Feeling the Ambiance: Using Smart Ambiance to Increase Contextual Awareness in Game Agents

Colm Sloan
Dublin Institute of Technology
Dublin, Ireland
colm.sloan@student.dit.ie

John D. Kelleher
Dublin Institute of Technology
Dublin, Ireland
johnd.kelleher@dit.ie

Brian Mac Namee
Dublin Institute of Technology
Dublin, Ireland
brian.macnamee@dit.ie

ABSTRACT

The behaviour of non-player character game agents can be made more interesting and believable through the use of increased contextual awareness. In this paper, we present *smart ambiance* which allows information about the ambiance of an environment (determined by the environment itself, objects in the environment and recent events) to be used in agent plan generation. We demonstrate how this leads to contextually influenced action selection and, in turn, more interesting and believable character behaviour.

1. INTRODUCTION

In many commercial games non-player characters act the same regardless of their context. For example, in *Fallout 3* (fallout.bethsoft.com) a village populated with friendly townsfolk is attacked by mutants whom the player must repel. After the attack surviving villagers still give the player a friendly greeting even though they are surrounded by the corpses of their neighbours. Furthermore, people newly arriving into the village have no misgivings despite the fact that they have to walk past a pile of dead bodies to enter. This inappropriate behaviour hurts the believability of the characters and the immersion of the gameplay.

This paper introduces *smart ambiance* as an attempt to address the problem of contextually unrealistic behaviour by having the objects in the environment of an agent implicitly affect the actions selected by the agent. We will present the mechanics of smart ambiance and through examples illustrate how it can create more interesting and believable game agent behaviours.

2. RELATED WORK

Amongst a variety of other techniques, planning systems have become a popular choice to drive the behaviours of non-player characters in games. The literature contains numerous examples of approaches using reactive planning [15], deliberative planning [13, 7], or some hybrid of both [4, 9].

This research uses a planning system based on smart objects, augmented with contextual information.

A *smart object* is an object within an intelligent virtual environment that contains more information than its inherent properties (e.g. position), usually containing information about how an agent can interact with it [6]. For example, a smart object could tell an agent how to grasp it or gaze at it [12]. Smart objects have also been used for planning. Early implementations embedded the entire plan in the object [5] but later representations used a more flexible and scalable method of STRIPS-like [3] conditions and effects associated with performing an action with the object [1]. Brom [2] used role passing [8] with a hierarchical task network to create plans for virtual humans using smart objects.

There is some overlap with the contextual aspect of this work and Paanakker's work where agents could stay away from (or go to) certain parts of a map that, for example, had dead allied soldiers in them [11]. Perhaps the work bearing most similarity to the approach outlined in this paper is that of Sung et al. [14]. Their approach uses *situations* which can be defined spatially or non-spatially. The *spatial situations* use smart objects and predefined zones that give possible actions to agents when they come within a certain range of them. For example, a bus stop area would add the possible actions of getting on or alighting from a bus. The *non-spatial situations* use smart objects that provide all information the character will need to take actions and may contain information about the relationships between characters.

None of these approaches have altered the actions an agent may take through planning with the use of the ambiance of an area nor do they use smart objects or action effects to change a virtual ambiance. We examine if it is possible to create more interesting and emergent behaviours using these ideas.

3. APPROACH

This research uses Goal-Oriented Action Planning (GOAP) [10], a popular planning system for game agents. Each action in GOAP has an associated cost and STRIPS-like conditions and effects. In GOAP, the A* search algorithm is used to search through a library of available actions, with associated costs, for the lowest-cost action sequence that will bring about a goal state.

Our approach extends GOAP to include *smart ambiance*, a representation of the general mood or atmosphere within an environment in which an agent is planning actions. The ambiance can arise from the type of an environment, the types of objects in an environment, and recent events that have taken place in an environment, and is defined through a collection of property-value pairs, or *ambiance effects*. The property is a descriptor of the ambiance effect. An ambiance effect indicates the type of ambiance present (e.g. subdued, seriousness, happy, sad, fearful, exciting[1]) and the strength of this ambiance. Each agent has a unique smart ambiance based on the environment it is in, nearby objects, and the events the agent has witnessed.

Actions in a GOAP system normally have an associated cost and a list of conditions and effects. We have extended GOAP actions to also contain a list of *action ambiance details* which are ambiance effects that a planning agent should take into account when considering the action for inclusion in a plan. If an ambiance effect is present in the smart ambiance in which an agent is forming a plan, and the same ambiance effect is defined to impact an action through the inclusion of an action ambiance detail, then the likelihood of including the action in a plan will be altered. This change is achieved by altering the cost associated with an action according to the action ambiance detail (which indicates whether the likelihood of inclusion of an action in a plan should increase or decrease in the presence of an ambience effect) and the strength of the ambiance effect.

As an example of how the smart ambiance alters action costs consider a man in a library attempting to fulfil his goal of socializing with two possible actions available: **chat** and **check_social_network**. Figure 1 shows a representation of the man, the actions, the smart ambiance he is in and the details relevant to calculating action costs. The smart ambiance contains a list of action effects with the origin of each effect specified in brackets beside it. For example, twice has someone performed the **shush** action, (an event) which has added ambiance effects to the smart ambiance with cost functions that may alter the cost of actions associated with the **volume** descriptor. The planner gets the default cost of the **chat** action. It then checks to see if the action shares a common descriptor with the agent and the smart ambiance. The **volume** is found to be shared by all and so the cost modification of each ambiance action effect is applied. In this case, the **library** and two **shush** action effects will apply their cost function, which will alter (increase) the cost of the **chat** action, making it less likely to occur.

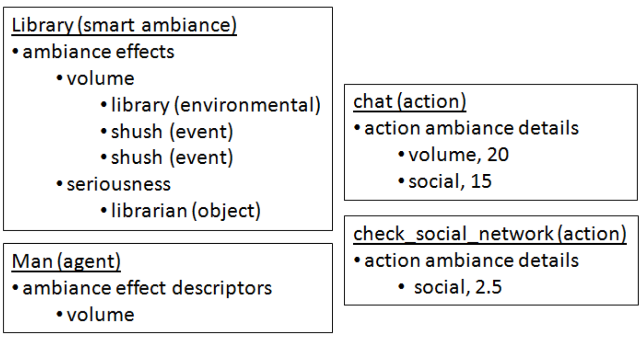

Figure 1: An example of how smart ambiance can impact action selection during planning

[1]At present we use an arbitrary, and unbounded set of ambiance types but accept the issue of which types to allow must be examined in the future

There are three sources of ambiance used in the smart ambiance system: environmental ambiance, event ambiance and object ambiance. *Environmental ambiance* is the inherent ambiance of a space. It is implemented by attaching action ambiance effects directly to a space in the virtual world. For example, a library might have an environmental ambiance associated with it to favour quiet actions. Environmental ambiance is defined by a game designer to capture the general mood of different environments.

Using this library example we can illustrate how environmental ambiance effects and action ambiance details interact to produce more contextually appropriate agent behaviour. The library has a smart ambiance with the **volume** descriptor. The **chat** action bears the same descriptor. The **check_social_network** action has no relevant descriptors. When cost is being calculated for **chat** while in the library, it increases to become higher than the cost of the **check_social_network** action, which just uses its default cost because it has no descriptors in common with the agent environment. This will cause the ambiance aware agent to select the more contextually sound action of using the computer in the library.

Event ambiance is an ambiance action effect produced through the execution of an action. For example, a pedestrian crossing has several people standing at the edge of a road. They all have the goal of getting to the other side. Some agents are nervous and others aren't. The light turns green and all agents start to walk across the road. When the light turns amber, nervous agents begin to run. Soon after, the agents who weren't nervous to begin with will break into a run because seeing the other agents run made them nervous.

No agents would have run at all if there had been no nervous agents present at the crossing but their actions affected the ambiance of the crossing that caused other agents to run with them. The **walk** and **run** actions have the **nervousness** descriptor. The smart ambiance of agents at the crossing will share this descriptor which means that the cost of these actions will be altered by the **nervousness** ambiance action effect cost modification function. This means that agents will choose the **walk** or **run** actions depending on the value of the **nervousness** property within their smart ambiance.

Object ambiance is the effect an object has on the smart ambiance. New ambiance action effects are added to a smart ambiance as soon as an object enters that smart ambiance. This allows objects to dynamically generate and alter a smart ambiance. For example, a man is in a room in house and wants to entertain himself. He knows that he could play games on his computer but his creative surroundings from the books of Shakespeare to the paintings on his walls seem to compel him to do something more interesting and unusual, causing him to instead write a funny limerick.

The man in the example had the goal of being entertained. In this scenario, two of the possible actions that result in being entertained are the **play_computer** and **write_limerick** actions. The **write_limerick** action had **creative** as a descriptor. The smart ambiance of an agent planning in the room will have been given the **creative** ambiance effect by the paintings and books in the room. The function for this ambiance action effect relating to the **creative** descriptor greatly decreased the cost of all creative actions, making them more likely to be selected.

Objects automatically generating an ambiance is useful because it means that designers wouldn't have to manually specify the ambiance for each area. Instead, they could just place objects inside an area and the combined ambiance effects would create the smart ambiance. This is also interesting because the cost functions will interact in unpredictable ways, creating emergent behaviour. Actions will always be influenced by the surroundings of the agent, making them more contextually logical.

4. CONCLUSIONS & FUTURE WORK

The behaviour of autonomous agents is becoming a more important part of computer games. While there are a few techniques that combine smart objects and planning to control these agents, none have used the ambiance of an area with an industry standard planner to create more contextually appropriate behaviour. Using the proposed framework, agent action selection is altered by a smart ambiance modifying the costs of particular actions.

There are several advantages to using smart ambiance: (1) emergent behaviour is quite likely to occur through the interactions of several objects altering ambiance properties, consequently less scripting on agent behaviour will be needed; (2) agents will perform more contextually sensitive actions; (3) information about how an agent should behave is decentralised away from the agent and across the environment; (4) and many popular game engines already have the data structures (such as triggers) required to implement smart ambiance.

But smart ambiance also has several disadvantages: (1) a new cost function must be made for each ambiance property, (2) smart ambiance can at times be computationally a little bulkier than what is strictly necessary to create the effects that it leads to; (3) and emergent behaviour is often seen as bad by game designers as there is a loss of control from the perspective of the designer. The system is also more computationally demanding than ordinary GOAP, which is already considerably more demanding than other agent behaviour control systems.

Despite these weaknesses, we believe that the smart ambiance presented in this paper offers interesting and realistic improvements to the behaviours of planning agents and that the demonstrations illustrate the usefulness of our approach.

The development of smart ambiance is still in its earlier stages and so there is work remaining in further developing it. For example, we are working on the composition of the list of possible ambiance effects and the interactions between different ambiance effects. The work has a number of interesting potential additions that could make smart ambiance even more powerful. The first of these is in exploring the different uses of object ambiance beyond automatically creating a smart ambiance. Objects could also have a proximity-based ambiance that is felt more strongly as an agent is closer to the object. For example, a Dementor from *Harry Potter* causes anyone nearby to feel great fear, which would make them more likely to perform actions associated with fear, such as screaming or fleeing. A focus on planning through the manipulation of ambiance to increase the probability of action success could help create more realistic behaviour too. For example, a boy is in a house with a girl and he has the intention of kissing her. Rather than just trying to immediately kiss her, he could make her more likely to kiss him by dimming the lights and putting on some romantic music. We will also develop how smart objects can be used to express inter-agent relationships. For example, a man may act differently in a room full of only men as he would in a room full of only women. Before any of these extensions are developed, we will first analyse how the different types of ambiance can combine together in a single environment.

5. REFERENCES

[1] T. Abaci, J. Ciger, and D. Thalmann. Planning with Smart Objects. *WSCG (Short Papers)*, pages 25–28, 2005.

[2] C Brom. *Action Selection for Virtual Humans in Large Environments*. PhD thesis, Charles University in Prague, 2007.

[3] R.E. Fikes and N.J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[4] N.A. Hawes and University of Birmingham. *Anytime deliberation for computer game agents*. Citeseer, 2003.

[5] M. Kallmann. *Object Interaction in Real-Time Virtual Environments*. PhD thesis, Citeseer, 2001.

[6] M. Kallmann and D. Thalmann. Modeling Objects for Interaction Tasks. In *Computer Animation and Simulation'98: Proceedings of the Eurographics Workshop in Lisbon, Portugal, August 31-September 1, 1998*, page 73. Springer Verlag Wien, 1999.

[7] J.E. Laird and P.S. Rosenbloom. Integrating execution, planning, and learning in Soar for external environments. *Ann Arbor*, 1001:48109–2110.

[8] B. MacNamee, S. Dobbyn, P. Cunningham, and C. OŠSullivan. Men behaving appropriately: Integrating the role passing technique into the aloha system. In *Proceedings of the AISBŠ02 symposium: Animating Expressive Characters for Social Interactions (short paper)*, pages 59–62. Citeseer, 2002.

[9] A. Nareyek. Intelligent agents for computer games. In *Computers and Games, Second International Conference, CG*, volume 2063, pages 414–422, 2000.

[10] J. Orkin. Agent Architecture Considerations for Real-Time Planning in Games. *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment*, 2005.

[11] F. Paanakker. Risk-Adverse Pathfinding Using Influence Maps. *AI Game Programming Wisdom*, 4, 2008.

[12] C. Peters, S. Dobbyn, B. Mac Namee, and C. OŠSullivan. Smart Objects for Attentive Agents. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, volume 13, page 14. Citeseer, 2003.

[13] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009.

[14] M. Sung, M. Gleicher, and S. Chenney. Scalable behaviors for crowd simulation. In *Computer Graphics Forum*, volume 23, pages 519–528. Wiley Online Library, 2004.

[15] B. Weber, P. Mawhorter, M. Mateas, and A. Jhala. Reactive Planning Idioms for Multi-Scale Game AI. 2005.