Component-based multimodal dialog interfaces for mobile knowledge creation

Georg Niklfeld
Telecommunications Research Center Vienna (ftw.)
Maderstr. 1/9
1040 Vienna, Austria
niklfeld@ftw.at

Robert Finan
mobilkom austria AG
Obere Donaustrasse 29
1020 Vienna, Austria
r.finan@mobilkom.at

Michael Pucher
Telecommunications Research Center Vienna (ftw.)
Maderstr. 1/9
1040 Vienna, Austria
pucher@ftw.at

Abstract

This paper addresses two related topics: Firstly, it presents building-blocks for flexible multimodal dialog interfaces based on standardized components (VoiceXML, XML) to indicate that thanks to well-supported standardizations, mobile multimodal interfaces to heterogeneous data sources are becoming ready for mass-market deployment, provided that adequate modularization is respected. Secondly, this is put in the perspective of a discussion of knowledge management in firms, and the paper argues that multimodal dialog systems and the naturalized mobile access to company data they offer will trigger a new knowledge management practice of importance for knowledge-intensive companies.

1 Introduction

Knowledge management is concerned with promoting the creation and dissemination of knowledge in organizations. A variety of technologies, in particular information technologies can support this process. One way in which information technology has contributed to knowledge management is through computer networks, which have provided individuals with easy access to information not stored in their own office. The knowledge worker was now able to remain seated at her desk and share data and information with other individuals seated at desks thousands of kilometers away. Mobile data communication technology brings another improvement, as access to the entry points to the network is no longer confined to the office: networking anytime, anywhere, as the slogan goes. The next step in this process of removing access restrictions to information should target the user interface of the entry points, viz. the information interfaces. For historical reasons their design is still inspired partly by the communication needs of machines, and certainly by the office-bound scenario featuring large computer screens and keyboards. Human beings in movement however use speech for the exchange of information along with visual representations such as text and graphics. To bring such multimodal communication to mobile computer devices is an important issue for naturalizing information access. However, this paper argues that much beyond such principled considerations, multimodal interfaces are imperative for information access from mobile devices. The paper presents building-blocks and constraints for the development of such multimodal interfaces and systems, and describes the emerging knowledge management scenario that companies should understand well in order to be prepared for this new strategic technology.

Section 2 develops the case for multimodal interfaces to mobile data services, e.g. in the approaching 3G telecommunication infrastructures. Section 3 presents our view on what are the building-blocks for a technological infrastructure that will foster ubiquitous multimodal interfaces to large numbers of data services in the business environment. Section 4 is more technically spe-

cific and describes our proposal for a general software architecture that provides for easy implementation of multimodal dialog interfaces. Section 5 discusses the question of how the multimodal interfaces can be mapped to heterogeneous data sources. With the technical picture in place, section 6 reviews the management theory concept of knowledge management practices, to prepare the ground for section 7, which considers whether multimodal interface techniques will trigger new knowledge management practices in companies.

2 The importance of multimodality

Human face-to-face communication is multimodal, combining the acoustic channel with visual and occasionally tactile channels. Human beings are therefore well equipped to communicate multimodally, and multimodal communication is perceived as natural. In human computer interaction, the use of spoken or written natural language poses complexities that have led to a dominance of visual interfaces consisting of text and usually 2D graphics. Visual interfaces are very efficient under some circumstances: written text allows rapid information uptake; visual interfaces on large screens allow to present large amounts of information simultaneously, with users able to focus on bits that interest them without having to process all the rest in-depth; finally, visual interfaces are preferred consistently for input and output of spatial information (Oviatt et al., 1997). In visual interfaces, text output can be used for information for which graphical metaphors are not available, and text input can be used for unrestricted content or inherently linguistic information such as names.

Yet, considering data services on 3G mobile devices, the following factors constitute obstacles for relying solely on visual interfaces, and imply a real usefulness of added speech interface capabilities:

- The terminal devices where 3G data services run will have small displays in terms of size and resolution, although a significant improvement over current WAP phones is expected. Still, it will not be possible to mirror the user experience of today's commercial web-sites in terms of the amount of information that can be presented simultaneously. (Note also that the speed disadvantage of TTS is less problematic for small amounts of information.)

- Although 3G terminals will be produced in a variety of form-factors, many devices will be too small to provide comfortable alphanumeric keyboards, relying instead on pointing devices such as a pen/touch-screen combination. Without a keyboard, text input becomes cumbersome, even when handwriting recognition is provided. Speech input is a logical alternative. The input dilemma of 3G devices is made yet more severe where even a pointing device is lacking, as in current mobile phones. This WAP-like scenario makes data services without speech support so unattractive that we consider it unlikely that large numbers of 3G devices intended for data access will be in this category.

- In mobile usage situations, visual access to the display may be impossible in certain situations, such as when driving a car. Also, even where pointing-device and keyboard are provided, access to them is not possible when a user has her hands busy for other activities. A speech interface may still be usable in such situations.

We contend that the combination of these considerations takes the case for multimodal interfaces for 3G services beyond a nice-to-have status to that of a significant building-block that needs to be put in place for successful deployment scenarios of 3G infrastructures to emerge. This is also the motivation for application oriented research like the one described here, which attempts to identify development models for multimodal interfaces that are feasible both technically and economically.

3 Building-blocks for multimodal dialog interfaces

It is our belief that a successful scenario where multimodal dialog interfaces become commonplace in the business and consumer environments requires that the development of a multimodal interface for an existing data service must be ac-

complishable by software developers without a specialized background in speech or natural language processing. Developers that have such a specialized background will likely be too hard to come by for small run-of-the-mill projects.

In 1999, the EU-sponsored DISC project undertook a comparison of seven toolkits for dialog management in spoken dialog systems (DISC, 1999). The dialog management component is the central part of a dialog system and of particular importance in a discussion about streamlined development processes for multimodal interfaces, because it is the bridge between the intelligent interface technologies and the underlying application logic of application servers and databases. The DISC-survey finds platforms that offer rapid prototyping support, partly via integrated development environments. As most of the toolkits are shipped together with speech recognition and speech synthesis engines, this frees application developers to focus, firstly, on the dialog design, and secondly, on interfacing to the application core.

In May 2000, the VoiceXML Forum, a multiparty industry organization, submitted version 1.0 of the VoiceXML standard to the W3C (W3C, 2000c). VoiceXML is a specification language for spoken dialogs including functional requirements on the platforms that implement the standard. The work done on the VoiceXML standard represents a major step forward from the state of affairs reported in the DISC-study for two reasons. Firstly, being an internet standard, VoiceXML goes beyond manufacturer-specific toolkits that are not compatible with each other by providing a credible set of target interfaces for all players in the industry. Second, it chooses XML and the associated web-programming technology both as a format for specification of the voice dialogs, and for interfacing to the application logic. The two most important aspects for a standardized component framework for dialog systems have thus been brought in line with web technology, which is what was needed to create that promise of a platform for speech-enabled applications that is easily accessible to developers with a general, and not speech processing specific, background.

Today one can add that VoiceXML has received important support from major players in the industry who have made development platforms for VoiceXML-based applications available to developers free of charge. The only potential downside of recent developments is that the platform manufacturers have also included some proprietary elements in their implementations, making direct transfer of VoiceXML applications from one platform to another again impossible.

While an early requirements document for VoiceXML explicitly treats multimodality in the context of the discussion on the inclusion of DTMF input in the standard (W3C, 1999), the standard is not written to cover general multimodal applications. Subsequently, W3C has drafted a requirements document for a multimodal dialog language (W3C, 2000b) and currently a new working group on that topic is being assembled. At present however, the standard provides no intentional support for general multimodal systems. The most important drawback that we found in our attempts to nevertheless build a multimodal architecture *around* VoiceXML, is that it is not possible to make an active voice dialog running in a VoiceXML browser aware of events that occur outside the voice browser, e.g. at a visual interface: VoiceXML neither allows for linking in Java applets that could receive pushed notifications, nor does it provide any other interface for external events. Our architecture for multimodal dialog systems based on VoiceXML is considerably influenced by this fact.

With VoiceXML, a standard for voice components is available which seems ready for the mass-market. What is needed in addition is a simple reference architecture that shows how functionality supported by VoiceXML can be integrated into a system architecture for multimodal dialog interfaces.

4 Architecture

In a project at our institution that follows the longer term goal to develop architectures for multimodal dialog systems for 3G telecommunications infrastructures, for the benefit of our supporting partner companies from the telecommunications industry, we are currently developing an architecture that shall: use mainstream technologies and standards as far as possible, to test their capabilities and limitations; be general enough to

scale from our first small prototypes to larger systems that are close to market-readiness; provide the basis for usability research on multimodal interfaces.

The architecture shall support a multimodal interface that shall combine a visual interface via HTML and Java applets in a visual web browser with a voice interface built using VoiceXML. Communication between the visual browser and the voice browser is mediated via a central application server that is built using Java servlet technology in combination with a web server.

In (W3C, 2000b), three types of multimodal input are distinguished:

1. *sequential multimodal input* is the simplest type, where at each step of the interaction, either one or the other input modality is active, but never more than one simultaneously;

2. *uncoordinated, simultaneous multimodal input* allows concurrent activation of more than one modality. However, should the user provide input on more than one modality, these informations are not integrated but will be processed in isolation, in random order;

3. *coordinated, simultaneous multimodal input* exploits multimodality to the full, providing for the integration of complementary input signals from different modalities into joint events, based on timestamping.

Because we cannot send events about the visual interface to the dialogs in the voice browser (cf. section 3), we maintain that only the *sequential multimodal input* pattern can be properly realized with the current version of the VoiceXML standard: The other patterns require that even while the voice interface is active, e.g. listening for user speech, it must be possible for the multimodal dialog to change state based on inputs received from the visual interface. In the sequential mode on the other hand, it is possible to deactivate the visual interface whenever voice input is activated.

In this case then, the choice of multimodal interaction pattern is determined by features of the components used. In many realistic application development efforts, the interaction pattern will be determined by user-level requirements that have to be met. Anyhow, the choice of multimodal interaction pattern will certainly be a dimension in which variation occurs. For the purposes of the demonstrator development in our project, it was important to find a software architecture that can remain stable across different patterns and across interface types.

This can be accomplished by a modular or object-oriented design which separates the central application server into the following functions:

visual communicator: a modular handler for the visual interface;

voice communicator: a modular handler for the voice interface;

transaction module: encapsulates the transaction needed for the application logic;

multimodal integrator: handles all message flows between the interface modules, and between the interface modules and the transaction module.

The resulting system architecture is shown in Fig. 1.

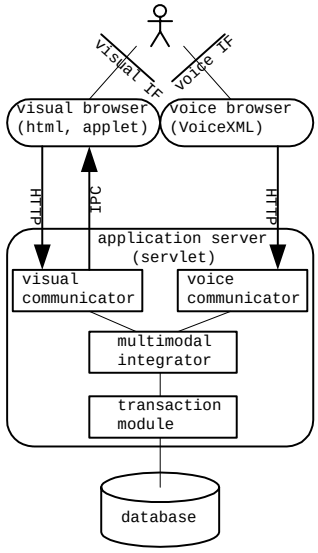

Figure 1: *Multimodal Architecture*

Both the visual and the voice user interface are realized by standardized components in line with general web technologies (visual web browser and VoiceXML browser). Within the application server, the architecture stipulates a specialized interface handler for each modality.

For example in our prototype, the *voice communicator* prepares VoiceXML documents which it puts on a web server that is associated to the application server. Once the VoiceXML interpreter of the voice browser has been started (after a user-triggered event, e.g. an incoming phone call), each of the VoiceXML documents processed by the interpreter is programmed so as to terminate with a load of a successor document from the web server. Our *voice communicator* simply prepares these successor documents based on messages it receives from the multimodal integrator. When the voice interface is not active, the prepared documents contain just an idle loop that terminates after some time, e.g. 0.5 seconds. When the *multimodal integrator* decides (based on user input) that a chunk of the interaction shall be performed via voice, it sends a message indicating field labels and data-types (e.g. the range of an enumeration) to the *voice communicator*, which instead of an idle VoiceXML document now produces a VoiceXML dialog document that executes the respective voice dialog with the user and returns results to the *multimodal integrator*.

The *visual communicator* is designed similarly to prepare HTML pages and applets for the visual browser. In our prototype, the visual interface includes controls that allow the user to explicitly activate the voice interface for a group of input fields. When this is done, the *visual communicator* deactivates all control elements on the visual interface, and sends a message with the user request for voice dialog to the *multimodal integrator*.

The *multimodal integrator* is the only part in the proposed architecture where information from more than one modality is processed. The way this processing is done then defines the multimodal interaction pattern. To change the pattern, the architecture envisages that one implementation of the *multimodal integrator* would simply be replaced by another, without any changes to other parts of the systems. This of course presupposes that the interfaces between the *multimodal integrator* and the interface handlers have been been defined so generally that all occurring multimodal interaction patterns are covered. A description of this interface is for further study.

In our view, these preliminary results show that VoiceXML can be used to build simple, but nevertheless useful multimodal interfaces for typical data services for mobile access. Once first implementations of the *voice communicator* and the *multimodal integrator* are available, it should become quite easy for the general web programmer to generate further multimodal interfaces for existing data services.

A simple working example for a multimodal interface (not in the domain of knowledge management) that we have already implemented is an interface to existing route-finder applications in the web for the city of Vienna. It would be hard to enter street names without a keyboard, so we provide the choice to use speech for any of the input fields, but results of route-queries are inherently graphical, and therefore are displayed visually. We have developed the multimodal interface without access to the route-finder applications themselves, but just by replicating the CGI requests sent to the applications from their provided visual browser interfaces, which demonstrates the potential of a modular architecture for easy multimodal interface development.

One drawback of the modularity of the VoiceXML standard itself is the resulting lack of support for some types of interface adaptivity. Information about environment characteristics that is easily obtainable in speech recognition, such as the level of ambiance noise and confidence scores in speech recognition, are kept local in the speech recognition component used by the VoiceXML browser (as part of the *implementation platform*, which is not further considered by the standard). They are not accessible in the VoiceXML browser, and therefore neither in the *voice communicator*, nor in the *multimodal integrator*, where they could be used to influence modality selection. This shortcoming should be addressed in future versions of VoiceXML.

5 Mapping data to speech using XML

To further illustrate the web-oriented technology framework for multimodal dialog interfaces that we advocate, this section briefly reviews the use of the XML family of technologies for the mapping of data to representations at a speech interface, although we do not have much new to say on this.

XML (W3C, 2000a) is a markup language that can be used to represent data and data schemata in text files. Processors for XML-files are available in common web-browsers and servers. There exist also associated transformation languages (e.g. XSLT) and style sheet languages (e.g. XSL) that make it possible to associate presentation formats to data that follows a defined schema. Note that VoiceXML itself is just one XML-schema, with defined semantics.

XML technologies are widely used to represent data that comes from databases or applications in a format that is universally interpretable in web infrastructures. A possible implementation of the *voice communicator* of our architecture would be an XML-based translator that takes general XML-representations and produces VoiceXML dialogs that play or query these data. Such an approach would reduce the development of a multimodal dialog interface to selection of a multimodal interaction pattern and provision of the respective *multimodal integrator*. It has to be said that automatic generation of good voice interfaces from arbitrary data is certainly not easy. However, if the complexity of each of the voice dialogs is kept low (in multimodal dialogs this is easier than in voice-only dialogs), the goal appears within reach.

6 Knowledge Management Practices

Within management theory literature, discussions of knowledge management have often focused on the discussion of different types of knowledge, such as 'tacit' vs. 'explicit' knowledge, and the possibilities to convert between the types in order to maximize usefulness to organizations (Polanyi, 1958; Nonaka, 1994). When the knowledge management of individual organizations is studied, it is however difficult to identify and classify the knowledge held by an organization.

This among others is the motivation for the proposal to study knowledge management practices (KMPs) instead (Coombs and Hull, 1998b; Coombs and Hull, 1998a). KMPs in a firm are regular activities by which knowledge is processed in some way, and which play an important role in shaping the knowledge base of the firm and making it available in the innovation process. They are distinctive ways in which a firm deals with knowledge, and can be established by questionnaire or interview studies with employees.

Coombs and Hull (1998a) distinguish five groups of KMPs: KMPs located in the formal R&D management process; KMPs for managing intellectual property positions; KMPs for 'mapping' knowledge relationships; KMPs for serial transfer of project experience; and KMPs contingent on information technology applications. In the latter group the authors distinguish KMPs that are supported by IT, but have a strong independent existence (e.g. electronic patent watch bulletins) from others that are triggered by innovations in IT, as is the case with across-site cluster building among research staff that is triggered by the ease of communication using email or intranet solutions.

The question that arises is whether mobile access to information is a new, IT-triggered KMP. This requires that such activities, presumably performed by employees that perform a significant part of their work off-site, plays a significant rather than just anecdotal role in shaping the knowledge base of firms and their innovative potential. Here are some thoughts on this topic:

- The mobile information access of today is mainly notebook-based. Employees access their company email while traveling and read company news on the intranet via secure internet connections. This is essentially a quantitative improvement of intra-company networking.

- Field engineers can feed data observed at a customer installation to corporate application servers and use results immediately to make changes to the customer installation. This is a quantitative improvement of response times.

- Professional mobile speech communication is a ubiquitous phenomenon, which makes groups whose individual members are highly mobile more cohesive. A minor qualification here could be that topics of mobile speech calls are usually administrative and do not touch the core of the firms' innovative processes.

It seems that mobile access intensifies data and information use and the associated creation and

sharing of knowledge, but that qualitatively new practices are restricted to the IT departments that have to plan and manage the supporting applications. These activities may be significant enough to constitute new KMPs. The next section considers the same issue for mobile multimodal interfaces.

7 Outlook on mobile multimodal interfaces

We assume that by 2004, business users will have smart phones for 3G telecommunication networks that will include speech input/output capabilities like on a telephone, and also colour displays of approx. 6x6cm size and 320x320 dots resolution, with touch-screen functionality. Most users will not have a comfortable keyboard for their device when mobile, although some handwriting recognition technology will be available.

We imagine that the following use cases are realistic:[1]

- At a working lunch, a researcher discusses work with a colleague. He is reminded of a relevant paper, but cannot recall the authors. Handling knife, fork and 3G device in turns, he queries the library database of his institution, entering keywords for the search by voice and browsing through the results list using sometimes voice, sometimes the touch-screen.

- Driving back to the airport on the highway after a meeting, a sales manager starts working on a report, alternately dictating notes and voice-navigating through web-pages, and checking back on data on internal-access pages of his company. He checks the screen of his smart-phone occasionally for a quick glance to check progress.

- During the train ride home from work, a project manager has been working on a resource plan on her notebook, which has a mobile internet connection. When she has to get off and walk over to the bus stop, the plan is nearly finished, so she decides to continue work using the multimodal interface to the corporate project planning application on her smart phone. She uploads the project plan to the multimodal interface server of her company and packs up the notebook. She then connects to the multimodal interface server from her smart phone and enters the remaining commands via voice while walking, and via the visual interface once she has entered the bus and found a seat (being reluctant to use voice control in crowded places).

What these imaginary examples demonstrate are firstly, new situations in which work which requires access to data and information resources is possible (intensifying knowledge creation), and secondly, a reliance on multimodal interfaces for tasks that probably would be performed via purely visual interfaces in more office-like situations. This suggests a need for the general component-based multimodal interface development model presented earlier, to enable the IT department of e.g. the firm in the last example to tailor multimodal interfaces for each of the applications offered on the company network.

Accordingly, we see a new KMP for multimodal interfaces within the IT management function of firms. This new KMP is concerned with supporting knowledge creation through the planning and provision of multimodal interfaces to company data and information resources. It requires decision-making on the following questions:

1. What types of mobile access to company resources should be supported via telephone speech, and what types via data services?

2. Of the data services, what services should be supported at notebooks, and what services at visual-only interfaces, speech-only interfaces, and multimodal dialog interfaces at smart phones, respectively?

3. Which multimodal interfaces can be developed in-house, preferably using component-based methods, and what interfaces need to be procured externally?

Companies for which knowledge creation by mobile employees is strategically important should consider these questions carefully.

[1] Although further progress in ASR in noisy environments is a precondition. This is an important challenge for mobile voice interfaces in general.

8 Conclusions

Using VoiceXML and XML technologies, the development of simple multimodal interfaces to company applications is today possible without involvement of significant speech and natural language processing expertise, by following architecture models like the one proposed in this paper. It is important for knowledge-intensive companies to define a stance on how to deal with this technology, because given the parameters of mobile data access, multimodal interfaces and the resulting improvements to mobile data and information access, which foster knowledge creation, will likely become competitively relevant.

Future work of our group will elaborate the proposed reference architecture, by providing prototype implementations of the *visual communicator*, *voice communicator*, and *multimodal integrator* for different multimodal interaction patterns, and by describing general interfaces between them.

The standardization work for a multimodal dialog language to supersede VoiceXML will bring further improvements to the development models for multimodal interfaces.

Acknowledgements

This work was supported within the Austrian competence center program **K***plus*, and by the companies Alcatel, Connect Austria, Kapsch, Mobilkom Austria, and Nokia.

References

Coombs, R. and R. Hull. 1998. 'Knowledge Management Practices' and path-dependency in innovation. *Research Policy*, 27(3):237–253.

Coombs, R. and R. Hull. 1998. Knowledge management practices for innovation: an audit tool for improvement. Technical report, CRIC, Univ. Manchester.

DISC. 1999. Deliverable d2.7a: State-of-the-art survey of dialogue management tools. Technical report, Esprit Long-Term Research Concerted Action No. 24823.

Nonaka, I. 1994. A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1):14–17.

Oviatt, S.L., A. DeAngeli, and K. Kuhn. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proc. of CHI97*, pages 415–422, New York. ACM Press.

Polanyi, M. 1958. *Personal Knowledge*. Univ. Chicago Press, Chicago.

W3C. 1999. Dialog requirements for voice markup languages. Working draft 23 dec 1999. http://www.w3.org/TR/voice-dialog-reqs/.

W3C. 2000a. Extensible Markup Language (XML) 1.0 (Second Edition). Recommendation 6 oct 2000. http://www.w3.org/TR/2000/REC-xml-20001006.

W3C. 2000b. Multimodal requirements for voice markup languages. Working draft 10 jul 2000. http://www.w3.org/TR/multimodal-reqs.

W3C. 2000c. Voice eXtensible Markup Language (VoiceXML) version 1.0. http://www.w3.org/TR/2000/NOTE-voicexml-20000505/.