Attention Based, Naive Strategies, for Guiding Intelligent Virtual Agents

Damien Clauzel[1], Claudia Roda[2], Laurent Ach[3], and Benoît Morel[4]

[1] American University of Paris, Paris, France, dClauzel@AUP.fr
[2] American University of Paris, Paris, France, cRoda@AUP.fr
[3] La Cantoche, Paris, France, lAch@Cantoche.fr
[4] La Cantoche, Paris, France, bMorel@Cantoche.fr

Abstract. Guiding the behaviour of Intelligent Virtual Agents so that a pleasant and productive interaction is established with the user has been the aim of a very large body of research work. In this paper we propose that by basing this guidance on a set of attention indicators it is possible to define strategies to effectively guide IVA's behaviours.

Keywords: learning, attention, motivation, avatar, intelligence, interaction, reasoning, intelligent virtual agent, interface agent

1 Introduction

One of the objectives of e-learning platforms designers is to provide interfaces that better support learning and collaboration. This paper discusses the interface design issues related to the implementation of an Intelligent Virtual Agent (IVA) in the AtGentive project[1,2]; AtGentive focuses on the support of attention in learning environments. To achieve this objective the system analyses the learners' computer activities and physical states and, on the basis of this analysis, it generates interventions. Such interventions either supply learners with information useful to support their current attentional focus, or are aimed at attracting the user's attention to new foci.

In this paper we briefly describe how the attentional indicators evaluated to support learner's attentional choices are also used to implement simple strategies to guide the behaviour of IVAs.

The AtGentive system communicates with the user using a wide range of modalities including an IVA controlled by an external Reasoning Module (the system's component generating the interventions).

The IVA's behaviour is guided by parametrized scripts. The scripts' parameters (mood, strength) are computed by the Reasoning Module based on models of the users and of the tasks they are performing.

In section 2 we describe the overall organisation of the AtGentive system. Section 3 introduces the guiding strategies for the IVA's reasoning and behaviour, and consequently how and when interactions with learners are computed. Details about the IVA, its behaviour according to the instructions it receives, and its internal components, are discussed in section 4. Finally, section 5 concludes and introduces future work.

2 The Reasoning Module

Within AtGentive systems, IVAs are piloted by an external component, the Reasoning Module. The Reasoning Module is in charge of processing the data sent by the application and other external components in order to produce interventions aimed at supporting users' attentional processes These interventions are used for providing suggestion to the application and triggering the IVA (see figure 1).

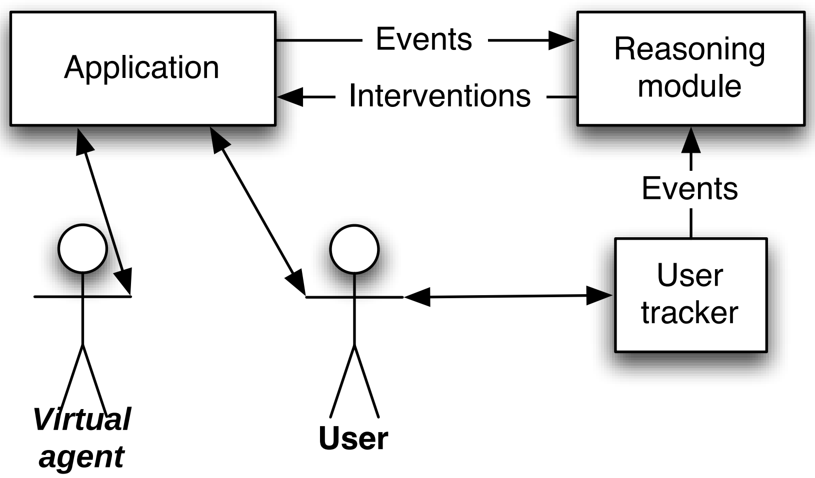

Fig. 1. General communication between the components

This separation between the brain and the body of the IVA allows externalizing the reasoning behind the virtual agent into a separate software component. Then by rendering it generic and pluggable into any application, we are able to propose a general purpose tool for controlling the virtual agent by customising the reasoning process. The multi-agents[5] approach chosen to develop the Reasoning

[5] Note that in this paper we use the names Intelligent Virtual Agent (IVA), or virtual agent, to refer to the physical agent present on screen. We use instead the name

Module allows us to easily extend the reasoning module by adding new agents, to customize it by creating several types of agents (rule based, neural network based) and even to externalize some of its components. The agents implementing the Reasoning Module can serve different purposes; in this paper we focus on the reasoning allowing the system to decide how and when to make suggestions to the learner and consequently guide the virtual agent.

Currently, each agent uses a rule-based engine for processing the information about the environment and the user activity. Agents select the information they want to receive by registering with a dispatcher; for example one agent may register so that it will be informed when the user has started a new activity, or when the user has received an email, or when the user has been inactive for a certain length of time.

Each agent may generate user foci. For example, one agent may suggest that the user reads a newly received email because it is relevant to the user's current task, whilst another agent may suggest that the user performs a task that may help him complete a task with which he is having difficulties. Agents associate to each focus a certainty level and a priority. For example, when suggesting a help task to the user, the certainty level of the proposal may be a function of the number of possible alternative help task, and the level of appropriateness of the help task for the specific user. The priority of the intervention would instead be related to the urgency of the task for which help is provided.

With these tools the system may represent a situation in which one knows that a learner needs some help to complete an urgent task but is uncertain about what help to provide.

When all agents have processed the information, the global result is optimized by an integration agent capable, for example, of removing duplicates, merging similar items, selecting amongst incompatible suggestions, removing foci that are not feasible for lack of resources, etc. All the reasoning agents are part of the same global process, each one of them adding its own contribution to the general reasoning. Some agents only work on a small subset of the information while some others do transversal reasoning.

The event feeding the agents' reasoning may be generated by the application, the tracking devices, or the learner. The application may generate events when the learner reads an email, start to work on an exercise or is chatting with somebody. The tracking devices monitoring the user physical state and activity may generate events describing the user keyboard activity, the level of noise in the room, or the presence (or absence) of the user from the screen. The learner may generate events by indicating his desires (e.g. being notified about something, not being interrupted. . .), or by giving direct feedback on the system's interventions.

"agent" to refer to the software agents (invisible to the user) implementing the Reasoning Module.

By reasoning on these events, the agents maintain an user model, and generate interventions. In order to generate interventions the agents must address three main issues. First, obviously, the system must decide what to say to the user (content aspect), secondly it must decide when to say it (temporal aspect), and finally it must decide how to say it (modality aspect). The latter two aspects, which are most directly related to guiding the IVA's behaviour, are discussed below.

3 Guiding strategies

Several studies have demonstrated that human's feelings towards virtual agent's interventions play an important role on whether the help and suggestions will actually be taken in consideration by the user. Supplying useful information is a necessary but not sufficient condition for a successful interaction. A study conducted on the interventions types generated by the AtGentive system suggests that, in order to establish productive interactions, the IVA should be likeable and offer advice that is timely and believable[3]. This section briefly introduces the strategies applied by the AtGentive system in order to achieve these objectives.

AtGentive systems select the mode of the intervention depending on the desired impact on the user's current attentional state. AtGentive systems currently interact with the user through text areas, pop-up message, sound (pre-recorded MP3), or a virtual agent. AtGentive may use several modalities at the same time (for example, since children cannot read text messages very easily, such messages are also spoken when interacting with children). The AtGentive system chooses the mode of intervention depending on the user activity and the information that needs to be communicated.

The Reasoning Module's agents make the decisions necessary to guide the IVA behaviour on the basis of a set of naive strategies aimed at making the learner's interaction with the agent more pleasant and productive. We refer to these strategies as naive because, for their implementation, we rely on simple rules. Having such simple rules is possible because the AtGentive system, being designed to support attention, natively provides a set of attention indicators on which these rules can be based.

These strategies are respectively aimed at obtaining: Trustworthiness, Politeness, and Appropriateness.

3.1 Trustworthiness

Trustworthiness requires that the learner perceives the agent as being worth of trust. In the AtGentive system trustworthiness is associated to the reliability of the value to the user of the IVA intervention.

On the basis of this strategy, the first criteria for establishing whether an intervention should be presented by the IVA or through other communication modes is the certainty level of the focus being proposed: only foci with high certainty level are proposed to the user via the IVA. The second criteria for selecting an IVA as a mode of communication is that the related focus should have a medium to high priority. This also ensures that, the IVA being a very intrusive communication system, the user is distracted from his/her current activity only if the communication is timely as the IVA is used for providing important and certain information.

3.2 Politeness

Politeness requires that the learner perceives the IVA as being respectful for his activity. In our model politeness requires that the IVA chooses the most appropriate time to interact with the user.

For doing this, the AtGentive system relies on "Breakpoint" events. Breakpoint events represent a shift in the user's attention, meaning that the learner is mentally available for receiving information; breakpoints can be graduated according to their strength. For example, when the user is reading a document spanning across several web pages, changing from one page to another modifies the attention of the user; also, the time between a user completing a task and starting the next one represents a breakpoint; the latter breakpoint being stronger than the former.

By defining a graduation for the breakpoints, we are able to adapt behaviour of the IVA to the user's activities so that IVA's intervention induce minimal cognitive load.

Breakpoint events may be generated by the application, or they may be internally generated by the Reasoning Module. The application is able to detect some modification of the user's attentional state by looking at his activity, and to quantify this change (for example, the user will be more mentally available when he finishes a task than when he changes to another web page).

Breakpoints may be generated by the Reasoning Module by observing how the user moves through a tree-based task structure (this methodology is inspired by [4,5]). When the user completes a task closer to the root of the task hierarchy – e.g. the user has just finished sending an email – agents may generate stronger breakpoints than when the user completes a task that is lower in the task hierarchy – e.g. the user has written the subject field of an email.

A further strategy used by the reasoning module to establish breakpoint situations is inspired by the work of Fogarty [6] and is based on sensory observation. In AtGentive, when the Reasoning Module is informed that a task is started, it may activate a keyboard-and-mouse tracking device that will inform it if the user is inactive longer than a given amount of time. Such inactivity is also interpreted as a breakpoint. In all situations in which there is no particular urgency to communicate with the user, AtGentive's IVA only intervene when there is a breakpoint in the user activity.

3.3 Appropriateness

Appropriateness requires that the learner perceives the IVA as being appropriate with respect to the situation, this entails that the IVA should place the correct emphasis on the messages it transmit. The appropriateness of presentation is controlled through two parameters: mood and strength.

The apparent mood of the IVA is altered in an attempt to match the rhetorical value of the intervention and enable processes such as positive re-enforcement or empathy. Current possible mood specifications are: angry, sad, neutral and happy. Moods are calculated as a function of the content of the intervention and of their level of certainty. The following table exemplifies some of the moods specifications:

Intervention Content	**High Certainty**	**Medium Certainty**
read email	happy	neutral
congratulate	happy	happy
re-attract attention	angry/neutral	sad/neutral
propose help	happy/neutral	neutral

Fig. 2. Sample modalities for interventions

The strength of the intervention is a generic modality and is not tight to the virtual agent. For text-based interventions the strength is *translated* in italic or bold font; in the case of a virtual agent it is translated by using an appropriate animation. As a general rule, the higher the priority or the relevance of the intervention, the strongest is the strength. For example, reading an email from the learner's teacher might be proposed in a strong manner because this information is generally important, but when the learner is actively working on a task and that the information is not directly relevant to what he is doing, AtGentive would postpone the intervention or propose it in a weak fashion by using a gentle animation not interrupting the learner.

4 The IVA

4.1 Presentation of the IVA

The purpose of the IVA Module is to receive and process requests for IVA interventions according to a previously defined general intervention model and to adapt these interventions to different environments and graphical contexts. This module is integrated in the application in a way that clearly separates the content of IVA's generic interventions from application processes, and also separates the necessary references to graphical user interface from the generic intervention data (see figure 3).

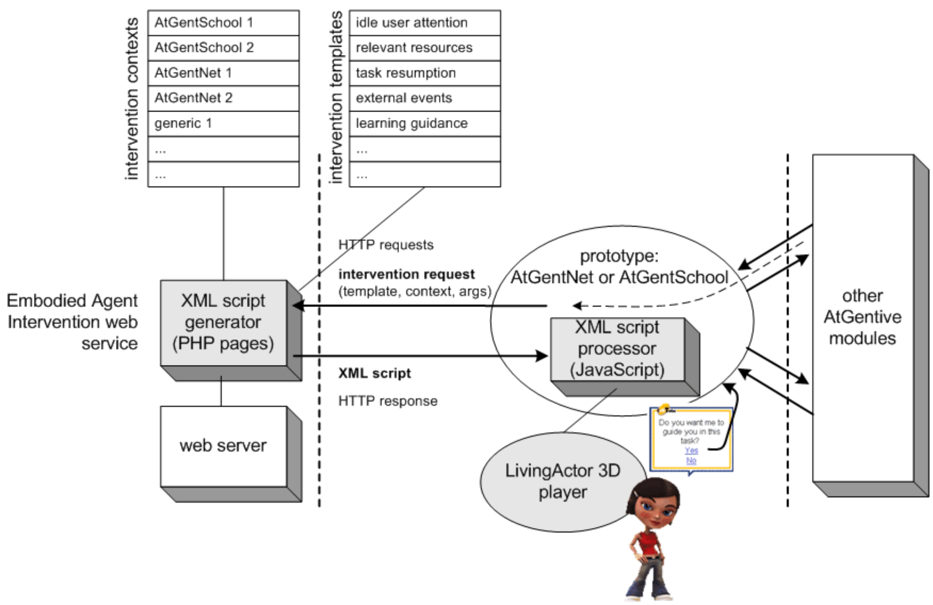

Fig. 3. Architecture of the virtual agent module

Two components constitute the AtGentive Embodied Agent Module:

– The script generator: produces a page describing an Embodied Agent intervention
– The script player: is responsible for playing the script in the browser

The Script Generator.

An intervention of the embodied agent is described by an XML script (or template script). It is composed of speech, animations, expressions and other actions. Any action in the script may be assigned a name (as an XML attribute) so that it can be changed from a context script (another XML file). Context scripts tell to replace some actions in template scripts by others, according to parameters mood (of the virtual character) and strength (of the intervention).

The process consists of:

- replacing some of the sequence actions according to the given external parameters: these parameters may come directly from intervention request arguments or from a context specified as argument in intervention request
- choosing an animation in the category specified in the template script: the choice is random and avoids having too much the same animation in successive interventions

The replacement consists of matching the names of XML elements in the template script with the names of parameters and then replacing the default values by the values from these parameters. The default values are simply the content of elements in the template script.

This mechanism allows choosing between generic or specific actions for each element of a template script like the name of an expression, text or audio data for a speech, the name of an animation, the name of a function to be triggered by a user event or the name of a graphical object for character positioning.

The resulting script is a fixed sequence of actions described in the returned XML page. This file is directly playable by the script player to be integrated in the platform as an embedded part of Embodied Agent module.

The Script Player.

The script player is implemented in JavaScript and encapsulates Living Actor™ 3D player, which is an ActiveX component. It triggers the actual intervention of the Embodied Agent according the XML script given as argument.

Such a script is called a fixed script because all actions it contains are to be performed directly without any change. All actions from the sequence described in the script are performed by through the script player, and when some interaction is requested with users, their actions trigger callback functions allowing the embedding application to manage the consequences of user choices.

References to callback functions are provided to the Script generator through their names, the same way as any parameter, whether as direct arguments or through context files. Users may click on links in Embodied Agent dialogue balloon and this generates a callback to a function associated with the link through the script.

4.2 Animation Selection

The current implementation of Living Actor™ scripting language allows commands telling the character to speak (with a specified text or audio file), to play a predefined sequence of actions (toward a specified target state), to show a particular item in the graphical user interface, and to perform other tasks at this same level of abstraction.

Embodied intervention model are defined from a set of template interventions corresponding for instance to event notification, navigation support or attracting user attention. These templates are matching a conceptual intervention model for embodied agent, which derives from the general intervention model. In addition, all interventions are taking into account the context in which they occur and several parameters specified in intervention requests.

4.3 Controlling the Embodied Agent

Intervention templates consist in small linear scenarios described using a XML base language or script code. Parts of these scenarios have variable parameters with values depending on particular instances of intervention and also possibly depending on random processes.

The random part of the scenario is an important point to avoid repetition in embodied agent intervention, as it aims at behaving partly like a living character (though human characters often repeat themselves). However, to preserve the quality of character animations, only the choice between a set of possible actions will be made at random and these actions will be visually compatible with rest of the sequence. The smooth transitions between actions are guaranteed by Living Actor^{TM} character production line.

4.4 Production of Script and Context Files

Scripts are written directly using any text editor and corresponding interventions can be visualized through a simple web page including the Script player.

Contexts are used to adapt intervention templates to particular situations and environment but they can also be used to choose between different modalities. For instance a context could correspond to a situation where the Embodied Agent makes a rude intervention and another one could correspond to the same intervention with more pleasant manners. A similar approach could be to distinguish between low and high priority contexts or between other types of intervention modalities.

Different contexts may be used for the same template script, as they are designed for this purpose. But also, the same context may be used for several template scripts just by concatenating parameters from the different scripts.

5 Conclusions and Future Work

Attention indicators play a very important role in any type of interaction (from the guide of turn-taking in conversations, to the deliberation on rhetorical value of future statements in natural language).

We have briefly described how attention indicators are used in the AtGentive system to guide the behaviour of an IVA. We have identified three major strategies for guiding the behaviour and indicated how these strategies are implemented in our system both at the level of reasoning and at the level of the IVA design.

We are currently in the process of evaluating a first prototype of the system in a learning environment for children. We expect that the results of this evaluation will allow us to refine the rules piloting the definition of the parameters guiding the IVA's behaviour. We are particularly interested in assessing whether our strategies are in fact sufficient to produce pleasant and productive interactions.

References

1. Roda, C., Nabeth, T.: The atgentive project: Attentive agents for collaborative learners. In: First European Conference on Technology Enhanced Learning EC-TEL'06. (2006) Crete, Greece: Springer.
2. : The AtGentive project (2005-2007) http://www.AtGentive.com. Partially sponsored by the European Community under the FP6 framework project IST-4-027529-STP.
3. Rudman, P., Zajicek, M.: Autonomous agent as helper - helpful or annoying? In: IAT 2006 - IEEE/WIC/ACM International Conference on Intelligent Agent Technology. (2006) Honk Kong.
4. Bailey, B., al: A framework for specifying and monitoring user tasks. In: Computers in Human Behavior. (2006) 709–732
5. Bailey, B., Konstan, J.: On the need for attention aware systems: Measuring the effects of interruption on task - performance, error rate, and affective state. In: Computers in Human Behavior. (2006) 685–708
6. Fogarty, J., al: Predicting human interruptibility with sensors. In: ACM Transactions on Computer-Human Interaction. (2005) 119–146