An Implementation of the Output Module for a Multimodal Man-Machine Communication System

Darío Alonso

Infineon Technologies AG. Corporate Research, Systems Technology. Otto-Hahn-Ring 6, 81739 Munich, Germany.
Dario.Alonso@infineon.com

Abstract

We present an output system for an intelligent man-machine-communication system in the form of an animated speaking 3D human head model, which interacts with humans by means of natural communication channels. The integration of these channels eases the communication and makes it more robust. Driven by recognized sentences and by the user's head position, a virtual 3D head behaves accordingly controlled by a behavioral organization mechanism based on the Dynamic Approach. As an example environment, we propose a virtual furniture shop.

Introduction

Human communication is not based on speech only: hand and body-gestures, mimics, gaze, lips movement, etc., are additional channels of information. A set of important functions can be assigned to them. First, the robustness of the information transfer is enhanced by exploiting the redundancy among these a priori independent channels (a misinterpretation of one channel can be corrected by the other ones). Second, these misinterpretations can often be discovered in an early stage of the dialog by analyzing the non-verbal channels. Third, the non-verbal channels often allow for a clarification of ambiguous situations. Besides, integrating non-verbal communication channels in the dialog eases the communication since it becomes more natural. Further, the whole set of communication channels defines a certain *context* of the dialog.

Compared to human-to-human dialogs, the human-computer interaction has a number of deficits: mostly, communication is limited to text or mouse input and text or graphics output. In some situations this is uncomfortable and requires the user's adaptation to rather unnatural forms of communication. The integration of multiple natural communication channels in the man-machine interaction (MMI) is a way to transfer the aforementioned advantages of the human-to-human dialog to human-computer interfaces. This is the topic the *Dialog-Modem Project*. The aim of such an initiative is to develop a user friendly interface between a user and an information source, such as the *virtual shop* scenario proposed here, for instance. The idea is to implement image-processing algorithms to recognize gestures, mimics, and gaze by means of a camera and to implement a so-called Virtual Personal Assistant (VPA), which is a human-like communication partner able to display natural communication forms such as speech, gesture, and mimics. We present a step towards this integration of input information, where a speech recognizer and a head tracker filter are the communication channels. These two input paths provide the system with the information used by the architecture organizing the VPA's behavior, which interacts with the user in a natural way based on its natural look and its natural behavior generated as sequences of logical and situation-dependent actions. Its behavior is organized my means of a scheme based on the theory of dynamical systems. This approach is also well known in robotics.

Behavioral Organization of the Output System for the Virtual Shop Scenario

For this concrete implementation scenario, a user in front of a screen wants to purchase items, and communicates with the VPA. The architecture organizing its behavior is able to carry out the following tasks:

- Displaying the 3-D virtual human head: a so-called human-like *softbot* [1] is displayed on the screen.
- Furniture presentation: furniture is shown to allow the user to explore it. The appearance of this environment, which builds the context of the dialog, will also be controlled by the behavioral architecture. See Figure 1 for a snapshot of the setup.

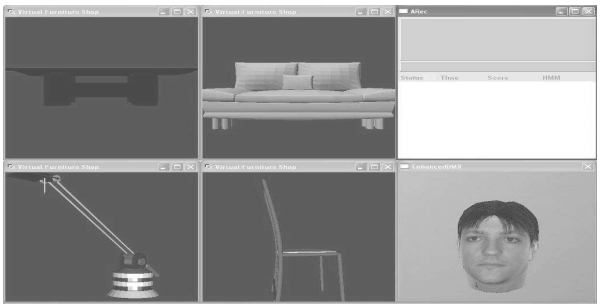

Figure 1: Screen shot of the virtual shop scenario: by organizing elementary behaviors like "speak", "listen", etc., the system can generate the complex behavior "presenting a piece of furniture to the user".

Two information channels are present at the system input: **the user's speech and head position**. They provide the VPA with the *sensor context information,* i.e. the abstract representation of the information coming from the sensors, which is used to organize the behavior of the VPA and coded by the variables s_i, $i=1, \ldots, N$. We have selected a set of N simple actions, the combination and sequential order of which represent a complex overall behavior. One of these complex tasks is *presenting a piece of furniture to the user*, for instance. This task can be divided into smaller elements, so-called *elementary behaviors* (EBs) such as speak (talking

about the piece of furniture), listen (waiting for multi-channel user inputs), etc. But how do we produce these complex sequences? After exploring the state of the art (see [2] for a detailed description), we chose the **Dynamic Approach to Robotics** [1][3][4][5][6], the architecture of which has several modules running in parallel and each of them establishes a continuous relationship among a set of qualities by means of differential equations. This approach provides with several advantages: one is the possibility to define the concept of "behavior", which can be associated to state variables of the dynamics. Hence, the overall complex behavior of the VPA is seen as a compound of "bricks" of behavior, the **elementary behaviors (EB)**, which are organized to produce **complex behavioral sequences**.

The task of a mechanism for behavioral organization is to activate/deactivate these EBs depending on the user's input and the state of the VPA such that a natural dialog is produced. Solving this task does imply a) to respect the logical interrelations coded in a matrix **L** between the elementary behaviors, b) that internal events of the system – namely activations/deactivations of EBs- can trigger activations/deactivations of other EBs, which is coded by another matrix **A**, and c) the sensor context has to be considered, the aforementioned s_i. The complex behavior is understood as a trajectory in a high-dimensional *behavioral space* spanned by the dynamical variables associated to the EBs. From the infinite set of different trajectories in the behavioral space, only a limited subset generates complex behaviors which an observer would call *logical* or *natural*.

The activity of every EB_i is represented mathematically by a variable n_i. We define that when $|n_i| \approx 1$, the EB_i is active, and if $|n_i| \approx 0$, inactive. The following equation is used to control its activity over time:

$$\tau_n \dot{n}_i(t) = \tau_n \frac{\partial n_i(t)}{\partial t} = \alpha_i n_i(t) - |\alpha_i| n_i^3(t) + \xi_i \quad (1)$$

The **state variable** n_i reflects the state dependency of the behavioral organization mechanism. This state changes over time on the time scale τ_n, controlled by the parameter α_i, which we assign to an external input. The term ξ_i is a small stochastic noise term needed to prevent the system from getting stuck in the repellers. Since there is a set of N elementary behaviors, only the behavioral patterns fulfilling certain logical requirements, coded by **L**, among the EBs are allowed. The activations/deactivations of an EB_i can lead to the activation/deactivation of EB_j, what is coded by another matrix, **A**. The use of these two matrices and the sensor inputs, represented by the variables s_i, allows us to produce the α_i-dynamics. The whole architecture can be summarized in a few words: EB_i becomes active/inactive when either its sensor context (s_i) or another related EB_j (**A**) votes for its activation/deactivation. Simultaneously all logical requirements (**L**) must be fulfilled. The entire architecture results surprisingly simple and has an abstract representation of all the sensor inputs (s_i), which allows us to integrate them within the system, since all the signals provided by the sensors are dealt with in the same way. Let us show some specific examples. In the actual implementation we have a couple of EBs called w_table and table, for instance. w_table indicates when the word "table" was recognized by the speech recognizer, meaning it codes the fact that the user wants to refer to this object. w_table gets activated by means of its sensor context s_{w_table}. If this activation happens, it triggers -due to the codes in **A**- the one of table, the EB indicating that the VPA is focused on displaying information about the table and the user has the possibility to explore it. The gaze output channel is used to provide the VPA with a more human-fashioned overall behavior. It has an associated EB, i.e. look_at_user that if active, makes the VPA look into the user's eyes. It can only be active while the VPA is listening to the user. This is coded by the EB listen which inhibits look_at_user when it is inactive by means of the logical inter-relationships in **L**. This way, while the VPA is waiting for some information from the user, it is looking into his/her eyes as in a human to human dialogue.

Conclusions & Future Work

We presented an MMI architecture, where two input channels are used to get a more natural dialogue with the user. The organization of the VPA's overall behavior is achieved by means of dynamical systems. The input channels produce the sensor signals which are associated the abstract representation we call the sensor context, which allows us to incorporate them into the behavioral architecture and use each input channel when recommendable to produce the VPA's behavior. In the future, more input channels will be included, like the user's pointing direction and lips movement and the set of EB for the VPA will be enriched.

References

[1] Software Agents. Seminar in New Artificial Intelligence Learning in Embodied Systems. 2001

[2] Self-calibration based on invariant view recognition: Dynamic approach to navigation. Robotics and Autonomous Systems 20 (1997), 133-156

[3] Generating Interactive Robot Behavior: A Mathematical Approach. From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior (2000), 135-144

[4] The dynamic approach to autonomous robot navigation. ISIE´97, IEEE International Symposium on Industrial Electronics (1997)

[5] Learning by doing: A dynamic architecture for generating adaptive behavioral sequences. Proceedings of the 2nd International ICSC Symposium on Neural Computation, NC´2000 (2000)

[6] An architecture for behavioral organization using dynamical systems. Abstracting and Synthesizing the Principles of Living Systems, 3Rd German Workshop On Artificial Life, GWAL ´98, 31-42

[7] Tracking Human Hand Movements by Fusing Early Visual Cues. 'Dynamic Perception', Workshop of GI section 1.0.4 "Image Understanding" and the European Networks MUHCI and ECOVISION (2002), 139-144