Scalable Adaptive Serious Games using Agent Organizations

(Extended Abstract)

Joost Westra
Utrecht University
joostwestra@gmail.com

Frank Dignum
Utrecht University
dignum@cs.uu.nl

Virginia Dignum
Delft University of Technology
m.v.dignum@tudelft.nl

Categories and Subject Descriptors

K.8 [**Personal Computing**]: Games

General Terms

Design

Keywords

Adaptation, Serious Games, Scalable, Agent organization

1. INTRODUCTION

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. These centralized approaches become very impractical and will not scale if the complexity of the game increases. Agents can be used in serious game implementations as a means to reduce complexity and increase believability but without some centralized coordination it becomes practically impossible to follow the intended storyline of the game and select suitable difficulties for the trainee. In this abstract we show that using agent organizations to coordinate the agents is scalable and allows adaptation in very complex scenarios while making sure that both the storyline and the right difficulty level for the trainee are preserved.

We argue that a system without any coordination will not result in good adaptation if the complexity of the game and the number of different adaptable elements increase. Multiple elements could adapt in the same direction and will overshoot the desired target difficulty for the trainee. Or the agents all adapt in a very similar way, resulting in state where the NPC's are not performing all the tasks required by the scenario. We will also show in this abstract that a naïve centralized approach will become too slow if the number of available tasks that NPC's can choose becomes too big. While this might not be problematic with the current entertainment games yet (where adaptation to the user is very limited), it will be a problem with more complex serious games. In previous work [2, 1] we proposed to use agent organizations plus a related adaptation engine to control the coordination and adaptation of the agents, while leaving them enough autonomy to determine their next actions. We will show that this gives the right balance between distributing decision making (leading to scalability) and keeping the game believable and immersive. This approach has the benefits of direct adaptation without the need for the designer to directly specify how the adaptation should be done. The designer is able to specify certain conditions on the adaptation to guarantee the game flow but does not have to specify which implementations are chosen after each state. In this abstract we focus on the scalability aspect of the framework.

Cite as: Scalable Adaptive Serious Games using Agent Organizations (Extended Abstract), Joost Westra, Frank Dignum and Virginia Dignum, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1291-1292.

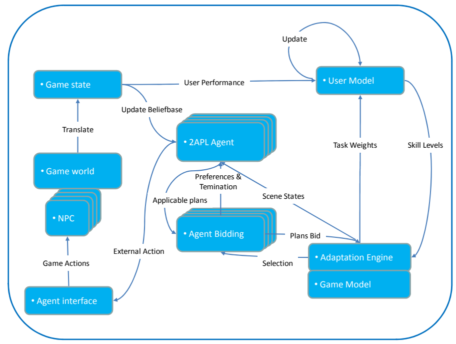

Figure 1: Framework overview

2. SCALABLE FRAMEWORK

To get a better understanding of the different elements of the whole framework we first 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. The *NPC's* and other dynamic game elements in 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 whole storyline of the game is build from a collection of partially ordered different scenes (the interaction structure). In each scene we specify the scene objective and the roles that are being played in this scene. Each participating agent plays one of these roles and therefore helps to complete the scene objective. This results in agents goals and plans that are very natural and relevant to the scene and therefore relevant to the storyline.

The scenes are defined by scene scripts that specify which roles participate and how they interact with each other. In these scenes the results of the entire scene are specified and how and in what order the different agents should interact. In our approach we use NPC's that are based on BDI agents. This means that agent behavior is specified using high level goals and act according to their internal believes. This makes it much easier to identify why an NPC why an agent performs a certain plan. We specially use the term "high level" goals because some of the lower level behaviors can better by specified by other approaches then BDI. Using a combination of BDI agents with an agent organization architecture, results in very natural agent objectives.

An obvious danger of coordinating actions between agents is that, if all possibilities are always sent to a central point which decides the best the combination, we can run into scaling problems and you might as well use completely central control instead of an agent based approach. There are two main differences between a completely centralized approach and our approach. The first is that the agents control when adaptation is initiated. The second is that the agents make a pre-selection of the plans that are applicable in regards to their internal state and the current game state. The numbers of plans combinations that need to be considered is much lower than a fully centralized system. Because pruning is performed on the agent level, even more on the scene level and also on the combination level because of *game model* boundaries.

We analyze the scaling difference between a naïve centralized approach and our coordinated distributed approach. Both approaches will have a very similar approach of combining the actions of the NPC's but the main difference will be in the remaining number of plans proposed by the agents. We aim to use reasonable assumptions that correspond to the type of serious games we have encountered during our research. The validations and explanations of these assumptions are beyond the scope of this abstract. Using the assumptions we get the following results. The purely naïve approach will have 720 (30 scenes * 4 sub-scenes * 6 actions per sub-scene) different plans for each agent active at the same time. Our approach will have 12 (6 actions per sub-scene *2 sub-scenes active per scene * 2 active scenes /2 for believability filtering) In figure 2 we plotted out the number of combinations for both approaches depending on the number of agents. As can be seen the number of combinations already add up very quickly with our distributed filtering but it is much more manageable then without the filtering. Even with four agents the filtered approach is already 12960000 times as slow. With more than four agents the naïve approach becomes completely impractical.

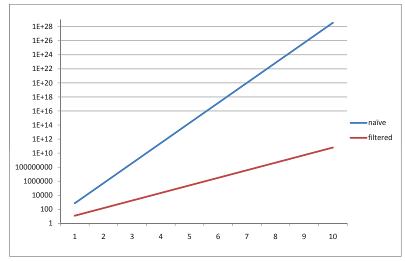

Figure 2: Number of possible action combinations

In practice our distributed approach will be much faster because we are also efficiently filtering out impossible combinations. This means that in practice the number of combinations that will be evaluated will be much lower than the estimations from our graph. We, however, also realize that the term scaling is relative. The coordination is fast enough by using our distributed approach for the type of games we are investigating and is much faster than the naïve approach. But because of the exponential nature of the remaining coordination it will not scale to games with massive numbers of NPC's.

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. We have shown that by using an agent organization framework we can segment the game in scenes in a natural way to describe which of the possible actions of the agents are relevant at the current moment. Every selection phases reduces the number of plans that need to be coordinated. This greatly reduces the scaling problems when coordinating multiple agents with a large variety of possible actions.

3. REFERENCES

[1] J. Westra, F. Dignum, and V. Dignum. Modeling agent adaptation in games. *Proceedings of OAMAS 2008*, 2008.

[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.

This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).