

Architecture for Multimodal Mobile Applications

Roman Englert and Gregor Glass

Deutsche Telekom Laboratories, Berlin, Germany

T-Systems International, Berlin, Germany

Roman.Englert@telekom.de

Gregor.Glass@t-systems.com

Abstract

Applications that are used on mobile terminals require a sophisticated input / output interface in order to ease the user application dialogue. Convenient data I/O can be achieved by the introduction of multimodal channels like stylus, speech recognition / synthesis, and gestures. A complex requirement is that the channels have to be applicable in parallel, e.g. tapping on a map and speaking “zoom in here”. We present an architecture for mobile applications that is distributed between the terminal, the network, and the backbone. The architecture is demonstrated by a prototype that enables users to maintain messaging applications in a comfortable manner.

Key words: Usability, Multimodal Interfaces, Multimodal Application Architecture, Mobile Applications, Usage Scenarios

1. Introduction

The development of future end user services needs to take into account growing complexity and capability of applications and devices, while human ability to cope with these is limited. Thus the gap between systems’ capabilities and human ability to make proper use of them widens. A promising solution is to utilize emerging technologies in order to enable a more intuitive, more natural, user centric, man machine interaction and – communication.

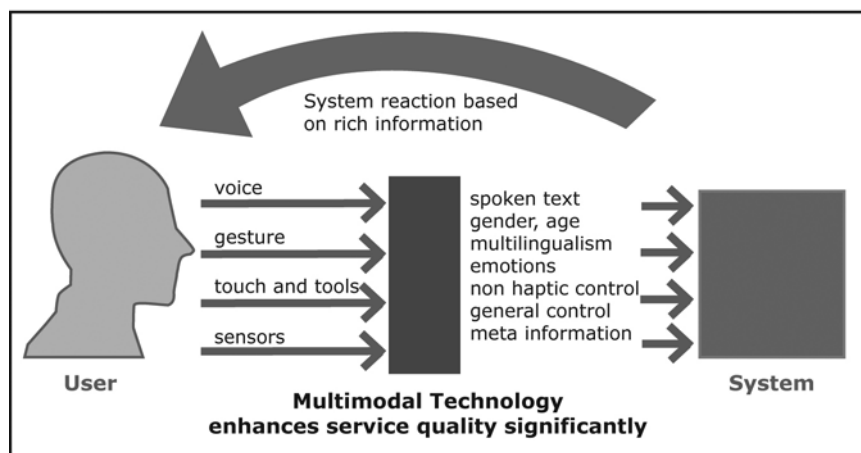


Figure 1: Multimodal input and output channels for applications.

This can be done by providing the user with interfaces that fit to the specific user needs in their current usage context – and to enable the user to choose how to most appropriately interact with the system in a multimodal manner (Figure 1). Multimodal interfaces are the system/human interaction panel for the sequential or parallel application of several input/output modalities like voice, keyboard, stylus, gesture, mimic, and/or kinesthetic. Furthermore, multimodal interfaces have to be symmetric for input and output. Thus, for the input several modalities have to be analyzed and formulated in a consistent hypothesis (fusion) and for the output several actions have to be generated (fission). The application of multimodal interfaces requires the definition of the customer needs and their segmentation. The architecture for a multimodal interface system for mobile applications is described. As prototype an implementation for a mobile multimodal customer self-service solution is depicted. In the scenario a mobile user with a Mobile Windows based MDA III of T-Mobile Germany can utilize pen- as well as speech input (sequential or parallel / composite) in order to register and administer customer care campaigns (i.e. free SMS, and personalized call connected signal) and to type or dictate SMS.

The paper is organized as follows: Section 2 contains the description of the multimodal architecture for mobile applications, and in Section 3 the application scenario for the prototype is described. The implementation of the prototype is detailed (Section 4), and finally, we conclude in Section 5.

2. Multimodal Architecture

The multimodal architecture for mobile applications consists of four components that require a distributed infrastructure as is known for telecommunication systems (Figure 2):

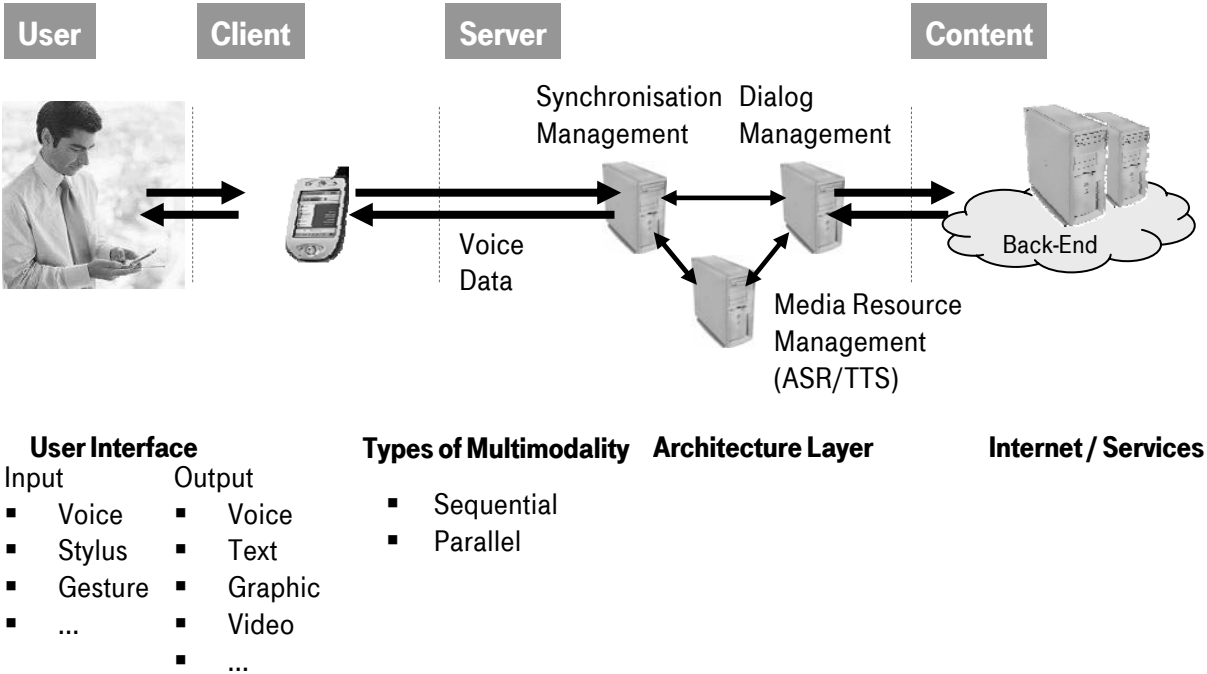


Figure 2: Multimodal architecture for mobile applications.

The user domain contains the interaction of an user with a (mobile) terminal or an interface. Hereby is the interaction given by a task that the user wants to perform depending on an application scenario as described in the subsequent section. Then the client domain contains the interaction media (terminal / interface) that provide the interaction channels for the input / output and whether these channels are applied subsequently or in parallel. Interaction channels are voice, keyboard, stylus, gesture, mimic, and/or kinesthetic. These interactions are synchronized by a server family that manages also dialogues and resources like text-to-speech. Content for dialogue prompts is maintained in the back-end.

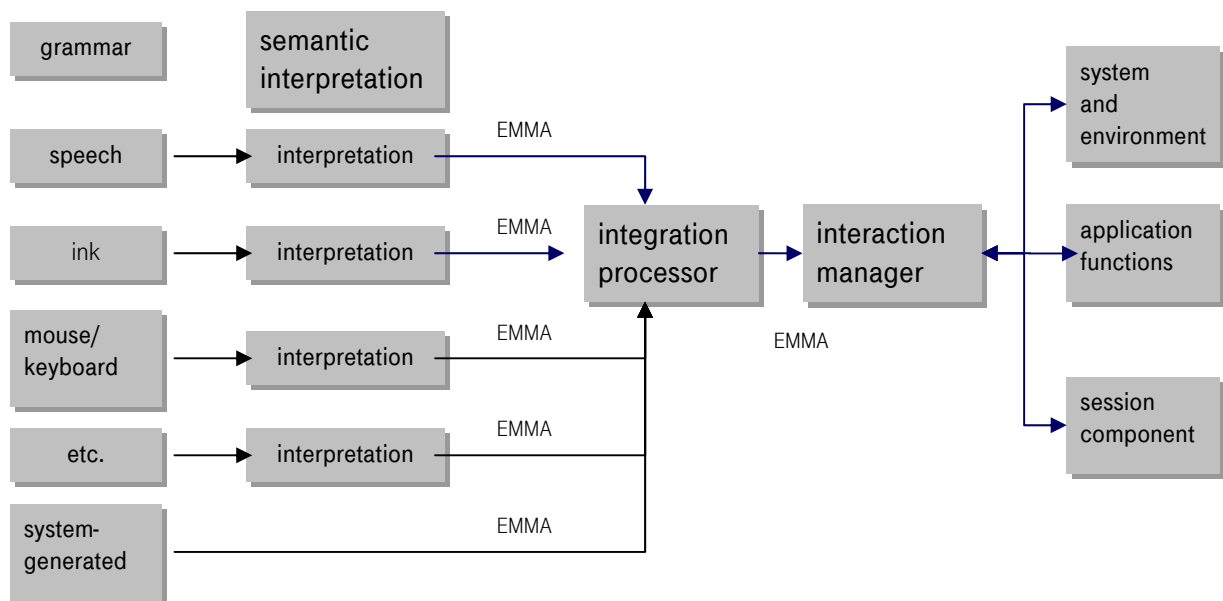


Figure 3: Generation of a consistent system task from multiple input channels.

The application of multiple channels in parallel enables an user, e.g., to provide input via stylus and speech in order to tap onto a map and to speak "zoom in here". However, the system has to combine both actions and to generate consistent tasks for the system. This complex challenge is described in Figure 3: on the left side the various interaction channels are depicted. The interaction channels are interpreted according to their meaning and subsequently they are combined to system task hypotheses. In our reference model the Extended MultiModal Annotation (EMMA) language [EMMA (2006)] is applied, that annotates XML code with multimodal features (see Section 4). Then the interaction manager has to analyse the system task hypotheses, whether they are consistent:

1. Proof of input data
 - a. Integrated input?
 - b. Speech only?
 - c. Ink/stylus only?

2. Proof of suitability of Integration results
 - a. Input data compatible? (e.g. are the real number of stylus input the same like the expected value)
3. Mapping of recognition results from different modalities e.g.
 - a. Speech recognition error but stylus correct
 - b. Speech recognition OK but stylus incorrect
 - c. Confidence ok and stylus ok
4. Decision for error handling output
 - a. Graphical, audio, prompt, TTS
5. Handling of redundant information and creation of related user reaction
 - a. Prioritisation of modalities

The above five steps are applied to check whether hypotheses are consistent, or not. Inconsistent hypotheses violate an integrity constraint, eg. tapping on a map and the x/y coordinates of the stylus do not belong to the window containing the map, or have an incomplete input, eg. speaking "zoom in here" and no tapping action followed. As a consequence inconsistent hypotheses are skipped. The remaining hypotheses are consistent according to the above described five steps. Then the consistent hypotheses have to be ranked with regard to the most probable action that the user has done or intended to do. This can be done by prioritising the input modalities, eg. keyboard, stylus, speech, gestures, and mimic. The order reflects the input accuracy of a modality: speech is more difficult to recognize and to interpret compared to keyboard input, and gestures are more complex to recognize and to analyze compared to speech input. The composite case where a multimodal output has to be generated has the challenge to choose the appropriate modalities. A ranking of the modalities and combinations of them enables a proper selection, e.g. a gesture-based avatar that points onto a map and speech synthesis for the utterance "look here" [Schreer et al. (2005)].

As a result specific system tasks are generated from this analysis. An example of an usage scenario and its application is described in the following section.

3. Usage Scenario of Prototype

As multimodal systems extend the so far established interaction options and thus enable new usage - and service multiverses. Main driver to utilize a multimodal approach was the idea to empower the mobile user to choose the most appropriate in- and output channel according to the specific usage situation and the actual usage goals. As scenario for the prototype an example implementation of a mobile multimodal customer self-service solution was chosen: To investigate and to demonstrate the possibilities, benefits and advantages of a mobile multimodal user interfaces for certain usage situations and self service cases which are too complex, or perceived by the user as too awkward to be automated by a pure voice application. The specification of the usage scenarios took into account general experiences and user requirements from earlier work on voice portals; while the main driving factor were concrete expectations of the responsible process owners at T-Mobile: Usage of a top model mobile device to develop a multimodal solution for an enhanced customer self service to support its marketing campaigns. These campaigns are run to increase customer satisfaction and to introduce existing and new products to customers, providing them with a joyful user experience using these free campaigns/products on offer.

Example:

User selects a sound logo
by clicking on the title with
a stylus and speaking in
order to hear it

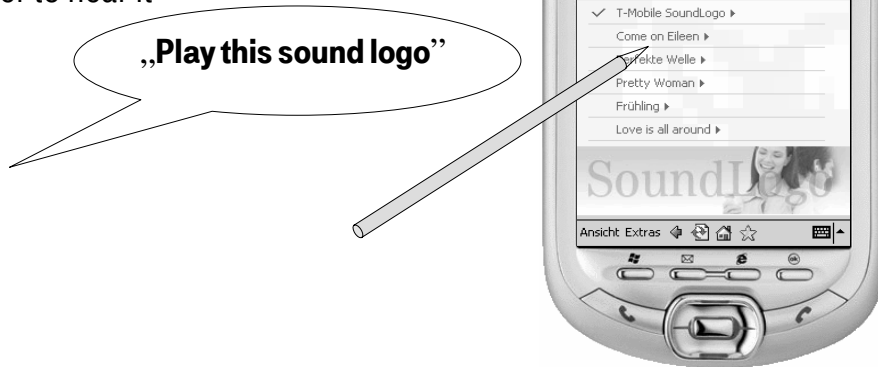


Figure 4: Composite input of selecting item by stylus and command by voice.

In the first step a subset of the T-Mobile “Service Manager” with the options for tariff change and tariff information was implemented. For the system to understand, when the user intends to talk to it, a push-to-talk voice recognition strategy was used. When the user wants to use speech control, he triggers the voice recognition by pressing a hardware button on the device. This was a proof of feasibility for a mobile multimodal application – providing support for sequential pen- and voice input. After an iteration of usability engineering this limited use case was elaborated to a full service usage and – administration. Additionally the multimodal application platform was extended to enable composite (i.e. parallel) input of both modalities (Figure 4), so that the user is now able e.g. to point at an item and verbally express his request, what to do with it. As a result two campaigns “SMS 20” (i.e. 20 free SMS) and “Soundlogo” (i.e. personalized call connected signal) are completely implemented. The user may inform himself about, register for-, administrate and use both campaigns. This includes acquisition of additional information about current and upcoming events and receiving status information for his registered campaigns. The choice in which modality to interact with the system lies completely with the user: Pen- as well as speech input (sequential or composite) may be used and the system will react accordingly, providing feedback in the respective manner.

4. Implementation

In the preceding section we have introduced the idea and architecture of mobile multimodal interaction as well as our specific usage scenarios. In this section the requirements for the platform are given, the software architecture is described, the prototype implementation is explained, and the exemplary usage scenario of the prototype is outlined.

4.1 Implementation Requirements

The implementation of the demonstrator requires a platform that provides capabilities for a number of issues: Firstly, the implementation has to support dynamic application building in order to enable a user-dependent login, service subscription, and service usage. Secondly, storage for user data is needed, like subscribed services and numbers of sent SMS. And

thirdly, the implementation requires the capability to deal with parallel input of modalities (input of voice and stylus), and finally, an Automatic Speech Recognition (ASR) for the German language and Text-To-Speech (TTS) synthesis is necessary.

4.2 Multimodal Platform Architecture

A distributed server structure is necessary to deal with the different I/O of the applicable modalities, and to constitute the basis for an intelligent interaction management (see Figure 2). In order to meet the above described requirements, the following multimodal architecture consists of the multimodal platform, the application server, and the client device (Figure 5).

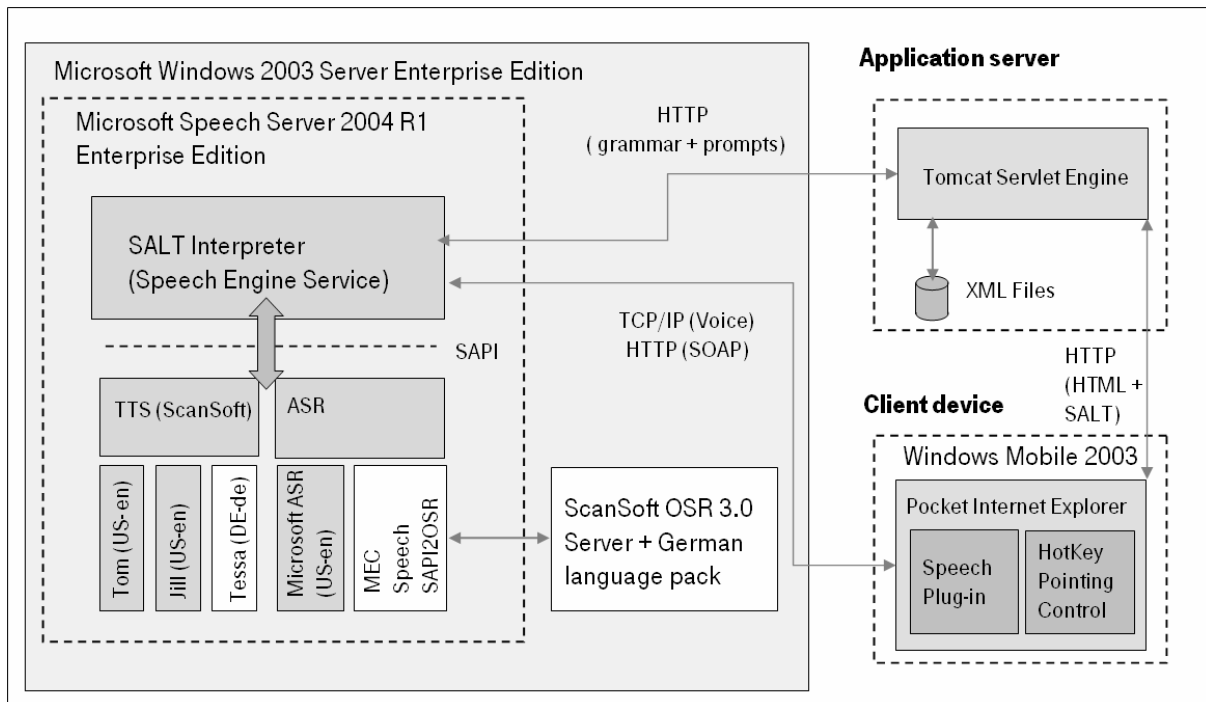


Figure 5: Software Architecture of mobile multimodal application platform.

The Microsoft Speech Server (MSS 2004 R1) [MICROSOFT (2006a)] uses Speech Application Language Tags (SALT) and connects to TTS & ASR via its Speech Application Programming Interface (SAPI). Even though it provides no native German language support, it is chosen as platform due to the fact that the integration of German ScanSoft Open Speech Recognizer (OSR) 3.0 [NUANCE (2006)] (with German language pack) is possible through the MEC Speech SAPI to Media Resource Control Protocol (MRCP) - adapter [MEC (2006)], thus enabling German TTS.

To support the dynamic application building, a Tomcat Servlet Engine (5.0.28) [APACHE (2006)] is implemented according to the MVC model through a Servlet/JSP. The user data is stored in XML files utilizing the Java Architecture for XML Binding (JAXB) [SUN (2006)]. On the client device, the handling of the parallel input is supported through PointingControl and HotkeyControl, both are ActiveX [MICROSOFT (2006b)] components for Windows Mobile [MICROSOFT (2006c)]. The fusion of the parallel input is done with JavaScript [MOZILLA (2006)] in the mobile terminal using EMMA. Finally, the resulting interaction is distributed to the system, application, and session (Figure 2, right side).

4.3 Prototype

The implementation of the prototype requires a server, a multimodal platform and a mobile terminal. The applied server runs the Microsoft Windows 2003 Enterprise version as operating system, the tomcat application server, and the multimodal platform. As mobile terminal the T-Mobile MDA III was chosen. It is a PDA that operates the Mobile Windows 2003 operating system [MICROSOFT (2006c)], features an Intel XScale 400 Mhz processor with Wireless LAN and GSM/GPRS connectivity. As input devices it supports a microphone, touch screen, a miniaturized keyboard, and a functional key for push-to-talk. To enable it for multimodal input the Microsoft Speech Plug-in for Pocket Internet Explorer is installed.

How does this work in practice? First let us have a look at the speech input: The user presses the push-to-talk button of the MDA III. This triggers the ASR to open the microphone and to record the expected speech input: The user's utterances are piped to the server side ASR, where the input is collected and analyzed according to the grammar, until the push-to-talk button is released. Then, the reliably recognized results of the user statements are routed in EMMA mode:speech (see Figure 3) back to the integration processor in the device. As soon as another modality appears, e.g. stylus input, it has to be combined with the speech input. This is done in the integration processor, which is running in our case as a javascript implementation on the mobile terminal. Input from different channels arriving conjointly within a certain time frame (in our case the timeout after a voice input is set to 0.75 seconds) are interpreted as belonging together (composite multimodal input). Input with longer gaps is handled as multiple input, that is queued and processed sequentially. For composite inputs the speech part from the ASR in the EMMA mode:speech is processed together with the stylus input in the mode:ink: If there is a gap of less than 0.75 seconds, the integration processor merges these two inputs to a multimodal, composite input and forwards this information as multimodal information (again in EMMA) to the interaction manager.

Considering again the scenario depicted in Figure 4, the flow of the interaction between the user and the multimodal system for listening to the sound logo "Come on Eileen" is as follows: The user presses the push-to-talk button and says "Play...". S/he continues to speak, at the same time s/he uses the stylus to simultaneously tap onto the logo of her/his request "<this> soundlogo". S/he taps the stylus once onto the name of one displayed logo on the screen, triggering another input event in the EMMA mode:ink. This delivers the ID of the soundlogo. As the user has completed his/her sentence, s/he releases the push-to-talk button. This closes the ASR which now directly processes the voice input. The applied grammar understands the "play", "this" and "soundlogo" and promptly delivers these in EMMA mode:speech to the integration processor, which also receives the x/y coordinate in EMMA mode:ink. As the pen input happened during the speech input, both of these inputs are interpreted as belonging together. Then the integration processor triggers the interaction manager to call the appropriate XML files with the soundlogo via JAXB from the content server and streams it via the network to the mobile terminal.

5. Conclusion

We have shown how to construct mobile multimodal applications in order to enable a convenient data I/O for the user. The architecture is distributed between the mobile terminal, the network, and the backbone. Multimodal data are represented based on the EMMA language that allows to model parallel actions based on different modalities. During the I/O hypotheses have to be checked, whether the multimodal actions (e.g. tapping on a map and speaking "zoom in here") are consistent.

Therefore the so-called interaction manager is applied. The prototype to maintain messaging applications demonstrates the power and flexibility of the described architecture.

However, the approach has limitations. Generally, multimodal interfaces have to be designed individually for each application. Therefore a model for multimodal interfaces is desirable that only has to be instantiated once for a new multimodal application. This topic is currently investigated.

Acknowledgements

The authors thank Gesche Joost for providing the graphic in Figure 1 and Ingmar Kliche for helpful comments. Finally, they thank Thomas Scheerbarth for fruitful discussions.

References

- APACHE (2006). <http://tomcat.apache.org/>
- Bolt, R.A. (1980). Put-that-There: Voice and Gesture at the Graphics Interface. *Computer Graphics*. 14(3), p. 262-270.
- EMMA (2006). <http://www.w3.org/TR/EMMAreqs>
- MEC (2006). http://www.mec.at/content/produkte/tel/tell_en_speech.htm
- MICROSOFT (2006a). <http://www.microsoft.com/speech/default.mspx>
- MICROSOFT (2006b). <http://activex.microsoft.com/activex/activex/>
- MICROSOFT (2006c). <http://www.microsoft.com/windowsmobile/pocketpc/default.mspx>
- MOZILLA (2006). <http://www.mozilla.org/js/>
- NUANCE (2006). <http://www.nuance.com/recognizer/>
- Oviatt, S.L., Cohen, P. R., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J., Ferro, D. (2000). *Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions*. *Human-Computer Interaction*. 15(4), p. 263-322.
- Plomp, J. Mayora-Ibarra, O. and Yli-Nikkola, H.(2001). *Graphical and speech-driven user interface generation from a single source format*. In *Proceedings of the first annual VoiceXML Forum User Group Meeting (AVIOS 2001)*.
- Schreer, O., and Tanager, R., and Eisert, P., and Kauff, P., and B. Kaspar and Englert, R., (2005). *Real-time avatar animation steered by live body motion*, *Proceedings of 13th Int. Conference on Image Analysis and Processing (ICIAP 2005)*, pp. 147 - 154.
- SUN (2006). <http://java.sun.com/webservices/jaxb/index.jsp>