Realizing Seamless Interaction: a Cognitive Agent Architecture for Virtual and Smart Environments

Youngho Lee, Hedda R. Schmidtke, Youngjung Suh, and Woontack Woo
GIST, U-VR Lab.
Email: {ylee,schmidtk,ysuh,wwoo}@gist.ac.kr

***Abstract*— We propose a cognitively motivated vertically layered two-pass agent architecture for realizing responsiveness, reactivity, and pro-activeness of smart objects, smart environments, virtual characters, and virtual place controllers, that is controllers for lighting and weather conditions. Being cognitively motivated, our approach aims to respect the cognitive demands of a human being, in order to deliver an adequate human-computer interface to ubiquitous and virtual environments. The vertically layered two-pass architecture allows to realize reaction within a certain time frame, as necessary for generating responsiveness and reactivity required for natural interaction, while at the same time providing a layer for further processing of information for creating pro-active, intelligent responses.**

I. Introduction

A key problem for realizing ubiquitous virtual reality is seamless interaction between objects in real and virtual space. Seamless Interaction is an interation when a user accesses and customizes objects seamlessly, and experiences responses of them in real and virtual environments. To achieve this goal, we need to consider physical synchronization, contextual synchronization, and user-awareness. If the real and virtual objects are not registered, users in a real environment have difficulties to manipulate them. Contextual information such as time, location, lighting condition should be matched in both real and virtual environments. User's information affects responses of objects in real and virtual environments.

Agent architectures have been used successfully for modelling virtual characters [3]. Our motivation for constructing four types of applications – virtual characters, smart objects, virtual place control[1], and smart spaces – in the same framework, was that in each case the user's cognitive demands determine three crucial time frames. *Direct responsiveness* is bound to the time frame of visual continuity. A mouse cursor, for instance, should move without any perceivable delay. Likewise the movement of virtual characters or visualization of gestures has to stay within this time frame. The threshold for imperceivable delay we chose, was the threshold of 40ms used in animation movies, which corresponds to generating 25 frames per second. *Immediate reaction* to a user's commands, on the other hand, can take more than 40ms. However, in human-human interaction, we usually require some response from a dialog partner within the time frame of a second, even if it is just a nod or 'hmm' for signalling demand for a larger time frame. *Pro-activity* is needed when a computer system schedules events, or when a human being constructs a piece of work. This may require any amout of time: five seconds, a minute, or a day.

In ubiquitous computing two main types of tasks have to be handled: *Context Integration* is the task of deriving aggregated and abstracted information about a situation from sets of singular data, particularly, sensory data and facts; *Context Management* is the task of invoking services/actions appropriate in a context with appropriate contextual information.

Architectures that fullfil the above time requirements have been investigated in the domain of artificial intelligence and within the agent-oriented programming paradigm. Wooldridge summarizes several advantages in comparison to object-oriented programming, which are crucial for realizing ubiquitous computing applications [1, p.10f]. Objects have only control over their state, but not over their behaviour, as methods are called from the outside, i.e. by other objects. Agents, in contrast, act according to some request, and can thus, for instance, include refined strategies for avoiding security issues. The second point Wooldridge discusses is the higher flexibility and autonomy of agents. Agents can be reactive, pro-active, and even social for use in multi-agent systems. Only the first two points are currently covered in our architecture. Social knowledge and strategies can be encoded in our framework as special types of knowledge and strategies. The third point compared in [1] is that agents in contrast to object are always executed concurrently, i.e. that each agent in a multi-agent system has its own threads.

II. A Cognitive Agent Architecture

We propose a vertically layered two-pass architecture with three layers, as shown in Fig. 1. The main benefit of using a two-pass architecture is that we can get immediate reactions to sensory input while still performing further processing and accumulation of knowledge.

- A virtual character might run from point A to point B, in order to retrieve a certain object at B. While running, it will need to do low-level actuator control, be prepared for immediate reaction to unforseen situations, such as obstacles on its path. And at the same time, it might already try to make a plan how to obtain the object.
- A smart calendar with a camera and a display might track the hand of a user for gesture recognition and directly

This research was supported by the CTI development project, MCT and KOCCA in Korea, and by the UCN Project, the MIC 21C Frontier R&D Program in Korea.

[1]Under this notion, we summarize all kinds of controllers for the virtual environment, such as control of lighting and weather conditions.

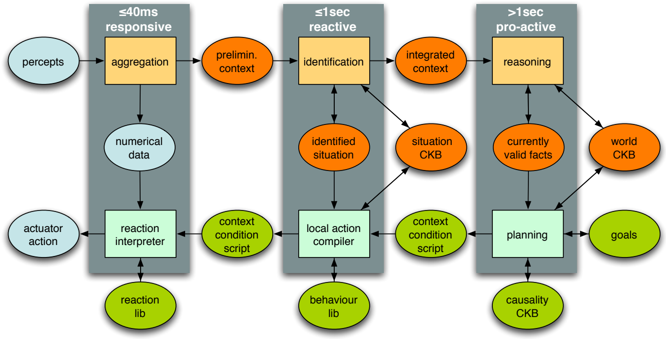

Fig. 1. A vertically layered two-pass agent architecture for virtual characters, virtual place control, smart objects, and smart spaces. Although by definition according to [2] the architecture is a vertically layered architecture, the layers are here drawn horizontally, with the bottom layer (input and output via sensors and actuators, respectively) on the left.

visualize each movement, while still accumulating the gesture. In the background, it might be executing complex functions, such as scheduling.

The agent architecture thus helps developers of virtual characters, virtual place controllers, smart objects, and smart spaces to provide natural responsive interaction, reactive beaviour, and pro-active capabilites.

In our layered architecture, *context integration* is a step-wise process of abstraction (shown in the upper row of Fig. 1). Contextual information can be integrated from low-level sensory information, to object-based and situation-based knowledge, and finally to high-level, situation-independent logic-based knowledge:

1) On the level of responsiveness, the intelligence of smart sensors can be implemented: an *aggregation component* would be able, for instance, to identify positions in successive frames for object tracking, or to aggregate basic coordinate data and sensor values with clustering techniques into object regions.
2) On the reactive level, basic situation and object recognition can be implemented: *identification* includes recognizing gestures from hand movements, i.e. matching shapes and other sensory input to objects and states of objects stored in the object-oriented knowledge-base or recognizing situations from a particular layout of objects or a particular succession of events
3) On the pro-active level, facts about the world are extracted from the constant flow of situations and objects detected at the lower level: *reasoning* is required, so that an agent can keep track of changes occuring in the world, generate representations of the current state of the world at varying levels of detail, and identify courses of events and their possible outcomes to avert disasters and reach goals.

In order to invoke services and actions appropriate in a context with appropriate contextual information, not only the *external*, sensed context has to be handled but also *internal*, contextual information, such as the status of the system with respect to fulfilling a user's more long-term goals. Accordingly, *context management* is performed as step-wise refinement and realization of plans (shown in the lower row of Fig. 1): from generating local action plans in response to a user's overall goals, to generating concrete instantiations of those actions within a local context, to translation of these actions to actuator responses appropriate to a dynamically changing world.

1) On the pro-active level, goals of an agent, facts about the world, and rules of causality are employed to generate plans, i.e. short parameterized sequences of commands: *planning* is required, so that an agent can solve complex problems, and achieve goals involving currently not existing objects, and situations.
2) On the reactive level, condition-based action triggering can be implemented: *local action planning* means linking parameters in an action plan to objects and situations of a context and triggering the next action in a plan based on identified parameters and preferred behaviours.
3) On the level of responsiveness, the intelligence of smart actuators can be implemented: a *reaction interpreter* would be able to generate involuntary actuator response directly from perceptual input, to generate voluntary actuator response by interpreting a local action, and to interpret the local action with respect to continuously updated coordinates retrieved from perception, such as tracking data.

Three types of representations are transmitted in the system. *Numerical data* are produced by sensors and consumed by actuators. *Context Representations* reflect knowledge about the current state of a local environment: what is happening *here* and *now*? *Context Conditions* are rules encoded as condition-action pairs. The granularity of context parameters is not fixed, that is *now* (the current state) can have the duration of a fraction of a second, it can mean *today*, *this weak*, or *this year*. Likewise *here* can mean *this room*, *this city*, *this country*.

III. Conclusions

We proposed a cognitively motivated vertically layered two-pass agent architecture for realizing responsiveness, reactivity, and pro-activeness of smart objects, smart environments, virtual characters, and virtual place controllers. Being cognitively motivated our approach aims to respect the cognitive demands of a human being, in order to deliver an adequate human-computer interface to ubiquitous and virtual environments. The vertically layered two-pass architecture allows to realize reaction within a certain time frame, as necessary for generating responsiveness and reactivity required for natural interaction, while at the same time providing a layer for further processing of information for creating pro-active, intelligent responses.

References

[1] M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems*. MIT Press, 1999.

[2] M. Wooldridge. Intelligent agents: The key concepts. In V. Marík, O. Stepánková, H. Krautwurmova, and M. Luck, editors, *Multi-Agent-Systems and Applications*, volume 2322 of *LNCS*, pages 3–43. Springer, 2001.

[3] S.-Y. Yoon. *Affective Synthetic Characters*. PhD thesis, Massachusetts Institute of Technology, 2000.