

MATCH: MULTIMODAL ACCESS TO CITY HELP

Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy

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

ABSTRACT

Interfaces to mobile information access devices need to allow users to interact using whichever mode or combination of modes are most appropriate, given their user preference, task at hand, and physical and social environment. This paper describes a multimodal application architecture which facilitates rapid prototyping of flexible next-generation multimodal interfaces. Our sample application MATCH (Multimodal Access To City Help) provides a mobile multimodal speech-pen interface to restaurant and subway information for New York City. Finite-state multimodal language processing technology enables input in pen, speech, or integrated combinations of the two. The system also features multimodal generation capabilities providing speech output synchronized with dynamic graphical displays.

1. INTRODUCTION

Since mobile information access devices (PDAs, tablet computers, next-generation phones) offer limited screen real estate and no keyboard or mouse, complex graphical user interfaces are not feasible and users must interact using recognition-based modalities such as speech or pen. Mobile devices by their nature are used in a broad variety of different environments, for different tasks, and by different users. Multimodal interfaces address this flexibility by providing multiple different channels for interaction (See [1] for a detailed overview of research on multimodal input and output). Critically multimodal interfaces need to enable users to interact using whichever mode or combination of modes is most appropriate given their user preference, the task at hand, and their physical and social environment. This poses significant challenges for work in natural language understanding, generation, and dialogue management. We describe here a multimodal testbed application MATCH which we have developed to support our research in these areas (Section 2). The multimodal application architecture underlying MATCH is described in Section 3 and in Section 4 we describe the finite-state approach to multimodal language understanding which enables MATCH to accept any command in either speech, or pen, or a dynamic combination of the two modes.

2. THE MATCH APPLICATION DOMAIN

In urban environments tourists and residents alike need access to a complex and constantly changing body of information regarding restaurants, cinema and theatre schedules, transportation topology

and timetables. This information is most valuable if it can be delivered effectively while mobile, since places close and plans change. Our testbed multimodal application MATCH (Multimodal Access To City Help), is a working city guide and navigation system which currently enables users to access information about New York City (NYC) restaurants and subway routes. The system runs standalone in an embedded mode on a Fujitsu pen computer (Figure 1). The system can also run in a client-server mode if a wireless network is available. The ability to run standalone was an important design decision since it enables collection of multimodal data in realistic mobile environments without relying on the availability of a wireless data network.



Fig. 1. MATCH running standalone on Fujitsu PDA

The user interacts with a graphical interface displaying restaurant listings and a dynamic map showing locations and street information (Figure 2). They are free to give commands using speech, by drawing on the display with a stylus or using synchronous multimodal combinations of the two modes. For example, they can request to see restaurants using the unimodal spoken command *show cheap italian restaurants in chelsea*. Alternatively, they could give the same command multimodally by circling an area on the map and saying *show cheap italian restaurants in this neighborhood*. If the immediate environment is too noisy or public, the same command can be given completely in pen as in Figure 3, by circling an area and writing *cheap and italian*.

Having identified some restaurants the user can ask for the review, cuisine, phone number, address, or other information for a restaurant or set of restaurants. The system responds with graphical

Thanks to AT&T Labs and DARPA ITO (Contract No. MDA972-99-3-0003) for supporting this research. Thanks also to Patrick Ehlen, Noemie Elhadad, Giuseppe DiFabrizio, Candace Kamm, Elliot Pinson, Mazin Rahim, Owen Rambow, Nika Smith, Marilyn Walker, and Steve Whittaker.

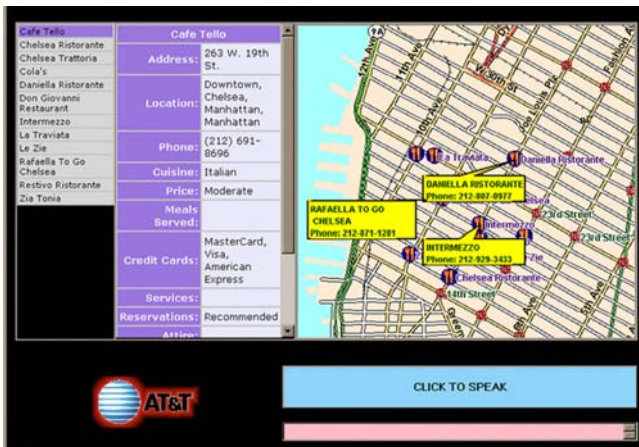


Fig. 2. MATCH user interface

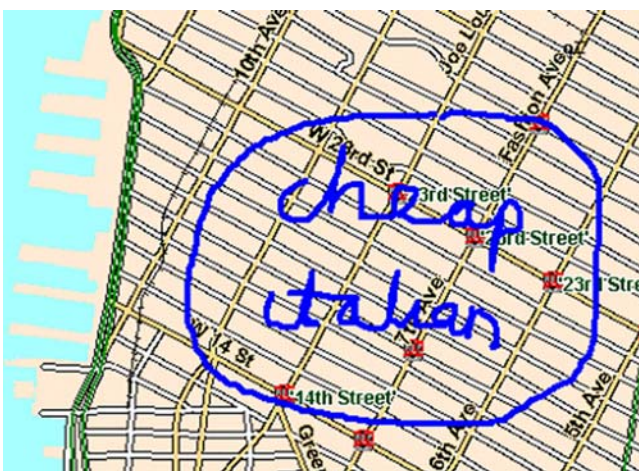


Fig. 3. Unimodal pen command to show restaurants

callouts on the display, synchronized with synthetic speech output. For example, the display in Figure 2 resulted from the user giving the multimodal command *what are the phone numbers for these places* and circling several restaurants on the map. These information seeking commands can also be issued solely with pen. For example, the user can circle a restaurant and write *review*. The user can also pan and zoom around the map. For example, they can say *show upper west side* or circle an area and say *zoom in here*. The system also provides subway directions. For example, if the user says *how do I get here* and circles or points at a location the system will ask for their current location. The user can respond either multimodally by saying *I am here* and pointing at their location, or unimodally with pen by pointing and writing *here*, or unimodally with speech by saying their location, for example *I'm at 57th St and Broadway*. The system then calculates the optimal subway route and dynamically generates a multimodal presentation indicating the series of actions the user needs to take. The systems starts by zooming in on the first station and then gradually zooms out graphically showing each stage of the route along with a series of synchronized TTS prompts. Figure 4 shows a subway route with two transfers.

MATCH works effectively using the in-built microphone on the

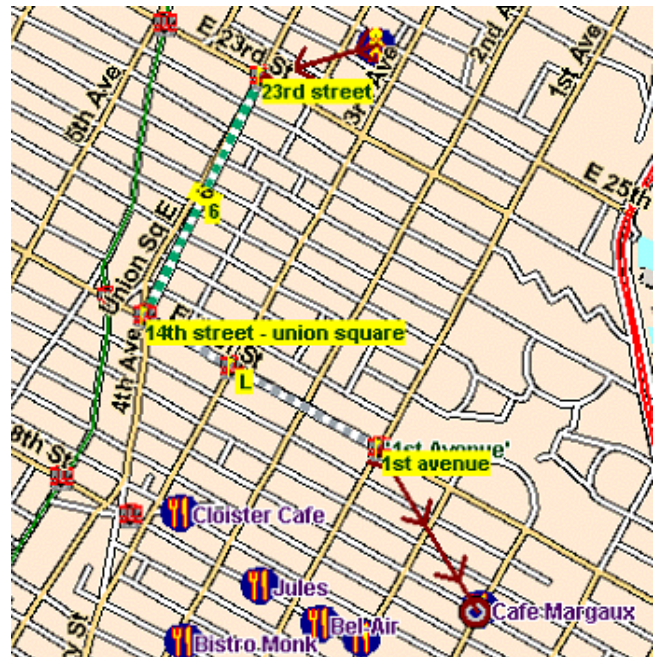


Fig. 4. Multimodal generation of subway route

top left of the Fujitsu device. It has been used in an initial trial and user study in New York City and is currently being used for multimodal data collection. In future work we will add additional functionality including access to cinema information, places of interest, bars and music venues. The next section provides details on the multimodal architecture which underlies MATCH.

3. MULTIMODAL ARCHITECTURE

The multimodal architecture which supports MATCH consists of a series of agents which communicate using a java-based facilitator agent MCUBE. The system architecture is as in Figure 5. Communications among agents are encoded in XML.

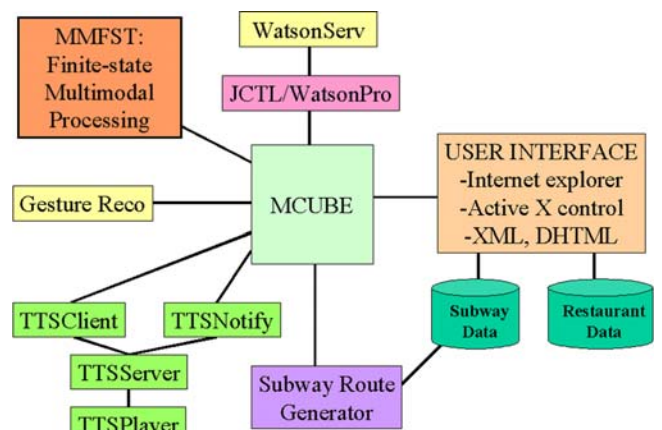


Fig. 5. Multimodal Architecture

3.1. Multimodal User Interface Client

The multimodal user interface client is browser-based and runs in Internet Explorer¹. This greatly facilitates rapid prototyping, authoring, and reuse of the system for different applications since anything that can appear on a webpage can be used in the multimodal user interface. System developers can use dynamic HTML and javascript to build multimodal UIs for new applications.

An ActiveX control provides a dynamic pan-able, zoomable map display. The map has been augmented with TCP/IP capability for communication with the MCUBE facilitator. It is augmented with ink handling capability. When the user draws on the map, their ink is captured, any objects potentially selected are determined, and the ink is processed using gesture recognition and handwriting recognition algorithms. The result is a gesture lattice indicating the different possible meanings of the ink input stream. This lattice is passed to the multimodal understanding component for composition with spoken input and determination of the joint interpretation of speech and gesture. The map component has also been augmented so it can respond to requests in XML to show particular subsets of restaurants or other entities. In essence, a query is received by the page and rendered both as a dynamic HTML listing and as graphical presentation of restaurant locations. The UI also has a system for intelligent placement of graphical callouts used for displaying restaurant information and other messages (Figure 2).

3.2. Speech Recognition and Text-to-speech

The system utilizes AT&T's Watson speech recognition engine. A local client running on the device (WatsonPro) gathers and audio and communicates with a Watson server running either on the device or on a server. AT&T's next-generation text-to-speech engine is integrated into the architecture and used to provide spoken output of restaurant information such as addresses and reviews, and for subway directions.

3.3. Subway Route Generator

The subway route generator component contains an exhaustive database of the NYC subway system. A constraint solving algorithm is used to identify the optimal route between two points, minimizing transfers and stops. This algorithm returns a list of the actions needed to complete the route and this is passed to a natural language generation component which assigns an appropriate prompt to each action. The actions and prompts are sent to the Multimodal UI which then coordinates presentation of graphical segments of the route which the appropriate TTS prompts.

4. FINITE-STATE MULTIMODAL PROCESSING

Multimodal integration involves merging semantic content from multiple streams to build a joint interpretation for a multimodal utterance. We employ a finite-state device to parse multiple input streams and to combine their content into a single semantic representation [3, 4]. For an interface with n modes, a finite-state device operating over $n + 1$ tapes is needed. The first n tapes represent the input streams and $n + 1$ is an output stream representing their composition. In the case of speech and pen input there are three tapes, one for speech, one for pen gesture, and a third for their combined meaning.

¹This approach follows on in part from work at AT&T on GMMI (Generic MultiModal Interface) [2].

In MATCH, users issue spoken commands such as: *tell me about these two restaurants* while gesturing on icons on the dynamically generated map display, or *show italian restaurants in this area* while drawing an area on the display, or *show restaurants along this route* while drawing a line along a street. The structure and interpretation of multimodal commands of this kind is captured declaratively in a multimodal context-free grammar. We present a fragment capable of handling such commands in Figure 6. The non-terminals in the multimodal grammar are atomic symbols. The multimodal aspects of the grammar become apparent in the terminals. Each terminal contains three components $W:G:M$ corresponding to the $n + 1$ tapes, where W is for the spoken language stream, G is the gesture stream, and M is the combined meaning. The epsilon symbol (ϵ) is used to indicate when one of these is empty in a given terminal. The symbols in W are words from the speech stream. The symbols in G are of two types. Sequences of symbols such as $G\ area\ location$ indicate the presence of a particular kind of gesture in the gesture stream, while those like SEM are used as references to entities referred to by the gesture. The meaning tape contains symbols which when concatenated together form coherent XML expressions. The multimodal grammar specification is compiled into an FSA using standard approximation techniques [5]. The transition symbols of the approximated FSA are the terminals of the context-free grammar and in the case of multimodal CFG as defined here, these terminals contain three components, W , G and M .

While a three-tape finite-state automaton is feasible in principle [6], currently available tools for finite-state language processing [7] only support finite-state transducers (FSTs) (two tapes). In order to implement our approach, we convert the three-tape FSA into an FST, by decomposing the transition symbols into an input component ($G \times W$) and output component M , resulting in a function, $T:(G \times W) \rightarrow M$. This corresponds to a transducer in which gesture symbols and words are on the input tape and the meaning is on the output tape. The domain of this function T can be further carried to result in a transducer that maps $\mathcal{R}:G \rightarrow W$. This transducer captures the constraints that gesture places on the speech stream and we use it as a language model for constraining the speech recognizer based on the recognized gesture string.

In order to capture multimodal integration using finite-state methods, it is necessary to abstract over certain aspects of the gestural content. For example, it is not possible to capture all of different possible sequences of coordinates that occur in gesture so that they can be copied from the gesture input tape to the meaning output tape. The gestural input is represented as a transducer \mathcal{S} that maps $I \rightarrow G$ where G are gesture symbols and I are the specific interpretations (see Figure 7). The G side contains indication of the type of gesture and its properties. In any place where there is specific content such as a list of entities or points in the gesture stream the symbol in G is the reserved symbol SEM. The specific content is placed on the I side opposite SEM. All G symbols other than SEM match with an identical symbol on I .

Individual gestures are decomposed into gesture symbol complexes of this basic form: $G\ FORM\ MEANING\ (NUMBER\ TYPE)\ SEM$. $FORM$ indicates the physical form of the gesture: and has values such as area, point, line, arrow. $MEANING$ indicates the specific meaning of that form for example an *area* can be either a *location* or a *selection*. $NUMBER$ and $TYPE$ are only found with *selection*. They indicate the number of entities selected (*1,2,3, many*) and the specific type of entity (*restaurant, theatre*). The $TYPE\ mixed$ is used for gestures at collections of entities of varied different types. As an example, if the user draws an area on the screen which contains a restaurant and a theatre, with identifiers *id1* and *id2*, the resulting

COMMAND	→	show:eps:<show> NP eps:eps:</show>
COMMAND	→	tell:eps:<info> me:eps:eps about:eps:eps DEICTICNP eps:eps:</info>
NP	→	eps:eps:<restaurant> CUISMOD restaurants:eps:eps LOCMOD eps:eps:</restaurant>
DEICTICNP	→	DDETSG SELECTION eps:1:eps RESTSG eps:eps:<restaurant> eps:SEM:SEM eps:eps:</restaurant>
DEICTICNP	→	DDETPL SELECTION NUM RESTPL eps:eps:<restaurant> eps:SEM:SEM eps:eps:</restaurant>
SELECTION	→	eps:area:eps eps:selection:eps
CUISMOD	→	eps:eps:<cuisine> CUISINE eps:eps:</cuisine>
CUISINE	→	italian:eps:italian chinese:eps:chinese
LOCMOD	→	eps:eps:<location> LOCATION eps:eps:</location> eps:eps:eps
LOCATION	→	in:eps:eps this:G:eps area:area:eps eps:location:eps eps:SEM:SEM
LOCATION	→	along:eps:eps this:G:eps route:line:eps eps:location:eps eps:SEM:SEM
DDETSG	→	this:G:eps RESTSG → restaurant:restaurant:eps
DDETPL	→	these:G:eps RESTPL → restaurants:restaurant:eps
		NUM → two:2:eps three:3:eps

Fig. 6. Multimodal grammar fragment

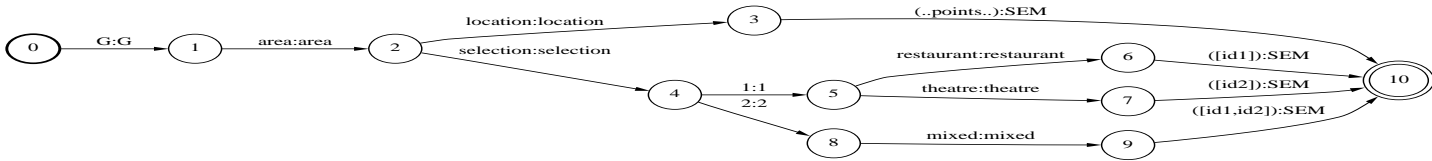


Fig. 7. Gesture Lattice: Restaurant and theatre

gesture lattice will be as in Figure 7. If the speech is *show chinese restaurants in this area* then the first path will be chosen when the multimodal finite-state device is applied. If the speech is *tell me about this restaurant* then the second, *selection*, path will be chosen. If they say *tell me about this theatre* the third path will be taken. But if they say *tell me about these two* the fourth path will be taken.

In order to carry out the multimodal composition with the transducer $\mathcal{R} : G \rightarrow W$ and $\mathcal{T} : (G \times W) \rightarrow M$, the output projection G of $\mathcal{S} : I \rightarrow G$ is taken. After composition we take a projection $U : G \rightarrow M$ of the resulting $G.W:M$ machine, basically we factor out the speech W information. We then compose \mathcal{S} and U yielding $I:M$. In order to read off the meaning we concatenate symbols from the M side. If the M symbol is SEM we instead take the I symbol for that arc. As an example, when the user says *show chinese restaurants in this area* and the gesture is as in Figure 7 the result is a $G.W:M$ machine. In order to compose this back with Figure 7 the $_W$ is removed. Reading of the meaning from the resulting $I:M$ we have the XML representation `<show><restaurant><cuisine>chinese</cuisine><location> (...points...) </location></restaurant></show>`. In [8], we present speech recognition accuracy results with and without multimodal input in the context of a directory assistance application. Our approach yields a average 23% relative sentence error reduction on clean speech.

5. CONCLUSION

MATCH gives users the flexibility to access NYC restaurant and subway information using speech, pen, or synchronized combinations of the two depending on their preferences and physical and social environment. The system responds by generating coordinated multimodal presentations incorporating dynamic graphics and text-to-speech output. All of the functions of MATCH can be used in the office, in the street, on a train, or in a meeting. It is built within a reusable multimodal application framework with a browser-based

UI which facilitates rapid prototyping and reuse for different applications. We have conducted initial user trials of MATCH in New York City and in ongoing work we are conducting a multimodal data collection exercise and evaluation of the system.

6. REFERENCES

- [1] Elisabeth André, “Natural language in multimedia/multimodal systems,” in *Handbook of Computational Linguistics*, Ruslan Mitkov, Ed. Oxford University Press, to appear.
- [2] Giuseppe DiFabrizio, Candace Kamm, Paolo Ruscitti, Shrikanth Narayanan, Bruce Buntschuh, Alicia Abella, Jim Hubbell, and Jerry Wright, “Extending a standard-based ip and computer telephony platform to support multi-modal services,” in *ESCA Workshop on Interactive Dialogue in Multimodal Systems*, Kloster Irsee, Germany, 1999.
- [3] Michael Johnston and Srinivas Bangalore, “Finite-state multimodal parsing and understanding,” in *Proceedings of COLING 2000*, Saarbrücken, Germany, 2000.
- [4] Michael Johnston and Srinivas Bangalore, “Finite-state methods for multimodal parsing and integration,” in *ESSLLI Workshop on Finite-state Methods*, Helsinki, Finland, 2001.
- [5] Mark-Jan Nederhof, “Regular approximations of cfls: A grammatical view,” in *Proceedings of the International Workshop on Parsing Technology*, Boston, 1997.
- [6] A.L. Rosenberg, “On n-tape finite state acceptors,” *FOCS*, pp. 76–81, 1964.
- [7] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley., *A rational design for a weighted finite-state transducer library*, Number 1436 in Lecture notes in computer science. Springer, Berlin ; New York, 1998.
- [8] Srinivas Bangalore and Michael Johnston, “Tight-coupling of multimodal language processing with speech recognition,” in *Proceedings of ICSLP*, Beijing, China, 2000.