The MITRE Multi-Modal Logger:

Its Use in Evaluation of Collaborative Systems

Samuel Bayer, Laurie E. Damianos, Robyn Kozierok, James Mokwa
The MITRE Corporation; 202 Burlington Road; Bedford, MA 01730

Introduction

Collaborative environments have grown quite sophisticated over the years. They have evolved from simple text conferencing tools [2] to full suites of integrated multi-modal tools such as Habanero [9] or MITRE's room-based Collaborative Virtual Workspace [3, 4]. As a result, evaluating multi-modal collaborative systems is an order of magnitude more complex than single-user, single-application HCI evaluation. In the context of the DARPA Intelligent Collaboration &Visualization (IC&V) program, the MITRE HCI group has adapted its logging technology to the task of evaluating collaborative systems.* The MITRE Multi-Modal Logger (MML) is a system for recording, retrieving, annotating, and visualizing data. In this paper, we describe the capabilities of the MML and its application to research including the IC&V Evaluation Working Group's work [5, 6, 7, 8, 13].

Background

The range of data which can be tracked in HCI is potentially broad. These data can be fine-grained (individual X events, for instance) or coarse-grained (a record of which windows the user interacts with). The data can be automatically gathered via instrumentation or manually created via post-hoc annotation. The data can be at the level of the physical or virtual device (i.e., X events or the output of an audio device), the interface (i.e., a record of menu selections made and the content of text entry fields), or the application (i.e., a record of actions taken, such as a retrieval of information or a command issued). This information can be used in a number of ways, such as usability testing or statistical adaptation by the system itself. This information can also be collected from a variety of modalities such as text conferencing, image sharing and manipulation, and audio conferencing.

HCI also imposes requirements on the structure of the information gathered. For instance, the information is typically gathered for multiple users and multiple interactions with the system in question. As a result, some notion of a "trial" or "session" is important. In addition, each trial might require information to be gathered from multiple components simultaneously. An example of this is when a speech recognizer is used in conjunction with an independent multi-modal system.

While there are many tools which address some slice of this problem ([10], [11]), their scope is limited either by the type of data they collect, or the type of data analysis they support, or both. The MML, on the other hand, separates collection and analysis, and gives the developer maximal control over both. We describe, in detail below, how the Logger accomplishes these goals.

Capabilities

The Multi-Modal Logger provides a relational database schema for housing the data and an easy-to-use API for instrumenting existing applications. It also provides a set of tools for visualizing, reviewing and annotating data collected via instrumentation or other means.

The database schema groups datapoints by application and applications by session. This schema can be implemented either in a relational database or a flat file database.** The strength of this schema derives most

---

• This work was funded, in part, by DARPA under contract number DAAB07-97-C-E601.

** We currently use a flat file database, both for speed and ease of administration.

notably from the way that it houses both data and metadata (i.e., ways of describing the data's type and structure). It also facilitates multimedia data handling by supporting MIME types as an aspect of the metadata.

The MML API can be conceptually divided into three major sections: setting up a database and session (which includes defining the metadata), logging the data, and retrieving the data. This API is available in several programming languages: C, C++, Tcl, Java, and Python.

The set of tools provided by the MML uses the API to explore and augment the data. The current version of the MML is distributed with a general-purpose visualization, annotation and replay tool and an SQL-like interpreter for exploring the database. Two other, simpler tools for viewing the contents of the database are included as well.

The capabilities of the MML can be exploited in a number of ways. The instrumentation API can be used to instrument applications to record programmer-determined events automatically. The visualization and annotation tool can be used to analyze and augment the collected data. The retrieval and instrumentation APIs allow the building of new tools in order to examine and augment the data in different ways. Exporting the data off-line into various other tools can be done through the retrieval API. In order to exploit the MML visualization tools and retrieval API, data recorded by other tools can be imported into an MML database using the instrumentation API. The MML tools can also be used to export and import an ASCII record of a session. This facilitates the sharing of data among sites.

In the remainder of this paper, we further explore these capabilities and provide detailed examples of some applications.

Illustrations

Example 1: Instrumented CVW

A good introduction to the strength of the MML is its most demanding MITRE-internal application: the instrumentation of MITRE's Collaborative Virtual Workspace (CVW). CVW was instrumented in support of a DARPA-funded effort to validate an evaluation methodology for collaborative technologies [6, 8, 13].

We instrumented two applications with the MML: the central CVW server, which monitors text conferencing and whiteboard traffic between participants, and the Visual Audio Tool (VAT) [15], an audio conferencing tool that does not communicate via the CVW server. In the MITRE Map Navigation Experiment [6, 13], data was collected from the CVW server and two instances of VAT. The data were then analyzed in an effort to compare and evaluate two versions of a system. We transcribed the speech in order to investigate some interesting discourse issues, and we also used a tool for further text-based annotations.

Defining Data

Our model of CVW data organizes events into major categories such as public communications, private communications, and navigation. Each datapoint has a number of different dimensions which are determined by the type of the data. For example, a public communication datapoint records the user, the user's location, and the contents of the communication whereas a navigation datapoint records the source and destination as well as the user. VAT audio traffic defines another category of data with an associated MIME type to control playback behavior.

Visualizing the Collected Data

The visualization and annotation tool can be used to visualize and replay interactions. Figure 1 is an example of a CVW session with two participants using audio, text chat, and a whiteboard. This tool can also be used for annotating data. We have also used it to mark topic shifts in dialogue and to group utterances into single turn units (not shown).

Building Visualization and Annotation Tools

We have also used the MML retrieval API to write other annotation and translation tools. One such tool supports adding manually-entered speech transcriptions as annotations to the session record. (See Figure 2.)

Exporting Data

The retrieval API was used to write scripts that translate the logged data into tab-delimited format for spreadsheet analysis. We used the same tool to translate the textual data into SGML and import it into MITRE's Alembic Workbench [1], a MITRE-developed tool for visualization and annotation of text corpora. Importing the data into the Alembic Workbench has provided us with invaluable insight into multi-modal interactions. Figure 3 illustrates a collaborative session between two participants using multiple modalities to solve a problem efficiently.

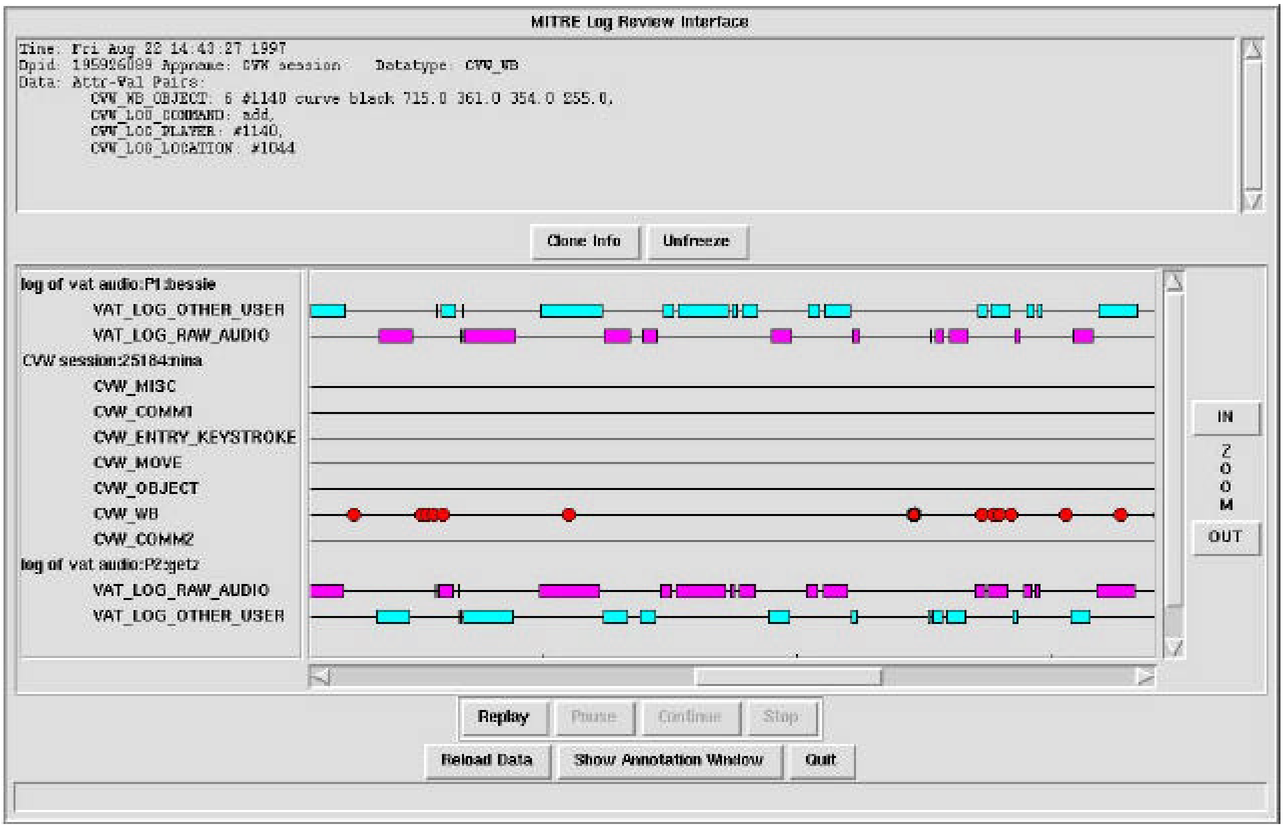

Figure 1: Visualization of data in the MML's annotation tool. Data represented here includes audio, typed text, and whiteboard activity.

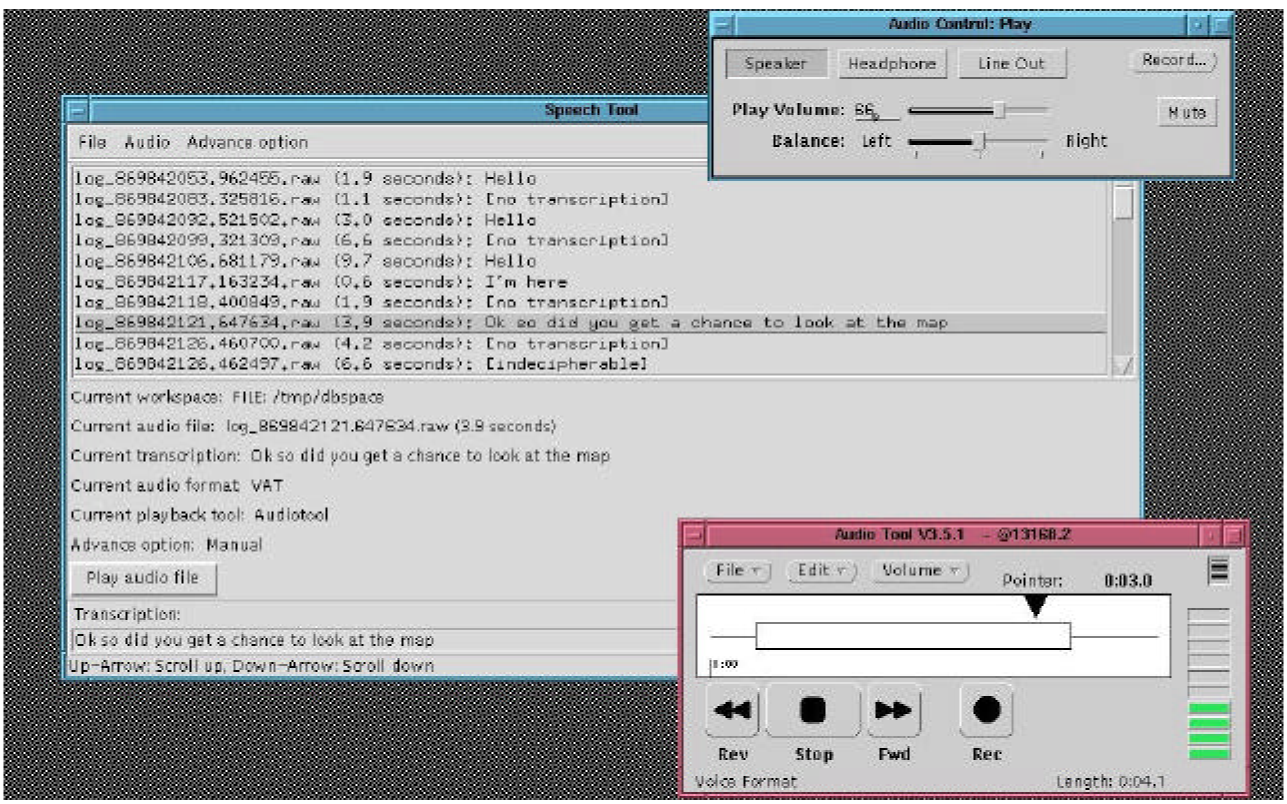

Figure 2: Adding speech transcriptions to a logged session using the Speech Annotation Tool.

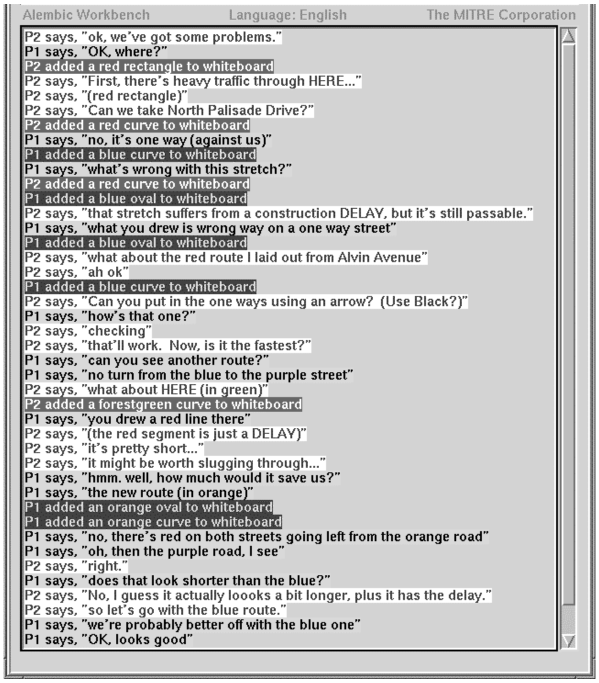

Figure 3: Example of multi-user, multi-modal collaboration as viewed as text in MITRE's Alembic Workbench. Participants P1 and P2 used text conferencing together with a shared whiteboard to establish a route on a road map.

Distributing Data

We are using the MML as the format for distributing data we have collected. Data from a collaborative problem-solving experiment, the MITRE Map Navigation Experiment, are available for downloading at http://snad.ncsl.nist.gov/~icv-ewg/experiments/mapnav/data/data.html.

Example 2: A Minute-Taking IMP

Another research project at MITRE that exploits the capabilities of the MML is the Intelligent Multi-Modal Participant, or IMP. Some of the objectives of the IMP research are to explore the relationships between human-human and human-machine interactions in a collaborative environment and to implement a software agent capable of adaptive learning.

A minute-taking IMP has been implemented as a CVW participant as part of a technology integration initiative at MITRE. The IMP uses the MML to log CVW interactions (speech, text, and whiteboard annotations) without requiring changes to the CVW server itself. The IMP has a specialized replay interface that allows the IMP to retrieve and replay logged data via the MML. The replay interface was built as a Python [14] subclass of the MML visualization and annotation tool.

Minute-taking IMPs have been used at MITRE to collect data from CVW training sessions. Users are asked to perform exercises during training and then given the same set of exercises at a later date. Data are collected and analyzed in an attempt to reveal how much of the training is retained over time. It is hoped that the results of the study will provide insight into the effectiveness of the training as well as into the usability of the system.

An IMP can also use the MML to keep a record of all interactions it has with other users. It could then look for patterns in the logs of the interactions, learn to make predictions from those patterns, and either simplify future actions for human participants, or even take over and automate some actions.

Other Examples

Importing Data

We have used the Multi-Modal Logger to translate saved CVW internal logs into MML databases for CVW usability analysis.

We have also done work with non-MITRE systems such as MIT's Jupiter system. Jupiter is a telephone-based spoken dialogue system that provides current weather information for hundreds of cities [11]. We have imported some of the collected Jupiter data into MML format for visualization and use case analysis of interactions.

Conclusion

The MITRE Multi-Modal Logger is a general, flexible tool which has proven to be well-suited to the task of evaluation of collaborative systems. Its capabilities support the sort of automated data collection which is crucial to efficient evaluation of software systems. Its visualization and analysis capabilities also help to provide new insights into collaborative interactions and patterns of usage. We believe that the MML componentized approach to data collection, annotation, visualization, and analysis is an essential dimension to progress in evaluation of these multi-modal, multi-user systems.

Appendix: Obtaining the MITRE Multi-Modal Logger

Version 1.0 of the MML is available from http://www.mitre.org/technology/logger. It is distributed with documentation which includes detailed instructions for use of the API for logging and retrieval, use of the database exploration tools, the structure of the database, and release notes. The MML runs under Solaris 2.5 and SunOS 4.1.3 on Sun workstations as well as Intel Linux and mkLinux for the Macintosh.

References

1. Alembic Workbench: http://www.mitre.org/technology/alembic-workbench
2. Bartle, R. A. (1990). Interactive Multi-Player Computer Games. MUSE Ltd, Colchester, Essex, UK. Available: ftp://ftp.lambda.moo.mud.org/pub/MOO/papers/mudreport.txt
3. Collaborative Virtual Workspace: Spellman, P. J. and Carlson, J. Technology for Virtual Organizations. Extended abstract in Proceedings GroupWare '95, Boston, MA, March 1995.
4. Collaborative Virtual Workspace: http://www-i.mitre.org/resources/centers/it/g061/cvw.html
5. Damianos, L., Hirschman, L., Kozierok, R., Kurtz, J., Greenberg, A., Walls, K., Laskowski, S., Scholtz, J. (1999) Evaluation of Collaborative Systems. ACM Computing Surveys, this issue.
6. Kurtz, J., Damianos, L., Kozierok, R., and Hirschman, L. (1999) The MITRE Map Navigation Experiment. ACM Computing Surveys, this issue.
7. Damianos, L., and Kurtz, J. Map Navigation Experiment. MITRE technical report 98-031. Feb. 1998.
8. Evaluation Working Group: http://www.antd.nist.gov/~icv-ewg/
9. Habanero, NCSA: http://www.ncsa.uiuc.edu/SDG/Software/Habanero/index.html
10. IICON Evaluator: James Hicinbothom et al. (1994) "A Toolset for Systematic Observation and Evaluation of Computer-Human Interaction". CHI 94 Conference Companion, pp. 5-6.
11. Jupiter: http://www.sls.lcs.mit.edu/jupiter
12. MACShapa: Sanderson, P. M., J. J. P. Scott, T. Johnston, J. Mainzer, L. M. Watanabe, & J. M. James (1994). MacSHAPA and the enterprise of Exploratory Sequential Data Analysis (ESDA). International Journal of Human-Computer Studies, 41, 633-68.
13. The MITRE Map Navigation Experiment: http://www.antd.nist.gov/~icv-ewg/experiments/mapnav/mapnav.html
14. Python: http://www.python.org
15. VAT: Van Jacobson, Steven McCanne. vat - Lawrence Berkeley National Laboratory Audio Conferencing Tool. http://www-nrg.ee.lbl.gov/vat.