Keeping the Trainee on Track[1]

Joost Westra [a] Frank Dignum [a] Virginia Dignum [b]

[a] *Universiteit Utrecht, P.O. Box 80.089 3508 TB Utrecht*
[b] *Delft University of Technology. P.O. Box 2600 GA Delft*

Abstract

Serious games and other training applications have the requirement that they should be suitable for trainees with different skill levels. Current approaches either use human experts or a completely centralized approach for this adaptation. Where agents are increasingly used in serious game implementations as a means to reduce complexity and increase believability, their use for adaptation can lead to situations in which the lack of coordination between the agents makes it practically impossible to follow the intended storyline of the game and select suitable difficulties for the trainee. In this paper we propose an architecture for game design that introduces a monitoring module to check the development of user skills and direct coordinated agent adaptation. Agents propose possible courses of action that are fitting their role and context, and the monitor module uses this information together with its evaluation of user level and storyline progress to determine the most suitable combination of proposals.

1 Introduction

Dynamic difficulty adjustment is an important aspect in training applications that need to be suitable for a large variety of users with different skill levels. Current approaches of dynamic difficulty adjustment in games use a purely centralized approach for this adaptation [1]. This becomes impractical if the complexity increases and especially if past actions of the non player characters (NPC's) need to be taken into account while trying to adapt to the skill level of the trainee (as is needed for serious games). Distributing the responsibility of staying believable and adjusting to game progress, over the different non player characters creates a much more manageable situation, but this might lead to unwanted situations if their adaptation is not well coordinated. In serious games, quality is measured in terms of how well the components in the game are composed, how they encourage the player (or trainee) to take certain actions, the extent to which they motivate the player and how well the gaming experience contributes to the learning goals of the trainee. The search for enhanced believability has increasingly led game developers to exploit agent technology in games in order to preserve believable storylines. However, this might be undermined if agents start to adapt individually to the trainee. In order to optimize learning, serious games should provide the trainee an ordered sequence of significantly different and believable tasks. Without a clear organization structure, adaptation can quickly lead to a disturbed storyline and the believability of the game will be diminished. Coordination of agent actions also becomes a lot more manageable if there is a facilitating system that allows the designer to put restrictions on the possible plans performed by the agents. In previous work [2, 4] we proposed the use of multi-agent organizations to define a storyline (defining coordination restrictions on the agents) in such a way that there is room for adaptation while making sure that believability of the game is preserved.

2 Framework

To get a better understanding of the different elements of the whole framework we briefly describe the different elements and the information that is passed between them. Figure 1 shows a schematic overview of all the different elements of the framework. We are currently using a custom Java environment as our *game world*, but our approach is also applicable to other games. The *NPC's* and other dynamic game elements in

[1]This is an extended abstract of our paper at CIG'10 [3]

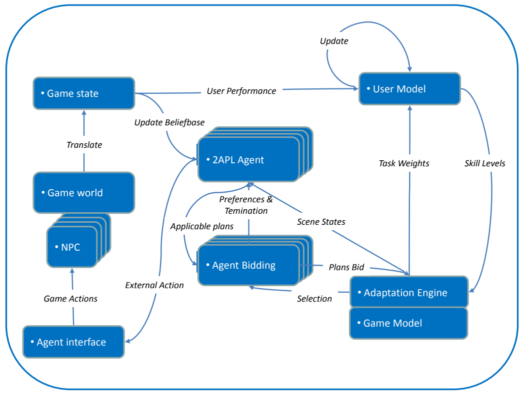

Figure 1: Framework overview

the game are controlled by *2APL agents*. The agents in the game have the capability to perform basic actions, like walking to a certain location or opening a door. The higher level behaviors are specified in the *2APL agents* which send the basic *external actions* to the *agent interface* which translates these commands to basic game actions. The *game state* is used to update the beliefs of the agents, update the progression of the game and pass the performance of the trainee to the user model. The *user model* uses this information and the *task weights* from the adaptation engine to update the estimated skill level for each state. These updated *skill levels* can then be used again to find better matching agent behaviors. The 2APL agents can perform different actions depending on their beliefs and dependent on the scene states. The *game model* contains information about the desired storyline of the game and keeps track of how far the game has progressed in the storyline. This information is passed to the *2APL agents* to influence the possible actions they can perform. The *agent bidding* module specifies the agent preferences for all the *applicable plans*. The *adaptation engine* uses this information and the information from the *user model* to find the plan assignment for the agents that best serves the situation for the trainee. The bidding module of the agent uses this information to control the plans that are selected by the agents. The *adaptation engine* is only used if new adaptation or coordination is necessary, otherwise the agents will continue executing their own plans.

In this abstract we discussed online adaptation in serious games. The adaptation is based on the use of learning agents. In order to coordinate the adaption of the agents we use an organizational framework that specifies the boundaries of the adaptation in each context. We argue that an agent based approach for adapting complex tasks is more practical than a centralized approach. It is much more natural when the different elements are implemented by separate software agents that are responsible for their own believability. The proposed model for game adaptation selects tasks that are most suitable for the trainee while following the specification of the game model and taking the preferences of the separate agents into account. The combination of adaptations selected at each moment is done through a kind of combinatorial auction that provides a balance between local optimization of the task and believability of the agent and overall difficulty of the situation for the trainee. We have shown that the system is able to not only adapt small elements in the game but also present the trainee with adapted storylines. We have also shown that adaptation sometimes needs to be coordinated and how this in handled by our framework. And finally we have shown that it relatively easy to make sure that all the tasks specified in the storylines are actually executed.

References

[1] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma. Adaptive game AI with dynamic scripting. *Machine Learning*, 63(3):217–248, 2006.

[2] J. Westra, H. Hasselt, F. Dignum, and V. Dignum. Adaptive Serious Games Using Agent Organizations. In *Agents for Games and Simulations*, pages 206–220. Springer, 2009.

[3] Joost Westra, Frank Dignum, and Virginia Dignum. Keeping the Trainee on Track. *CIG'10 IEEE Conference on Computational Intelligence and Games*.

[4] Joost Westra, Frank Dignum, and Virginia Dignum. Modeling agent adaptation in games. *Proceedings of OAMAS 2008*, 2008.