MULTI-MODAL INTERACTION IN THE AGE OF INFORMATION APPLIANCES

Stéphane H. Maes and T. V. Raman
IBM T. J. Watson Research Center
e-mail: {smaes, tvraman}@us.ibm.com

Abstract

The coming millenium will be characterized by the availability of multiple information appliances that make ubiquitous information access an accepted fact of life. The ability to access and transform information via a multiplicity of appliances, each designed to suit the user's specific usage environment, requires the exploitation of all available input and output modalities to maximize the band-width of man-machine communication.

There will be an increasingly strong demand for devices that present the same set of functionalities when accessing and manipulating the information, independently of the access device. The resulting uniform interface must be inherently multi-modal and dialog driven. During this evolution, conventional phones will remain a major type of access device. Also, the clients will offer more and more complex functions on ever more miniaturized devices. Eventually, we need to use these devices while performing other activities, and therefore often in hands-free or eyes-free mode.

This paper addresses the challenges of coordinated, synchronized multimodal user interaction that is inherent in designing user interfaces that work across these multiplicity of information appliances. Amongst the key issues to be addressed are the user's ability to interact in parallel with the same information via a multiplicity of appliances and user interfaces, and the need to present a unified, synchronized view of information across the various appliances that the user deploys to interact with information. We achieve such synchronized interactions and views by adopting the well-known Model, View, Controller *(MVC) design paradigm [7] and adapting it to conversational interactions.*

The resulting Conversational MVC (CMVC) is to be considered as the key underlying principle of any conversational multi-modal application.

1 Introduction

The computing world is presently evolving towards an era where billions of interconnected pervasive clients communicate with powerful information servers. This evolution will mean that soon, personal information devices will offer ubiquitous access, bringing with them the ability to create, manipulate and interchange information anywhere and anytime —using interaction modalities most suited to the user's current needs and abilities.

Such appliances will come to include familiar access devices such as telephones and pocket organizers and will vary widely in the interface peripherals they use to communicate with the user. At the same time, as this evolution progresses, users will demand a consistent *look, sound and feel* in the user experience provided by these plethora of information appliances.

The increasing availability of information, along with the rise in the computational power available to each user to manipulate this information, brings with it a concomitant need to increase the bandwidth of man-machine comunication; users will come to demand multimodal interaction in order to maximize their interaction with information appliances in hands-free, eyes-free environments. In addition, the availability of a plethora of information appliances will encourage multiple parallel interactions with electronic information akin to what users expect today in the world of traditional human-intermediated information interchange. Realizing these dreams will require fundamental changes in the user interface, as described in this paper and in [4, 3]; lacking this, users will be unable to access, act on, and transform information independently of the access device.

Additionally, application developers wishing to support information access through multiple devices must today heavily invest in explicit adaptation of information content and the user-interfaces used to interact with that content to every device category of interest — i.e., requiring multiple re-developement efforts. Whenever a new device class is considered, the development must be repeated. Even if limited to broad categories like HTML, WML, VoiceXML and synchronized versions of these mark-up languages, the investement required to maintain up-to-date code bases and content bases is enormous.

In the following sections, we outline the challenges of synchronized, coordinated user interactions using multiple modalities and multiple devices. Management of the resulting distributed systems, when multiple devices are used, will be described elsewhere.

2 Need For Conversational Computing

The evolution of the computer world towards billions of pervasive devices interconnected via the internet, wireless networks or spontaneous networks (see for example Bluetooth [1] and Jini [2]) will revolutionize the principles underlying man-machine interaction. These new devices define new market needs —the ability to access and manage information from anywhere, at anytime and through any device. This last requirement encompasses traditional phones, cell phones, smart phones and PCs.

Information being manipulated via such appliances might be located on the local device or accessible from a remote server via the network using open, interoperable protocols and standards. Usage of such open standards also leads to a seamless integration across multiple networks and multiple information sources such as an individual's personal information, corporate information available on private networks, and public information accessible via the global Internet. This availability of a unified information source will define productivity applications and tools of the future; users will increasingly interact with electronic information —as opposed to interacting with platform-specific software applications as is done today in the world of the desktop PC.

Information-centric computing carried out over a plethora of multimodal information devices will be essentially *conversational* in nature and will foster an explosion of conversational devices and applications. This trend towards pervasive computing goes hand-in-hand with the miniaturization of the devices and the dramatic increases in their capabilities.

With the pervasiveness of computing causing information appliances to merge into the users environment, the user's mental model of these devices is likely to undergo a drastic shift. Today, users regard com-

puting as an activity that is performed at a single device like the PC. As information appliances abound, user interaction with these multiple devices needs to be grounded on a different set of abstractions. The most intuitive and effective user model for such interaction will be based on what users are already familiar with in today's world of human-intermediated information interchange, where information transactions are modeled as a *conversation* amongst the various participants in the conversation. Notice that here the term *conversation* is used to mean more than speech interaction —here, *conversation* is used to encompass all forms of information interchange, where such interchange is typically embodied by one participant posing a request that is fulfilled by one or more participants in the *conversational* interchange.

The fact that such *conversational* interactions will include devices with varying I/O capabilities —ranging from the ubiquitous telephone characterized by speech-only access to personal organizers with limited visual displays— places traditional GUI-based desktop PC clients at a significant disadvantage; the user interface presented by such software maps poorly if at all to the more varied and constrained interaction environments presented by information appliances. Moreover, pervasive clients are more often deployed in mobile environments where hands-free or eyes-free interactions are desirable. Accordingly, conversational computing [4, 3] will become indispensible in the near future. As explained in [4, 5, 3, 6], conversational computing is inherently multi-modal and often expected to be distributed over a network.

Thus, conversational computing also defines an inflection point in personal information processing and is likely to lead to a revolution in all aspects of computing more significant than what was observed in the transition from mainframe based computing to graphical workstations in the mid-1980's.

The ability to access information via a multiplicity of appliances, each designed to suit the user's specific needs and abilities at any given time, necessarily means that these interactions will exploit all available input and output modalities to maximize the band-width of man-machine communication.

3 Design Challenges

This section enumerates the challenges inherent in synchronized multi-modal user interfaces that work across a multiplicity of information appliances.

The key ideas introduced in this paper are:

- Enable user to interact in parallel with the same information source via a multiplicity of appliances and user interfaces;
- Present a unified, synchronized and coordinated view of information across the various appliances.
- Synchronized interaction history across access devices.
- Uniform information access functionality and behavior independent of the device or modality.
- Tight synchronization across multiple parallel modalities.
- Coordination of the user interfaces, behaviors and services.
- Mechanisms to achieve synchronized interaction in a distributed environment.
- Client registration for appliances to announce available services, and on-the-fly content negotiation to enable dynamic, customized content delivery to clients.

4 Multiple Information Appliances

Multiple information applications —running either on the same or multiple devices— can be used simultaneously to gain sequential or parallel information access. Such coordinated, parallel user interaction implies shared application context and history to enable all participating devices to play equally well in the conversation with the user. Mapping this back to the model of a shared, multi-person conversation outlined earlier, this means that all appliances participating in the user interaction need to share a common context that is dynamically updated as user interaction proceeds via one or more devices. We call the device that the user is interacting with at any given instance the *active* device. To extend the analogy with human interaction, we call participants in the conversation that are not presently *active* as *listeners* to reflect the fact that these participants implicitly follow and reflect the information transactions being carried out with the currently *active* device.

The different devices need to provide similar and equivalent functionality while ensuring that the user gets consistent views of the underlying information that is being manipulated. In addition, interaction context and history needs to be synchronized across these devices in order to enable seamless transitions in the user interaction amongst the various devices. Thus, user interaction with a specific device needs to be reflected across all available devices; conversely, each available device needs to be primed to carry on the *conversation* with the user where the previously *active* device leaves off.

Parallel use of multiple access devices implies that transactions are shared across these different devices. In addition, updates to the underlying information via any given device or interface needs to be immediately reflected in all available views of the information.

The prerequisites outlined in this section necessarily postulate a *Model, View, Controller* view of the world, where a single information source *Model* residing on a server (and possibly mirrored on the client) is viewed via different *view-ports* and manipulated via different *controllers*. The significant departure from the traditional MVC paradigm adopted by graphical user interface environments as implemented in SmallTalk [7] is that conversational computing of the future is characterized by multiple controllers that share/negotiate conversational state.

Note that there are no fundamental differences between multiple devices and multiple modalities. Both have to be treated as different views of a same dialog.

5 Coordinated user interfaces, functions and behaviors

Parallel use of coordinated devices will be especially important among pervasive clients. Today, users juggle between cell phones, pagers, PDAs and laptops. Synchronization mechanisms are provided but they merely guarantee that part of the information is shared and kept up to date across the devices.

Spontaneous networking, as proposed by Bluetooth [1] and Jini [2] only guarantee discovery, connection and "remote controllability" of local devices. However, these mechanisms do not address the selection of the appropriate device(s) or the most suitable interface to carry out a transaction; nor do they help determine the device that should react to a given input from the user. They also do not address the issue of determining which device is *active* versus which devices are to be *listeners* during a given stage of the conversation. Note that silent partners give up some or all of their interaction capabilities. In addition, considerations of efficient resource utilization dictate that an intelligent *orchestrator* determine the devices that can *drop-out* during specific portions of the conversation.

In [6], we describe how speech I/O, controls, results and data files can be distributed. The details of the distribution of the other facets of conversational systems over other modalities like visual (GUI) interaction are outlined in this paper —this involves transport and control of the presentation as well as the synchronization information.

Actual selection of the role of each device is decided using registration and dynamic negotiation. The decision can be influenced by the

capability of the networked devices, the requirements of the applications and transactions, the preferences of the application developer, the preferences of the user and the state of the network.

6 Multimodal User Interaction

Given the underlying paradigm of the user participating in a *conversation* with the various available information appliances all of which communicate with a common information backend to manipulate a single synchronized model, multimodal interaction is a logical next step in designing the user interaction. Thus, different participants in the *conversation* —including the user— use the most appropriate modality to communicate with the target of the current portion of the conversation. Notice that when phrased as above, the role of the user and the various devices participating in the conversation is *symmetric* —a user can choose to point or use other visual gestures to interact with a particular device while using spoken commands to direct other portions of the conversation; the conversational interface driving the various devices can equivalently choose to display certain information visually while speaking other aspects of the conversation.

Key aspects of this form of conversational interaction include the ability of the distributed conversational system to use the best possible combination of interface modalities based on the user's current preferences, needs and abilities as well as the application requirements and device capabilities. At the same time, the distributed conversational system is characterized by the ability to dynamically update its choice of modalities based on what the user chooses to do. Thus, upon failure of the user to respond to a spoken prompt, the system might choose to revert to a visual interface –an implicit assumption that the user is in environment where speech interaction is inappropriate –equivalently, a spoken request from the user might cause the conversational network to update its behavior to switch from visual to spoken interaction.

It is important to emphasize the importance of supporting seamless transitions in the user interaction amongst the different modalities available to the user – that it be on one or across multiple devices. When appropriate conversational multi-modal user interface middleware become available, application developers and users will influence what information and under what preferred form is provided and acted upon in each modality. Automatic adaptation of the applications based on this consideration can be available on the server (application adaptation) or on the connected clients (user preferences, browser rendering features).

However, the user interfaces must always support dynamic and often unpredictable dynamic switches across modalities. Indeed, based on the user's activities and environment, the preferred modality may suddenly change. For example, a speech-driven (or speech and GUI) banking transaction will probably become GUI only if other people enter the room. Transactions that the user could not complete in his office are to be completed in voice only or voice only / GUI constrained mode in the car.

7 Presenting Unified Information Views

At any time, the dialog interaction must be in the same state in all the interacting views, that it be different devices or different modalities. The resulting conversational MVC or CMVC, is illustrated in figures 1 and 2.

Presenting unified views is first achieved by adopting the MVC paradigm for the distributed conversational system. Such synchronized views are further facilitated by adopting standardized mechanisms of information interchange amongst the various participants in the conversation and the back-end that is the information repository for the *model* being manipulated. Given wide-spread adoption of XML as an industry standard for information interchange, we postulate that devices participating in distributed conversational systems will use XML-based encodings to encapsulate both the information being transacted as well as the user interaction involved in completing such transactions.

The XML paradigm of a single modality-independent information representation that is filtered and otherwise transformed for delivery to different devices and applications is particularly well-suited for deployment across the conversational network; thus, participating devices will receive and process a single unified information representation to produce modality-specific presentations and interactions. Such transformations will be encapsulated in device-specific and modality-specific XSL stylesheet (or other transformation mechanism) that will be selectively shared or overridden by specific devices and applications to provide specialized behaviors, possibly requested by applications or user preferences. User interaction using a given modality and device will in turn be mapped back to the single universal information representation to be consequently reflected across all participating devices in the conversation.

Besides XML, implementations of dialogs and user interactions can be developed and transmitted imperatively and later appropriately rendered and synchronized through modality-dependent interfaces.

8 Synchronized User Interaction

A further consequence of the decision to embody the distributed conversational system as a collection of *controllers* all of which manipulate the same underlying *model* is to provide synchronized *views*. This synchronization of views is a direct consequence of generating *all* views from a single unified representation that is continuously updated; the single modality-independent representation provides the underpinnings for coordinating the various *views* —see 1.

To see this, consider each *view* as a transformation of the underlying modality-independent representation. Further, the modality-independent XML representation can be viewed as an abstract tree structure that is mapped to modality-specific (and device-specific) presentational tree structures. These transformations provide a natural mapping amongst the various *views* —since any portion of any given view can be mapped back to the generating portion of the underlying modality-independent representation, and this portion consequently mapped back to the corresponding view in a different modality by applying the appropriate transformation rules —see 2.

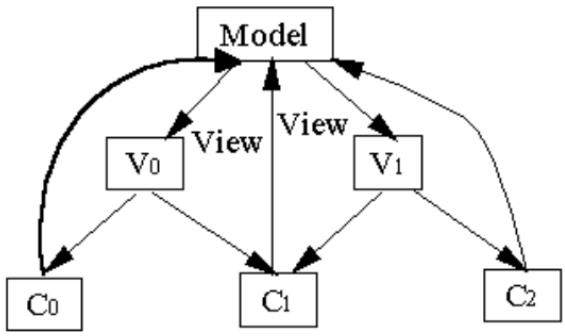

Figure 1: Single *model* mapped to multiple *views*. Multiple *controllers* act on and transform the underlying model by interacting via one or more views.

Multi-modal / conversational user interfaces must follow the CMVC paradigm. More specifically, there must always be a model of the conversation/dialog, independent of the rendering modality, that is the repository of the current dialog state, the dialog flow as currently known by the application and the whole conversation history

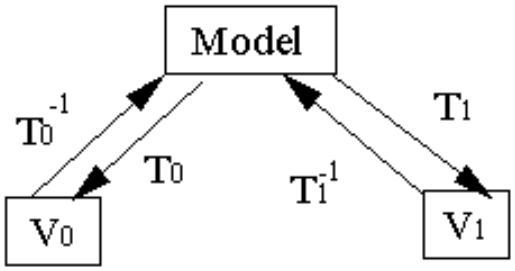

Figure 2: A single *model* is *transformed* to multiple synchronous views. These transformations T_0 and T_1 can be inverted to map specific portions of a given *view* to the underlying model. Composing T_i with T_j^{-1} for appropriate values of i, j enables us to synchronize amongst the various views.

and context. Any user interactions within a modality must act on the conversation model before being reflected on the different views. Any other approach, will result in multi-modal dialog inconsistencies: forms completed in multiple modalities will be rejected as incomplete; page navigation in a modality may not reach the same dialog state in the other. Fixing these inconsistencies, without implementing CMVC, requires overhead with numerous roundtrips to the server, multiple connections between the views or hard synchronization authoring; in the end, these solutions are weak version of CMVC.

9 Distribution of the interfaces

When multiple devices are involved (e.g., a speech browser on the server and a GUI browser on a client device), additional information must be exchanged between these devices:

- The information needed by each device view to render and possibly process each view
- The user interaction events with each view
- The control information to update the different views
- Registration and dynamic negotiation between devices

9.1 I/O management

In order to implement correct synchronization of the different interactions with the user, a mechanism must be provided to manage all the I/O events. In a conversational system, the exact order and nature of the I/O events directly impacts the disambiguation of user input —each I/O event must be appropriately accounted for in the context stack.
A consequence of the need to appropriately track and sort all I/O events is to introduce "clock synchronization" with a primary time server, as proposed in NTP (Network Time Protocol, RFC1119). Also, all events will be at least sent to a root repository before transmission to a controller or before transmission to a view.

9.2 Registration and Dynamic Negotiation

In order to provide coordination between the different views as defined in section 5, we need to provide a protocol which enables each device, view and conversational engine to describe its processing and I/O capabilities. The same protocol is used by each application (dialog or dialog components) to describe its processing and presentation needs. This is also associated with discovery protocols, when applications and devices must be discovered and identified before any registration or negotiation can occur.

Dynamic negotiation is required at multiple stages:

- To negotiate the conversational engines that are activated for each dialog component.
- To negotiate the views and devices associated with a given application.
- To communicate where the information required to view the application can be found, and possibly to communicate that information.
- To disambiguate user's input and appropriately route I/O events.

Methods like view cosmetization, application adaptation and application versioning can also contribute to appropriate dynamic negotiation.

10 Conclusion

In this paper, we have motivated the need for, and discussed the challenges inherent in coordinated, synchronized multi-modal interfaces. Such interfaces are an integral part of conversational computing, that we predict, will become the future prevalent computing paradigm. In future publications, we will discuss implementation options and proposals for the different components that we have introduced.
The introduction of CMVC should be considered as the main message that we want to convey: multi-modal / conversational user interfaces must follow the CMVC paradigm. More specifically, there must always be a model of the conversation/dialog, independent of the rendering modality, that is the repository of the current dialog state, the dialog flow as currently known by the application and the whole conversation history and context. Any user interactions within a modality must act on the conversation model before being reflected on the different views.

11 References

[1] http://www.bluetooth.com/.

[2] http://www.sun.com/jini/.

[3] S. H. Maes. Elements of conversational computing. Submitted to ICSLP 2000.

[4] S. H. Maes. Conversational computing. In *PvCC99*, June 1999.

[5] S. H. Maes. US Serial No 60/102,957, 1998, following applications and CVM proposal. Technical report, IBM Research, Nov. 1997, Oct. 1998 & Oct. 1999.

[6] S. H. Maes, D. Chazan, G. Cohen, R. Hoory, and M. Zibulski. Conversational networking: Conversational protocols for transport, coding and control. Submitted to ICSLP 2000.

[7] Stephen T. Pope and G. Krasner. A cookbook for using the model-view-controller user interface paradigm in smalltalk-80. *Journal of O-O Programming*, 1(3):26–49, 1987.