Modality management for multimodal human-machine interfaces

Katharina Bachfischer,[1] Moritz Neugebauer,[1] Niels Pinkwart,[2] and Tobias Brandt[1]

[1] Volkswagen AG, Forschung Elektronik und Fahrzeug, Bedienkonzepte und Fahrer, Brieffach 1777/0, D-38436 Wolfsburg
[2] Technische Universität Clausthal, Institut für Informatik, Julius-Albert-Str. 4, D-38678 Clausthal-Zellerfeld

Abstract Synergistic multimodal human-machine interfaces are characterised by their ability to interpret user input from more than one input modality. Such interfaces may contribute to better driver information systems in terms of efficieny and comfort of use. In this article we present an approach for the integration of voice and touchscreen input as well as capacitive proximity sensing for two scenarios: interaction with a map of points of interest and with a media player. We will present details of the system realisation and of the implementation of the scenarios. Finally, we will report results from a recent user study.

1 Introduction

The human machine interface (HMI) of the car of the future has to allow for time-efficient access to a broad range of functions while reducing the perceived complexity of the system. A promising approach to achieve this goal lies in multimodal HMIs. Multimodal HMIs provide a greater flexibility when interacting with a software system since a variety of input and output modalities are available to the user. Depending on the situational context and personal preferences the user can always select the most appropriate way to interact with the system. Thus, a well-desgined multimodal HMI makes the overall system seem less complex and less demanding on the user. This should hold true even during interactions where a lot of information is to be supplied to the system by the user.

Multimodal input options are already present in today's vehicles: when interacting with a certain function the user can choose between several alternative modalities in order to input a particular command. In contrast to an exclusive use of one out of various modalities, synergistic or alternating multimodal input offers an increase of efficiency and comfort since inputs made via different modalities are interpreted as one single command.

The many varieties of multimodal interaction have already been subclassified as either synergistic or alternating multimodality. Following we use the term *synergistic multimodality* if information input occurs via different modalities at the same time and is semantically fused to one command [1]. In contrast, the term *alternating multimodality* is used if parts of information are given successively and nevertheless are interpreted as one combined command. The step of fusion of multimodal commands is thus part of the larger task of managing more than one modality.

In synergistic-multimodal or alternating-multimodal touchscreen interaction, gesture and speech inputs are combined. This way the communicative characteristics of the different modalities are exploited in an ideal way. To specify objects, geometric dimensions and positions as well as to adjust scales (e.g. for volume control) manual pointing gestures and touchscreen input are particularly suitable. For input of names, terms and complex commands, speech input is more adequate. One common example is the *Put-That-There*-metaphor [2]. Commands including spatial-geometric information as often used in map interaction, are according to [3] the most common type of tasks for employing multimodal interaction.

In this article we present an approach to modality management and demonstrate its application to two use cases, namely interaction with a map of points of interest (POIs) and music selection from a database of music files. For both use cases, a specific combination of touch gestures and speech commands has been implemented while the focus will be on multimodal input by the user rather than multimodal output by the application in question. First, we are going to present an extensible system architecture for managing various input modalities. After a sketch of the two use cases—map interaction and music selection—multimodal integration and information fusion of touch and speech interaction will be treated in more detail. Finally, results from a recent user study will be

reported.

2 System overview

For the task of modality management we assume at least four main building blocks which are derived in a top-down fashion from the purpose of the application as a whole. Once we have identified the fundamental components of our system architecture, the properties of each of these components and the dependencies between these components will be explored.

The first component is a set of recognition engines for speech and gesture recognition, for example. These recognition engines represent the possible input channels which provide first-hand data about the user's intention when interacting with the system. Whenever these external software components need to pass on information to the system for further interpretation, a predefined set of command words is used for communication via TCP/IP. With respect to this abstract communication layer between the recognition engines and the remaining components, the system may be extended with other recognition engines with little effort. The recognition engines as a first component are depicted in Figure 4.1.

The core of the modality management system is comprised of a fusion component and a component for dialogue management. The fusion component takes care of the process of fusing the information of the recognition engines according to a given set of rules for integrating various information blocks. It is the task of the dialogue manager to coordinate system output and control the interaction between the human machine interface as a whole and the user. The separation of external software components (input channels) and internal components (dialogue management) is reminiscent of the concept of model-view-control (MVC). In our case, the model (the data) is represented by the external applications, the view is given by the GUI or other output. Finally, the controller is comprised of one or more recognition engines.

An important aspect of MVC is that the graphical user interface (view) of the application can be developed and maintained independently of the data model and the control of the application. In contrast to a monolithic system architecture, this results in reusable components and avoids redundant information across the components of the system.

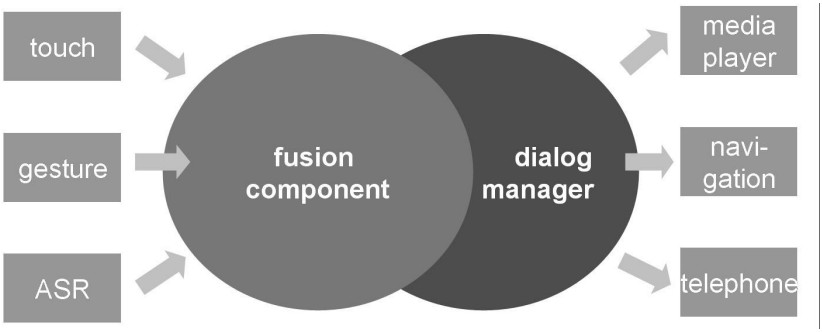

Figure 4.1: System architecture.

On the right hand side of Fig. 4.1 external applications, such as a media player, are depicted. The fusion component is responsible for triggering all dialogue actions since it delivers all information about the user input to the dialogue manager component. Similarly, the dialogue manager has full control of the external applications (media player, navigation, telephony), since commands to these applications are only carried out in accordance with the dialogue model that has been implemented. Communication with output modalities, such as text-to-speech and communication with the graphical user interface, are handled similar to the communication with recognisers.

Two components of our modality management system have been highlighted as being core to the task of modality management, namely the fusion component and the dialogue management component. In order to achieve a better understanding of the workflow inside the modality manager, Fig. 4.2 gives a detailed view on its internal components. Starting out from a view (input modality), socket network connections are being established which provide an interface to recognisers one the left hand side and applications on the right hand side. The incoming data from the recognition engines is preprocessed with respect to matching applications and application-specific adjustments may be made. Right after, the commands are ready to be manipulated by the fusion and dialogue manager component.

If problems should occur in the step of modality integration, for example if contradictions between two or more recognised commands cannot

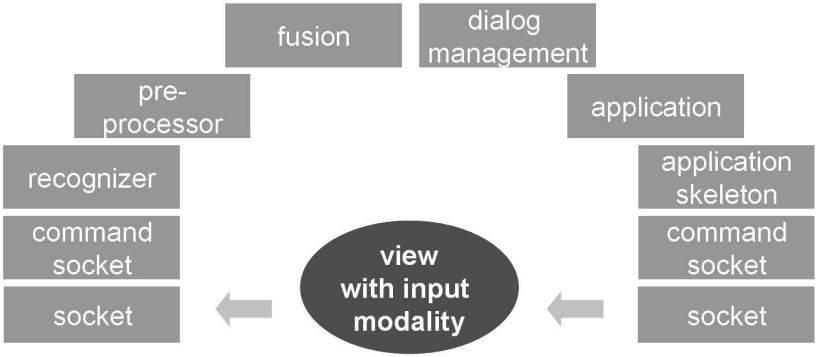

Figure 4.2: System architecture.

be resolved, error management would take place here in the dialogue manager. Application and application skeleton are responsible for controlling media player or other external applications; commands to these applications are sent via the socket connection. After an optional update of the view, one workflow cycle has been completed.

3 Use cases: map interaction and music selection

Following on from the presentation of our approach to modality management, we now present the use cases which demonstrate the system in use. Our implementation comprises two use cases: an area-related POI search and music selection by artist name or song title. In the description of these two use cases we are going to abstract from details of the temporal alignment of touch and speech interaction which distinguish alternating and synergistic multimodality. Instead, both use cases are presented with sufficient detail such that the demands on the user are highlighted for map interaction and music selection, respectively.

In the case of map interaction, the user defines the search area on the map with a drawing gesture on a touchscreen and specifies the requested POI category by speech. Given that the area on the map is defined

properly via the drawing gesture, the POIs found for this spatial context are displayed on the map. The two upper pictures in Fig. 4.3 show an example of an interactive map (a) augmented by a manually selected area with a few relevant hotels which are displayed on it (b).

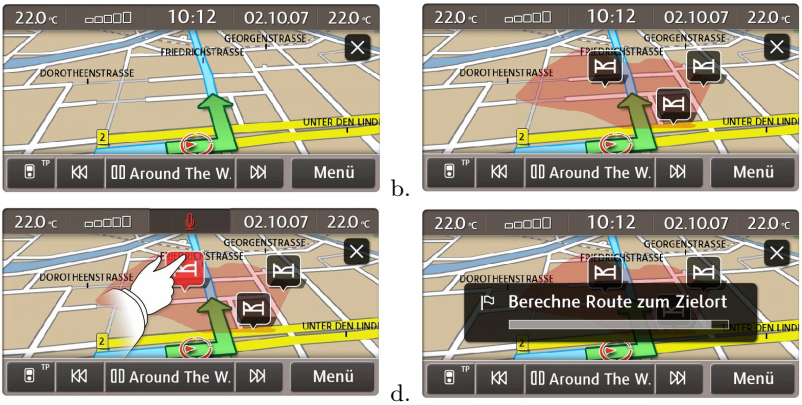

Figure 4.3: Screenshots of the navigation graphical interface.

By activating the speech recogniser in the context of navigation, only the navigation-specific vocabulary is taken into account for the recognition. Hence, error rates may be kept low. A second dialogue step involves selecting POIs via a second touch action by which phone calls, destination or additional information on this POI may be requested. This is depicted in the two pictures at the bottom of Fig. 4.3: in (c) the user selects one of the POIs found in the search area, in (d) the route to the selected POI is being calculated.

Generally, map interactions including the one just described may be classified in the following way [4]:

- *Informative map interaction* The current map (and the associated metadata) are used to identify, select, or request information about certain objects. The map itself is not being manipulated in this case.
- *Manipulative map interaction* The current map is being updated. Manipulative map interactions are comprised of:

- Map navigation: keep map content, change to different focus (zoom, pan).
- Show/hide objects: change map content, filter or expand the selection of object on the map.

With respect to this classification, our use case for POI selection may be subsumed under manipulative map interaction. While the focus remains in the same state throughout the interaction, map content is manipulated by the user. This holds true for both, the selected area which defines a local filter and also the POI category which leads to an update of the informational status of the map.

Music selection from a database of music files, our second use case, involves only one dialogue step which, broadly speaking, consists in manipulating the title that is played at the time of interaction. The commands which are available to the user refer to either control of the media player functions such as *play*, *pause* and *skip* or the user may select a new song by speaking the song title or by choosing a new artist. In the latter case, the first song in the artist's track list will be played.

Media player functions are available globally in this use case, meaning that they are part of the active grammar in both artist and song selection. However, by touching on either artist or title string, the user may further specify the context of use. A touch action on either field loads the list of artists or song titles, respectively. Just like in the case of POI selection, the touch action is intended to improve accuracy of recognition by limiting the list of available vocabulary items. The screenshot in Fig. 4.4(a) visualises the graphical interface for the music selection use case. First, speech recognition is activated (b) and, by pointing on song title and speaking the desired title to be played, the music player skips to a new state (c).

In the case of the skip functions (e.g. next, previous) there is an additional way to aid successful recognition of the user's speech command. By saying the category which is to be manipulated along with the skip command (e.g. next artist, previous title) both parts of the voice command are taken into account when fusing it with a touch action on the artist or title entry.

Figure 4.4: Screenshots of the media player graphical interface.

4 Interweaving touch and speech interaction

Earlier in this article we introduced the distinction between synergistic multimodality and alternating multimodality. For the implementation of the two use cases above, both variants of multimodal interaction have been explored and will now be described in detail.

The first variant implements alternating multimodality and will in the following be referred to as Talk-To-Object (TTO). In this variant, automatic speech recognition is activated by a long touch action on the touchscreen. The area which is marked by the haptic action is used to define the context which the user wants to operate in. In the context of music selection, for example, a long touch action right on the artist name indicates that the user intends to manipulate this particular information out of all the others which are available on the graphical user interface. By activating the speech recogniser in the context of artist names, only the specific vocabulary of this context is taken into account for the recognition.

For area-related POI search, the object to interact with has to be defined before automatic speech recognition comes into play. This is due the fact that the speech recogniser can only be activated by pressing on some GUI item which, in this situation, is supplied by a marked area.

While in the case of music selection the artist and music title slots are always present (and thus available directly for touch actions), a drawing gesture constitutes the first step in POI search.

After the desired area has been provided by the user, a long touch action somewhere in this area on the touchscreen activates automatic speech recognition. That is, once the area is selected, only the vocabulary for POI selection is loaded in order to reduce error rates. The user can now select the desired POI category. If the search for the given category was successful, further interaction with the displayed POI entity is possible. A long touch action on one of these icons activates speech recognition and phone calls, destination or additional information on this POI may be requested.

The second variant implements synergistic-multimodal input and is characterised by detecting the proximity of a hand to the touchscreen interface as well as pointing gestures; the integration of synergistic-multimodal inputs is realised on the basis of a "Point-To-Talk" (PTT) scenario. By capturing position and movement data of the user's interacting hand, the system can infer the interaction intention of the user. The detection of the movement track of the hand and its position can be realised, e.g. by infrared or capacitive proximity sensing. An approach to the display (touchscreen) by less than 10 cm distance is interpreted as a general interaction intention by the communication manager.

The crucial difference between the PTT variant and the TTO variant is the way the automatic speech recognition component is activated. While in TTO, the user is required to actually touch the display in order to set off a voice command; in the PTT variant the detection of interaction intention is used to activate the speech recogniser.

To manipulate the media player in the PTT variant, the user is not required to actually touch the screen but a pointing gesture combined with a speech command would suffice. In this specific interaction, all speech commands (artists, titles, global commands) will be available at the same time. However, by touching the artist name, for example, the same beneficial effect is achieved as in the TTO variant: the vocabulary will be reduced to a subset of commands and recogniser performance will be better. Note that this touch action really is optional in the PTT variant: since automatic speech recognition is activated by proximity of a hand to the touchscreen, there is no obligatory touch interaction involved.

In map interaction, the drawing gesture which selects the relevant area and the speech command to select the area are in a way disentangled for the PTT variant. Since automatic speech recognition is activated by proximity of the hand, the user may actually speak the POI category before, while or after drawing on the interactive map. This demonstrates that multimodal integration by the modality manager is in fact independent of temporal alignment. The next section is going to cover information fusion and multimodal integration in more detail.

5 Information fusion and multimodal integration

Information fusion

Information fusion describes the process of combining data of different sensors or information sources in order to create new or more precise knowledge of physical parameters, events or situations [5]. *Information* is everything what can potentially contribute to reduce existing uncertainty [6]. Useful information concerning the fusion process is specified by facts and their associated uncertainties as well as by a description of dependencies between information parts of different sources. Generally one can differentiate three levels of information fusion [5, 7]:

- *Lexical fusion* or *fusion on a signal level*: Signals of sensors are combined directly. Prerequisites are comparable signals as well as registration (identification of common features) and synchronisation (coordination of events or signals in order to operate a system in unison).
- *Syntactic fusion* or *fusion on a feature level*: If no temporal or spatial coherence can be guaranteed, it may be useful to fuse signal descriptors in order to achieve better numerical estimates of certain signal characteristics.
- *Semantic fusion* or *fusion on a symbol level*: On the basis of associated probabilities, symbolic signal descriptors as e.g. classification results are combined to make a decision.

As we fuse different modalities in our system and output of the recognition engines differs widely in our system, we standardise recognition results on a common abstraction level and in the latter fuse it on a symbol level. Referring to the classification of [8], our fusion approach on

a symbol level would be categorised as *soft-decision fusion*, as the confidence of each classifier is taken into account as well as the integration on an N-best list of each classifier.

Multimodal integration

Multimodal integration connotes the way in which information is fused. According to Althoff [9] there exist three different methods:

- *Temporal integration*: Information parts are considered with respect to their temporal relation. They are linked together if they coincide or arrive with short temporal delay. As the utilised speech adapter does not deliver the period of time in which the speech input was made, temporal integration could not be used here.
- *Rule-based integration*: Information parts are linked on the basis of different rules, e.g. a context-free grammar or a slot-filling-algorithm.
- *Stochastic integration*: Each recognition engine creates a ranking of results or provides associated probabilities. Integration is carried out on this basis.

In our system we use a combination of rule-based and stochastic integration.

Many information parts do not have to be integrated, but should be processed immediately. Only more complex input requires a detailed examination. Those commands, as mentioned in Sects. 3 and 4, are composed of two parts at most, where one part may be optional. Therefore, we defined three different categories of input:

- *Full commands* do not require further integration.
- *Functions* can be integrated with a parameter or can be processed on their own.
- *Parameters* can only be processed in combination with a function.

The combination of functions and parameters shows characteristics of the slot-filling methods: Each function has a defined set of slots with a predefined possible content. In contrast to classical slot-filling algorithms, parameters have only to be checked when a function appears. We do not have to test whether functions or full commands can be integrated with

other functions or full commands. Generally, with complex functions this fusion algorithm would be time consuming, as still many integration options would have to be tested. This would occur if a function had many slots and many parameters had to be fitted in. In our input scenarios this does not occur.

Hypothesis generation and processing

If the fusion component receives an information part (categorised as full command, function or parameter), a base hypothesis is created. Each hypothesis m is attributed with a confidence level $K_{\mathrm{hyp},m}$ which is determined by the recogniser. Touchscreen input is regarded as 100% confidence, the speech recogniser sends a N-best list of results with corresponding confidence values. Full commands cannot be further integrated, though they are directly passed on to the set of active hypotheses and—attributed with a 100% confidence—immediately executed.

Parameters and functions are, if possible, combined with respect to the rule base and form a new hypothesis while the original base hypothesis is kept. The confidence level K_{hyp} for a new hypothesis is calculated on the following basis:

$$K_{\mathrm{hyp}} = \frac{1}{M} \sum_{m=1}^{M} K_{\mathrm{hyp},m} + p\,(M-1)\,, \tag{4.1}$$

with $M-1$ parameters which are integrated with a function. p is a factor that serves to privilege combined hypotheses.

All new hypotheses are passed on to the set of active hypotheses, and for all active hypotheses we constantly check if they can be processed. However, the following criteria have to be assured:

1. For each (combined) command the best hypothesis should be processed and each command must only be executed once.
2. Out of each N-best list only one entry can be processed.
3. Functions have to be associated with all parameters before they can be executed. Parameters cannot be processed on their own.
4. The minimum time to live has to be reached. Some functions have optional parameters which may occur at the fusion component at a later point of time than the function. Therefore those functions

have to "wait" for a certain time to have the chance to integrate optional parameters.

5. The maximum time to live must not be exceeded. Functions which are lacking parameters or parameters which are lacking functions can "wait" for a certain time in the fusion component. After a certain amount of time, however, they are deleted from the set of active hypotheses.
6. The confidence of a hypothesis should exceed a certain level. A lower and a upper confidence level is defined. Hypotheses which exceed the upper confidence level are processed immediately if they comply with the criteria above. Hypotheses which do not reach the lower confidence level are deleted. Hypotheses which lie in between stay in the set of active hypotheses until their maximum time to live has expired. Therefore the hypothesis can be integrated with successive hypotheses.

It is important to notice that the temporal order of inputs does not necessarily correspond to the output order. Especially if the input order is function → full command → parameter, the function processing may be delayed while the command is executed immediately. As a possible example can serve the input order of *touch on artist* → *touch on play* → *speech input "Beatles"* that would be processed in the order of *start music* (touch on *play* is a full command) → *change of artist to "Beatles"*.

6 User evaluation

After realising the system with two options of interweaving touch and speech interaction (TalkToObject and PointToTalk) we conducted a user study in order to evaluate the two different ways of interaction. Therefore we designed a questionnaire and asked the test persons to interact with the system for about half an hour while driving a driving simulator.

In whole, 19 persons took part in the study of which 17 answered the questionnaire. The mean age was 28 years ($\sigma = 5.5$ years). Incorrect data as well as data in which user errors occurred were deleted. Moreover extreme outliers were eliminated by box-whisker plots. After all 361 data sets could be analysed.

Figure 4.5 shows the median (and the 95% confidence interval) of the interaction times for different tasks. The confidence intervals do not

overlap, and also a Mann-Whitney test shows that interaction times in TTO mode are significantly longer than in PTT mode for all tasks.

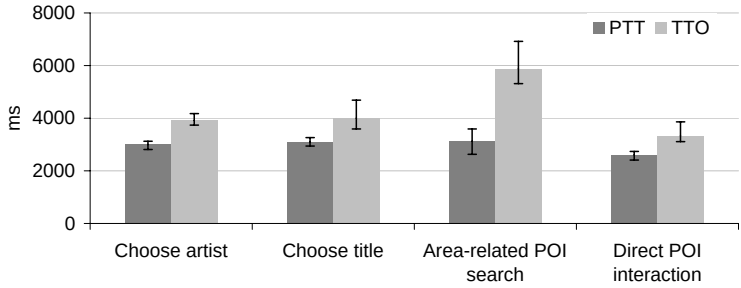

Figure 4.5: Interaction times for different tasks in TalkToObject (TTO) and PointToTalk (PTT) mode.

Concerning the subjective evaluation the PTT mode is clearly preferred (13 persons out of 17, see Fig. 4.6). Nearly the same number of people consider the PTT mode as the more comfortable one. Referring to the question which mode leads to shorter interaction times, again mostly the PTT mode is named. Only on the question where less interaction errors occur, no clear conclusion can be drawn.

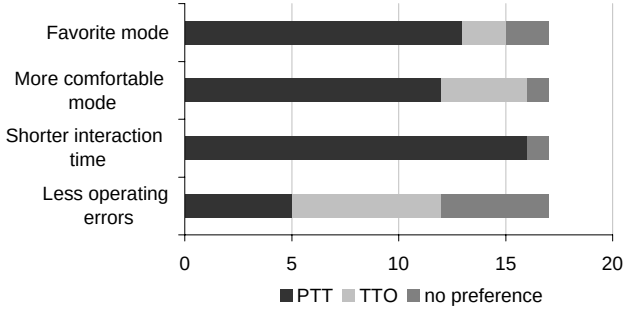

Figure 4.6: Subjective evaluation of TTO and PTT mode.

7 Conclusions

In this article we presented a prototypical implementation of modality management for multimodal human-machine interfaces along with two sample applications and some results from user studies. The underlying system architecture of the modality manager is aimed to be extensible and thus new input modalities as well as additional context information sources may be integrated with little effort.

Two application examples of map interaction and music selection have been implemented using the components described in this paper. For the first component, the recognition engines, we chose automatic speech recognition and touch interaction. Under the term touch interaction we subsume pointing gestures as well as drawing on the touchscreen. The fusion component has the task of combining data of different sensors or information sources. A dialogue manager receives the interpreted input and takes care of the flow of information between fusion component and the applications. The last component is represented by a media player and an interactive map application.

A first evaluation of our system in the form of a user study investigated parameters such as interaction times, error rates and acceptance. We compared two main variants for multimodal interaction, namely PointToTalk and TalkToObject. Our empirical study showed a user preference for the PointToTalk variant; here the detection of interaction intention is used to activate the speech recogniser. Of course, speech input may be combined with touch input resulting in multimodal interaction.

Next steps of our work include a closer analysis of the empirical data gained in the described user study. A follow up study on a broader range of PointToTalk use cases also seems promising. Given the extensible architecture for multimodal integration, it would be interesting to augment our current set of recognisers with additional input, such as eye tracking. To further analyze the applicability of our approach for an in-car scenario, user studies beyond a driving simulator environment are necessary.

References

1. L. Nigay and J. Coutaz, “A design space for multimodal systems: concurrent processing and data fusion,” in *CHI ’93: Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems, 1993, pp. 172–178.

2. R. Bolt, "Put-that-there: Voice and gesture at the graphics interface," in *7th Annual Conference on Computer Graphics and Interactive Techniques*, 1980.
3. S. Oviatt, A. DeAngeli, and K. Kuhn, "Integration and synchronization of input modes during multimodal human-computer interaction," in *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, 1997, pp. 415–422.
4. J. Haeussler and A. Zipf, "Multimodale Karteninteraktion zur Navigationsunterstützung für Fußgänger und Autofahrer," in *Proceedings of AGIT 2003*, 2003.
5. H. Ruser and F. Puente León, "Informationsfusion – eine Übersicht," *Technisches Messen*, vol. 74, no. 3, pp. 93–102, 2007.
6. J. Beyerer, J. Sander, and S. Werling, *Fusion heterogener Informationsquellen*. Universitätsverlag Karlsruhe, 2006, ch. Theoretische Grundlagen der Informationsfusion, pp. 21–38.
7. A. Gourdol, L. Nigay, D. Salber, and J. Coutaz, "Two case studies of software architecture for multimodal interactive systems: Voicepaint and a voice-enabled graphical notebook," in *Conference on Engineering for Human-Computer Interaction*, 1992.
8. B. Schuller, M. Ablaßmeier, R. R. Müller, T. S. Poitschke, and G. Rigoll, *Speech Communication and Multimodal Interfaces. Advanced Man-Machine Interaction*. Springer, 2006, pp. 141–190.
9. F. Althoff, "Ein generischer Ansatz zur Integration multimodaler Benutzereingaben," Ph.D. dissertation, TU München, 2004.