On Autonomous Behavior Control for Robotic Agent in Persuasive Technology Applications

Blagovest Vladimirov, NamShik Park and Hyun Kim

ETRI, 138 Gajeongno, Yuseong-gu, 305-700 Daejeon, Korea
{vladimirov, nspark, hyunkim}@etrise.kr

Abstract. Successful development of a robotic computer as a mediator in smart environments requires providing a certain level of behavior autonomy to the robot and a capability to adapt its behavior in long-term interaction with the users. We present our implementation of autonomous behavior control subsystem for a robotic agent and discuss its suitability for persuasive technology applications.

1 Introduction

In our work on the Future Robotic Computer (FRC) project [1], we explore the possibility of using a robotic computer as a mediator in smart environments. By building a robotic computer we aim at supporting flexible, adaptive information services with multi-modal, context-aware interactions. As an extension to the FRC's Software Framework for creating applications on the platform, we developed an Autonomous Behavior control Subsystem (ABS). In this paper we describe ABS and the possibility to support persuasive technology applications.

2 Behavioral Autonomy for FRC

The behavioral autonomy of a system is concerned with external behavior and is related to the stability and flexibility of the system's interactions with the environment [2]. In robot development, an important issue related to the behavioral autonomy is finding a balance between the robot's independence from human intervention and our ability to control it from outside to perform desired tasks or obtain desired behaviors. In this respect, Vernon [3] suggests useful guidelines for designing cognitive architecture that is autonomous and inherently trainable.

We are interested primarily in the role of robot's autonomy in supporting useful behaviors while relieving us from the need to specify every detail in advance. Therefore, we focus on behavior autonomy with relative independence from human intervention, while still retaining the ability to guide the behavior adaptation process. In ABS, we consider the following aspects of behavior autonomy: autonomous behavior execution; autonomous behavior selection; autonomous adaptation of existing behaviors; and autonomous initiation of behaviors.

3 Autonomous Behavior Subsystem Architecture

Cognition is necessary for behavior autonomy because it provides adaptive mechanisms for action selection based not only on past and present events but also on possible future consequences of the selected actions. The Hybrid approaches to cognitive system development attempt to combine the strengths of the other two groups - Cognitivist and Emergent approaches - so that we can retain the ability to supply the system with relatively advanced initial knowledge and rely on the system's capabilities for adaptation and self-development in the process of interaction with the environment for further tuning of the desired behavior [4]. In the ABS design, we follow the general ideas of Hybrid approaches [5], using a dual-level architecture with both connectionist modules and symbolic rules. The interactions among these two levels, including top-down learning and bottom-up rule extraction, support the desired ability to specify some initial behaviors and later to adapt or refine them through appropriate interaction with the robot.

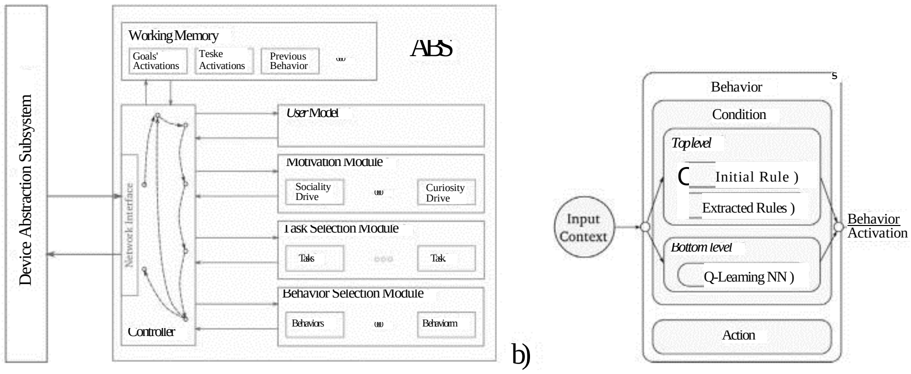
a) 

Fig. 1. a) ABS structure, communication among modules, and external communication with the Device Abstraction Subsystem. b) Dual-level Behavior representation in ABS.

The ABS structure, the internal communication among its modules and the external communication with the Device Abstraction Subsystem (DAS) are shown in Fig. 1 a). A brief description of the main steps in ABS's operation is as follows. Initially, ABS receives a sensory event from DAS. This sensory event is augmented with information maintained in the Working Memory (WM) to form the current input context. The current input context is sent to the User Model (UM) and the user preferred service in the current situation, is added back to the input context. The augmented input context is sent to the Motivation Module (MM) to update the internal drives' and goals' activations, and the changes are reflected back in the input context. The Task Selection Module (TSM) selects a task based on the input context, updating back the active task part of the input context. Finally, the behavior

Selection Module (BSM) selects the appropriate behavior for execution. The action specified by the selected behavior is sent to DAS, while simultaneously the WM is updated with information necessary for the system's functionality in the future.

3.1 Behavior and Task Selection

The primary purpose of the BSM is to select an appropriate behavior in a given context. In addition, BSM supports learning of behavior selection rules and adaptation of rules specified in advance based on reward signal computed by MM. In ABS, we view behaviors as situated actions, where actions are basic primitives specified and implemented in DAS. Consequently, as shown in Fig. 1 b), behaviors have one component specifying the action with its parameters and a dual-level structure specifying the conditions for executing that action. The top level of that structure consists of an initial symbolic rule and an optional collection of candidate rules extracted from the bottom level. The bottom level consists of a multi-layer neural network trained with Q-Learning [6].

The behavior's activation is computed as a weighted, linear combination of the maximum rule activation from the top-level and the estimated Q-value. The behavior selection is performed by comparing the behaviors' activations and selecting the maximum activated behavior following an s-Greedy policy.

Behavior adaptation is provided by optional learning processes in the top level, through maintaining performance statistics for each rule, and in the bottom level, through Q-learning based on reward signals from the MM. Top-down learning can be performed by using only the top-level rules to compute the behavior's activation and at the next step using the obtained reward to train the bottom-level network. Currently, we use a simple approach for bottom-up rule extraction, where if the reward is above a pre-specified threshold, the input context used to compute the behavior's activation is transformed into a new candidate rule condition.

The TSM supplements BSM in implementing the required ABS functionality to provide behavior selection and adaptation capabilities. For each task there are task-control behaviors with fixed actions for starting, suspending, resuming, and stopping the task, which modify the task-activation state in the WM. The task selection functionality is provided by appropriate selection of task-control behaviors in a manner similar to BSM.

3.2 Motivation Module, User Model, Working Memory and Controller

The MM supports autonomous behavior initiation and behavior adaptation functionality by providing internal state that can be used in the behavior conditions to trigger or to inhibit behaviors, and by computing reward signals used in the behavior conditions' learning process. This is achieved by implementing internal drives' models and a mechanism for computing reward signals from drives' activations and activation changes.

Currently, in ABS we have implemented two main internal drives: a *Sociality drive* and a *Curiosity drive*. The Sociality drive gives the users a mechanism for influencing the behavior adaptation process effectively, while providing for a certain level of autonomy. On the other hand, the Curiosity drive makes exploration-based behavior adaptation possible even without user intervention.

The UM learns users' preferences from interaction. When a user requests some service (internally represented by a task), the UM associates the requested service with the perceived current context. With time, the salient associations are used to extract explicit rules. A detailed description of the UM is given in [7].

The WM maintains relevant information necessary for setting drives' and goals' activations, suggesting services, and selecting tasks and behaviors.

Through the interaction with DAS, the Controller provides the capabilities for collecting sensory information and for behavior execution.

4 Implementation

The current implementation of FRC includes an Agent Unit shown in Fig. 2 and a server. DAS is a part of the FRC's Software Framework implementation ICARS (integrated control architecture for robotic mediator in smart environments) is described in [8]. The ABS is implemented in C++ and the Controller communicates with ICARS over a TCP connection.

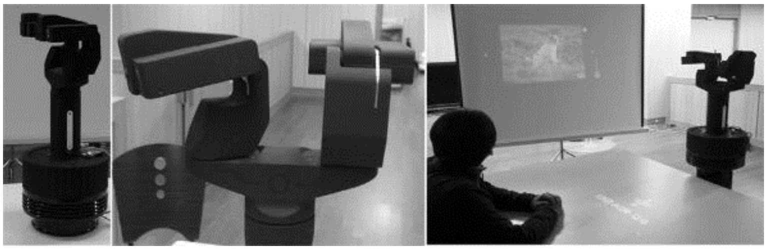


Fig. 2. The FRC's Agent Unit.

5 Possibilities for Persuasive Technology Applications

In addition to providing information services, using FRC in Persuasive Technology [9] applications opens many new possibilities for helping the users. We are interested in applications that assist the user's efforts to adopt attitudes and behaviors leading to healthier and more productive lifestyle. FRC could be used to make the target behavior easier to perform, to provide cues in the right time and in the right situation, to keep record of past performance, to help the user monitoring himself for inadvertent performance of undesirable behavior (e.g., using parasite words), etc.

The FRC's capabilities to acquire context information, to present information in a flexible manner (e.g., overlaying projections on everyday objects), and to adapt its behavior make it suitable for augmenting the explicit interactions with the users, in the process of providing a given service, with implicit interactions [10] that target complementary goals. For example, FRC could observe the user's posture while he is reading a book projected on the table. If for a certain period of time the user is maintaining a bad posture, FRC could try to shift slowly the projection to a different location eventually prompting the user to change his posture. The important aspect here is to try to achieve this interaction naturally, without interrupting the user's main task (reading the book). In the ideal case this could be achieved without the user noticing it. Such application depends on the robot's behavior adaptation capability to adjust its actions based on the user's response (positive: user maintaining a good posture; or negative: user saying "Keep still!"). Using such implicit interaction, the user could achieve his goal to maintain a good posture while concentrating on his main activities.

References

1. Kim, H., Suh, Y.-H., Lee, K., Vladimirov, B.: Introduction to system architecture for a robotic computer. In: Int. Conf. Ubiquitous Robots and Ambient Intelligence (URAI), pp. 607-611, Nov. (2011)
2. Froese, T., Virgo, N., Izquierdo, E.: Autonomy: a review and a reappraisal. In: Proc. of 9th European Conference on Artificial Life, Berlin, Germany, (2007)
3. Vernon, D.: Reconciling autonomy with utility: A roadmap and architecture for cognitive development. In: Proc. of Int. Conf. on Biologically-Inspired Cognitive Architectures, pp. 412-418, IOS Press, (2011)
4. Vernon, D., Metta, G., Sandini, G.: A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. IEEE Trans. Evolutionary Computation, vol. 11, no. 2, pp. 151-180, (2007)
5. Sun, R.: The importance of cognitive architectures: An analysis based on CLARION. J. Experimental and Theoretical Artificial Intelligence, vol. 19, no. 2, pp. 159-193, (2007)
6. Rummery, G. A., Niranjan, M.: On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, University of Cambridge, Cambridge, England, (1994)
7. Koo, S.-Y., Park, K., Kwon, D.-S.: A dual-layer user model based cognitive system for user-adaptive service robots. In: 20th IEEE International Symposium on Robot and Human Interactive Communication (Ro-Man 2011), pp. 59-64, (2011)
8. Suh, Y.-H., Lee, K.-W., Lee, M., Kin, H., Cho E.-S.: ICARS : Integrated Control Architecture for the Robotic mediator in Smart environments A Software Framework for the Robotic Mediator collaborating with Smart Environments. (submitted for publication)
9. Fogg, B. J.: Persuasive Technology: Using Computers to Change What We Think and Do. Morgan Kaufmann, San Francisco, (2003)
10. Schmidt, A.: Interactive Context-Aware Systems--Interacting with Ambient Intelligence. Ambient Intelligence, G. Riva et al., eds., IOS Press, pp. 159-178, (2005)