Proposal of a Hierarchical Architecture for Multimodal Interactive Systems

Masahiro Araki[*1] Tsuneo Nitta[*2] Kouichi Katsurada[*2]
Takuya Nishimoto[*3] Tetsuo Amakasu[*4] Shinnichi Kawamoto[*5]

[*1]Kyoto Institute of Technology [*2]Toyohashi University of Technology
[*3]The University of Tokyo [*4]NTT Cyber Space Labs. [*5]ATR
araki@kit.jp

Abstract: In the present paper, we propose a hierarchical MMI system architecture that is currently being discussed by the speech interface committee of the Information Technology Standards Commission of Japan (ITSCJ). First, we present an overview of the proposed hierarchical architecture of multimodal dialogue systems. Next, we explain the first draft of informative descriptions of each layer. The proposed architecture is intended not only to support practical system development by complying with the existing description language and development framework, but also to function as a research platform by specifying the role of each component.

1. Introduction

Multimodal interaction (MMI) technology is expected to be a core element of near future human-computer interaction, such as mobile devices, intelligent car navigation systems, ubiquitous home equipment, and personal robots. Although some advanced MMI systems are developed for various research purposes, there are few commercial MMI systems for practical use. One reason for this is thought to be a lack of well-recognized methodologies of multiple modality fusion and fission. In addition, compared to GUI-based Web applications, there is no established framework for developing MMI systems.

In the present paper, we propose a hierarchical architecture for MMI systems. The goal of the proposed architecture is to support practical system development by complying with the existing design language and development framework, and to function as a research platform (e.g., for the Galatea toolkit [1]) by specifying the role of each component.

The remainder of this paper is organized as follows. Section 2 describes overview of proposed MMI system architecture. Section 3 explains requirements of each component of the architecture. Section 4 discusses the advantages and disadvantages of the proposed architecture through comparison with previous research. The present paper is concluded in Section 5 with a discussion of future research.

2. Overview of MMI system architecture

The target MMI system of the proposed architecture is not only a practical system but is also an advanced research system. Therefore, the basic requirements of the proposed architecture are as follows:

(1) to aggregate modality dependent processing,
(2) to facilitate the addition of new modality, and
(3) to enable timing-sensitive modality control.

Model-View-Component (MVC) architecture, which separates the application logic to the user interface description intermediated by the controller, is suitable for above requirements. Modality dependent processing can be aggregated in the user interface description. The separation of the user interface description facilitates the addition of new modality. Event-driven control of the MVC model realizes flexible control of each

modality. Based on these investigations, the proposed architecture is the MVC-based hierarchical model shown in Figure 1.

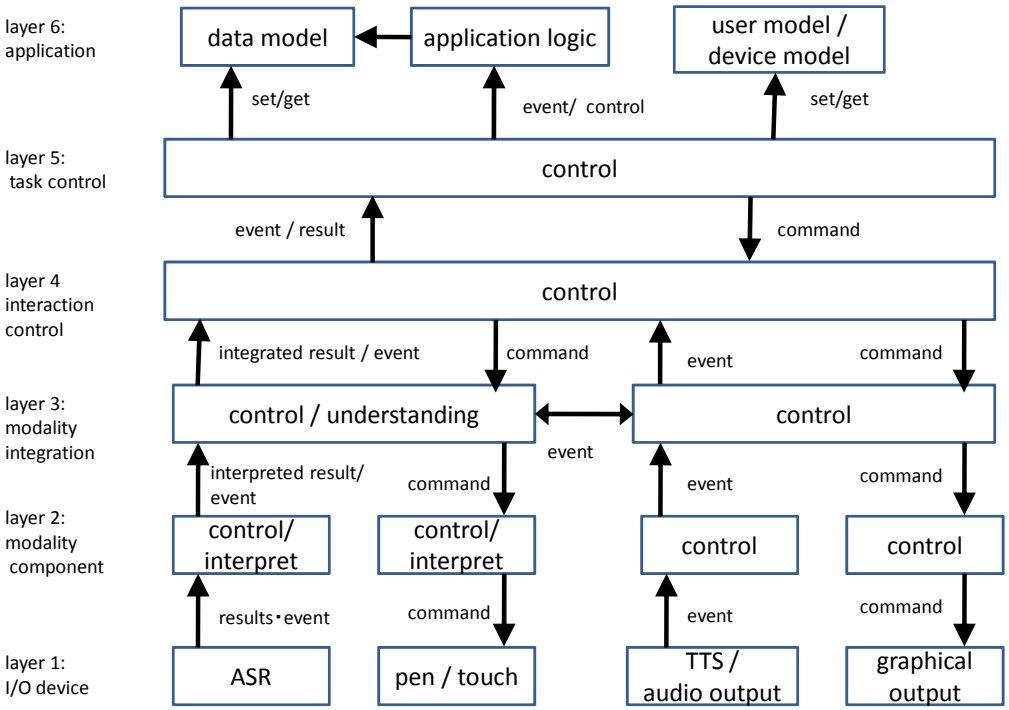

Figure 1. Hierarchical architecture of the MMI system.

3. Requirements of each layer

In order to specify the requirements of each layer, we examine several use case scenarios for MMI systems, such as online-shopping, voice directory search, human-robot interaction, and car-navigation system. Based on these considerations, we examine the necessary function of each layer and the appropriate interface with adjacent layers.

3.1 I/O device layer

The bottom layer is the I/O device layer, which is in charge of signal input/output, recognition/generation of single modality information, and event generation. The actual software components of this level include the ASR engine, the TTS engine, the animated character controller, and the Web browser.

3.2 Modality component layer

The next layer is the modality component layer. Although the main concept of this layer is similar to the modality component of W3C multimodal architecture and Interfaces[1], the proposed architecture includes additional functions.

(1) The modality component layer provides abstract API to the modality integration layer. The input of this layer is a modality-dependent specification of input (e.g., SRGS[2] for ASR), and the output is EMMA[3].

---

[1] http://www.w3.org/TR/mmi-arch/

[2] http://www.w3c.org/TR/speech-grammar/

[3] http://www.w3c.org/TR/emma/

(2) The modality component layer can represent a black-box component that is actually made of several I/O device components (e.g., an animated face that is lip-synchronized to a voice).

3.3 Modality integration layer

The third layer is the modality integration layer, which is responsible for modality fusion for input processing and modality fission for output processing. Since there are various methods of modality fusion/fission for MMI systems, we do not restrict the method of modality fusion/fission, but rather specify the interface between the upper layer and the lower layer.

The modality fusion component receives the start message for input from the upper layer and passes the modality specific input definition to the lower layer. The modality fusion component then returns EMMA as a result of modality fusion processing. The modality fission component receives an output message from the upper layer, and passes the modality specific output definition to the lower layer. The modality fission component then returns the finished event. The specification of the output message remains flexible in the present model. The output message can be an a-modal (i.e., modality independent) representation or an existing modality specific markup language.

3.4 Interaction control layer

The fourth layer is the interaction control layer, which deals with a small segment of dialogue that can be managed without interacting with the server side program. The fourth layer typically corresponds to a form that consists of a group of field elements. The user-initiative behavior of the user is to be dealt with by, for example, the form interpretation algorithm (FIA) of VoiceXML[4].

3.5 Task control layer

The fifth layer is the task control layer, which corresponds to the C (controller) component in the MVC-model. The fifth layer manages the flow of the V (view) component (below the fourth layer) and handles the sub-dialogue, which is typically invoked by an event (e.g., help) from the interaction controller.

If the developer wants to write the explicit state transition, the candidate description language would be SCXML[5]. On the other hand, if the developer wants to use the Rails application, simple script languages (e.g., Ruby, groovy) offer another choice. The Rails application framework enables seamless data management with a back-end database.

3.6 Application layer

The upper-most layer is the application layer. We locate the supportive component for interaction at this layer. These are the DB access components, which correspond to the M (model) component in the MVC-model, the device model component that holds the device information (specified by DCCI[6]), and the user model component, which stores the user properties (e.g., expertise with the MMI system, knowledge level for the domain, etc.).

The latter two components can be accessed from the interaction control layer for determining a dialogue strategy (system-directive or user-initiative), or from the modality integration layer for planning a multimodal presentation.

---

[4] http://www.w3c.org/TR/voicexml21/

[5] http://www.w3c.org/TR/scxml/

[6] http://www.w3c.org/TR/DPF

4. Related research

In this research field, Galaxy architecture [2] is the best known architecture for the spoken dialogue system. The Galaxy architecture can be easily applied to MMI systems because other modality components are pluggable, as long as it has a designated interface. However, Galaxy's hub-and-spoke architecture depends heavily on the script of the central hub component. When a new modality is added, this hub script must be modified considerably.

In a practical system, the W3C's multimodal architecture is a clear-cut suggestion. Maximum use is made of the existing description language and modality component. However, we believe that the Russian doll model is not sufficient for handling multiple modality fusion/fission because the description language for interaction manager (e.g., SCXML) grasps only one side of the modality fusion/fission, which is the timing control. Additional declarative elements, such as the unification of the results from each modality component, and a planning capability for modality fission are needed.

5. Conclusion

We reported the intermediate status of the speech interface committee of the ITSCJ. The proposed architecture is intended not only to support practical system development by complying with the existing description language and development framework, but also to function as a research platform by specifying the role of each component.

In the near future, we plan to construct a specifications document for the proposed architecture and examine the validity of this specification by implementing several types of MMI systems.

Acknowledgements

The authors thank Mr. Kazuyuki Ashimura (W3C) for his participation in this group as an observer and for his fruitful discussion.

Reference

[1] T. Nitta, S. Sagayama, Y. Yamashita, T. Kawahara, S. Morishima, S. Nakamura, A. Yamada, K. Ito, M. Kai, A. Li, M. Mimura, K. Hirose, T. Kobayashi, K. Tokuda, N. Minematsu, Y. Den, T. Utsuro, T. Yotsukura, H. Shimodaira, M. Araki, T. Nishimoto, N. Kawaguchi, H. Banno, K. Katsurada: Activities of Interactive Speech Technology Consortium (ISTC) Targeting Open Software Development for MMI Systems, In Proc. 13th IEEE International Workshop on Robot and Human Interactive Communication (2004)

[2] J. Polifroni, and S. Seneff: Galaxy-II as an Architecture for Spoken Dialogue Evaluation Proc. LREC, pp.42-50 (2000)