

Multimodal Interactions with an Instrumented Shelf

Rainer Wasinger¹, Michael Schneider¹, Jörg Baus², Antonio Krüger²

¹ DFKI, GmbH,
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
{rainer.wasinger, michael.schneider}@dfki.de

² Saarland University
P.O.Box 15 11 50
66041 Saarbrücken, Germany
{baus, krueger}@cs.uni-sb.de

Abstract. In this paper, we describe the initial implementation of our application demonstrator called ‘ShopAssist’. This application will aid users in product queries within a shopping scenario context. We describe the wide range of input modalities that our application supports such as speech, handwriting, intra- and extra gestures, and the mixed modality combinations that promote advanced user interaction with real-world and virtual objects.

1 Introduction

Users are no longer limited to the “desktop computing” paradigm. Applications are now mobile and ubiquitous, and may span multiple and changing contexts. Depending on a user’s current state, these contexts often require the use of different input and output modalities such as speech, handwriting and gesture. In addition, users often have a preference for the input modalities they wish to communicate through at any instance in time, and this is also largely influenced by surrounding environmental factors such as background noise, crowds, and access to the underlying physical and virtual data spaces. Data spaces are no longer limited to graphical objects on a computer display. Modern ubiquitous and mobile computing scenarios now require user interactions that span both virtual and physical spaces.

It is these concepts that this paper will discuss. In section 2, we describe the scenario that we are implementing, and include motivations for our work. This is followed in section 3 by an outline of the architecture on which the scenario is based. In section 4, we discuss the interaction forms that we are implementing, and the flexibility that can arise by mixing and matching input types. We provide our conclusions in section 5.

2 Scenario

Although research is now being conducted on interfaces for ubiquitous computing [1], and for the domain of shopping [5, 2], the combination of these areas with that of multimodal interaction [6, 4] is only slowly gaining momentum. Our ShopAssist application combines the context of shopping, with that of mobile and ubiquitous computing and multimodal interaction. Two different scenarios are currently being catered for. The first scenario supports the use of a shopping trolley in a grocery shop, and aids the user through plan recognition strategies [5]. The second scenario supports the use of a pocket PC within for example an electronics shop, and supports the user by providing a rich range of interaction possibilities. It is the second scenario that we will focus on in this paper.

Consider a user in possession of a pocket PC, browsing through a “real-world” electronics store. The user connects to a data container of their liking (i.e. a shelf), and then waits for the relevant product information for each of the objects in the shelf to be automatically downloaded (e.g. digital cameras). If the shelf does not contain all of the models that the user is currently interested in (perhaps because they are out-of-stock, or because the store does not stock them), the user will still be able to download the required product information from the environment’s server, and use this for example in product comparisons.



Fig. 1. A combined virtual-physical interaction used to compare two products

Upon synchronization with the data container, the user has the ability to interact with the *virtual* set of products downloaded onto the pocket PC’s display, with the *physical* set of products on the shelf, or with a *combination of physical and virtual products* making up the data space. Fig. 1 shows an example of a combined virtual-physical interaction in the form of a product comparison query. Upon downloading the data container, the user also has the option of disconnecting from the shelf and the surrounding environment and browsing entirely offline. This limits the range of inter-

actions that are available to the user, but guarantees a high level of user privacy, especially if the user is dubious about the integrity of the shop.

Depending on background noise levels within the electronics store, the user may decide to interact with the products through the medium of *speech*. If the store becomes very crowded, the user may switch to combined *handwriting-gesture* interaction. They may use *extra-gestures* to access the product (i.e. picking up or putting down a real-world object), or if the shop is too crowded or the products are locked away, the user may instead decide to use *intra-gestures* (i.e. pointing via stylus to objects on the pocket PC's display).

A motivation for shops to provide new interaction types such as extra gesture is that this supports *interaction shopping*. In contrast to *window shopping* (where a user is generally limited to viewing products locked behind a glass window), interaction shopping permits a user to physically interact with and query the objects around them. Based on our initial observations, we believe that tangible product queries provide the user with a certain fun factor less commonly found when browsing products solely on a computer display. Even without physically being able to touch an object, multimodal interaction can still benefit the more limiting form of *window shopping*. As an example, consider shopping on a Sunday when the shops are closed, or even during the week outside of business hours. Interaction through modalities such as speech, handwriting and intra gesture would in this case still permit a user with enough flexibility to purchase products.

In our scenario, the environment also contains infrastructure to support public displays. When such displays are not currently in use by other users, a user may request detailed information on a particular product(s) to be displayed here, instead of on the comparably smaller display of the pocket PC (see Fig. 2). Upon deciding on a product to purchase, the user can finally add the product to their shopping list and either continue to browse, or progress to the counter. Our public display infrastructure works equally well for interaction shopping and window shopping, but currently only supports a single user at a time.



Fig. 2. Both private and public screens can be used to provide product information

3 Architecture

As seen in the scenario above, there are three main components to this ubiquitous computing environment – the user, the device, and the environment [3]. The environment can be further decomposed into rooms, data containers and data objects, in which data containers refer to items such as shelves and tables, and data objects refer to the individual products contained within a data container. The shelves are fitted with RFID antennas, while the products are fitted with RFID tags. In this way, the environment's server can identify which products have been picked up or put back into a shelf. The server runs on an RMI infrastructure and has access to the underlying product databases, which may be distributed throughout the environment. The server also supports multiple client types such as shopping trolleys and pocket PCs.

In this shopping scenario, the pocket PC is the central communication portal between the user and the environment. Data is communicated in XML format over a TCP/IP socket connection, and contains information on the products within a container, as well as the associated dynamic language grammars for each product type. With the exception of extra gesture recognition, all of the interaction processing is performed locally and in real time on the pocket PC itself. In situations where multiple shelves exist in a single room, infra-red beacons are used to identify each shelf. The downloading of the container data onto the pocket PC takes place over a wireless LAN connection.

The ShopAssist's *input modalities* include speech, handwriting, intra and extra gesture, while its *output modalities* include speech, text and graphics. IBM's Embedded ViaVoice V4.2¹ is used for speech recognition, while Microsoft Transcriber V1.51 is used for character recognition. The grammars are derived from product types within the product database, and are dynamically loaded upon synchronization with the shelf. Each grammar contains separate values for the modalities of speech and handwriting. The speech grammars may theoretically contain any number of words, however our grammars only contain and have only been tested using around 50-100 unique words. Fig. 3 shows a combined handwriting-gesture interaction that has been mapped to valid grammar entries. "CMD_H" refers to a handwriting event that has been mapped to a COMMAND, and "OBJ_GI" refers to an intra-gesture event that has been mapped to an OBJECT referent.

The system uses ScanSoft's RealSpeak Solo² for concatenative speech synthesis output, and can also fall back to IBM's Embedded ViaVoice formant synthesizer depending on the available memory. Whereas the concatenative synthesizer sounds more natural, the formant synthesizer sounds more robotic, but requires much less memory.

¹ IBM EVV, http://www-306.ibm.com/software/pervasive/embedded_viavoice_enterprise/

² ScanSoft RealSpeak Solo, <http://www.scansoft.com/realspeak/mobility/>



Fig. 3. An example of how a handwriting event (1) is recognized by the character recognizer (2), and then mapped to a valid grammar entry (3). Also note the summarized modality recognition results (4)

4 Virtual and physical user interactions

Our system supports the following input types: - *speech*, *handwriting*, and *gesture*, whereby gesture can be further categorized as being either of type *intra-gesture* or *extra-gesture*. Intra gestures refer to stylus input on the touch screen of the pocket PC, while extra gestures refer to physical real-world interaction with products. We currently only have one intra-gesture type (i.e. “intra-point”), and two extra gesture types (i.e. “extra-pickup”, and “extra-putback”). Intra-gestures could be extended in the future to also account for simple commands like “delete-from-shopping-list”, and “add-to-shopping-list”, while extra-gestures could be extended to include functionality like “extra-point”. In comparison to an “extra-pickup” command which requires physically touching a real-world object (based on RFID technology), an “extra-point” command would allow the user to select an object from a distance (for example, based on barcode scanning for very small distances, or based on optical marker recognition for longer distances).

A primary goal of multimodal systems is to convert multimodal input into a language that is not dependent on any modality (i.e. uni-modal). All user input in our system is converted into the following uni-modal user input type, via a modality fusion module (first versions of which are published under [8]):

$$\langle \text{COMMAND} \rangle \langle \text{OBJECT} \rangle + \quad (1)$$

For the context domain “digital cameras”, COMMAND refers to values such as “compare”, “price” or “mega pixels”, and OBJECT refers to values such as “PowerShot S1” or “PowerShot S60”. As an example (see Fig. 4) the combined speech and gesture input command: “How many mega pixels does this camera have <G =EOS 10D>?” is mapped to:
 <COMMAND=“mega pixels”><OBJECT=“EOS10D”>.



Fig. 4. The circled parse information shows a mapping of the modalities *speech* (S) and *intra gesture* (GI) to that of the uni-modal language elements *command* (CMD) and *object* (OBJ)

Although we have stated above that speech, handwriting and gesture input can all be used to form a uni-modal result, not all of the combinatorics in combining these inputs are currently available in our system. As an example, if three different modalities are available for both the COMMAND and a single OBJECT referent, we would have 9 modality combinations that arise from mixing modalities together, and 27 modality combinations for a single COMMAND and two OBJECT references. This is even before considering the effects of overlaid modality information as in the case of the following example: “How many mega pixels does the PowerShot S50 <G=PowerShotS50> have?”, in which both speech and gesture are used to define the same object referent. Fig. 5 shows the input modality combinations that are being implemented.

	COMMAND	OBJECT	
1	Speech	Speech	} Implemented modalities
2	Speech	Gesture	
3	Speech	Handwriting	} Implemented modalities
4	Handwriting	Speech	
5	Handwriting	Gesture	} Implemented modalities
6	Handwriting	Handwriting	
7	Gesture	Speech	} Implemented modalities
8	Gesture	Gesture	
9	Gesture	Handwriting	

Fig. 5. Input modality combinations currently being implemented in the ShopAssist demonstrator

Indeed some of the implemented modality combinations require new interaction metaphors to work. For example, by providing the user with a visual “What Can I Say” (see Fig. 6), the system can evaluate modality combinations in which not only the OBJECT but also the COMMAND are obtained via intra-gesture:



Fig. 6. Displayed as scrolling text, a visual “what can I say” can allow the user to access command functionality through stylus intra-gestures

As described in [7], *situational statements* about the surrounding *environment* may affect which input modalities to use. For example, a very noisy environment may require the use of combined gesture-handwriting interaction, and a very crowded environment may require the use of sole intra-gesture interaction. Aside from environment characteristics, *user requirements* also affect multimodal input. For example if the user is on the move, speech interaction may be the best form of human-computer-interaction to use. *Device requirements* further affect which modalities should be used, for example in our scenario, speech input requires the user to press a button on the PDA to start and stop an utterance³. Handwriting input requires a touch screen and a stylus. Intra-gestures require a touch screen and finger input, and extra-gestures require only the use of a hand (and of course the availability of a real-world object to interact with).

³ It is interesting to note that unlike office environments where background noise may be negligible enough for a recognizer to be always actively listening, mobile scenarios are much less likely to support such ideal conditions.

Each situational statement can contribute in determining which modalities are best to use. A future extension to the ShopAssist will be to automate the recognition of situational statements and to alert the user of the most appropriate input modalities to use, or alternatively pro-actively bias certain input types based on these situational statements to aid in increased user input recognition accuracy.

5 Future Work and Conclusions

Future work will now turn towards usability testing. We will try to determine the level of learning required for multimodal interaction by unfamiliar users, and the acceptance level that users will have for new interaction types, especially within a public environment. The systems ability to sustain acceptable recognition rates when placed outside of a controlled test environment will also need to be studied.

This paper has described the potential use of multimodal interactions within a ubiquitous shopping scenario. We have described the architecture required for such an implementation, and have also outlined the interaction modalities and modality combinations that may be relevant for physical and virtual data spaces.

References

1. Dey, A., Ljungstrand, P., Schmidt, A., "Distributed and Disappearing User Interfaces in Ubiquitous Computing", CHI Workshop, (2001).
2. FutureStore. Future store initiative, June 2003. Official website: <http://www.future-store.org>
3. Kray, C., Wasinger, R., Kortuem, G., "Concepts and issues in interfaces for multiple users and multiple devices", Workshop on Multi-User and Ubiquitous User Interfaces (MU3I) at IUI/CADUI, (2004), pp. 7-12.
4. Oviatt, S.L., "Multimodal interfaces", In *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, (2003), pp. 286-304.
5. Schneider, M., "A Smart Shopping Assistant utilizing Adaptive Plan Recognition", ABIS Workshop on adaptivity and user modelling in interactive software systems, (2003), pp.331-334.
6. Wahlster, W., "Towards Symmetric Multimodality: Fusion and Fission of Speech, Gesture, and Facial Expression", *Proceedings of the 26th German Conference on Artificial Intelligence*, (2003), pp. 1-18.
7. Wasinger, R., Oliver, D., Heckmann, D., Braun, B., Brandherm, B., Stahl, C., "Adapting Spoken and Visual Output for a Pedestrian Navigation System, based on given Situational Statements", ABIS Workshop on adaptivity and user modelling in interactive software systems, (2003), pp.343-346.
8. Wasinger, R., Stahl, C., Krüger, A., "Robust speech interaction in a mobile environment through the use of multiple and different media input types", *Proc. of EuroSpeech*, (2003), pp. 1049-1052.