Social Agents for Serious Games

(Extended Abstract)

Joost van Oijen
Utrecht University
PO Box 80.089, 3508 TB
Utrecht, the Netherlands
J.vanOijen@uu.nl

Frank Dignum
Utrecht University
PO Box 80.089, 3508 TB
Utrecht, the Netherlands
F.P.M.Dignum@uu.nl

ABSTRACT

In this paper we propose the use of a *shared* social context to facilitate agent perception of social dynamics within a virtual environment. We argue that through a shared context it can be more feasible to equip agents with social behaviors and reasoning rules for social intelligence, better suited for real-time constrained serious games with multiple agents.

Categories and Subject Descriptors

I.2.0 [**Artificial Intelligence**]: General—*Cognitive Simulation*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent Agents, Multiagent Systems* ; I.6.3 [**Simulation and Modeling**]: Applications

General Terms

Design, Human Factors

Keywords

Intelligent Virtual Agents, Middleware, Social Perception

1. INTRODUCTION

Social reasoning for agents in a real-time (virtual) environment often depends on a rich social context including a good understanding of the environment and social activities of other agents [1]. Composing a social context fully autonomously involves computationally expensive inference procedures concerning environment perception, action recognition and intent recognition. In this paper we propose to employ a partly *shared* social context to facilitate agents in performing such procedures in order to keep their behavior both socially realistic and also efficient. Since actions and intents are already assumed to be available within an actor agent, we make use of this information by allowing other agents to use it directly in their perception process. The shared social context is managed by a middleware coupling multiagent systems (MASs) and game engines.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

2. A MIDDLEWARE APPROACH

Figure 1 illustrates the architectural design of a middleware approach for managing a *shared social context* for virtual agents whose design is distributed among a MAS and a game engine. Concerning the middleware, the vertical layer provides an inter-process communication mechanism for agents to communicate with their embodiment. The horizontal layer introduces a social layer that offers an information source for agents to directly read aspects of the social context, hereby eliminating the need for individual agents to construct a similar context fully autonomously. Communication with the social layer is regulated through publish/subscribe mechanisms. Ontologies are used for specifying concepts employed in the shared social context. Employed models within the social layer are described next.

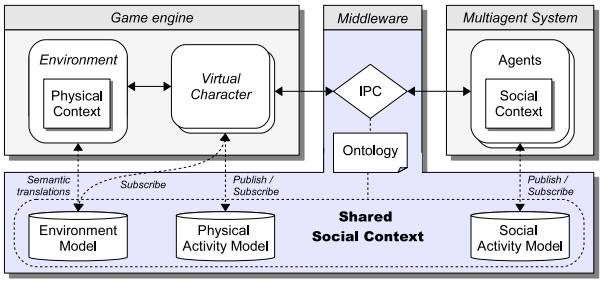

Figure 1: Middleware Approach

3. SHARED SOCIAL CONTEXT

Social Environment Model This model represents the virtual environment defined at an abstraction level at which agents can reason efficiently based on meaningful concepts. It is built on ontologically-grounded concepts representing environment objects, properties and events. Semantic translations are required to convert data from the game engine's game state to semantic concepts (e.g. a collection of fire particles translates to a *fire* concept). The benefits of managing this model globally is that *semantic processing* can be optimized by sharing translations between multiple agents.

Physical Activity Model This model comprises all currently active physical actions of all embodied agents. Here an action represents some bodily movement of a virtual character like locomotion, grabbing an object, performing a gesture or speech action. In this model, agents can be facilitated in performing *action recognition* where they can be notified about the physical actions of others. Without such explicit information, agents would individually need to infer these

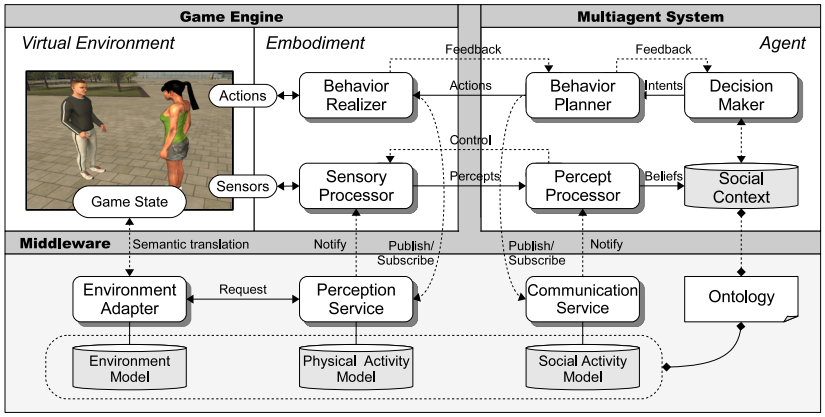

Figure 2: System Implementation

actions themselves, which would involve inspecting their perceivable part of the game state over time while searching for patterns that may represent some physical action.

Social Activity Model Similar to its physical counterpart though now at the cognitive level, this model contains a collective overview of all agents' current intents. Here, intents can represent both task-related intents (e.g. an intent to pick up an object) or communicative intents (e.g. a dialogue move for requesting information). With this model, agents can be facilitated in performing *intent recognition* where they can be notified about active intents of other agents. Without this facility, agents would have to infer intents autonomously which would involve associating the type and timing of one or more physical actions to some possible underlying intent of the actor agent.

One of the implications of using a shared context is that an agent may loose the ability to make certain interpretations of its own. In case of environment perception, a *red car* is perceived as a *red car* by all agents; or in case of intent recognition, the meaning of a certain gesture is universally understood. In our view, it is up to a designer to decide what model(s) to use (e.g. facilitate in action recognition but let agents perform intent recognition autonomously).

Further, for the sake of believability, it is not desirable for agents to have unrestricted access to the shared context and control mechanisms are required to deal with this aspect (e.g. make sure agents cannot sense other agent's physical activities when they fall outside sensory range; or that agents cannot recognize communicative intents when they have not observed the physical actions involved in conveying the intent). As the required level of believability differs between different applications, the design of any such control mechanism is left for designers.

4. EVALUATION

To evaluate our model, we present a system implementation in which a scenario was run. Figure 2 shows an abstract design of the system. A C++-based game engine was used connected to a Java based MAS. Internally TCP/IP is employed for inter-process communication. Protégé is used to develop ontologies for the separate models in the shared context. The scenario run is as follows.

"Agent James is walking towards the train station to catch a train. Along the way he overhears a conversation between two other agents, Mike and Sally, who he is both acquainted to. Sally has just asked Mike about the current time. She is a bit early for an appointment with a friend of hers and she is wondering if she still has time to pick up her laundry from the laundry shop. Now Mike replied that he did not know the time. James however does know the answer and since he is not in a hurry, he walks over to Sally and informs her about the information she sought. At that time Sally sees her friend walking and reasons that she does not have any time left, regardless of the current time. She politely interrupts the conversation and starts approaching her friend."

The scope of this scenario is fairly small though it does illustrate agents requiring good situation and social awareness to behave as described: Sally is aware of her situatedness in the environment, surrounded by other social agents. She initiates a social interaction to obtain missing information she can use for prioritizing on task-related (picking up the laundry) and social goals (meeting with a friend). Mike and James perceive intentions of Sally through direct and overheared communication. Last, Sally notices her friend within sensory range and pursues her original goal (meeting with a friend), though after naturally terminating her current interactions (e.g. expressing gratitude). Using the middleware's shared social context, agents were able to efficiently build models of their environment and the activities of others. These models were then used in dialogue processing or after belief formation in social reasoning.

5. CONCLUSION

In this paper we proposed the use of a shared social context managed by a middleware to facilitate agents in perceiving their environment and physical and social activities of other agents in this environment. Still, providing this shared context does not preclude agents to keep an individual context in which they might keep subjective information about the social situation and other agents. E.g. they can keep track of the reputation, reliability or status of other agents based on their own interactions with them. So, the shared social context should be seen as an easy obtainable, but general basis for creating a personal social context rather than an all-compassing and restrictive model.

6. REFERENCES

[1] J. Gratch. Socially situated planning. In K. Dautenhahn, editor, *Socially Intelligent Agents: The Human in the Loop*, pages 61–64. AAAI Press, Menlo Park, California, 2000.