Mobile multi-modal data services for GPRS phones and beyond

Georg Niklfeld
ftw. Telecommunications Research Center Vienna
Donau City Str 1
1220 Vienna, Austria
niklfeld@ftw.at

Michael Pucher
ftw. Telecommunications Research Center Vienna
Donau City Str 1
1220 Vienna, Austria
pucher@ftw.at

Robert Finan
Mobilkom Austria AG & Co KG
Obere Donaustr. 29
1020 Vienna, Austria
r.finan@mobilkom.at

Wolfgang Eckhart
Sonorys Technology GmbH
Industriestr. 1
2100 Korneuburg, Austria
wolfgang.eckhart@sonorys.at

Abstract

The paper discusses means to build multi-modal data services in existing GPRS infrastructures, and it puts the proposed simple solutions into the perspective of technological possibilities that will become available in public mobile communications networks over the next few years along the progression path from 2G/GSM systems, through GPRS, to 3G systems like UMTS, or equivalently to 802.11 networks. Three demonstrators are presented, which were developed by the authors in an application-oriented research project co-financed by telecommunications companies. The first two, push-to-talk address entry for a route-finder, and an open-microphone map-content navigator, simulate a UMTS or WLAN scenario. The third demonstrator implements a multi-modal map finder in a live public GPRS network using WAP-Push. Some indications on usability are given. The paper argues for the importance of open, standards-based architectures that will spur attractive multi-modal services for the short term, as the current economic difficulties in the telecommunications industry put support for long term research into more advanced forms of multi-modality in question.

1. Introduction

Multi-modal interfaces combining speech recognition and keypad touch-screen input have the potential to alleviate the input bottleneck in mobile data services for 2.5G GPRS and 3G (in Europe, this means UMTS)–or equivalently to the latter, 802.11 W(ireless)LAN networks. Yet there exists so far no easily discernible road-map regarding how and when these interfaces will be ready for real services in public telecommunications networks. At the same time, due to the recent dramatic changes of the economic environment in telecommunications, many telecom firms are under pressure to search for models to generate new revenues from data services in GPRS and 3G/UMTS networks, which they need to do in order to recuperate the large capital expenditures for GPRS and particularly 3G/UMTS. As they struggle to survive the next three to five years, many firms restrict their financial support to technologies that promise to create revenues in such a short time-frame. They are interested in convincing applications, not mere potential interface capabilities. In this paper we present three demonstrators that we have built in order to explore various possible building blocks for multi-modal data services, and in order to learn what can and what cannot be done in existing infrastructures. The demonstrators are: (1) alternate mode, push-to-talk address entry for a route finder for a UMTS scenario; (2) an open-microphone map content navigator for UMTS; and (3) a voice/WAP-Push demonstrator for GPRS. We also discuss the range of applications and interface types that should be implemented first; give a short overview of architectures, standardisation work, and classes of terminals; and propose a road-map for various activities leading up to successful multi-modal data services in real-world networks.

2. Application considerations

In the current difficult economic environment of the telecommunications industry, many companies strongly support innovation in data service technology where the innovations are preceived to translate directly into a value-generation opportunity, whereas nice-to-have innovation that is risky or relies on too many unproven assumptions about user acceptance and device capabilities is often put on hold. For new multi-modal data services this means that firstly, every deviation in the user interface from familiar display-only paradigms should be justifiable by a significant advantage that is either obvious or can at least be demonstrated in a usability study; and secondly terminal requirements should be kept low, in particular no capabilities beyond what is already available in a target class of widely used consumer devices should be stipulated. Therefore, in line with the ongoing standardisation efforts, for our work we assume voice I/O capabilities, keypads or touch-screens, and small text or graphics displays for a first generation of network-ready multi-modal interfaces. We do not consider video input.

From the point of view of a network operator, interface technologies alone will not generate revenue, thus it is necessary to present multi-modal interfaces as part of powerful applications. Some applications that are used in multi-modality research projects do not scale to telephony-oriented mobile terminals because they rely on rich graphic information that will not fit on most mobile displays - e.g. military resource planning or animated communication agents. During the specification phase for our demonstrators we have learnt that elaborate forms of multi-modal input integration (co-ordinated, concurrent input) can only bring substantial benefits when the visual interface supports information-rich input events, such as gestures that designate arbitrary regions on the display. We expect that a large number of multi-modal interfaces to mobile data services will operate without such advanced forms of input, but will exploit only the simpler types of multi-modal input integration (cf. the three types of multi-modal integration in [16]). Applications that seem favourable for multi-modal telecommunications services combine visual information such as maps with speech recognition of proper names from a closed set (street names, telephone registers, product catalogues) [2]. Some examples of suitable types of applications are described in [6, 13].

3. Technologies and mobile terminals

3.1. Architectures

Architectures that have been proposed for multi-modal systems [8] can be distinguished

- by the way processing (modality-specific recognisers and renderers, modality fusion and fission) is distributed between terminal and server.
- by the layering. In most recognition systems, late (semantic) fusion is applied rather than early feature fusion [11]. This enables the use of modality-specific recognition modules and implies the existence of at least an additional module for modality integration. More articulated layerings propose up to four layers from multi-modal to modality-specific processing [3].
- by the communication model among different system modules. A number of research projects have developed flexible communication models for component interfaces [15, 1, 7].

3.2. Standards

The VoiceXML standard [17, 19] promotes a development model for spoken language applications that is similar to web programming. Despite its benefits, which include the availability of developments tools, VoiceXML is not well suited for implementing interfaces that support co-ordinated, concurrent multi-modal input, due to the lack of a notification model through which an ongoing voice dialog could be informed about external events. Once a dialog specified in a VoiceXML document is executing, it operates independent of any events occurring in other modalities such as a GUI.

Nevertheless, until recently no comparable standard has existed for an easily accessible development model for voice interfaces, and therefore we chose VoiceXML for building three simple multi-modal demonstrators (cf. section 4). Current standardisation efforts are: (1) The W3C consortium has announced plans for a multi-modal markup language since 2000, but there is little public documentation of progress [18]. (2) In July 2002 the SALT initiative for a multi-modal markup language [14] released a version 1.0 specification. SALT has substantial support from industry and aims to tie development of multi-modal applications closely to visual development using HTML, and to bridge different device characteristics ranging from PC notebooks to smart-phones and even audio-only devices. (3) The ETSI AURORA initiative for Distributed Speech Recognition (DSR, [4]) has also produced proposals for integrating DSR with UMTS for multi-modal applications [12].

3.3. Mobile terminal capabilities

In table 3.3 we give a classification of some mobile devices according to their wireless connectivity (by rows), and

to the best GUI they offer (by columns). We also mention representative products.

Radio link	GUI: SMS	WAP	smart-phone	Quarter-VGA	Standard-size VGA
GSM	Nokia 8210	most recent GSM phones	SonyEricsson R380, Nokia 9210	PDA+GSM-Pocket PC, Handspring Treo	Notebook HSCSD/ GSM card
GPRS		SonyEricsson T68	Nokia 7650, SonyEricsson P800	PDA+GPRS Pocket PC	Notebook GPRS card
WLAN UMTS				PDA WLAN card	Notebook WLAN

Table 1. Mobile device types for multi-modality

Some implications of specific features along both the ”Radio link” and the ”GUI” axis for multi-modal interfaces including speech are: (1) Existing terminals to the left of the grayed columns for GUI-type (Quarter-VGA and VGA) provide insufficient computing resources for continuous speech recognition on the device, although some PDAs can support isolated-word vocabularies of a few hundred entries. Therefore speech recognition solutions for the devices in the white columns require either server-based speech recognition or DSR. (2) Only GPRS and UMTS/WLAN terminals (bottom two rows) provide sufficiently fast switching from voice to data connections to make multi-modal interfaces that combine voice and GUI feasible. This in turn is a needed capability for all devices that do not provide a software environment that would allow for the transfer of speech over a data link, using VoIP, e.g. DSR. (3) The darker two shades of gray in the table indicate the device types that are in our opinion the most promising for multi-modal service deployments in the short term. We refer to the devices that support GPRS, but not UMTS or WLAN connectivity, as the ”stage 1” target group of devices. Some of these devices (e.g. the SonyEricsson T68 phone) are already present in the high-volume mobile phone market. (4) Only UMTS/WLAN terminals can provide concurrent voice and data connections to enable server-side fusion of co-ordinated multi-modal inputs, and thus advanced forms of multi-modality. This group of devices is marked in the darkest shade in the table, we refer to it as the ”stage 2” target group of devices. These devices are not yet used in large numbers. (5) Large-size VGA displays (rightmost column) are used on bigger devices that effectively remove the comparative input bottleneck of mobile devices, reducing the improvement potential of multi-modality.

4. Three multi-modal demonstrators

4.1. Push-to-Talk address entry for route-finder

Map-oriented data services call for visual displays. In one familiar type of service a map-area around a specified address is shown, or a route between two specified addresses (one address may be supplied automatically through handset localisation). In text input of the addresses, relatively long strings of characters occur, while terminal-based predictive typing aids will work less effectively for proper names. For this demonstrator, we assumed a small mobile device with inconvenient text input, but with a touch-screen and the possibility of a voice call concurrent to a data connection, as in UMTS. This scenario was simulated on a notebook PC in a WLAN. To make the input of the addresses easier, we provided push-to-talk buttons next to each of the visual form-fields and blocks of fields for text entry of addresses, slightly different from the ”Tap and Talk” metaphor in MIPAD [5], see figure 1.

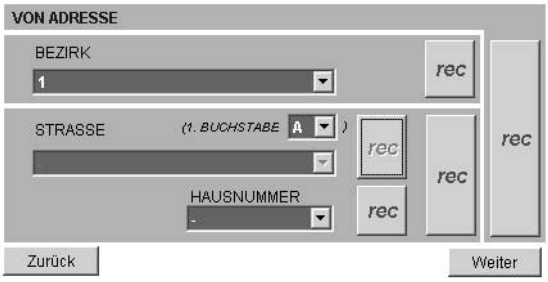

Figure 1. Push-to-talk address entry with simultaneous voice and data connections

Speech recognition (implemented using a VoiceXML server) is active only after the user presses the push-to-talk button, until an input is understood and presented at the display. During this time, the GUI is inactive. Multi-modality is therefore only sequential. All feedback is given on the display to save time. Users can choose freely among text or speech input for each field depending on preferences or situation. The VoiceXML approach lent itself naturally to a simple re-use of an existing route finder web service for the city of Vienna by attaching to the CGI-interface. Initial results from user tests with 12 subjects show a preference for having both speech and GUI input available as options, compared with an interface that would provide only one of the two throughout.

4.2. Open-microphone map content navigator

The objective in our second demonstrator was to test how far the VoiceXML approach could be taken in terms of more sophisticated multi-modal input integration. The user selects a map area, which is presented on the display. He/she can activate location marks on the display using a pointing device such as a pen on a touch-screen (in the demonstrator, mouse-pointing on a notebook PC was used for simulation).

Figure 2. GUI-only map selection in the QuickMap WAP page: in the example, at least 60 key-presses are required before the map is displayed, including 44 key-presses just for entering the lengthy street name "Abraham-a-Sancta-Clara-G" (compare figure 3).

Each mark remains active for a short time, which is indicated by visual cues. A speech recognition channel is open permanently. The user can give speech or GUI commands for map navigation such as zooming or centering, or request to display content such as the nearest shops or facilities in a number of categories. Some commands take variable numbers of location marks as arguments. For example "route" will display the shortest route between two active marks, or between the current centre of the display and the active mark if there is only one in the current time-window. When a particular type of content has been selected for display, additional commands become available, such as the program at a cinema, or opening hours of a shop. While this demonstrator implements concurrent multi-modality to some extent, it puts more demands on the speech recognition than would be necessary if VoiceXML provided a model for notification about external events that could provide context for the dialogue [10]. An implementation of the demonstrator was completed with most of the described functionality.

4.3. QuickMap: voice input, GPRS, WAP-push

To see what can be implemented in infrastructures now ("stage 1"), this demonstrator puts the voice address entry for a route finder on a GPRS phone in a live public network in Austria. To use GUI entry, users can browse directly to a WAP 1.2.1 [20] page that presents a visual form (see figure 2).

To use voice entry, users call a phone number and are engaged in a voice dialog (built using VoiceXML) in which they are asked for the required form fields: city district, street name, and street number. DTMF is also available as an option during the voice dialog, to enable safe entry of the numeric values for district and street number using the mobile phone keypad. For most street names however, voice entry is much more convenient than text entry. A long street name such as "Abraham-a-Sancta-Clara-Gasse" is recognised by the speech recogniser with very high confidence, but requires 44 key-presses on a typical mobile phone keypad, even though the used map server accepts "G" as an abbreviation for "Gasse" (small street). We have found that users nearly always makes several mistakes in the typing process, where each correction requires further key-presses. On the SonyEricsson T68 device used for our tests, when entering the text-input field the user must also first switch off the T9 predictive-typing aid, which would not work for unknown names. This requires another three key-presses (in addition to good command of the mobile phone's features). It should be noted that for some short street names such as "Hegelgasse" speech recognition does not work well and therefore users may prefer text entry. To accommodate this, the demonstrator provides the possibility to switch the entry method in both directions: the WAP page provides a link to "Spracheingabe" (speech input), which triggers the phone to set up a voice call to the dialog system (using the WTAI-protocol [22] that is part of the WAP 1.2.1 suite of protocols). Conversely, the user can request a "Textformular" (text form) in the voice dialog, which will trigger a WAP-Push Service Indication message [21] being sent to the device with a link to the QuickMap WAP page. If voice-input is used and an address is recognised, the system sends a WAP-Push message with a link to the required map to the user's phone. If a user has permitted WAP-Push on his/her device, incoming WAP-Push messages seize control of the display without further user intervention. On the T68 phone, the user can then select to proceed and load the document referenced by the link in the WAP-Push message. (cf. figure 3).

We believe that in a commercial implementation, charging for the service invocation (beyond the connection and data transfer fees) should occur when the user follows the link to the map, after he/she has verified that the address was understood correctly by the speech recognition.

A number of unsystematic tests on a live public network have shown some variation in the times various steps of the application take (cf. table 2). The complete voice dialogs (including confirmation of the recognition results by the user) take about half as long as GUI-only input, for short street street names like "Maderstr(asse)". In our tests, the delivery of a WAP-Push message usually took about 15 seconds, in good cases only c. 3-5 seconds. However, there were outliers when delivery of a WAP-Push message took

Figure 3. Multi-modal or voice-only address entry in the QuickMap demonstrator. The user can initiate the voice call to the dialog system either directly or through the WAP page. DTMF can be used during the dialog for safe entry of numeric form-field values (district and street number, third picture). When the voice-form is complete, the application server requests the network to send a WAP-Push message to the mobile phone. The WAP-Push message seizes control of the display (fourth picture). After selecting Proceed, the T68 phone asks the user whether to load the document linked in the WAP-Push message through a GPRS connection.

Action	GUI input	Voice input
GPRS connection setup	3-5 (GSM: 15-20)	
Voice call setup		5-10
Address input	typically 60	typically 30
WAP-Push delivery		min. 3-5, typically 15, max. 300
GPRS connection setup		3-5 (GSM: 15-20)
Loading of map	max. 10	

Table 2. Rough observed timings for the QuickMap demonstrator in seconds

up to approximately 5 minutes. Setup of a GPRS data connection usually takes 3-5 seconds (compared to 15-20 for setup of a GSM data connection). Loading the WAP-page with the map image over an established GPRS or GSM connection always took less than 10 seconds.

WAP-Push messages are delivered as SMS messages to current mobile phones, therefore there are no timing guarantees. Because of this, applications should be designed to use WAP-Push sparingly and only where timing is not critical. Waiting for a final result, such as the map in our demonstrator, may be an acceptable example. An alternative way to transport the WAP-Push messages would be by IP-Push, which can currently be supported by some PDAs running a special push client software.

5. Alternative way of WAP-voice integration

As the timing results with the QuickMap demonstrator show, the WAP-Push implementation for current GPRS phones does not provide a reliable means for smooth switching from voice calls to WAP sessions. We have therefore started to work with another way of achieving that integration. This approach is based on URL-rewriting and dynamic generation of all WAP- and VoiceXML-pages, and tight management of user session states at a server that performs multi-modal integration. To switch from voice to WAP, users terminate their voice-call (during which the server can identify them at least by their calling party information), and re-establish a GPRS connection to their last visited WAP-page (which includes user identification in the dynamically generated link URLs). The server uses centrally stored user state information together with the parameter values transmitted in the HTTP requests to identify the appropriate next content. The time for the switch from voice to WAP is then the sum of GPRS connection setup, user selection of the continuation link, and loading of the successor page, typically totalling around 20 seconds.

6. Discussion

To promote the cause of multi-modal data services in public mobile telecommunication networks, it is necessary to synthesise (1) the work on standards and architectures, (2) the insights from multi-modality research, and (3) the constantly changing landscape of commercially available mobile terminals onto a single axis of technical development and activities, as we have tried to do in the work described here. Different roles can be fulfilled in this complex scenario by various actors ranging from academically oriented long-term research projects to private software companies. As an application-oriented research project co-financed by the telecommunication industry, we see our

own role between these poles and try to focus on innovative forms of multi-modality that can be implemented in public networks over the next one to three years, as the mass-market devices transit from "stage 1" to "stage 2". But it seems to us that multi-modal data services will still not become substantial revenue producers in telecommunications unless and until more appealing applications and services are developed. This is a challenge for independent software developers, but the R&D and technology providers must contribute by providing APIs, SDKs, demonstrators and usability analyses that inspire the creativity of the application developers and let them concentrate on the content they provide rather than on the technological infrastructure they use.

The three described demonstrators show that already simple forms of multi-modality can bring usability advantages on specific tasks. In particular the final QuickMap demonstrator can be implemented in many existing GSM/GPRS networks, and can be used with popular mobile phones.

Acknowledgments

This work was supported within the Austrian competence center program **K***plus*, and by the companies Alcatel, Connect Austria, Kapsch, Mobilkom Austria, Nokia, and Siemens. The authors would also like to thank Szabolcs Hodossy for technical advice.

References

[1] A. Cheyer and D. Martin. The Open Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1/2):143–148, 2001.

[2] P. Cohen and S. Oviatt. The role of voice input for human-machine communication. In *Proc. of the National Academy of Sciences*, volume 92, pages 9921–9927, 1995.

[3] C. Elting and G. Michelitsch. A multimodal presentation planner for a home entertainment environment. In *Workshop on Perceptive User Interfaces*. ACM Digital Library, November 2001. ISBN 1-58113-448-7.

[4] ETSI. Speech processing, transmission and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithms, 2000. RES/STQ-00018.

[5] X. Huang, A. Acero, C. Chelba, L. Deng, D. Duchene, J. Goodman, H. Hon, D. Jacoby, L. Jiang, R. Loynd, M. Mahajan, P. Mau, S. Meredith, S. Mughal, S. Neto, M. Plumpe, K. Wang, and Y. Wang. MIPAD: A next generation PDA prototype. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, Beijing, 2000.

[6] ISCA. *Proc. of ISCA Tutorial and Research Workshop - Multi-modal Dialog in Mobile Environments*, Irsee, Germany, June 2002.

[7] S. Kumar, P. Cohen, and H. Levesque. The Adaptive Agent Architecture: Achieving fault-tolerance using persistent broker teams. In *Proc. of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000)*, pages 159–166. IEEE, 2000.

[8] M. T. Maybury. Multimodal systems, resources, and evaluation. In *Proc. of LREC02*, volume III, pages g–n, 2002.

[9] G. Niklfeld, R. Finan, and M. Pucher. Architecture for adaptive multimodal dialog systems based on VoiceXML. In *Proc. of Eurospeech2001*, Aalborg, DK, 2001.

[10] G. Niklfeld, R. Finan, and M. Pucher. Multimodal interface architecture for mobile data services. In *Proc. of TCMC2001 workshop on Wearable Computing*, Graz, 2001.

[11] S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and pen-based gesture applications: State-of-the-art systems and future research directions. *Human Computer Interaction*, 2000.

[12] D. Pearce and D. Kopp. ETSI STQ Aurora presentation to 3GPP. Slide presentation, July 2001.

[13] G. Pospischil, M. Umlauft, and E. Michlmayr. Designing Lol@, a mobile tourist guide for UMTS. In *Proc. of MobileHCI02*, 2002.

[14] SALT Forum. http://www.saltforum.org, 2002.

[15] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue. Galaxy-II: a reference architecture for conversational system development. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998.

[16] W3C. Multimodal requirements for voice markup languages W3C working draft 10 july 2000. http://www.w3.org/TR/multimodal-reqs, 2000.

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

[18] W3C. Multimodal interaction activity. http://www.w3.org/2002/mmi, 2002.

[19] W3C. Voice eXtensible Markup Language (VoiceXML) version 2.0. http://www.w3.org/ TR/voicexml20/, 2002.

[20] WAP Forum. WAP-100, Wireless Application Protocol Architecture Specification . http://www1.wapforum.org/tech/terms.asp?doc=WAP-100-WAPArch-19980430-a.pdf, 2000.

[21] WAP Forum. WAP 165, Push Architectural Overview. http://www1.wapforum.org/ tech/terms.asp?doc=WAP-165-PushArchOverview-19991108-a.pdf, 2000.

[22] WAP Forum. WAP-170, Wireless Telephony Application Interface Specification . http://www1.wapforum.org/tech/terms.asp?doc=WAP-170-WTAI-20000707-a.pdf, 2000.