

COLLECTING MOBILE MULTIMODAL DATA FOR MATCH

Patrick Ehlen, Michael Johnston, Gunaranjan Vasireddy

AT&T Labs-Research
180 Park Ave, Florham Park, NJ 07932
{ehlen, johnston, guna}@research.att.com

ABSTRACT

Next-generation multimodal systems designed for use in mobile environments present challenges to the task of data collection that are not faced by speech-based systems. We discuss some established data collection and evaluation methods and their limitations in the context of a mobile multimodal system. These limitations are addressed by the “on-board” multimodal data collection method developed for MATCH, a multimodal mobile city guide. Our approach exploits MATCH’s component architecture in that each component can be redeployed in evaluation and annotation tools, allowing user test sessions to be replayed with a high degree of fidelity without the use of recorded video. Instead, the components themselves perform a dynamic re-enactment of test sessions directed by the script of a comprehensive log file. This method enabled continual user testing and piloting to inform the iterative development process for MATCH.

1. INTRODUCTION

As developing technology converges on portable devices for use in a variety of environments, traditional methods of evaluation—often conducted in a static lab environment—become less viable. More versatile approaches are needed to evaluate systems in mobile environments. Multimodal interfaces are well-suited candidates for mobile devices, as they permit expanded flexibility for the user. Effectively designing for these expanded possibilities requires flexible and portable methods of evaluation, exemplified by MATCH’s on-board logging and evaluation paradigm that handles much of the process of evaluation locally, obviating the need for an extensive, external evaluation setup.

2. THE MATCH SYSTEM

MATCH (Multimodal Access To City Help) is a multimodal speech-pen interface that provides mobile access to restaurant and subway information for New York City [1]. As a mobile and flexible interface, it meets the demands of modern urban environments where access to a complex and rapidly changing body of information on restaurants, theatre schedules, and transportation details is best delivered on the go. The multimodal map-based interface permits any combination of speech and pen input, allowing users to inscribe deictic and symbolic gestures on the map, as well as



Fig. 1. MATCH running on Fujitsu PDA

handwritten words. System output is multimodal as well, combining synthesized speech with icons and descriptive call-outs presented dynamically on the map. MATCH’s architecture is modular and distributed, with multiple components that work in concert by communicating through XML strings over TCP/IP. The user interface runs from an ActiveX control embedded in a browser. MATCH functions either as a thin client on a small portable device, or can run standalone on a tablet PC (Figure 1).

3. APPROACH TO DATA COLLECTION

As a multimodal system, MATCH demands a break from established evaluation methods for speech-based systems, which are often designed to test telephony interfaces that require only a single channel of information about the user’s actions (the audio channel) to be stored with system state data. The DARPA Communicator spoken language system, for instance, evaluates test user call-ins by logging recorded audio from a call with a logfile of system state data [2]. Speech-only evaluation methods prove insufficient for multimodal data, which requires a method that can monitor and log events in multiple modes—adding graphical and gestural data to speech data—making the task of obtaining and analyzing user data more complicated. The narrow channel of information offered by recording audio may be augmented by adding recorded video to user test data. This also adds the necessity of synchronizing video, audio, and system data channels for analysis and archiving. In SmartKom, where hand gestures and facial expressions are be-

Thanks to AT&T Labs and DARPA ITO (contract No. MDA972-99-3-0003) for supporting this research. We would also like to thank Marilyn Walker, Candy Kamm, and Srinivas Bangalore.

ing annotated, this is done by hand with video editing software [3]. In STAMP [4], where the input modes are speech and pen, an automated synchronization method using video timecodes is used to synchronize log data and video playback for analysis.

Data collection for MATCH is further complicated by its design as a mobile system, which should therefore be evaluated in mobile environments. This limits the utility of video methods that record the test user in a lab setting with a fixed video setup. Such limitations to mobility were partially overcome for the multimodal QuickSet system [5] by employing a mobile video setup in which the test user wore a portable camera and scan converter that transmitted video and system data to an observation station in the lab via wireless LAN. However, this mobile video method limits the mobility of the user to the vicinity of the lab, while MATCH is designed to be used throughout an entire city. If MATCH is ultimately to be used on the street, it should be evaluated on the street, using little more than the device itself. The problem we faced in designing an evaluation process for MATCH was how to collect high-fidelity data on each test user's multimodal interactions without incurring the procedural overhead of recording video for every user test.

3.1. Data Collection Solution

Our primary goals in collecting MATCH data were to evaluate and improve speech, gesture, and handwriting recognition models and multimodal grammars, and to examine mode compensation and modality convergence [6]. Given the substantial development of MATCH simultaneous to the process of data collection, we were also interested in using collected information to debug MATCH's numerous components. With these goals in mind, it was clear that much of the required data could be obtained without the use of video if the system logged enough information about the user's and the system's actions to allow those actions to be "replayed" dynamically at a later time. Since there are two channels by which a user can interact with MATCH—by speech and by pen—video recordings could be supplanted by a system that logged both the audio of the user's actions and sufficient detail about the user's gestural pen actions for that information to be recreated with a high degree of fidelity. Our logging efforts then proceeded in two steps: the creation of a high-fidelity logger, and the re-instrumentation of the various MATCH components so they would not only *send* detailed messages to the logger, but could also *receive* and replay those messages, allowing individual MATCH components to be re-used in evaluation tools like log viewers and annotators.

3.2. Logging MATCH Messages

The MATCH logger collects and stores system messages on the state of each component, maintaining an integrated and detailed log file of events describing the state of the system throughout the test session—so detailed, in fact, that the log file could also be thought of as a *script* that can direct the MATCH components to replay the user's actions. The logger first establishes a "cast list" that records which components are running, the version number of each component, and the parameters invoked. Every salient event that occurs in MATCH sends a message to the logger, which records the handle of the component delivering the message, a timestamp for the event, and enough information about the event to allow it to be recreated. Recorded details include information about speech recognition results, meaning result lattices from a multimodal fi-

nite state transducer, system responses in both graphic and text-to-speech form, timeout triggers, map details, and pointers to audio files of the test user's speech. The user's electronic ink (both gestures and handwriting) are recorded with a high-resolution sampling of points and timing values associated with those points, so gestures and handwriting can be replayed dynamically on the user interface in exactly the same manner in which they were drawn. All of this data is stored in an XML log file for the session, a section of which is shown in Figure 2, resulting in a series of snapshots of the "stage"—that is, of the state of the system.

```
<MMExchange ID="6">
  <click_to_speak ID="6" FROM="UI">
    <time><secs>1005247426</secs><msecs>316</msecs></time>
  </click_to_speak>
  <ASR_begin ID="6" FROM="JCTL">
    <time><secs>1005247426</secs><msecs>396</msecs></time>
  </ASR_begin>
  <ASR_end ID="6" FROM="JCTL">
    <time><secs>1005247429</secs><msecs>860</msecs></time>
  </ASR_end>
  <audiologname ID="6" FROM="JCTL">helen1108/helen1108005</audiologname>
  <speechresult ID="6" FROM="JCTL">
    <time><secs>1005247431</secs>
    <msecs>894</msecs></time>
    <string>I am in thirty fourth street and park</string>
    <score>71</score>
  </speechresult>
  <type ID="6" FROM="MMFST">speech_only</type>
  <unimodal_speech_timeout>
    <timeout>1</timeout>
    <time><secs>1005247432</secs><msecs>865</msecs></time>
  </unimodal_speech_timeout>
  <meaning_result ID="6" FROM="MMFST">
    <meaning><command><userlocation>
      <crossstreets>
        <crossstreet1>_30_A_st</crossstreet1>
        <crossstreet2>park_ave</crossstreet2>
      </crossstreets>
      </userlocation></command></meaning>
    <time><secs>1005247433</secs><msecs>326</msecs></time>
  </meaning_result>
  <gesturelog ID="6" FROM="UI">
    <ftsbeforeink></ftsbeforeink>
  </gesturelog>
  <inkstyle>
    <inkcolor>16711680</inkcolor>
    <penwidth>3</penwidth>
    <penstyle>0</penstyle>
  </inkstyle>
  <mapdisplay>
    <mapcenterx>-73.98144</mapcenterx>
    <mapcentery>40.74715</mapcentery>
    <mapzoomlevel>0.299999999999653</mapzoomlevel>
    <mapscreenheight>420</mapscreenheight>
    <mapscreenwidth>351</mapscreenwidth>
  </mapdisplay></gesturelog>
```

Fig. 2. A Section of the MATCH Log File

Each snapshot represents a *multimodal exchange*, which roughly describes the user doing something (making a request), and MATCH doing something in response. These actions are grouped and logged as a single unit which is conceptually adapted from the notion of an exchange in transactional analysis, put forward by Sinclair and Coulthard [7], and approaching what has been called a *multimodal contribution* by Villaseñor, Massé & Pineda [8] (also see Clark & Schaefer [9]). While a contribution describes a higher level of abstraction and implies some directedness toward a higher goal, an exchange implies merely a user utterance and a system response to that utterance; a simple exchange of presentation and acceptance.

3.3. The Log Viewer / Annotator

The XML logfile allows data access at either high or low levels of scrutiny through XSL transformations, providing a more generalized evaluation framework than video. The limited perspective offered by video is replaced by a framework where evaluators can choose which aspects of the data they will observe, re-deploying individual system components that receive and execute the same messages they had originally transmitted, replaying each event. Since MATCH runs from a browser based interface that controls

its components through a Java facilitator, these components can be readily recombined into evaluation tools for various purposes, such as the log viewer we designed to replay recorded multimodal interactions.

After opening a log file in the log viewer, an evaluator can step through a user session exchange by exchange, or skip through non-consecutive exchanges, allowing a comprehensive review of the test user's actions and MATCH's responses. The log viewer first presents the state of the system prior to the user's actions in the exchange, placing any previous markings and selections on the map that would have been visible to the test user. The user's actions can then be played and replayed, either at the exact rate employed by the user, or at a reduced rate (slow motion). Because the map interface is the same ActiveX control as that used in the test session, the map appears as it did for the user, drawing ink on the map exactly as the user originally drew it, even at the same pace. Audio files of the user's speech are also replayed in the log viewer through a Tcl/Tk plug-in with a waveform display. The ability to plug in different components like this is handy, allowing different evaluation tools to be rapidly constructed and tailored to the specific needs of the annotation or analysis session. Finally, an evaluator can review the system's responses to the user's actions in that exchange, with graphics and audio occurring just as they occurred for the user.

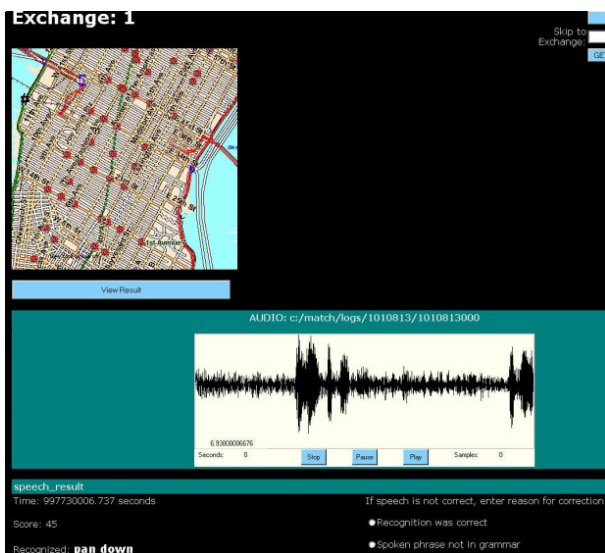


Fig. 3. The MATCH Log Viewer / Annotator

One version of this log viewer serves as a rapid annotation tool, allowing post-test hand-annotations to supplement the log files for each session with information about the test user's actual inputs. Any annotations made to a log file are appended to that file and stored within the multimodal exchange to which the annotation refers. Multiple annotations can be appended to any exchange (as a new XML node), so interpretations of the logged user's multimodal utterances can be compared to interpretations made by previous annotators. Annotations can also be automatically compared to the multimodal grammar by plugging in a grammar-checking component that provides immediate feedback about whether a mis-recognized user utterance failed because of poor recognition conditions or because the utterance was out of grammar. For a more detailed look at the paths considered by the system, a lattice view-

ing component also enables display of a graphical lattice representation of the gesture and speech inputs. All of these components contribute to our overall goal of creating a flexible, modular evaluation system that makes the process of annotation and analysis as simple and yet as robust as possible, permitting us to focus our efforts more on collecting and learning from user test data than on processing that data.

4. ITERATIVE DEVELOPMENT AND USER TESTS

The instantiation of a rapid logging and evaluation process within MATCH enhanced our ability to evaluate the system's strengths and shortcomings as development proceeded, leading to an iterative development process. Small pilot studies with a handful of test users (mostly summer interns and AT&T researchers) were conducted initially both in the lab and on the streets of New York (Figure 4), during which consistent problems in the system were identified using the log viewer. The ability to identify and analyze problems almost as soon as the user had finished proved invaluable for making rapid improvements that could be re-tested on other users.



Fig. 4. Testing MATCH in NYC

Under this iterative process, we caught several problems early on. For example, while we originally thought most users would navigate by specifying the names of New York's many neighborhoods, as in "Show Italian restaurants in Chelsea," a few early pilots on location in the city revealed that a user's need for neighborhood names in the grammar was almost negligible when compared to the need to specify cross-streets and landmarks; as a result, a sizable list of cross-streets and landmarks was added to the grammar. Other early tests revealed the need for easily accessible "cancel" and "undo" features so users can make quick recoveries from system misunderstandings.

After some initial open-ended piloting trials, more structured user tests were conducted, for which we developed a set of six fictional scenarios that required the test user to solve everyday problems using MATCH. The scenarios were left as open-ended as possible, anticipating that test users would try to solve the problems by whatever means came most naturally to them, whether using pen, voice, or both simultaneously. Test users received minimal training on how to approach the system, and our brief tutorial was

intentionally broad in scope and also somewhat vague in the hope that test users would overestimate the system’s capabilities and approach problems in ways the developers had yet to conceive. This “minimal cueing” approach resulted in substantial additions to the speech grammar and gesture repertoire, but had the drawback of being more frustrating for users than a highly structured (or wizarded) task would have been. One sample scenario is illustrated in Figure 5. The text is:

You are walking down 34th Street in Murray Hill near Park Avenue South, and you have a strange craving for onion soup. You recall once having great onion soup at a French bistro on the Upper East Side that was fairly inexpensive. Unfortunately, you don’t remember what it was called or where it was. Use MATCH to help you remember the name of the restaurant, and write down directions for how to get there by subway.

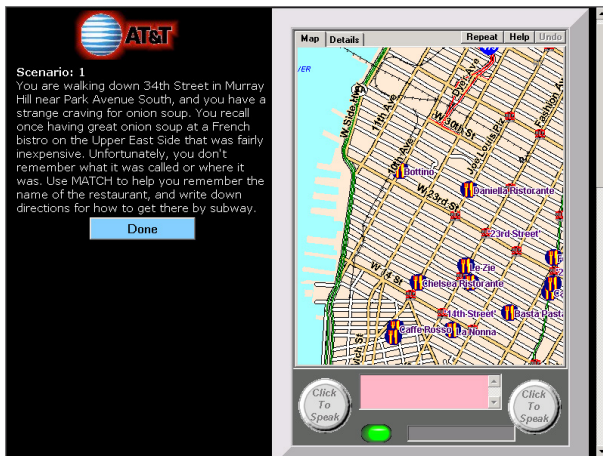


Fig. 5. MATCH with a User Test Scenario

Our method of using the evaluated system as the primary tool for data collection and evaluation did reveal certain limitations, for which video collection or a live evaluator would prove helpful. A few speakers had problems with recognition that remained mysterious until one of the evaluators noted that the placement of the “click-to-speak” button on the bottom-right side of the screen was leading left-handed users to block the microphone—located at the top-left of the device—with their arms as they spoke. A few other exploratory tests revealed that the placement of this button and the recognition feedback box at the lower-right corner was actually leading many users to speak “to” that corner, rather than toward the tiny microphone, hindering speech recognition. Moving the button and the feedback box to the top-left of the device resolved both problems. These device-specific issues could not be identified by recording from the device itself, and require either video recording or the presence of an unobtrusive yet observant evaluator who can identify such problems. In our experience, a live evaluator proved essential for initial pilot studies, and the portability of MATCH’s evaluation system facilitated that option.

5. CONCLUSION

To evaluate multimodal systems in diverse and changing environments, certain advantages can be realized by forsaking a lab-based video evaluation method in favor of an on-board test data logging

system that records and stores data using the system itself. The primary advantage of this approach is the increased mobility and flexibility in evaluation studies, allowing tests of the system to be conducted in environments where the system ultimately will be used. MATCH’s component-based architecture was critical in enabling the rapid construction of evaluation tools, including a log viewer that re-deploys MATCH’s components to recreate events from test sessions, using the multimodal log as a script. This robust logging capacity proves useful for debugging a mobile, distributed system like MATCH, for conducting usability tests, and for collecting data about multimodal interactions to use in testing models for language, gestures, and multimodal integration.

6. REFERENCES

- [1] M. Johnston, S. Bangalore, A. Stent, G. Vasireddy, and P. Ehlen, “Multimodal language processing for mobile information access,” in *Proceedings of ICSLP*, Denver, Colorado, 2002.
- [2] M. Walker, L. Hirschman, and J. Aberdeen, “Evaluation for DARPA Communicator spoken dialogue systems,” in *Proc. of the Second Int’l Conference on Language Resources and Evaluation*, Athens, Greece, 2000.
- [3] U. Türk, “The technical processing in SmartKom data collection: A case study,” in *Proc. of EUROSPEECH Scandinavia*, Aalborg, 2001, pp. 1541–1544.
- [4] S. L. Oviatt and J. Clow, “STAMP: A suite of tools for analyzing multimodal system processing,” in *Proceedings of ICSLP*, Sydney, Australia, 1998, vol. 2, pp. 277–280, ASSTA Inc.
- [5] S. L. Oviatt, “Multimodal system processing in mobile environments,” in *Proceedings of the Thirteenth Annual ACM Symposium on User Interface Software Technology*, New York, 2000, pp. 21–30, ACM Press.
- [6] L. Bell, J. Boye, J. Gustafson, and M. Wirén, “Modality convergence in a multimodal dialogue system,” in *Proc. of Gotalog 2000, Fourth Workshop on the Semantics and Pragmatics of Dialogue*, 2000, pp. 29–34.
- [7] J. M. Sinclair and M. Coulthard, *Towards an Analysis of Discourse: The English Used by Teachers and Pupils*, Oxford University Press, Oxford, 1975.
- [8] L. Villaseñor, A. Massé, and L. Pineda, “A multimodal dialog contribution coding scheme,” in *Proc. of LREC-2000 Workshop on Meta-Descriptions and Annotation Schemes for Multimodal Language Resources*, Athens, Greece, 2000, pp. 52–56.
- [9] H. H. Clark and E. F. Schaefer, “Contributing to discourse,” *Cognitive Science*, vol. 13, 1989.