44] Yingen Xiong and Francis K. H. Quek. Meeting room configuration and multiple camera calibration in meeting analysis. In *ICMI '05: Proceedings of the 7th International Conference on Multimodal Interfaces*, pages 37–44, 2005.