# The Effect of Input Mode on Inactivity and Interaction Times of Multimodal Systems

Manolis Perakakis
Dept. of Elec. & Comp. Engineering
Technical Univ. of Crete
Chania 73100, Greece
perak@telecom.tuc.gr

Alexandros Potamianos
Dept. of Elec. & Comp. Engineering
Technical Univ. of Crete
Chania 73100, Greece
potam@telecom.tuc.gr

## ABSTRACT

In this paper, the efficiency and usage patterns of input modes in multimodal dialogue systems is investigated for both desktop and personal digital assistant (PDA) working environments. For this purpose a form-filling travel reservation application is evaluated that combines the speech and visual modalities; three multimodal modes of interaction are implemented, namely: “Click-To-Talk”, “Open-Mike” and “Modality-Selection”. The three multimodal systems are evaluated and compared with the “GUI-Only” and “Speech-Only” unimodal systems. Mode and duration statistics are computed for each system, for each turn and for each attribute in the form. Turn time is decomposed in interaction and inactivity time and the statistics for each input mode are computed. Results show that multimodal and adaptive interfaces are superior in terms of interaction time, but not always in terms of inactivity time. Also users tend to use the most efficient input mode, although our experiments show a bias towards the speech modality.

## Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology; Voice I/O; Natural language; Graphical user interfaces (GUI)*

## General Terms

Experimentation, Human Factors, Measurement, Performance

## Keywords

Input Modality Selection, Mobile Multimodal Interfaces

## 1. INTRODUCTION

The emergence of powerful mobile devices, such as personal digital assistants (PDAs) and smart-phones raises new design challenges and constraints that could be better addressed by a combination of more than one modalities. Combining multiple modalities efficiently is a complex task; one needs to identify and exploit the synergies that exist between the various interaction modalities, in order to build powerful multimodal interfaces.

It is widely supported that speech and graphical user interface (GUI) modalities are highly complementary ([1, 2, 3, 4]). As far as the style of interaction is concerned, GUI interfaces entail a constrained interaction style, while speech interfaces support a more natural style. As far as input is concerned, GUI interfaces have low error rates and offer easy error correction. In contrast, speech interfaces have high error rates and errors are harder for the users to fix, frequently causing frustration. Also they are inconsistent in terms of input, since they may produce different recognition results for the same user utterances, causing an uncertainty feeling to users. Finally, as output is concerned, visual output is fast (parallel) compared to much slower (sequential) speech output.

Few guidelines exist for selecting the appropriate mix of modalities in multimodal systems [5, 6]. It is established that visual modality is more efficient than speech [4], while speech is a more natural interaction mode. However, it is often the case when designing multimodal user interfaces, that the developer is biased either toward the voice, or the visual modalities. This is especially true, if the developer is voice-enabling an existing GUI-based application or building a GUI for an existing voice-only service. Our goal in this paper is to investigate input modality usage from the user point of view and to better understand efficiency considerations and user biases in input mode selection. Such information would be valuable for user modeling and multimodal dialogue system design in general.

For this purpose, we have implemented a travel reservation form-filling multimodal dialogue system that combines the visual and speech modalities on PDA environments, using three different configurations, namely: “Click-To-Talk”, “Open-Mike” and “Modality-Selection”. The two unimodal (“Speech-Only”, “GUI-Only”) and three multimodal systems are described in the next section. Our goal is not only to compare the efficiency and the objective metrics among the different systems, as is typically done in the literature, but to also measure the various factors that could affect the efficiency and modality choice by the user. For this purpose, we compute interaction and inactivity times within a turn to better understand the effect of input modality on interface efficiency. In addition, we measure modality usage

ICMI’07, November 12-15, 2007, Nagoya, Aichi, Japan.

for different levels of relative efficiency of the input modes. General conclusions can be drawn from these experiments that can guide us through the multimodal interface design process.

The rest of this paper is organized as follows. In Section 2, the two unimodal and three multimodal systems are described. Objective evaluation metrics are presented in Section 3 and evaluation results are presented in Section 4. We conclude with an analysis of the most important results in Section 5 and present our conclusions and future work directions in Section 6.

## 2. UNIMODAL AND MULTIMODAL INTERACTION

In this paper, we investigate unimodal and multimodal interaction for a generic form-filling system that implements a travel reservation system application (flight, hotel and car reservation). The spoken dialogue system used and the application scenario is based on the Bell Labs Communicator system described in [7, 8]. The following input modalities are available to the user: keyboard, mouse and speech input for the desktop version of the system, and pen and speech input for the personal digital assistant (PDA) version. The following systems were implemented and are evaluated in this paper:

- Two unimodal ones, namely, "GUI-Only" (GO) and "Speech-Only" (SO).
- Three fully multimodal systems with the following interaction modes: "Click-To-Talk" (CTT), "Open-Mike"(OM) and "Modality-Selection"(MS).
- A constrained multimodal system "Open-Mike-Speech-Input" (OMSI) that allows unimodal speech input and multimodal output.

The various interfaces and interaction modes are described next. Only the form-filling part of the travel reservation application is analyzed in this paper.

### 2.1 Unimodal GUI Interaction

The graphical user interface (GUI) is implemented as a series of forms; each form contains attribute-value pairs, each employing label and text-field/combo-box components, respectively. Two versions of the GUI are implemented: a desktop version which allows for keyboard and mouse input (GUI uses both text fields and combo boxes - see [9]) and a PDA version which only allows for pen input (GUI uses only combo boxes - see Fig. 1). The choice of using text field or combo-box for a certain attribute field, is based on efficiency considerations; that is the number of values that an attribute takes. For attributes with less than 25 options, a combo-box is used, otherwise a text-field (see Table 2). For the PDA GUI on the other hand, *all* data entry fields are implemented as combo boxes (a pull-down menu that contains all possible values) due to the slow text input methods available on such devices. The number of options available to the user in some of these combo boxes is quite large, e.g., 250 choices for the "hotelname" attribute. The following features are common for both the desktop and PDA GUI: (1) ambiguity is shown as a pull-down box with a list of choices and highlighted in red, (2) error messages are represented in the GUI as pop-up windows, (3) fields and buttons that become inaccessible in the course of the interaction are "grayed out", and (4) the context (or focus) of the interaction is highlighted.

### 2.2 Unimodal Speech Interaction

The "Speech-Only" interface is identical to the one described in [10, 7, 8]. In brief, the spoken dialogue manager promotes mixed-initiative system-user interaction. All types of user requests and user input are allowed at any point in the dialogue, i.e., the full application grammar is active throughout the interaction. The system prompts are focused and try to elicit specific information from the user, e.g., the value of an attribute. Explicit confirmation is used only to confirm the values of the attribute at the form level, e.g., for all flight leg user supplied information. Implicit confirmation is used in all other cases throughout the interaction. The audio subsystem we developed for the speech interface allows for both *Voice Activity Detection* (VAD) and *barge-in*, i.e., users speaking over system prompts.

### 2.3 Multimodal Interaction

Three different multimodal (MM) interaction modes have been implemented for combining the visual and speech modalities. The output interface is common for all interaction modes to allow us to better investigate the effectiveness of the "optimum" input modality mix. The visual output is identical to the corresponding "GUI-Only" mode.

Audio output prompts were significantly shortened compared with the unimodal "Speech-Only" case. Specifically, implicit confirmation prompts were not used in the multimodal case because confirmation was efficiently done via the visual modality. In addition, form creation prompts and explicit confirmation prompts were significantly shortened or not used at all, depending on the interaction context. Finally, information request prompts were shortened too (typically to the name of the attribute requested, e.g., "Departure city?"). In general, speech output was mainly used as a way to grab the attention of the user, emphasizing information already appearing on the screen. The speech interface was identical for all three multimodal modes. Note that in all three multimodal modes only one modality is active at a time, i.e., the system does not allow for concurrent multimodal input[1]. In our current multimodal implementation, visual input is not allowed (GUI is "grayed-out") while speech input is active. For all multimodal modes, users are free to override the system's proposed modality, that is use a modality other than system's default, e.g. GUI input for "Open-Mike" mode. Next, each multimodal mode is presented in more detail for the PDA (the analysis holds also for the desktop system).

#### 2.3.1 Multimodal Interaction Modes

The main difference between the three multimodal interaction modes is the default input modality. For "Click-To-Talk" interaction, pen is the default input; the user needs to click the "Speech Input" button to override the default input modality and use speech input. For "Open-Mike" interaction, speech is the default input modality; the system is always listening and a VAD event activates the recognizer. Again the user can override by pressing with the pen anywhere on the GUI; "Modality-Selection" is a mix of

[1] For information-seeking/ form-filling multimodal applications this is not a major limitation.

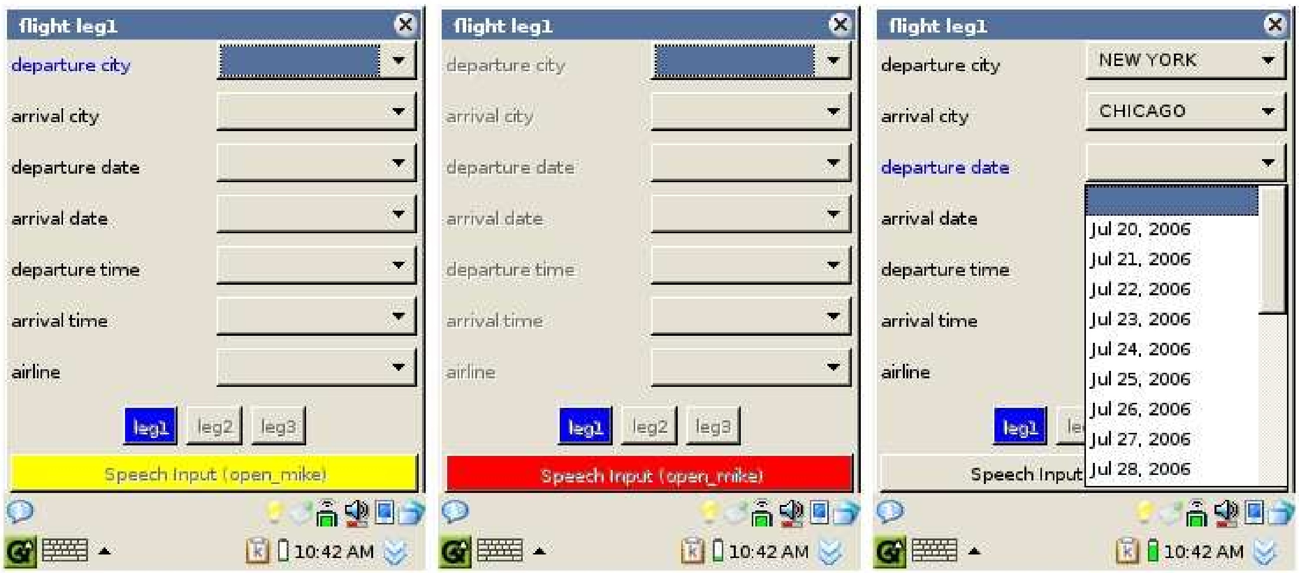

**Figure 1: "Modality-Selection" interaction mode examples on the PDA. System is in "Open-Mike" mode in the first frame (speech button is yellow indicating waiting for input), receives user input "From New York to Chicago" during the second frame (speech button is red showing activity) and switches to "Click-To-Talk" mode in the third frame. The speech/pen input default mode is selected by the system in the first/third frame, respectively, due to the large/small number of options in the combo-box.**

the "Click-To-Talk" and "Open-Mike" interaction; the system switches between the two multimodal modes depending on efficiency considerations (the size of the current combo-box). For combo-boxes with less than 25 values, the system goes into "Click-To-Talk" mode and visual input is the default mode, otherwise the system goes into "Open-Mike" mode where speech input is the default. Speech input is faster compared to pen input for long combo-boxes on the PDA and the threshold of 25 options was chosen based on the input mode efficiency of the stereotypical user. Overall, "Click-To-Talk" is GUI-biased, while "Open-Mike" is speech biased; "Modality-Selection" tries to balance the two input modes based on efficiency considerations.

In Fig. 1, examples from the "Modality-Selection" mode running on the PDA, are shown. Initially the system is in "speech waiting" state; the user input "from New York to Chicago" activates the speech recognizer (VAD event) and the GUI becomes disabled. Once the recognizer returns the recognized utterance, the GUI is updated and the modality is selected for the next turn ("modality selection" state). For the next turn, visual input is selected (focus is on "departure date" for which a combo box with less than 25 choices is available) and the system goes to the "GUI input" state.

#### 2.3.2 *Speech-only input and multimodal output*

For the purposes of completeness and to better investigate the effect of "visual feedback" in spoken dialogue interaction a system with limited multimodal capabilities was also implemented, namely "Open-Mike-Speech-Input" (OMSI). The user is allowed only speech input (unimodal input), while the system output is multimodal including both speech and visual feedback. OMSI interaction is equivalent to "Open-Mike" interaction with visual (GUI) input disabled. Alternatively OMSI can be seen as a "Speech-Only" system with visual feedback and shortened prompts. Note that the OMSI prompts are identical to the MM system prompts. Table 1 shows the six systems in terms of input and output modes.

**Table 1: Input and ouput modes in the six implemented systems.**

| | input modes | | output modes | |
|---|---|---|---|---|
| **system** | **GUI** | **speech** | **GUI** | **speech** |
| GO | √ | x | √ | x |
| SO | x | √ | x | √ |
| OMSI | x | √ | √ | √ |
| CTT/OM/MS | √ | √ | √ | √ |

## 3. EVALUATION METHODOLOGY

### 3.1 Evaluation Setting

The "GUI-Only" and the three multimodal interaction modes were evaluated for both the desktop and PDA environments. The two speech-only input modes, namely "Speech-Only" and "Open-Mike-Speech-Input" (OMSI), were also evaluated independently. Thus a total of ten systems were evaluated. Evaluation took place in an office environment, with all software (spoken dialogue system, speech platform, visual interface) running on the same host computer for the desktop and speech-only systems. For the PDA system, evaluation took place with all the back-end software (spoken dialogue system, speech platform) running on the same host desktop computer and the front-end (visual interface) running on a Zaurus Linux PDA device.

All systems were evaluated using five scenarios of varying complexity: one/two/three-legged flight reservations and

**Table 2: Attributes for the travel reservation application, ordered by the number of available values in their combo box. Attribute usage per scenario and total usage is also shown.**

| attribute name | number of values | scenario usage | total usage |
|---|---|---|---|
| hotelname | 250 | (0/0/1/0/0) | 1 |
| city | 135 | (2/3/3/3/3) | 14 |
| airline | 93 | (1/1/1/1/1) | 5 |
| date | 22 | (1/2/2/2/3) | 10 |
| car type | 15 | (0/0/0/1/0) | 1 |
| car rental | 10 | (0/0/0/1/0) | 1 |
| time | 9 | (1/2/2/2/3) | 10 |

round trip flights with hotel/car reservation. The attributes size and usage for all five scenarios is shown in Table 2. Note, that the sum (total usage) of attribute occurrences in the evaluation scenarios, is very similar for the "long" and "short" attributes (20 for long and 22 for short ones)[2]. Assuming that the users select input modes solely based on efficiency considerations, we would expect the use of the speech and GUI input modes to be equally balanced when averaged over all scenarios (more on this in Section 3). Eight non-native English-speaking users evaluated all systems on all five scenarios. All users, had some prior experience using spoken dialogue systems; these users were also used in the pilot studies described in [10].

The evaluation procedure was as follows. First, users were given a short introductory document which explained the system functionality with emphasis on the modes to be evaluated. In order to familiarize users with the system before actual evaluation takes place, users were asked to complete a demo scenario using all different systems (on average the users spent 20 minutes familiarizing themselves with the systems). Finally evaluation took place; users were asked to complete all five scenarios using all ten systems, a total of 50 sessions per user. Systems were evaluated in random order and detailed logs for each session were saved for off-line processing by our analysis software (objective evaluation). Upon completion of each session, users are asked to evaluate the system by filling out a questionnaire (subjective evaluation). Upon completion of all runs, an exit interview is conducted. In this paper, we focus only on the objective evaluation results.

## 3.2 Objective Evaluation Metrics

Interface evaluation of multimodal dialogue systems is a fairly complex task and different metrics may be used to evaluate aspects of such systems. Our goal in this paper is to investigate the relationship between modality usage and the efficiency of each mode, in terms of time required to complete a task. As a result, we mainly focus our objective evaluation on two dependent variables: modality usage (GUI vs speech) and user turn duration. To better understand the relationship between mode and turn duration we further decompose the turn duration into user inactivity and user interaction time. Furthermore, the mode usage and turn duration is analyzed not only as a function of the different MM systems, but also as a function of the context, i.e., the main attribute that is being filled at each turn.

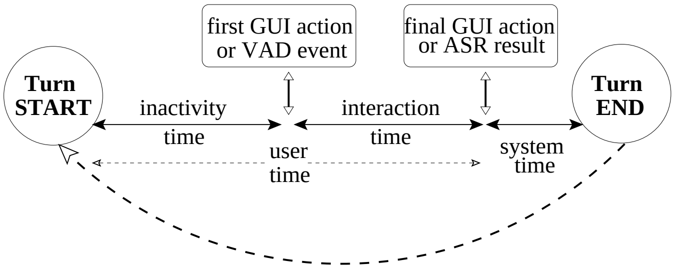

**Figure 2: Turn decomposition into user and system time. User time decomposition into inactivity and interaction times.**

In this paper, we focus in the form filling part of the interaction and most specifically on how the user provides attribute-value pairs to the system. Other parts of the interaction such as confirmation questions, verification requests, and navigation among forms were not included in our current analysis[3]. The main reason for this is that for the vast majority of these actions users used GUI input, as it was clearly the faster and easier way to respond, e.g. click "Yes" on a dialog window. By excluding the navigation, confirmation and verification actions we avoid biasing the evaluation results.

Out of the 40 runs per system (eight users times five scenarios) there are only two non-completed runs for the OMSI system and one for the "Speech-Only" system. We focus our analysis on the completed tasks, i.e., on 38 out of the 40 runs that were completed for all ten systems.

In Fig. 2, the decomposition of a turn into user and system time is shown. System time refers to internal system actions that are common to all implemented systems and can thus be safely ignored. User time is further broken down into user inactivity and interaction time. Interaction time is defined as the time during which the user is providing input to the system, while inactivity time is the remainder of the user turn.

### 3.2.1 *Duration and mode statistics*

The following objective measures are computed for all ten systems and for the 38 completed runs: (i) the total time to completion for each system, time to completion is summed over all scenarios, and (ii) the total number of turns per system for all scenarios. Mode statistics are computed for the two metrics defined above, i.e., time to completion is summed up separately for the GUI and speech input turns. Duration, turns and mode statistics are also computed per context (main elicited attribute) but are not shown in this paper.

### 3.2.2 *Inactivity and interaction time statistics*

The task duration and mode statistics are further refined into interaction and inactivity times and corresponding mode usage. As discussed above, inactivity time[4] refers to

[2] We use the term "short" for attributes that have less than 25 value options in their combo box; for those attributes pen is the most efficient input mode.

[3] Note that error correction turns are included. We exclude from our analysis only turns that are responses to YES-NO questions such as "Is this a one way trip?" or "Is this correct?" (after filling each form).

[4] The term "inactivity" refers to the fact that the user *ap-*

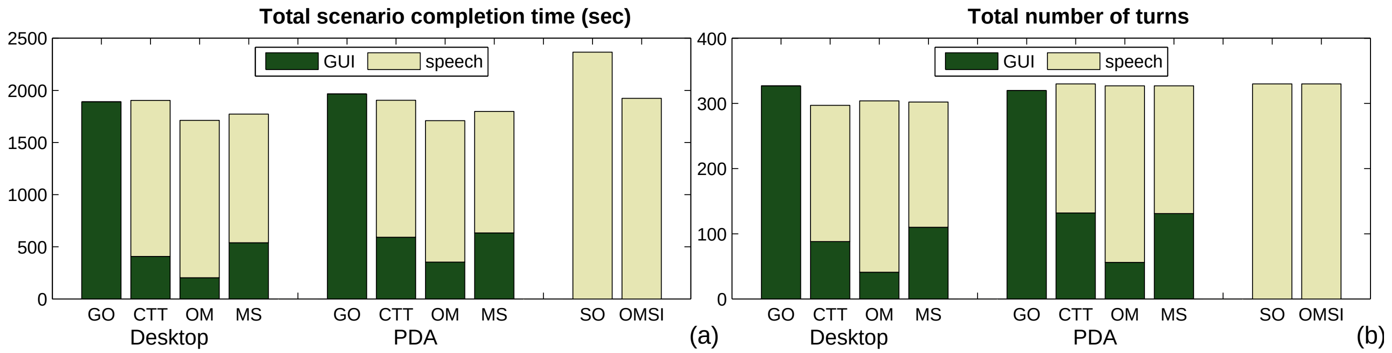

**Figure 3: Duration and turn cummulative statistics shown for each of the desktop and PDA systems summed over all scenarios: (a) total time to completion in seconds, (b) total number of turns. The color-codes for each system bar show the total time and number of turns for GUI and speech input respectively.**

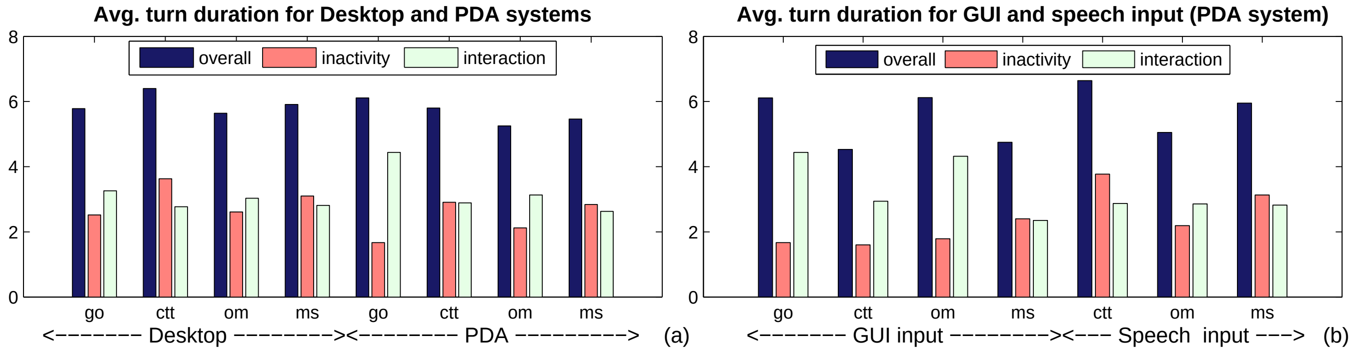

**Figure 4: (a) Inactivity and interaction times in secs for Desktop and PDA systems. (b) Inactivity and interaction times for GUI and speech input for PDA systems only.**

the time interval starting at the beginning of each turn, until the moment the user actually interacts with the system using GUI or speech input. During this interval, the user has to comprehend system's response and state and then plan his own response. The response typically includes entering information requested by the system (after reading the scenario information) using the preferred modality for that specific turn.

For GUI input, the inactivity time is defined as the time interval between the turn start time and the moment the user clicks on the combo-box (PDA case). For speech input, inactivity time is defined as the time interval between the turn start time and the moment of a VAD event ("Click-To-Talk" has voice activity detection enabled in this evaluation), that is the moment the audio subsystem has detected speech activity. For GUI input, interaction time is defined as the time interval between the instant the user clicks on the combo-box and the moment the user selects the desired value using pen input. For speech input, interaction time is defined as the time interval between the moment of the VAD event and the moment speech recognition result becomes available (assuming recognition nearly real-time, as is the case for our system).

We have computed the inactivity and interaction time for each of the ten systems. In addition, the distribution of inactivity and interaction times as a function of context was computed for the four main attributes: city, airline, date, time. The distributions are computed over all four PDA systems. Kernel density functions were used to estimate the distributions.

# 4. EVALUATION RESULTS

## 4.1 Duration and Mode Statistics

Task duration and number of turns for all ten systems (four for Desktop, four for PDA and two speech input modes) are shown in Fig. 3(a) and Fig. 3(b) respectively for the 38 completed runs per system.

For the speech-only systems, one way ANOVA shows that the task duration times significantly differ (F=76.82, $p < 0.05$), i.e., OMSI is significantly faster than "Speech-Only". For the PDA systems, one way ANOVA shows that the task durations of the four systems differ (F=5.8095, $p < 0.05$). A multiple comparison reveals that "GUI-Only" significantly differs compared to both "Open-Mike" and "Modality-Selection" modes but not compared to the "Click-To-Talk" mode. For the Desktop environment, one way ANOVA shows that the task durations differ (F=5.9134, $p < 0.05$). A multiple comparison reveals that "Click-To-Talk" significantly differs compared to all three other modes. Overall, "Speech-Only" is the less efficient mode. For both desktop and PDA environments, "Open-Mike" is the fastest mode, closely followed by "Modality-Selection" mode and then by the slower "GUI-Only" and "Click-To-Talk" modes.

*pears* inactive to the system.

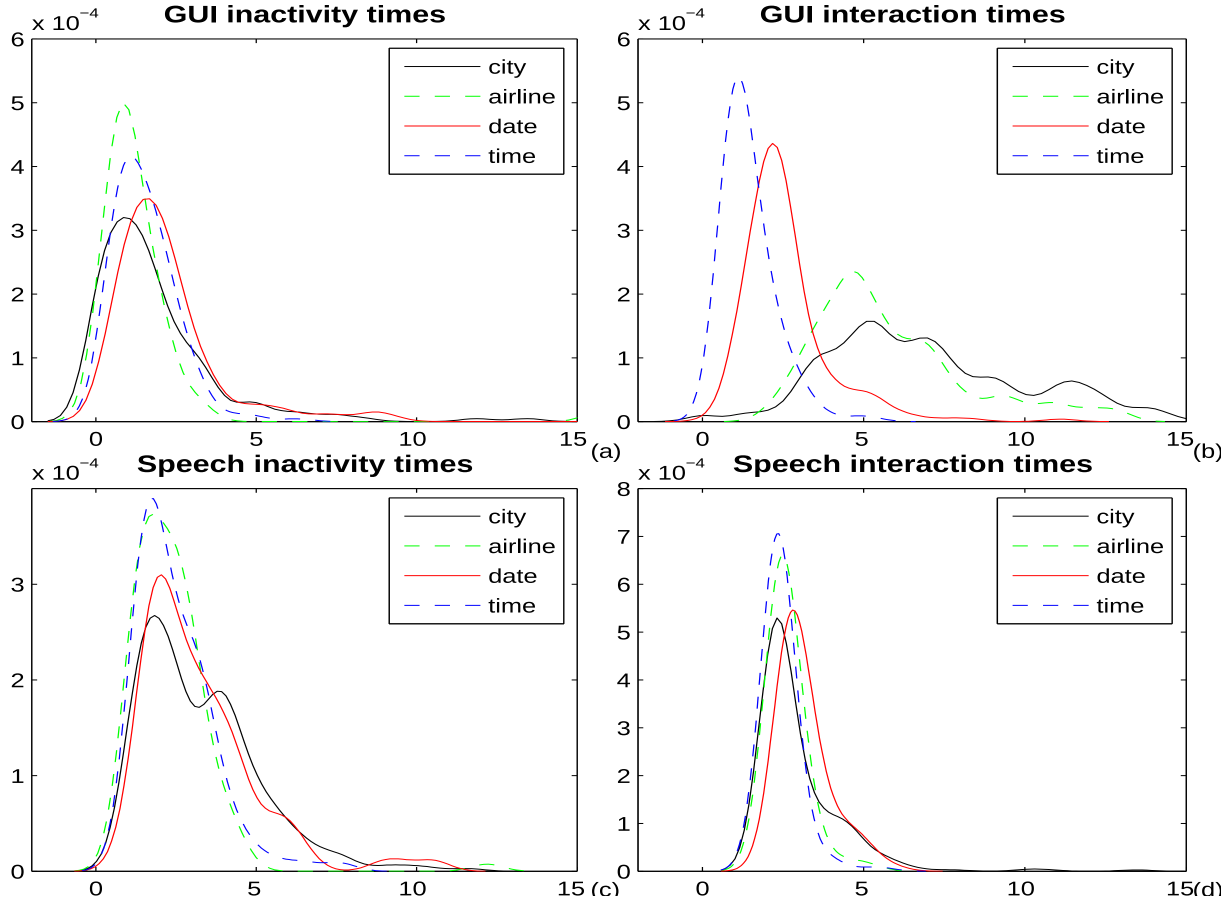

**Figure 5: Distributions of average turn duration in seconds broken down into inactivity and interaction times for the four most frequently-used contexts (city, airline, date, time). Results are cummulative for the following PDA systems: GO, CTT, OM, MS. Distributions approximated using kernel density functions. (a) Avg. inactivity time distribution for pen input. (b) Avg. interaction time distribution for pen input. (c) Avg. inactivity time distribution for speech input. (d) Avg. interaction time distribution for speech input.**

In terms of input mode usage, speech was used more often for the "Open-Mike" systems, while the "Click-To-Talk" and "Modality-Selection" system show non-significant differences in mode usage on the PDA. In all three multimodal systems, speech was used much more often than GUI input.

## 4.2 Inactivity and Interaction Time Statistics

In Fig. 4(a), interaction and inactivity times are shown for both desktop and PDA environments. The three multimodal systems have shorter interaction times compared to the "GUI-Only" unimodal system (also holds for the "Speech-Only" system not shown in the figure). The differences are significant and more pronounced on the PDA than on the desktop. The opposite is true for the inactivity time; the "GUI-Only" system has significantly shorter inactivity times compared to the three MM systems. All differences are significant with the exception of the "Open-Mike" system on the desktop.

In Fig. 4(b), the inactivity and interaction times are shown only for the PDA, as a function of system type and input mode (GUI vs. speech). For all evaluated systems (with the notable exception of the "Modality-Selection" system), the interaction times for GUI input are significantly higher than the corresponding interaction times for speech input, while the opposite holds for inactivity times. Comparing the inactivity times among systems for GUI input, there is no significant difference among them, with the exception of "Modality-Selection". However, the interaction times vary greatly among systems for GUI input. Conversely for speech input, the interaction times are not significantly different among the three MM systems, but the inactivity times are significantly different.

In Fig. 5(a)-(d) the estimated probability density functions (PDFs) for inactivity and interaction turn times are shown for GUI and speech input. The cumulative distributions are computed over the GO, CTT, OM and MS systems running on the PDA, for the four most common attributes. The means of the four attribute-dependent PDFs are similar for the four contexts in plots (a), (c) and (d). However, the GUI input average turn interaction time shown in (b) is very much attribute dependent and orders attributes from high to low interaction time as follows: city, airline, date, time. The variance of the PDFs for city and date attributes is also higher for speech input in (c), (d). Other fine differences between the PDFs are discussed in the next section.

# 5. ANALYSIS OF RESULTS

Numerous interesting conclusions can be drawn from the evaluation results presented above. Namely:

- The task duration results in Fig. 3(a) clearly show the importance of having "visual feedback" in a spoken dialogue system. By adding visual output to the "Open-Mike-Speech-Input" (OMSI) the efficiency increases dramatically compared to the "Speech-Only" (SO) system.
- Among the three multimodal systems the "Click-To-Talk" system is clearly the least efficient. This is due to the numerous default modality overrides (pressing the "Speech Input" button) as can be seen in Fig. 3(b) and Fig. 4(b).
- The mode statistics results in Fig. 3(b) clearly show that the multimodal system biases the input mode usage (CTT vs. OM). Users tend to use GUI input more often when it is the default input mode (in CTT), compared to the OM system where speech in the default input mode.
- From Fig. 4(b), we can see that GUI input has on average lower inactivity times, while speech input has lower interaction times. Although speech is the most efficient in terms of input (interaction times), recognition errors and context switching incurs higher cognitive load to the user resulting in higher inactivity times for speech input.
- The "adaptive" "Modality-Selection" system, which at each turn suggests to the user the most efficient input mode, is better compared to the other multimodal systems in terms of interaction times, however it typically has higher inactivity times. This is due to the increased cognitive load that adaptivity incurs on the user; automatically switching between default input modes is sometimes inconsistent and confusing. This is a common problem for adaptive interfaces.
- In Fig. 5(b), we see that the mean interaction times for GUI input are shorter for attributes with fewer options in the combo box, as expected. For speech input, the PDFs shown in Fig. 5(d) are very similar for all attributes (the attribute "date" requires multi word input and thus has slightly longer interaction times). Comparing the interaction times per attribute, it is clear that GUI input is more efficient for "time" and "date", while speech input is more efficient for "city" and "airline".
- Based on the observation above and given the almost 50-50% balancing between "GUI-efficient" and "speech-efficient" attributes in the scenarios, one would expect a 50-50% input mode usage split between GUI and speech. However, the results show that for all multimodal systems speech input is used for over 60% of the turns. Although users tend to use the more efficient mode at each turn, these results indicate a clear speech bias; speech input is used for attributes where GUI would be more efficient. It is unclear if this is a "novelty effect" or the bias indicates a preference for the more "natural" modality.

Combining multiple modalities efficiently is a complex task and requires both good interface design and experimentation to determine the appropriate modality mix. From the analysis of the relative efficiency of the input modes and from the mode usage results it is clear that a relationship between input mode selection and mode efficiency exists but is not perfectly linear. More research is needed to quantify the nature of this relationship.

# 6. CONCLUSIONS

In this paper, we have evaluated two unimodal and three multimodal form-filling systems on the desktop and PDA environments. The objective evaluation metrics used included mode and task duration statistics. These objective metrics were also calculated on a per attribute basis and broken down into the interaction and inactivity part of a dialogue turn. This detailed evaluation yielded some obvious and not-so-obvious results that can help us better understand human-machine interaction for multimodal dialogue systems. Here are some important conclusions from our analysis: (1) Synergies between the speech and visual interaction modes exist in multimodal interfaces; among these synergies visual feedback (GUI output) plays an important role. (2) When changing the relative efficiency of the input modes in multimodal interfaces, user input mode usage also changes; users tend to use the most efficient modality but biases also exist. (3) Multimodal and adaptive interfaces are almost always better in terms of shorter interaction times, but inactivity time may increase due to increased cognitive load of the user. Keeping these points in mind can help us design better multimodal systems.

Future work will focus on evaluating the unimodal and multimodal systems for varying levels of task complexity and unimodal interface efficiency (e.g., different speech recognition error levels). Through these experiments multiple measurement points for mode usage, unimodal and multimodal interface efficiency will be obtained; these results will help us better understand the relationship between efficiency, user satisfaction and input mode usage. We will also perform longitudinal studies to investigate possible "novelty effects". By incorporating this knowledge into the multimodal dialogue system design process we aim at building adaptive multimodal interfaces that are natural, efficient and improve on the state-of-the-art.

# 7. ACKNOWLEDGMENTS

This work was partially supported by the EU-IST-FP6 MUSCLE network of excellence and the GSRT 03ED375 PENED research project. The PENED research project is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%). The authors wish to thank Prof. Aikaterini Mania for valuable discussions and comment on evaluation issues, Michail Toutoudakis and Fotis Garitos for project involvement and all the users who participated in the evaluation.

# 8. REFERENCES

[1] J. Lai and N. Yankelovich, "Conversational speech interfaces," pp. 698–713, In The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications, Lawrence Erlbaum Associates, NJ, 2003.

[2] M.A. Grasso, D.S. Ebert, and T.W. Finin, “The integrality of speech in multimodal interfaces,” *ACM Trans. Comput.-Hum. Interact.*, vol. 5, no. 4, pp. 303–325, 1998.

[3] Cohen, P., Oviatt, S., “The Role of Voice in Human-Machine Communication,” In Voice Communication Between Humans and Machines. Roe, D., Wilpon, J. (editors). National Academy Press, Washington D.C.: 34-75, 1994.

[4] P. Cohen, M. Johnston, D. McGee, S. Oviatt J. Clow, and J. Smith, “The efficiency of multimodal interaction: A case study,” in *Proc. Internat. Conf. Speech Language Processing*, 1998.

[5] V. Bilici, E. Krahmer, S. teRiele, and R. Veldhuis, “Preferred modalities in dialogue systems,” Beijng, 2000.

[6] N.O. Bernsen and L. Dybkajer, “Is speech the right thing for your application?,” in *Proc. Internat. Conf. Speech Language Processing*, Sydney, Australia, Dec. 1998.

[7] A. Potamianos, E. Ammicht, and H.-K. Kuo, “Dialogue management in the Bell Labs communicator system,” Beijng, 2000.

[8] A. Potamianos, E. Ammicht, and E. Fosler-Lussier, “Modality tracking in the multimodal Bell Labs Communicator,” in *Proc. ASRU Workshop*, 2003.

[9] M. Perakakis, M. Toudoudakis, and A. Potamianos, “Modality selection for multimodal dialogue systems,” Internat. Conf. on Multimodal Interfaces, 2005.

[10] A. Potamianos, E. Fosler-Lussier, and E. Ammicht, “Information Seeking Spoken Dialogue Systems-Part II: Multimodal Dialogue,” *IEEE Transactions on Multimedia*, Apr. 2007.

[11] A. Potamianos et al, “Design principles and tools for multimodal dialog systems,” in *Proc. ESCA Workshop Interact. Dialog. Multi-Modal Syst.*, Kloster Irsee, Germany, June 1999.

[12] M. Perakakis, M. Toutoudakis, and A. Potamianos, “Blending speech and visual input in multimodal dialogue systems,” in *IEEE/ACM Workshop on Spoken Language Technology*, Aruba, Dec. 2006.